

Конспект лекций

Глобальная сеть состоит из *сайтов* доступных для общего пользования, закрытых частных сайтов, корпоративных и локальных (доступных на уровне какой-либо локальной сети).

Сайт (веб-сайт англ. *website*, от *web* – паутина, и *site* – «место») – это место во всемирной сети (интернете), которое имеет свой адрес, собственного хозяина и состоит из отдельных *web*-страниц, которые мы видим как одно целое.

Все странички каждого сайта объединяются одним корневым адресом (то, что мы набираем в адресной строке *браузера*), тематикой, системой и дизайном. Каждая страница *web* – сайта это документ, структура которого описана при помощи языка разметки (X) HTML. Страницы сайтов могут быть как простыми наборами информации – тексты и картинки, так и сложными, с огромным количеством функций.

У каждого сайта свои *цели, задачи, функции, характеристики объемов и др.*, которые реализуются благодаря возможностям и преимуществам интернет-технологий.

Сайты можно условно *классифицировать* по:

1. *целям и задачам* его создания и функционирования – коммерческий, некоммерческий, информационный, рекламный, учебный, поисковая система и др.);
2. *объемности* – хоум-пейдж, т.е. домашняя страничка, сайты-визитки, интернет-представительства, веб-порталы и т.п.;
3. *интерактивности* – наличие интерактивных модулей (гостевые книги, форумы, модули голосования, формы заказов, авторизация...);
4. *методу их создания и функционирования* – динамические и статические (последние в настоящее время уже практически нигде не встречаются).

Создание сайта начинается с **концепции и структуры** - т.е. с постановки задачи в соответствии с целями сайта. И уже на основе этой информации необходимо подбирать нужные материалы. Очень важно представить всю информацию удобным и понятным образом, чтобы то, что вы хотите передать, узнал бы и пользователь сайта.

Очень важна - информационная структура сайта. Продуманная информационная структура гарантирует, что пользователи потратят меньше времени на поиск нужной информации, и никогда не скажут, что они чего-то не нашли. При хорошей структуре они всегда обнаружат, что один документ связан ссылками с другими документами по той же теме. Они всегда смогут легко переключаться с поиска документов на их просмотр и обратно. Они лучше будут понимать, какую информацию сайт может им предложить.

Содержание сайта – это совокупность информационных ресурсов, удовлетворяющих следующим условиям:

Во-первых, и это самое главное условие, информация, которую вы хотите донести до посетителей должна быть полезной для них. Информация по возможности должна быть уникальной, т.е. такой, чтобы ее нигде не могли бы найти кроме как у вас;

Во-вторых, постарайтесь наиболее полно охватить ту тему, которой посвящен ваш сайт. Постарайтесь специализироваться, т.е. давать информацию на какую-то конкретную тему. Лучше чтобы было много о малом, чем мало о многом. К тому же сейчас очень много серьезных и раскрученных порталов и с ними сложно конкурировать. Да и сами посудите, куда бы вы пошли, если бы хотели найти специализированную информацию?

Если ваш сайт посвящен бизнесу, публикуйте новости вашей компании или фирмы. Смотрите широко и предоставляйте информацию и новости не только по вашей компании, но и по всей отрасли в которой она работает. Если вы продаете товары, то предоставляйте новости о родственных им товарах. Эта информация может быть полезной для ваших посетителей и они, возможно, захотят вернуться к вам на сайт (полезным для этого может стать ссылка

"добавить в избранное"). Если же ваш сайт является информационным, то можно разместить ссылки на другие сайты той же тематики, что и ваш. Конечно, не все решатся на этот шаг - рекламировать собственных конкурентов, и если вы боитесь, что посетители уйдут от вас, то можно открывать ссылки на другие сайты в новых окнах.

Возможно, вы являетесь профессионалом своего дела, и тогда вы можете предоставить на своем сайте бесплатные консультации по интересующим посетителей вопросам. Для этого достаточно иметь гостевую книгу.

Теперь немного о том, как должна быть оформлена и подана информация на сайте. Постарайтесь не делать статьи очень длинными. В то же время не дробите их на маленькие кусочки так, чтобы посетителю приходилось бы все время переходить с ссылки на ссылку. Постарайтесь уместить всю статью на одной страничке, чтобы можно было бы ее сохранить и прочитать как отдельный рассказ. Отделяйте между собой разные смысловые кусочки статьи пустыми строками или названиями (подразделы). Используйте удобную навигацию по сайту. На каждой страничке должна быть хотя бы ссылка на главную страницу, а еще лучше будет, если вы поставите ссылки на другие свои статьи по ходу текста (если она уместна в этом месте) или в конце страницы. Так человек, заинтересовавшийся вашей статьей, захочет прочитать и другие ваши или чужие работы на данную тему. И еще один важный совет - старайтесь обновлять сайт по мере возможностей. Посетители должны быть уверены, что, вновь зайдя на ваш сайт, они найдут для себя что-то новенькое. Иначе они просто не будут больше вас посещать.

Дизайн сайта – это совокупность графических элементов, шрифтов и цветов, реализованных на сайте.

Основная задача дизайна сайта – объединение всех информационных блоков на художественно проработанной основе и формирование у посетителя приятного впечатления. По сути, дизайн задаёт общий стиль вашего сайта, помогает посетителю с первого взгляда понять, что его здесь ждёт. Грамотно

разработанный дизайн является одним из важнейших факторов, определяющих посещаемость вашей веб-страницы.

Как правило, дизайн сайта – это внешнее его оформление, которое призвано, как минимум не отпугнуть посетителя Вашего интернет-ресурса, и как максимум – завлечь, заинтересовать его. Как говорят опытные специалисты, *хороший дизайн сайта* – это незаметный, ненавязчивый дизайн, который не отвлекает посетителя от основного – от предоставленной целевой информации (вспомните хотя бы дизайн страниц известных поисковых систем - *Google и Яндекс*). Думаю, Вы согласитесь с тем, что человек, заходя на какой-либо ресурс Интернет, прежде всего, осуществляет поиск необходимой информации (исключением является лишь дизайнер, ищущий для себя новые решения и интересующийся именно дизайнерской тематикой). И в этой ситуации любая отвлекающая информация (слишком яркий, броский дизайн сайта, излишняя анимация, всплывающие рекламные окна) будут только мешать в достижении основной цели – получении необходимых данных, в поиске которых на Ваш сайт и заходил посетитель.

HTML (от англ. *HyperText Markup Language* – «язык разметки гипертекста») – стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создаются при помощи языка **HTML** (или **XHTML**). Язык HTML интерпретируется браузерами и отображается в виде документа, в удобной для человека форме. В настоящее время стабильная и самая распространенная версия языка **HTML 4.01**, а также активно развивается **HTML 5.0**, который поддерживает новейшие мультимедийные приложения.

Тег - это служебная инструкция, размещенная между символами меньше (<) больше (>). Теги могут быть открывающимися и закрывающимися. Например: <div> - открывающий тег, </div> - закрывающий тег. Открывающий тег указывает на начало какого либо блока, а закрывающий – на завершение этого блока.

Атрибут – это уточнение для браузеров, как «поточнее» задать тег; атрибуты описываются внутри открывающего тега в виде коллекции имя="значение". Например: <div class="header">.

Для комплексных разработок и ведения больших проектов существуют системы программных средств, которые называются **Интегрированными средами разработки** (англ. Integrated development environment или integrated debugging environment – IDE). IDE используются повсеместно: от разработки мелких статичных сайтов до полноценных игр и программных комплексов. Для веб-разработок IDE требует определенной настройки. Обычно среда веб-разработки в себя включает три компонента:

5. *текстовый редактор* – область для ввода кода;
6. *компилятор и/или интерпретатор* – компонент, который обрабатывает код;
7. *отладчик* – компонент, отвечающий за корректность кода и проверку синтаксиса.

Интегрированные среды разработки призваны максимально увеличить производительность программиста благодаря тесно связанным компонентам с простыми пользовательскими интерфейсами. Это позволяет разработчику сделать меньше действий для переключения различных режимов, в отличие от дискретных программ разработки. Поскольку IDE является сложным программным комплексом, то лишь после долгого процесса обучения среда разработки сможет качественно ускорить процесс разработки ПО. Обычно среда разработки предназначена и ориентирована на определенный язык программирования, предоставляя набор функций, который наиболее близко соответствует парадигмам этого языка программирования. Однако в настоящее время наиболее популярные IDE, такие как *Eclipse, ActiveState Komodo, NetBeans, Microsoft Visual Studio, WinDev u Xcode*, могут поддерживать несколько языков программирования, установив определенный модуль.

Графика на web-страницах

В настоящее время существует достаточно большое количество различных форматов графических файлов, которое можно разделить на две группы:

8. файлы, хранящие векторную графику;
9. файлы, хранящие растровую графику.

Растровая графика – это когда изображение хранится в виде маленьких ячеек – пикселей. Соответственно качество такой картинки ограничивается двумя факторами: это собственно размером самой картинке в пикселях, и разрешением изображения – то есть количеством пикселей на единицу длины (наиболее распространен показатель разрешения раstra в пикселях на дюйм – dpi). Файлы, хранящие растровую графику, это jpg, gif, bmp, tiff, png, psd и прочие.

Векторная графика – это когда изображение хранится в виде массива чисел, описывающих построение изображения в виде векторов, заданных координатами ключевых точек-вершин. Форматы векторной графики – dxf, swf, cdr, max, ai, svg, частично pdf.

В практике *web* – графики в основном используются три формата растровой графики – gif, jpeg и png, и один формат векторной графики – svg.

Форматы растровой графики различаются различными алгоритмами сжатия изображения. Так как в web размер файла до сих пор играет весьма существенное значение в силу ряда причин, следовательно, малый вес графического файла существенно повышает скорость загрузки изображения. Для разных «видов» картинок подходит тот или иной формат графики.

Определение необходимого формата – основное умение *web*-дизайнера. Для фотографий, портретов, картинок большого размера, насыщенных сложными деталями – лучше всего подходит формат *JPEG*.

Алгоритм сжатия этого формата *JPEG* работает таким образом, что при уменьшении «веса» картинке, а следовательно качества, изображение как бы «размывается», становятся плохо различимы четкие переходы между цветами, и появляются паразитные цветные пиксели как побочный эффект действия

алгоритма. Степень компрессии файла определяется каждый раз дизайнером исходя из его нужд, но оптимальным соотношением размер/качество изображения считается процент сжатия, равный 65.

Формат *GIF* наилучшим образом подходит для изображений небольшого размера, там, где необходима прозрачность (альфа-канал), и для анимированной растровой графики.

Алгоритм сжатия этого формата основывается на том, что изображению задается фиксированная цветовая палитра (от 2 до 256 цветов), а все близкие оттенки выкидываются, либо заменяются соседними цветами. Также алгоритм просчитывает изображение линиями – слева направо, и хранит информацию не о каждом пикселе в отдельности, а считает, сколько пикселей одинакового цвета стоят в ряд, и сохраняет информацию только о цвете и количестве пикселей. Это существенно снижает вес файла. Нетрудно заметить, что вертикальный градиент (сверху-вниз) будет весить в таком случае намного меньше градиента горизонтального (слева-направо). Это следует учитывать при создании изображений, особенно при создании анимированных баннеров, где вес складывается еще и из количества кадров и идет буквально война за каждый байт. Также возможно загружать *GIF*-изображения чересстрочным способом. Если картинка имеет очень большой размер и долго загружается, то будут сначала видны нечеткие контуры изображения, а по мере загрузки оно будет “проявляться”. Для этого при сохранении *GIF*-файла нужно включить метод *Interlaced* (Чересстрочный).

Формат *PNG* имеет большие возможности, он сочетает в себе свойства *GIF*- и *JPEG*- сжатия: отбирать только нужные цвета, использовать палитру шкалы полутонов, обеспечивать прозрачность, сложная последовательная загрузка. Хотя формат *PNG* поддерживается всеми современными браузерами, но все же некоторые браузеры могут некорректно его отображать.

При именовании файлов следует придерживаться простых правил. Во-первых, следует избегать как бессмысленных, так и «говорящих» названий. Название файла должно сразу определять его место в структуре *web* -страницы.

То есть, если это картинка к статье, то она должна располагаться в папке `articles` и ее названием служит `id` статьи. Если это позиция в каталоге, то в соответствующей папке (`items`, `groups`) картинка должна иметь названием `ID` группы, подгруппы или товара.

В то же время, если, к примеру, на сайте очень редко пишутся статьи, можно не вводить дополнительные папки, но тогда файл должен по названию однозначно ассоциироваться с данным разделом. Это можно сделать, к примеру, добавлением слова `news` или `art` перед `id` картинки (пример `news-34.jpeg`). Если к одному `id` относится несколько картинок разного размера – необходимо добавлять после `id` картинки расширение, обозначающее размер (для больших картинок – `b`, для маленьких – `s`) (пример `38-s.gif`, `art-08-b.jpeg`). Для нескольких картинок одного размера, можно ввести порядковые номера (пример: `art08-b_01.gif`).

SVG (от англ. *Scalable Vector Graphics* — масштабируемая векторная графика) — язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины (W3C) и входящий в подмножество расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно/растровой графики в формате XML. Поддерживает как неподвижную, так анимированную и интерактивную графику — или, в иных терминах, декларативную и скриптовую. Не поддерживает описание трёхмерных объектов (не путать с имитацией трёхмерности путём светотени).

Достоинства формата:

10. Текстовый формат — файлы SVG можно читать и редактировать (при наличии некоторых навыков) при помощи обычных текстовых редакторов. При просмотре документов, содержащих SVG графику, имеется доступ к просмотру кода просматриваемого файла и возможность сохранения всего документа. Кроме того, SVG файлы обычно получаются меньше по размеру, чем сравнимые по качеству изображения в форматах JPEG или GIF, а также хорошо поддаются сжатию.

11. Масштабируемость — SVG является векторным форматом. Существует возможность увеличить любую часть изображения SVG без потери качества. Дополнительно, к элементам SVG документа возможно применять фильтры — специальные модификаторы для создания эффектов, подобных применяемым при обработке растровых изображений (размытие, выдавливание, сложные системы трансформации и др.) В тексте SVG-кода фильтры описываются тегами, визуализацию которых обеспечивает средство просмотра, что не влияет на размер исходного файла, обеспечивая при этом необходимую иллюстративную выразительность.

12. Широко доступно использование растровой графики в SVG документах. Имеется возможность вставлять элементы с изображениями в форматах PNG, GIF или JPG.

13. Анимация реализована в SVG с помощью языка SMIL (Synchronized Multimedia Integration Language), разработанного также консорциумом W3C. Поддерживаются скриптовые языки на основе спецификации ECMAScript. SVG-элементами можно управлять с помощью JavaScript. Применение скриптов и анимации в SVG позволяет создавать динамичную и интерактивную графику. В SVG обеспечивается событийная модель, отслеживаются события (загрузка страницы, изменение ее параметров, события мыши, клавиатуры и др.) Анимация может запускаться по определенному событию (например «onmouseover» или «onclick»), что придает графике интерактивность. У каждого элемента есть свои собственные события, к которым можно привязывать отдельные скрипты.

14. SVG — открытый стандарт. В отличие от некоторых других форматов, SVG не является чьей-либо собственностью.

15. SVG документы легко интегрируются с HTML и XHTML документами.

16. Совместимость с CSS (англ. Cascading Style Sheets). Отображением (форматированием и декорированием) SVG элементов можно управлять с

помощью таблицы стилей CSS 2.0 и её расширений, либо напрямую с помощью атрибутов SVG элементов.

Недостатки формата:

17. SVG наследует все недостатки XML, такие как большой размер файла (впрочем, последний компенсируется существованием сжатого формата SVGZ).

18. Сложность использования в крупных картографических приложениях из-за того, что для правильного отображения маленькой части изображения документ необходимо прочитать целиком.

19. Чем больше в изображении мелких деталей, тем быстрее растёт размер SVG-данных. Предельный случай — когда изображение представляет собой белый шум. В этом случае SVG не только не даёт никаких преимуществ, но и даже обладает чрезмерно избыточным по отношению к растровому формату размером. На практике, SVG становится невыгоден уже задолго до того, как изображение дойдёт до стадии белого шума.

Звук на web-страницах

Есть четыре основных варианта на выбор: WAV, MP3, OGG и AAC. Но не все браузеры обеспечивают полную их поддержку.

WAV (в его наиболее распространенный формат PCM) является несжатым аудио. В результате, файлы, как правило, очень велики и из-за этого очень сильно возрастает время загрузки *web*-страницы.

Хоть *MP3* и является в данный момент самым распространенным, этот формат не открытый. За возможность с ним работать требуется платить определенную сумму обладателям патента.

Ogg Vorbis — это относительно новый универсальный формат аудио компрессии, официально вышедший летом 2002 года. Он принадлежит к тому же типу форматов, что и MP3, AAC, VQF и WMA, то есть к форматам компрессии с потерями. Психоакустическая модель, используемая в *Ogg*

Vorbis, по принципам действия близка к MP3 и иже с ними, но и только — математическая обработка и практическая реализация этой модели в корне отличаются, что позволяет авторам объявить свой формат совершенно независимым от всех предшественников.

Главное неоспоримое преимущество формата Ogg Vorbis — это его полная открытость и свобода. Более того, в нем использована новейшая и наиболее качественная психоакустическая модель, из-за чего соотношение битрейт/качество значительно ниже, чем у других форматов. Как результат — качество звука лучше, но размер файла меньше.

Advanced Audio Coding (AAC) — собственный (патентованный) формат аудиофайла с меньшей потерей качества при кодировании, чем MP3 при одинаковых размерах.

Также AAC — это широкополосный алгоритм кодирования аудио, который использует два основных принципа кодирования для сильного уменьшения количества данных, требуемых для передачи высококачественного цифрового аудио. Данный формат является одним из наиболее качественных, использующих сжатие с потерями, поддерживаемый большинством современного оборудования, в том числе портативного.

Видео на web-страницах

Видео файл в любом из видео форматов, следует воспринимать как zip-архив содержащий видео поток и аудио поток. Вот три видео формата, наиболее распространенных в сети:

20. mp4 = H.264 + AAC
21. .ogg/.ogv = Theora + Vorbis
22. .webm = VP8 + Vorbis

На звание кодека для HTML5 video в данный момент претендуют два кодека — Ogg Theora и H.264.

В основе *Ogg Theora* лежит кодек VP3, разработанный On2 Technologies. В 2002 году, On2 Technologies передали код VP3 под свободной BSD-подобной лицензией в руки Xiph.org Foundation, а также отказались от патентов на кодек (технически, не отказались, а просто передали право их использовать всем, но это по сути то же самое). С тех пор, Xiph.org продолжает развитие этого кодека.

Использовать Ogg Theora можно везде, всегда, без лицензионных или патентных отчислений.

H.264 — это лицензируемый стандарт сжатия видео. Его использование требует платы в странах, где действует патенты на него (в первую очередь, это США). Однако, на сегодняшний день, это один из самых лучших способов сжимать видео. Именно H.264 является стандартом де-факто сжатия HD-видео, к примеру. H.264 заметно эффективнее Ogg Theora по соотношению качество/битрейт.

Если кратко, H.264 — лучше, но даже его open-source реализации не могут быть использованы свободно в странах, где действуют патенты на него.

Другим вариантом воспроизведения видео является декодирования видео в браузере с использованием модульного подхода, не привязываясь к определённому кодеку. Мало того, в каждой операционной системе уже и так есть модульная инфраструктура кодеков. В Windows — это DirectShow, в Mac OS X — это QuickTime, в Linux — это gstreamer. А gstreamer ещё и кроссплатформенный, между прочим, и уже используется в кроссплатформенных программах, к примеру, Songbird для воспроизведения музыки использует именно gstreamer на всех платформах.

Использование gstreamer решит все проблемы с кодеками в браузерах один раз и навсегда. В частности, не будет никаких проблем с патентами, так как браузер будет распространяться без защищённых патентами кодеков, но на системе пользователя он сможет найти установленный плагин для этого кодека, и использовать его.

А мало того, в `gstreamer` предусмотрена возможность использовать кодеки, установленные в родном для данной системы фреймворке (для Windows — `DirectShow`, для Mac OS — `QuickTime`).

Для создания представления *web* –страниц предназначена технология **каскадных таблиц стилей** (`Cascading Style Sheets`, `CSS`), или просто *таблиц стилей*. Таблица стилей содержит набор правил (*стилей*), описывающих оформление самой *web* – страницы и отдельных ее фрагментов. Эти правила определяют цвет текста и выравнивание абзаца, отступы между графическим изображением и обтекающим его текстом, наличие и параметры рамки у таблицы, цвет фона *web* –страницы и многое другое. Стоит отметить, что в настоящее время при достаточно большом количестве *web*-браузеров, каждый из них интерпретирует `CSS`-стили со своими особенностями, что следует учитывать при верстке `HTML`-страниц.

Каждый стиль должен быть привязан к соответствующему элементу `Web`-страницы (или самой *web* –странице). После привязки описываемые выбранным стилем параметры начинают применяться к данному элементу. Привязка может быть явная, когда мы сами указываем, какой стиль к какому элементу *web* –страницы привязан, или неявная, когда стиль автоматически привязывается ко всем элементам *web* –страницы, созданным с помощью определенного тега.

Таблица стилей может храниться прямо в `HTML`-коде *web* –страницы или в отдельном файле. Последний подход более соответствует концепции `Web 2.0`; она требует, чтобы содержимое и представление *web* –страницы были разделены. Кроме того, отдельные стили можно поместить прямо в тег `HTML`, создающий элемент *web*-страницы; такой подход используется довольно редко и, в основном, при экспериментах со стилями.

Обычный формат определения стиля `CSS`:

Листинг 00.00

```
01 <селектор> {
02 <атрибут стиля 1>: <значение 1>;
03 <атрибут стиля 2>: <значение 2>;
04 . . .
05 <атрибут стиля n-1>: <значение n-1>;
06 <атрибут стиля n>: <значение n>
07 }
```

Селектор используется для привязки стиля к элементу *web*-страницы, на который он должен распространять свое действие. Фактически селектор

однозначно идентифицирует данный стиль. За селектором, через пробел, указывают список атрибутов стиля и их значений, заключенный в фигурные скобки.

Атрибут стиля представляет один из параметров элемента *web*-страницы: цвет шрифта, выравнивание текста, величину отступа, толщину рамки и др.

Значение атрибута стиля указывают после него через символ: (двоеточие). В некоторых случаях значение атрибута стиля заключают в кавычки. Пары `<атрибут стиля>:<значение>` отделяют друг от друга символом: (точка с запятой).

Объектная модель документа делает все элементы страницы программируемыми объектами. С ее помощью через языки сценариев можно получить доступ и управлять всем, что есть в документе. Каждый элемент HTML доступен как индивидуальный объект, а это означает, что можно изменять значение любого параметра любого тега HTML-страницы, и, как следствие, документ действительно становится динамическим. Любое действие пользователя (щелчок кнопкой мыши, перемещение мыши в окне браузера или нажатие клавиши клавиатуры) объектной моделью документа трактуется как событие, которое может быть перехвачено и обработано процедурой сценария.

Поведение, то есть набор правил, определяющих, как *web* –страница будет реагировать на действия посетителя. создается с помощью так называемых *web* –*сценариев* – программ, которые записывают прямо в HTML-коде *web* –страниц или, что предпочтительнее, в отдельных файлах. Эти программы пишут на языке *JavaScript*. *Web*-обозреватель считывает JavaScript-код и последовательно выполняет записанные в нем выражения, проводя вычисления и выполняя на основе полученного результата заданные манипуляции над *web* –страницей.

Общим заблуждением является то, что JavaScript аналогичен или тесно связан с Java, это не так. Оба языка имеют C-подобный синтаксис, являются объектно-ориентированными и, как правило, широко используются в клиентских веб-приложениях, на этом их сходство заканчивается:

23. Java реализует ООП подход, основанный на классах, JavaScript на прототипах;

24. Java имеет статическую типизацию, JavaScript динамическую типизацию;

25. Java загружается из скомпилированного байт-кода; JavaScript интерпретируется напрямую из файла (но часто с незаметной JIT-компиляцией).

Динамическое содержимое на *web* –странице можно разделить на автономное, независимое от пользователя (анимированный логотип, падающие снежинки) и интерактивное, напрямую зависящее от его действий (игры, галереи изображений, перемещение элементов).