

Лекция №2 ОСНОВЫ HTML

- 1 Основы языка HTML
2. История развития HTML
3. Структура <HTML>-документа
4. Форматирование
- 5 Изображения в HTML
- 6 Гиперссылки
- 7 Логическая и физическая структура сайта
- 8 Таблицы
- 9 Включение звука в Web-страницу
- 10 Включение видео в Web-страницу
- 11 &-последовательности
- 12 Режимы браузеров
- 13 Режимы Internet Explorer
- 14 Графика в web-дизайне

1 Основы языка HTML

HTML-документ представляет собой обычный текстовый файл в формате ANSI ASCII, содержащий собственно текст и специальные **теги** для его разметки, а также ссылки на другие документы, графические изображения и любые иные файлы. Теги задаются в файле и обрабатываются браузером в соответствии с правилами специального языка HTML (Hyper Text Markup Language – Язык Разметки Гипертекста).

HTML не является языком программирования.

HTML – это типичный язык разметки, т. е. с его помощью можно оформлять документы, создавать ссылки, но никак не писать программы.

Гипертекст это текст, в котором имеются ссылки для автоматического перехода на другие тексты – **гиперссылки**.

Тег (от англ., tag – ярлык, метка) – элемент языка HTML, используемый для разметки Web-страниц.

Элементы разметки состоят из заключённых в угловые скобки (< и >) **дескрипторов – тэгов (tags)** и их **атрибутов**. Совокупность открывающего (< >) и закрывающего (< / >) дескрипторов - есть **контейнер**. Элементы **HTML** подразделяются на *структурные* - которые организуют текст и на *форматирующие* - которые задают его стиль. Для создания документа **HTML** необходимо создать текстовый файл с содержимым, вставить нужные **тэги** и после внесения изменений текстовый файл сохраняется с расширением **.htm** или **.html**.

2 История развития HTML

Язык HTML был разработан Тимом Бернерс-Ли. В 1990-х годах HTML получил широкое признание благодаря быстрому росту Web.

HTML 1.0 – был направлен на представление языка как такового, где описание его возможностей носило скорее рекомендательный характер.

HTML 2.0 (ноябрь 1995) – был разработан под эгидой Internet для упорядочения общепринятых положений.

HTML+ (1993) и HTML 3.0 (1995) – это более широкие версии языка HTML. Несмотря на то, что в обычных дискуссиях по стандарту согласие

между разработчиками браузеров не было достигнуто, эти версии содержат ряд новых общих свойств.

HTML 3.2 (январь 1997) создан усилиями Рабочей группы World Wide Web Consortium по HTML (для упорядочения ряда общепринятых положений и изменений предыдущих версий).

HTML 4.0 (с 1999 г.) – вводятся механизмы таблиц стилей, скриптов, кадров, внедрения объектов, улучшенная поддержка разных направлений письма и направления справа налево, таблицы с большим количеством возможностей и новые свойства форм.

В стандарте XHTML используются те же синтаксические соглашения, что и HTML, но в нем ужесточены требования к следованию установленным правилам. Большая часть отступлений от правил разметки, которые сходят с рук в традиционном HTML, попросту неприемлемы в XHTML.

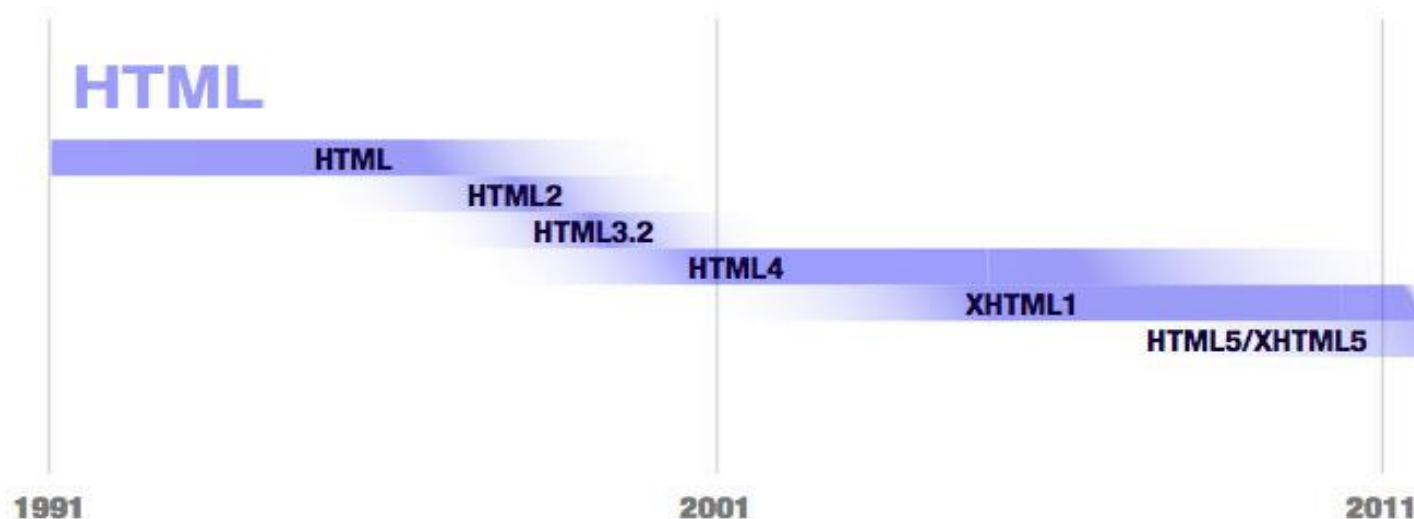
С целью работы над новыми решениями языка компании Opera, Mozilla и Apple создали рабочую группу по технологиям гипертекстовых web приложений сокращённо WHATWG (WHATWG - Web Hypertext Application Technology Working Group). Группа WHATWG ставила перед собой задачу постепенного расширения языка, в ходе которого были созданы две спецификации расширений: WebApplications1.0 и WebForms2.0. В итоге спецификации «эволюционировали» и появился HTML5.

В 2009 году W3C прекратил развитие XHTML 2.0 и начал разрабатывать HTML5 (решили писать без пробела) уже совместно с WHATWG.

К 2012-му году практически все современные браузеры в мире, начинают понимать язык HTML5, хотя еще и остаются некоторые теги, которые браузеры пока не понимают.

На данный момент, работа над языком HTML5 продолжается, создаются новые теги и технологии, всё это добавляется в спецификацию, сама спецификация HTML5 была опубликована 28 октября 2014 г.

В будущем решили не давать цифровых версий для HTML, если будут добавляться какие-либо новшества, то их просто будут внедрять в существующую спецификацию, а сам язык называть просто HTML.



3. Структура <HTML>-документа

Любой <HTML>-документ состоит из следующих основных частей:

- Строки с информацией о версии HTML, используемой в данном документе
- Блока заголовка документа
- Тела документа.

Каждый тег HTML принадлежит к определенной группе тегов, например, табличные теги направлены на формирование таблиц и не могут применяться для других целей.

Условно теги делятся на следующие типы:

- **теги верхнего уровня;**
- **теги заголовка документа;**
- **блочные элементы;**
- **строчные элементы;**
- **универсальные элементы;**
- **списки;**
- **таблицы;**
- **фреймы.**

Разница между блочными и строчными элементами следующая:

- ✚ Строчные элементы могут содержать только данные или другие строчные элементы, а в блочные допустимо вкладывать другие блочные элементы, строчные элементы, а также данные. Иными словами, строчные элементы никак не могут хранить блочные элементы.
- ✚ Блочные элементы всегда начинаются с новой строки, а строчные таким способом не акцентируются.
- ✚ Блочные элементы занимают всю доступную ширину, например, окна браузера, а ширина строчных элементов равна их содержимому плюс значения отступов, полей и границ.

Таблица 1. Основные тэги по типам

Тип	Тэг	Назначение
теги верхнего уровня	<html>	Начало и конец страницы
	<body>	Содержимое веб-страницы
	<head>	Описание страницы
теги заголовка	<title>	Имя страницы
	<meta>	Для хранения информации для браузеров
блочные элементы	<blockquote>	Для выделения длинных цитат
	<div>	универсальный блочный контейнер
	<h1>,...,<h6>	Заголовок (уровни от 1 до 6)
	<hr>	Горизонтальный разделитель

	<p> <pre>	Абзац Блок предварительно форматиров. текста
строчные элементы	<a> <big> <i> <small> <sub> <sup>	Ссылка на другую страницу Жирный Увеличивает размер шрифта на единицу Перевод строки Устанавливает курсивное начертание Для отображения изображений Уменьшает размер шрифта на единицу Универсальный тег, предназначенный для определения строчного элемента внутри документа Верхний индекс Нижний индекс
универсаль ные элементы	 <ins> <!-- ... -->	Для выделения текста, который был удален в новой версии документа Для акцентирования вновь добавленного текста Комментарий
списки	 <dd>, <dt>, <dl>	Нумерованный Ненумерованный Одельный элемент списка Список определений
таблицы	<table> <td> <th> <tr	Внешний элемент таблицы Элемент задающий ячейку таблицы Элемент, заголовочную ячейку Элемент, задающий строку таблицы

Информация о языке и версии HTML

Начиная с версии 3.2 каждый документ должен начинаться с декларации <!DOCTYPE>, которая необходима для того, чтобы сообщить браузеру версию языка HTML.

<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01 Strict//EN>

HTML 4.01 Strict – документ этого типа не должен содержать отмененных элементов и атрибутов, и не должен быть контейнером для фреймов.

<!DOCTYPE> используется для HTML5!

4 Форматирование

Оформление занимает особое место при создании WEB-сайта.

Оживляет страницу использование различных стилей шрифтов – полужирного, курсива, подчеркивания и т.д. Правильный выбор шрифта, его стиля, размера и цвета акцентирует содержимое страницы, не привлекая внимание посетителя сайта к самому шрифту.

Важное место при оформлении страниц занимают списки, которые позволяют выделить группы перечислений и оформить их различными маркерами.

Использование линий позволяет разделить смысловые части страницы и оживить ее.

Большое значение для повышения наглядности страницы имеют таблицы, позволяющие довести до посетителя сайта максимальное количество информации при минимальном занимаемом на странице месте. При этом необходимо выбирать толщину и цвет рамок так, чтобы они не подавляли текстовую информацию.

Особое внимание при оформлении страниц занимает размещение изображений и фоновых рисунков. Рисунки должны гармонировать с остальным оформлением страницы, не привлекая к себе чрезмерного внимания.

Очень оживляет оформление страницы использование спецэффектов, таких как бегущая строка, однако не следует злоупотреблять использованием спецэффектов, обычно на странице достаточно одной бегущей строки.

а. Изменение шрифта

Изменение шрифта текста в HTML может производиться с помощью тега **...**. При этом появляется возможность изменить у текста, оказавшегося внутри этого контейнера, начертание, размер и цвет шрифта.

...содержимое контейнера...

Тег **FONT** изменяет размер и цвет шрифта только для текста его содержимого, а для задания базовых значений (действующих по-умолчанию) для всего документа, используется тег **BASEFONT**, который устанавливает заранее значения для атрибутов тегов **FONT**, находящихся после него.

< BASEFONT FACE=...SIZE=...COLOR=...>

Тег **BASEFONT** не имеет закрывающего тега и должен располагаться в самом начале содержимого контейнера **BODY**.

б. Тип шрифта

Для изменения типа шрифта (гарнитуры) используется атрибут **FACE**, значением которого является список шрифтов, приведенных через запятую в порядке уменьшения приоритета. Если в системе клиента имеется шрифт с указанным названием, то он будет использоваться при воспроизведении данного фрагмента. Если в названии шрифта имеются пробелы, то желательно взять его в кавычки.

FACE = «Courier New, Arial, Tahoma»

Если атрибут **FACE** не указан, браузеры будут использовать по умолчанию шрифт **Times New Roman**

** Courier New<\FONT>
**

**Arial<\FONT>
**

** Tahoma <\FONT>
**

** System <\FONT>
**

в. Размер шрифта

Для изменения размера шрифта в тексте используется атрибут **SIZE**. Размер шрифта задаётся по некой шкале, самостоятельно определяемой браузером конечного пользователя и не имеющей прямой связи с размером типографской точки и другими единицами измерения.

SIZE= абсолютный размер

SIZE= относительный размер

Атрибут **SIZE** устанавливает размер шрифта текста, содержащегося в пределах тега **FONT**. Можно задать абсолютный размер шрифта, указав какое-либо целое число от 1 до 7. Для шрифта можно также указывать относительный размер (желательно в кавычках), присваивая атрибуту целое число со знаком, например, **SIZE= «+1»** или **SIZE= «-2»**. Это означает, что размер будет изменен относительно базового, заданного тегом **BASEFONT**, на 1 пункт больше или меньше, соответственно.

Если в теге **BASEFONT** не используется атрибут **SIZE**, то по умолчанию используется размер 3, что соответствует 13,5 пунктам для шрифта Times New Roman.

г. Цвет текста

Изменение цвета текста производится с использованием атрибута **COLOR**. Данный атрибут указывает цвет, которым будет выделен данный фрагмент текста. Цвета задаются в виде RGB-значения с шестнадцатеричной нотацией, либо точным названием цвета на английском языке.

Например,

COLOR = #FF0000

COLOR = Red.

Для получения любого другого оттенка цвета можно применить различные комбинации RGB-значений, записанные в шестнадцатеричном формате

Названия основных цветов и соответствующие им коды

Black или #000000 (черный)	Red или #FF0000 (красный)	Lime или #00FF00 (салатовый)	Yellow или #FFFF00 (желтый)
Maroon или #800000 (бордовый)	Gray или #808080 (серый)	Blue или #0000FF (синий)	Aqua или #00FFFF (морская волна)
Green или #008000 (зеленый)	Navy или #000080 (темно-синий)	Silver или #C0C0C0 (серебристый)	Fuchsia или#FF00FF (фиолетовый)
Olive или #808000 (оливковый)	Teal или #008080 (темная морская волна)	Purple или #800080 (пурпурный)	White или #FFFFFF (белый)

д. Выравнивание текста

Для выравнивания текста используется атрибут **ALIGN**, который может принимать значения: **Center** – выровнять по центру; **Left** – выровнять по левому краю; **Right**- выровнять по правому краю; **Justify** – выровнять по обоим краям (по ширине).

Необходимо отметить, что данный атрибут применяется не с тегом **FONT**, а с тегами форматирования и группировки текста такими как: **<P>**, **<H1>** - **<H6>** и др.

```
<H1 ALIGN = Center> Заголовок1 </H1>
<H1 ALIGN = Left> Подзаголовок1 </H2>
<H1 ALIGN = Right>Параграф№1.</P>
<H1 ALIGN =Justify> Параграф№2. </P>
<H1 ALIGN =Left> Параграф№3. </P>
```

Для отделения блока HTML-документа от остальной части документа или объединения несколько параграфов в группу и придания им общих свойств используется контейнер **<DIV>...</DIV>**. Находящиеся между начальным и конечным тегами текст оформляется как отдельный параграф.

```
<DIV ALIGN = Center>
<P> Параграф№1. </P>
<P> Параграф№2. </P>
</DIV>
<DIV ALIGN = Right>
<P> Параграф№3. </P>
<P> Параграф№4. </P>
</DIV>
```

5 Изображения в HTML

а. Изображения в HTML

Для размещения изображений на Web-страницах используются различные графические форматы, наиболее часто применяются - JPEG и GIF. JPEG используется для фотоизображений и полутоновой графики, живописи, градиентов и изображений с множеством мелких разноцветных деталей.

GIF используется для рисованных изображений, векторной графики, картинок без полутонов, градиентов (переходов от цвета к цвету) и большого количества мелких деталей разного цвета. Помимо того, формат GIF используется для сохранения анимированных картинок и изображений с прозрачными частями.

Для размещения изображений существует специальный тег ****, имеющий обязательный параметр **SRC= «адрес»**.

IMG – тег изображения

SRC= «адрес» - атрибут источника

атрибут1, атрибут2... - атрибуты изображения

В качестве параметра (адрес) может использоваться несколько типов аргументов:

- имя файла с изображением (для изображений, находящихся в текущем каталоге)
- адрес файла с изображением (для изображений, находящихся в других каталогах)
- URL изображения (для изображений, находящихся в сети Интернет или на вашем сервере)

б. Изменение параметров изображений

Для изменения параметров изображения на Web-странице используются дополнительные атрибуты:

HEIGHT и **WIDTH** – определяют ширину и высоту изображения соответственно. Если указанные значения не совпадают с реальным размером изображения, изображение масштабируется (иногда при этом заметно теряется качество).

HSPACE и **VSPACE** – определяют количество свободного пространства (в пикселях) по горизонтали и вертикали вокруг изображения.

BORDER – определяет ширину рамки вокруг изображения в пикселях.

ALT – определяет текст, отображаемый браузером на месте изображения, если браузер не может найти файл с изображением. Кроме того, текст, указанный в параметре ALT, будет отображаться во всплывающей подсказке, появляющейся при наведении указателя мыши на изображение.

в. Выравнивание изображений

Для выравнивания изображений и определения их положения относительно текста применяется атрибут **ALIGN**.

Он может принимать следующие значения:

left или **right** –выравнивает изображение по левому или правому краю документа.

top - выравнивает верхнюю линию текущей текстовой строки относительно верхней кромки изображения.

middle - выравнивает базовую (нижнюю) линию текущей текстовой строки с вертикальным центром изображения.

bottom - выравнивает базовую линию текстовой строки относительно нижней кромки изображения.

Кроме этого для центрирования изображения по горизонтали всего HTML – документа используется тег **<CENTER>**.

<CENTER></CENTER>

г. Изображение в качестве гиперссылки

Изображения можно использовать в качестве гиперссылок. Если тег **** находится внутри контейнера ** ... **, изображение фактически становится кнопкой, при нажатии на которую происходит переход по ссылке.

** **

"File1.htm" – адрес перехода

**** - тег изображения

Кроме того, довольно распространен случай, когда, вместо адреса перехода используют другое изображение. Например, на Web-странице находится маленький рисунок, при активизации которого в окне браузера открывается большой рисунок.

Если в теге **<A>** указать атрибут **TARGET="_blank"**, документ, на который указывает гиперссылка, будет открыт в новом окне.

<H3>Образец сертификата</H3>

6. Гиперссылки

а. Что такое гипертекст?

Гипертекст – способ структурирования документов путем размещения ссылок внутри одного документа или между документами. В отличие от обыкновенного текста гипертекст позволяет осуществлять мгновенный переход от одного фрагмента текста к другому. Системы помощи многих популярных программных продуктов устроены именно по гипертекстовому принципу. При нажатии левой кнопкой мыши на некоторый выделенный фрагмент – гиперссылку текущего документа – происходит переход к некоторому заранее назначенному документу или фрагменту документа.

В HTML переход по гиперссылке задается с помощью тега вида:

ссылка

"[адрес]" – адрес перехода;

ссылка - текст, активизация которого вызывает переход.

В качестве параметра [адреса перехода] может использоваться несколько типов аргументов:

- адрес документа
- адрес сайта сети Интернет или внешнего сервера
- адрес электронной почты
- адрес перехода к определенному месту внутри документа.

б. Гиперссылки на документ и сайты Интернет

Для перехода от одного документа к другому в качестве адреса перехода задается путь к файлу:

Перейти к документу1.

Причем, не обязательно это должен быть HTML-документ (например, это могут быть документы Microsoft Office или др.) Такой фрагмент HTML-текста приведет к появлению в документе выделенного фрагмента Перейти к документу1, при нажатии на который в текущее окно будет загружен файл Документ1.htm.

Если файл, на который указывает ссылка, и файл с самой ссылкой расположены в разных директориях, то создается относительная ссылка, в которой прописывается часть пути между директорией с исходным документом и файлом-назначением.

Перейти к документу1.

Чтобы создать ссылку на файл во внешнем каталоге относительно текущего, т. е. расположенном уровнем выше, используется горизонтальное двоеточие **"../пример.htm"**.

Если необходимо сделать ссылку на какой-либо сайт сети Интернет, внешний сервер или на какой-либо другой ресурс, то в качестве адреса перехода используется URL-строка:

[протокол][адрес][порт][путь][имя файла].

в. Гиперссылки на почтовый адрес

Существует возможность использовать гиперссылку для вызова почтовой программы. Для этого используется ссылка вида:

Существует возможность использовать гиперссылку для вызова почтовой программы. Для этого используются ссылка вида: `Послать письмо `, где: user@mail.ru - адрес электронной почты.

г. Переходы внутри документа

При необходимости можно задать переход не просто к некоторому документу, но и к определенному месту внутри документа. Для этого необходимо создать в документе некоторую опорную точку или анкер. Это тег `<A>` специального вида, в котором используется атрибут NAME. Допустим, что необходимо осуществить переход к слову "Параграф 1".

Прежде всего необходимо создать анкер, находящийся в месте расположения требуемого слова и «охватывающего» его:

` Параграф 1` где : **Метка 1**- имя анкера.

Слово "Параграф 1" при этом никак не будет выделен в тексте документа браузером. Затем необходимо определить ссылку для перехода на этот анкер: `< A HREF="#"Метка 1">Переход к слову – Параграф 1 `.

При активизации этой ссылки браузер переместит изображение страницы так, чтобы текст «Анкер1» оказался в верхней видимой строке окна.

7 Логическая и физическая структура сайта

Каждый ресурс Интернета, от любительской домашней странички до большого информационного портала, содержит несколько тематических рубрик, соединенных между собой гиперсвязями. Как правило, ссылки на все разделы сайта с краткими анонсами их содержимого приводятся на первой, так называемой стартовой странице, которой присваивается имя index.htm (.html). Если тематические рубрики содержат собственные подразделы, каждая из них также имеет свою стартовую страницу, называющуюся index.html.

Такое имя файла рекомендуется присваивать всем стартовым документам сайта, поскольку в противном случае при обращении к какому-либо разделу посредством сокращенного URL без указания названия стартовой страницы (например, `http://www.mysite.ru/photos/` вместо `http://www.mysite.ru/photos/startpage.html`) браузер отобразит не саму web-страницу, а перечень хранящихся в данной папке файлов.

Подобный набор тематических рубрик с распределенными по соответствующим разделам документами и заранее спроектированными гиперсвязями между всеми страницами ресурса и называется логической структурой сайта. Физическая структура, напротив, подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой

опубликован ваш сайт. Пример сравнения логической и физической структур одного и того же ресурса Интернета показан на рис. 1.

Очевидно, что логическая и физическая структуры могут не совпадать, поскольку в общем случае физическая структура ресурса разрабатывается, исходя из удобства размещения файлов. Однако более или менее точное сохранение порядка следования логических разделов в физической структуре сайта позволит вам избежать путаницы при последующем дополнении и обновлении материалов.

Рекомендуется размещать все графические изображения, являющиеся элементами проекта, в отдельной папке с названием «Images», расположенной в корневой директории сайта. Такой подход позволит обновлять хранящиеся в других тематических разделах документы HTML без переноса графики, использовать одни и те же графические файлы во всех разделах сайта и при необходимости удалять целые директории.

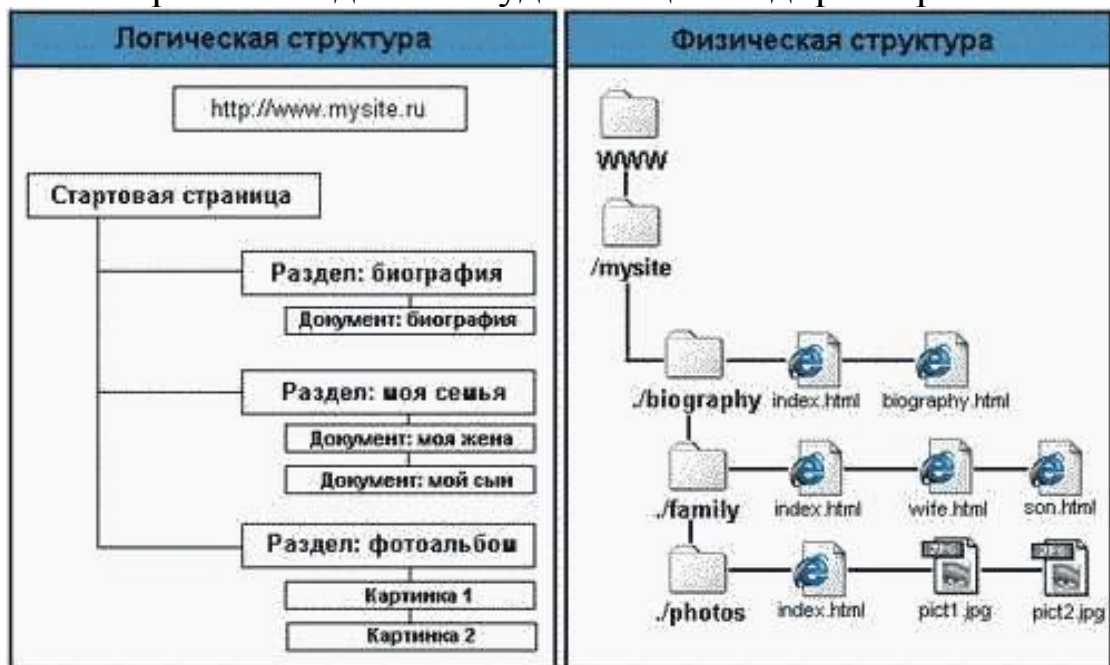


Рис. 1. Сравнение логической и физической структуры сайта

Для того чтобы все гиперссылки на вашей домашней страничке или web-сайте работали корректно, все документы открывались правильно и браузер не выдавал ошибок при обращении к каким-либо разделам ресурса, при создании его физической структуры следует соблюдать несколько простых правил.

Назначайте имена директорий, имена и расширения документов HTML и графических файлов с использованием символов только латинского алфавита и только в строчном регистре. Старайтесь, чтобы имена созданных вами файлов и директорий не превышали по длине восьми символов.

При присвоении имен файлов документам HTML старайтесь следить за тем, чтобы эти имена были «смысловыми»: впоследствии вы легко можете забыть содержимое и назначение какой-либо web-страницы, если имена файлов будут выглядеть, например, как 1.htm, 2.htm, 3.htm и т. д.

Физическая структура сайта скрыта от посетителей вашего ресурса: они могут наблюдать только логическую структуру, причем именно так, как она представлена при помощи элементов навигации. Отсюда следует вполне логический вывод: строение системы навигации должно если не полностью повторять, то хотя бы максимально соответствовать разработанной вами логической структуре сайта.

При выполнении лабораторных работ рекомендуется использовать следующую структуру папок:

Проектирование структуры каталогов



8 Таблицы

а. Оформление таблиц

HTML представляет большие возможности оформления таблиц. Оформление может выполняться как с помощью атрибутов основных тегов таблицы, так и с помощью специальных тегов.

Тег **<TABLE>** позволяет использовать следующие основные атрибуты:

BORDER – определяет ширину внешней рамки таблицы (в пикселях).

CELLPADDING – определяет расстояние (в пикселях) между рамкой каждой ячейки таблицы и содержащимся в ней материалом.

CELLSPACING – определяет расстояние (в пикселях) между границами соседних ячеек.

ALIGN – определяет способ горизонтального выравнивания таблицы. Возможные значения: **left** – по левому краю, **center** – по центру, **right** – по правому краю. Значение по умолчанию – left.

WIDTH и **HEIGHT** – определяет ширину и высоту таблицы, значение задается либо в пикселях, либо в процентном отношении к ширине окна браузера.

BGCOLOR – определяет цвет фона ячеек таблицы. Он задается либо RGB-значением в шестнадцатеричной системе, либо одним из 16 базовых цветов.

<CAPTION> – специальный парный тег для описания заголовка таблицы. Атрибут этого тега **ALIGN** определяет место размещения заголовка. Его значение **top** помещает заголовок над таблицей, а **bottom** – под таблицей.

```
<TABLE BORDER=15 WIDTH=400 HEIGHT=200 ALIGN=CENTER  
BGCOLOR=Yellow>
```

```
<CAPTION ALIGN="bottom"> Оформленная таблица</CAPTION>
```

```
<TR><TD>Ячейка1.1</TD><TD>Ячейка 1.2</TD></TR>
```

```
<TR><TD>Ячейка2.1</TD><TD>Ячейка 2.2</TD></TR>
```

```
<TR><TD>Ячейка3.1</TD><TD>Ячейка 3.2</TD></TR>
```

```
</TABLE>
```

При использовании темного фона таблицы, можно изменить цвет текста в ячейках, например:

```
<FONT COLOR=WHITE>Текст... </FONT>.
```

Необходимо иметь в виду, что тег **...** должен находиться внутри тегов изменяемой ячейки таблицы.

6. Сложные таблицы

Размещая внутри ячеек одной таблицы теги других таблиц или объединяя соседние ячейки, можно создавать сложные таблицы.

Кроме тега **<TD>** (Table Data) для создания ячейки можно применять тег **<TH>** (Table Head), который также создает ячейку, но определяет ее как заголовок с увеличенным шрифтом.

Для оформления содержимого ячеек можно использовать следующие атрибуты: **ALIGN**, **WIDTH**, **HEIGHT**, **BGCOLOR**, которые определяют способ горизонтального выравнивания содержимого ячейки, её ширину, высоту и цвет соответственно.

VALIGN - определяет способ вертикального выравнивания содержимого ячейки. Возможные значения: **top** - по верхнему краю, **bottom** - по нижнему краю, **middle** - по центру.

COLSPAN - определяет кол-во столбцов, на которые простирается данная ячейка (ширина ячейки в столбцах). По умолчанию имеет значение 1.

ROWSPAN - определяет количество рядов, на которые простирается данная ячейка (высота ячейки в строках). По умолчанию имеет значение 1.

```
<TABLE BORDER=15 ALIGN=CENTER BGCOLOR=Yellow>
```

```
<!--строка 1 большой таблицы-->
```

```
<TR> <TD>
```

```
<TABLE BORDER=10 WIDTH=300 HEIGHT=100> <TR>
```

```
<TH ROWSPAN=2>Высокая ячейка</TH><TD>Ячейка</TD>
```



```

<TD>Ячейка</TD> </TR><TR><TD>Ячейка</TD>
<TD>Ячейка</TD> </TR> <TABLE></TD> </TR>
<!--Строка 2 большой таблицы-->
<TR> <TD>
<TABLE BORDER=10 WIDTH=300 HEIGHT=100>
<TR> <TH COLSPAN=2> Широкая ячейка </TH> </TR>
<TR> <TD> Ячейка </TD> <TD> Ячейка </TD> </TR> <TR>
<TD> Ячейка <TD> Ячейка </TD> </TR></TABLE> </TD>
</TR>
</TABLE>

```

Иногда удобно использовать атрибут NOWRAP, который блокирует автоматический перенос слов в пределах текущей ячейки (текст в ячейке помещается в одну строку).

9. Включение звука в Web-страницу

Звуковой файл включается в Web-страницу путем указания его имени в теге . Не забудьте про закрывающий тег. После данного тега следует привести текст, характеризующий содержание звукового файла. Совсем не плохо указать его размер, чтобы посетитель мог решить, стоит ли его загружать. Тип данного файла wav. (имеется в виду расширение файла).

Пример 1: Воспроизведем рычание тигра. Имя звукового файла: tiger.wav
Его размер: 29,740 bytes.

Текст ссылки звукового файла: Рычание тигра.

Запись в блокноте: Рычание тигра. (29,740 bytes)

Для прослушивания файла необходимо щелкнуть мышью по тексту ссылки. Автоматически загружается универсальный проигрыватель, и воспроизводится звук.

Пример 2: Использование графики.

Можно использовать графику с пояснительным текстом для прослушивания файла.

Запись в блокноте:

 Рычание тигра
(29,740 bytes)

10. Включение видео в Web-страницу

Видеофайл включается в HTML-документ точно так же, как звуковой файл. Единственное различие, что теперь тип файла - MOV или AVI.

Пример 1:

 Вид горной долины.(1,600 bytes)

Так же можно использовать графику с пояснительным текстом.

Пример 2:

Вид горной долины.(1,600 bytes)

11 &-последовательности

В HTML имеется возможность вывести на экран специальные или зарезервированные символы, это делается с помощью &-последовательностей (их еще называют символьными объектами или эскейп-последовательностями). Браузер показывает на экране символ «<», когда встречается в тексте **последовательность** **<** (по первым буквам английских слов less than- меньше, чем).

Ниже приведено несколько часто используемых &-последовательностей

&larr	Стрелка влево	&cent	Цент
&uarr	Стрелка вверх	&pound	Фунт стерлингов
&rarr	Стрелка вправо	&curren	Денежная единица
&darr	Стрелка вниз	&yen	Иена или юань
&harr	Стрелка влево-вправо	&brvbar	Разорванная вертикальная черта
&iexcl	Перевернутый восклицательный знак	&sect	параграф
&gt	«>»	&amp	«&»
&quot	(")	&nbspsp	неразрывный пробел

12 Режимы браузеров

Во время противостояния браузеров Internet Explorer и Netscape каждый из разработчиков старался улучшить свой продукт. Netscape 4 и IE4 плохо поддерживали веб-стандарты, поэтому следующая версия, IE5 должна была не только исправить ошибки IE4, но и показать улучшенную поддержку спецификации CSS. Это было необходимо еще и по политическим мотивам, поскольку компания Microsoft вошла в группу W3C и начала оказывать сильное влияние на разработку HTML и CSS.

В процессе работы над браузером IE5 его разработчики столкнулись с неожиданной трудностью. Разница при отображении страницы в разных версиях браузера была настолько велика, что множество сайтов оказались бы неработоспособными при просмотре в IE5. Идея сделать кнопку для переключения в режим совместимости пришла только в версии 8.0, поэтому разработчики IE5 пошли другим путём. Все старые страницы отображались по старым правилам, а для включения режима поддержки стандартов в код страницы необходимо добавить элемент **<!DOCTYPE>** (доктайп).

Кроме переключения браузера в один из режимов, доктайп также сообщает, согласно каким правилам синтаксиса проводить проверку текущего документа. К примеру, для HTML 4.01 и XHTML 1.0 существует по три разных типа доктайпа, для HTML5 лишь один.

А) Почти стандартный режим

Этот режим основан на стандартном режиме с некоторыми исключениями: отображение изображений внутри ячеек таблицы и рисунков друг под другом происходит как в режиме совместимости.

Для переключения в почти стандартный режим применяется один из следующих доктайпов.

Для HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

Для XHTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
```

Б) Строгий доктайп

Применяется строгий синтаксис языка, допускается включать все теги и атрибуты, кроме осуждаемых.

Для HTML 4.01:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Для XHTML 1.0:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- Осуждается использование следующих тегов:

<applet>, <basefont>, <center>, <dir>, , <isindex>, <noframes>, <plaintext>, <s>, <strike>, <u>, <xmp>. Взамен по возможности рекомендуется использовать стили.

- Текст, изображения и элементы форм нельзя напрямую добавлять в <body>, эти элементы должны обязательно находиться внутри блочных элементов вроде <p> или <div>.

- Осуждается применение атрибутов target, start (тег), type (теги , ,) и др.

Естественно, также должен соблюдаться синтаксис языка — правильное вложение тегов, закрытие тегов, должны присутствовать обязательные теги и др. Документ с неявными ошибками приведён в примере 1

Пример 1. Невалидный код HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
<html>  
<head>  
<title>Заголовок</title>  
</head>  
<body>
```

```
<form>
  <input type="text">
</form>
</body></html>
```

В данном примере у тега `<form>` нет обязательного атрибута `action`, тег `<input>` располагается напрямую внутри формы, хотя должен быть заключён в блочный тег. Также не указана кодировка через метатег. Это не является ошибкой, но пользователям в некоторых случаях придется самостоятельно указывать её в браузере. Валидный код XHTML с учётом исправленных ошибок при строгом доктайпе продемонстрирован в примере 2.

Пример 2. Валидный код XHTML

```
<!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" content="text/html; charset=utf-8" />
    <title>Заголовок</title>
  </head>
  <body>
    <form action="">
      <p><input type="text" /></p>
    </form>
  </body>
</html>
```

В) Переходный доктайп

Применяется «мягкий» синтаксис языка, допускается использовать все теги и атрибуты, включая осуждаемые.

Для HTML 4.01:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Для XHTML 1.0:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Цель переходного доктайпа заключается в постепенном знакомстве с синтаксисом языка.

12 Режимы Internet Explorer

Вокруг Internet Explorer сложилась ситуация, которая больше не прослеживается ни с одним другим браузером — разброс версий начинается с 6.0 и заканчивается 9.0.

Сама компания Microsoft всячески поощряет переход на новые версии Internet Explorer и в 2010 году прекратила поддержку IE6 и IE7.

Что касается разработчиков сайтов, то для них такой большой набор версий является настоящей проблемой. Каждая версия IE содержит свои уникальные ошибки, особенности отображения веб-страниц, а также не поддерживает какие-то свойства CSS. Одним из радикальных вариантов решения проблемы является полный отказ разработчика от поддержки определенных версий IE.

В итоге было принято решение добавить режим представления совместимости; для быстрого переключения сайтов в этот режим возле адресной строки добавлена специальная кнопка (рис. 4а). В IE9 пошли еще дальше и в нём уже можно переключаться на IE8 или IE7 (рис. 4б)



Рис. 4. Кнопка для переключения в режим совместимости
а — в IE8, б — в IE9

Начиная с версии 8.0, переключение режимов в браузере делается через Средства разработчика (Сервис > Средства разработчика), которое проще вызвать при нажатии на клавишу F12. В этом инструменте доступно два пункта меню связанных с режимами: Режим браузера (рис. 5) и Режим документов.

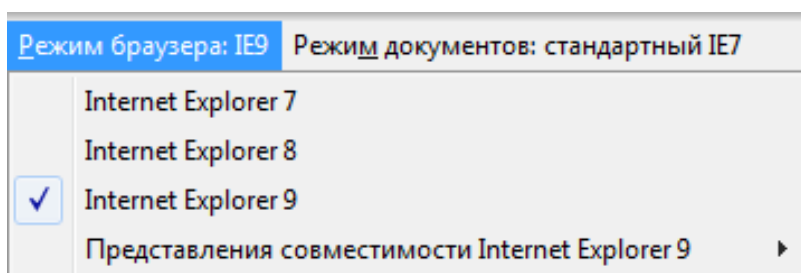


Рис. 5. Режимы браузера

Выбор режима браузера определяет следующее:

- строка User-Agent, которую браузер отправляет на сервер, в Microsoft называется «агент пользователя»; Значение User-Agent определяет не только браузер, но также поисковых пауков и сетевых роботов.
- версия для условных комментариев, которую в Microsoft называют «вектор версии»;
- режим документа.

Вектор версии позволяет с помощью условных комментариев определять версию IE и отдавать для неё отдельный код. Условные комментарии активно применяются в вёрстке для устранения различий в макете между IE и другими браузерами. Таким образом, вы можете ознакомиться с небольшим примером, который показывает текст только в IE8. Остальные браузеры игнорируют этот фрагмент, считая его комментарием.

```
<!--[if IE 8]>  
<p>У вас браузер IE8.</p>  
<![endif]-->
```

Кроме переключения непосредственно в браузере, задать режим можно через тег `<meta>`, отправив HTTP-заголовок X-UA-Compatible.

```
<meta http-equiv="X-UA-Compatible" content="IE=версия" />
```

Версия может принимать следующие значения (табл. 2).

Табл. 2. Режимы документа IE

Значение	Описание
5	Вынужденное переключение в режим IE5, доктайп при этом игнорируется.
7	Вынужденное переключение в режим IE7, доктайп при этом игнорируется.
8	Вынужденное переключение в режим IE8, доктайп при этом игнорируется.
9	Вынужденное переключение в режим IE9, доктайп при этом игнорируется.
EmulateIE7	При наличии доктайпа переключается в режим IE7, в противном случае в режим совместимости.
EmulateIE8	При наличии доктайпа переключается в режим IE8, в противном случае в режим совместимости.
EmulateIE9	При наличии доктайпа переключается в режим IE9, в противном случае в режим совместимости.
Edge	Устанавливает документ в наиболее новый доступный режим. Для версии 8.0 это режим IE8, для версии 9.0 это режим IE9.

К примеру, на сайте Яндекса применяется следующий код для эмуляции режима IE7

```
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7">
```


13 Графика в web-дизайне

Графические компоненты web-страниц по назначению можно условно разделить на три обширные категории: иллюстративная графика, включая дополняющие текст фотографии, пояснительные рисунки, чертежи и схемы, функциональная графика, представляющая собой элементы управления сайтом (кнопки навигации, счетчики и элементы интерактивных форм) и, наконец, декоративная графика — элементы дизайна страницы, включенные в нее «для красоты» и не несущие информационной нагрузки (фоновые рисунки, линии-разделители, выполненные в виде графических файлов заголовки и многое другое).

А) Сетевые графические форматы

Как говорили И. Ильф и Е. Петров, «статистика знает все». Эта самая всезнающая статистика утверждает, что большинство пользователей Интернета на нашей планете применяют для связи со Всемирной сетью соединение по коммутируемым телефонным каналам при помощи модема. Поскольку это весьма медленный способ связи, время загрузки графического изображения в клиентский браузер в данном случае должно быть как можно меньшим. Именно поэтому два наиболее популярных стандарта, в которых хранится 90 % всей графики, представленной ныне в Интернете, это GIF и JPEG. В них заложены различные алгоритмы сжатия изображения с потерей качества, благодаря использованию которых удастся значительно уменьшить размер целевого файла.

GIF - В далеком 1978 году двое израильских исследователей Якоб Зив (Jacob Ziv) и Абрахам Лемпел (Abraham Lempel) разработали принципиально новый для того времени алгоритм сжатия информации без потери данных, которому дали полученное из сокращения собственных фамилий и даты завершения своего проекта название: LZ78. Информация о принципах построения этого алгоритма была общедоступной, и спустя несколько лет американский программист Терри Уэлч (Terry Welch) усовершенствовал его, добавил в обозначение первую букву своей фамилии и запатентовал новый алгоритм под названием LZW, также предоставив свою разработку для использования всем желающим.

Одним из таких «желающих» оказался сотрудник компании CompuServe Inc. Боб Берри (Bob Berry), взявший LZW в качестве основы для созданного им в 1987 году принципиально нового графического формата GIF (Graphic Interchange Format). Созданная Терри Уэлчем компания Unisys, которой и принадлежали авторские права на алгоритм LZW, взимала плату за его использование только с производителей аппаратного обеспечения для компьютеров, в котором применялся данный стандарт, например, с изготовителей модемов. Разработчики программного обеспечения «комиссионными сборами» не облагались.

Однако зимой 1994 года компания Unisys, начавшая испытывать финансовые проблемы, объявила LZW коммерческим стандартом, использование которого требует оплаты. Это автоматически сделало GIF единственным в мире «платным» графическим форматом, что вызвало волну недовольства среди пользователей Интернета, поскольку практически на всех современных web-сайтах так или иначе применяются элементы GIF. Тем не менее GIF чрезвычайно широко используется в Интернете и сейчас, причем пользователи не обязаны оплачивать кому бы то ни было возможность разместить на своей страничке изображение в данном формате, так как упомянутые выше финансовые претензии касаются, в первую очередь, производителей работающего с GIF программного обеспечения. Ситуация с дальнейшей судьбой этого стандарта до сих пор остается не разрешенной.

Благодаря возможностям алгоритма LZW стандарт GIF позволяет значительно сокращать объем итогового графического файла по сравнению с исходным изображением. Достигается это методом смешения сходных оттенков в один. Если, например, в составе рисунка имеется участок, состоящий из нескольких сходных полутонов, к примеру, голубого, светло-голубого и темно-голубого цвета, они будут кодированы одним оттенком — голубым. Информация об изображении в файле стандарта GIF записывается построчно, то есть представляет собой массив описаний строк высотой в один пиксел. Именно это свойство GIF, а также то, что данный формат оперирует фиксированной, так называемой индексированной палитрой, причем число цветов в этой палитре не превышает 256, явилось основой для появления двух простых правил, применяющихся в современном web-дизайне.

Стандарт GIF используется в документах HTML только для отображения так называемой бизнес-графики: диаграмм, логотипов, кнопок, разделительных линий, других элементов оформления страницы. Для размещения на web-сайте фотографий, репродукций картин и изображений с большим количеством цветов и цветовых переходов используется стандарт JPEG.

В упрощенном виде данный закон «web-мастерства» можно сформулировать так: если изображение рисованное, его следует представлять в стандарте GIF, во всех остальных случаях лучше воспользоваться JPEG.

Подготавливая рисунок для сохранения его в формате GIF, необходимо избегать следующих художественных приемов: градиентных заливок, размытий, постепенных цветовых переходов с множеством оттенков, а также графических фильтров, обеспечивающих неравномерное смешение нескольких цветов на одном участке изображения, например,

эффектов изменения интенсивности освещения, подобных фильтру «блик» редактора Adobe Photoshop.

Это правило продиктовано тем, что алгоритм замещения схожих оттенков одним в формате GIF далеко не всегда работает корректно. Правильнее было бы, наверное, сказать «работает некорректно почти всегда». Поэтому участки со множеством различных оттенков на небольшом физическом пространстве рисунка после сохранения изображения в индексированной палитре будут выглядеть смазанными и «грязными». Этого можно избежать, применяя в своей иллюстрации по возможности однотонные и контрастные цвета.

Одно из замечательных свойств стандарта GIF — его уникальная особенность, названная разработчиками «interlace». Она позволяет загружать картинку с сервера в клиентский браузер не целиком, а частями, причем процедура считывания файла выглядит следующим образом: сначала на экране отображаются первая, пятая и десятая строки, составляющие изображение, затем — вторая, шестая и одиннадцатая и т. д. Таким образом, для пользователя создается иллюзия постепенной загрузки графического элемента: картинка как бы медленно «проявляется» на странице, что иногда бывает очень полезно при включении в документ изображений большого размера — психологически зрителю легче дождаться полной прорисовки иллюстрации, чем несколько минут скучать перед пустым экраном. К тому же еще до полной загрузки файла пользователь может оценить приблизительное содержимое картинки и решить, стоит ли ему дожидаться ее полной прорисовки или нет.

Через несколько лет после создания стандарта GIF, в 1989 году, компания CompuServe выпустила новую версию этого графического формата, получившую название GIF89a. Данная модификация включает в себя еще две уникальные особенности, широко используемые в современном Интернете. Первая называется «transparency» и подразумевает создание для изображения прозрачного фона методом сохранения вместе с файлом так называемого альфа-канала, представляющего собой маску прозрачности рисунка. Цвета, помеченные в альфа-канале как прозрачные, станут невидимы в браузерах и большинстве других программ, предназначенных для просмотра изображений. Эта функция необходима, например, при размещении картинок неправильной геометрической формы на странице со сложным фоновым рисунком, когда корректно «подогнать» части изображений друг к другу не представляется возможным.

Убрать ненужный фон из файла GIF можно, воспользовавшись встроенной процедурой графического редактора Adobe Photoshop. Для этого в меню «файл» необходимо выбрать пункт «экспорт», среди предложенных вариантов отметить «GIF89a» и в появившемся окне с помощью инструмента «пипетка» указать цвета, в которых вы больше не нуждаетесь.

Другая полезная особенность стандарта GIF89a заключается том, что этот формат позволяет сохранять в файле с одним физическим заголовком несколько разных изображений, демонстрируя их на экране последовательно одно за другим, причем с возможностью специфицировать порядок их чередования и временной интервал между сменами кадра. Именно на этом принципе построена весьма распространенная в Интернете GIF-анимация.

JPEG

JPEG (Joint Photographic Experts Group) — графический стандарт, созданный на основе одноименного алгоритма сжатия изображений с потерей качества, кодирующего не идентичные элементы, как алгоритм LZW, а межпиксельные интервалы. В упрощенном виде механизм сжатия изображения в файл формата JPEG выглядит следующим образом. Первой ступенью компрессии является преобразование изображения в цветовой образ LAB, раскладывающий картинку на три независимых канала, один из которых (Lightness) выделен для сохранения значений интенсивности цветов, а два других (A и B) — для запоминания непосредственно цветовой информации. Причем данные о цветах сохраняются в виде шкалы, организованной по принципу непрерывного спектра. Вторая ступень — собственно компрессия: из получившейся цветовой модели удаляются приблизительно три четверти информации о цвете, затем образ дробится на участки размером 8x8 точек и преобразуется в числовой массив данных. Заголовок каждого блока описывает доминирующий цвет участка, остальная информация — менее заметные оттенки. На третьей ступени сжатия из массива данных удаляется определенная часть информации, описывающей второстепенные оттенки, причем количество изымаемых данных зависит от выбранного пользователем качества результирующего изображения. И наконец, готовый файл сжимается согласно алгоритму Хаффмана, который предусматривает замену наиболее часто встречающихся в массиве данных знаков более компактной двух битной кодировкой. Декомпрессия файла JPEG происходит в обратном порядке.

Из всего сказанного можно сделать вполне очевидное заключение: JPEG оптимален для передачи фотографических изображений, а также картинок с большим количеством полутонов и цветовых переходов. Максимальное число цветов, которое может содержать изображение в формате JPEG, достигает 16 миллионов. Очевидно также, что чем выше степень компрессии такого изображения, тем ниже его качество.

При изготовлении иллюстраций в формате JPEG рекомендуется избегать использования больших участков, заполненных одним цветом, во избежание появления на изображении постороннего цветового «шума» и «грязи».

Несоблюдение этого правила может повлечь за собой ряд неприятных последствий: картинка в формате JPEG, содержащая достаточно

большое пространство, заполненное одним цветом, будет выглядеть на экране компьютера «грязно» и неопрятно, фон сохраненного в JPEG логотипа, «подогнанный» под фоновый цвет html-документа, «поплывет» при изменении свойств экранной палитры.

Существует несколько разновидностей стандарта JPEG. Среди них — формат JPEG Baseline Optimized, основанный на более совершенном алгоритме компрессии изображений. Недостатком данной реализации JPEG является то, что она не распознается целым рядом часто используемых приложений. Так называемый Progressive JPEG был оптимизирован специально для представления графики во Всемирной сети, изображения в этом формате сжимаются чуть лучше, чем в стандартном JPEG, но хуже, чем в JPEG Baseline Optimized. Отличительная особенность Progressive JPEG — возможность сохранять графику в чересстрочном режиме, как это реализовано в стандарте GIF.

При работе со стандартом JPEG следует учитывать, что сжатие с наименьшей потерей качества поддаются изображения, сохраненные с высоким разрешением, от 150 до 300 dpi. В процессе обработки иллюстраций лучше всего сохранять картинку в формате TIFF и переводить в JPEG лишь готовый результат, поскольку каждое промежуточное сохранение файла JPEG на диск приводит к удалению второстепенной информации и ухудшению качества рисунка.

PNG

Описанная чуть ранее «криминальная история» с неожиданным изменением статуса алгоритма сжатия LZW с бесплатного на платный спровоцировала стремление пользователей избавиться от формата GIF. С этой целью во второй половине девяностых годов была создана инициативная группа программистов и исследователей, которую возглавил американский специалист по электронным технологиям Том Боутелл (Thomas Bowtell). Перед инициативной группой стояла нелегкая задача: разработать стандарт, который не только включал бы в себя все лучшие качества GIF, но и превосходил бы его по всем характеристикам, оставаясь при этом совершенно бесплатным для пользователей. Основной областью применения данного формата должен был стать Интернет. Такой стандарт был создан и получил название Portable Network Graphics — PNG, хотя многие пользователи Всемирной сети придумали этой аббревиатуре иную расшифровку: Picture is Not GIF.

PNG, как и GIF, поддерживает interlacing (чересстрочность), но в отличие от последнего не только по горизонтали, но и по вертикали. Палитра PNG не ограничивается 256 цветами, данный формат позволяет создавать изображения с глубиной цвета до 48 бит. Весьма любопытные изменения внесены в режим прозрачности графики: PNG также поддерживает альфа-канал, по состоит он не из одного уровня, как альфа-канал файлов GIF, а из

254 уровней для каждого элемента изображения, с диапазоном прозрачности от 0 до 99 %. Благодаря этому дизайнер получил возможность создавать картинки с переменной прозрачностью, например, логотипы, фон которых плавно сливается с фоновым цветом html-документа, каким бы тот ни был.

Помимо альфа-канала изображения PNG содержат так называемый блок описания гамма-коррекции. Под переменной «гамма» в данном случае подразумевается некая величина, выражающая зависимость яркости свечения каждой точки экрана пользователя от напряжения на электродах кинескопа, которое берется из файлов управления видеоподсистемой компьютера. Таким образом, при открытии изображения, созданного с помощью IBM-совместимой машины, на компьютере другого производителя, например, на Apple Macintosh, в яркость цветопередачи вводится соответствующая поправка, и картинка отображается совершенно идентично на разных типах оборудования. Этим стандарт PNG еще более приблизился к основной идее Интернета единообразию отображения web-страниц на компьютерах от различных производителей.