
Лекция _3

Введение в программирование на языке гипертекстовой разметки HTML

-
1. Версии HTML. *Содержание*
 2. Введение в HTML.
 3. Инструментарий.
 4. Теги.
 5. Структура HTML-кода.

-
1. Теги верхнего **Топы тегов** уровня
 2. Теги заголовка документа.
 3. Блочные элементы
 4. Встроенные элементы.
 5. Универсальные элементы.
 6. Списки.
 7. Таблицы.
 8. Фреймы.

1. Цвет.

2. Шестнадцатеричные цвета. *Значения параметров у тегов*

3. Веб-цвета

4. Размер.

5. Адрес.

6. Текст.

- Особенности текста в HTML.
- Абзацы.
- Заголовки
- Выравнивание текста
- Начертания: жирное, Курсивное
- Верхний и нижний индекс

Ссылки

- Абсолютные и относительные ссылки.
- Виды ссылок.
- Правила вложений для тега <A>
- Параметры ссылок.
- Ссылка на адрес электронной почты.

Якоря

Изображения

- Формат GIF.
- Формат JPEG.
- Формат PNG-8 и Формат PNG-24
- Добавление рисунка .

Изображения

- Альтернативный текст.
- Изменение размеров рисунка.

Списки

- Маркированный список.
- Нумерованный список. Нумерация списка
- Список определений.

Таблицы

- Создание таблицы.
- Параметры тега <TABLE>.
- Параметры тега <td>.
- Особенности таблиц

Таблицы

- Выравнивание таблиц.
- Объединение ячеек.
- Вложенные таблицы.
- Заголовок таблицы.
- Достоинства и недостатки фреймов.
- Создание фреймов.
- Ссылки во фреймах.
- Границы между фреймами.
- Изменение размеров фреймов.
- Полосы прокрутки.
- Плавающие фреймы.

Версии HTML

RFC 1866 — HTML 2.0, одобренный как стандарт 22 сентября 1995 г.;

HTML 3.2 — 14 января 1997 г.;

HTML 4.0 — 18 декабря 1997 г.;

HTML 4.01 (значительные изменения) — 24 декабря 1999 г.;

ISO/IEC 15445:2000 (так называемый ISO HTML, основан на HTML 4.01 Strict) — 15 мая 2000 г.

HTML 5 — в разработке.

- Request For Comments

Версии HTML

Официальной спецификации HTML 1.0 не существует. До 1995 года существовало множество неофициальных стандартов HTML. Чтобы стандартная версия отличалась от них, ей сразу присвоили второй номер.

Версия 3 была предложена Консорциумом всемирной паутины (W3C) в марте 1995 года и обеспечивала много новых возможностей, таких как создание таблиц, «обтекание» изображений текстом и отображение сложных математических формул. Этот стандарт был совместим с версией 2.0, но его реализация была сложна для браузеров того времени. Версия 3.1 официально никогда не предлагалась, и следующей версией стандарта HTML стала 3.2, в которой были опущены многие нововведения версии 3.0, но добавлены нестандартные элементы, поддерживаемые браузерами «Netscape» и «Mosaic».

HTML — это не язык программирования в общепринятом смысле, а язык гипертекстовой разметки документа.

С его помощью на Web-странице можно разместить текст, графические изображения, звуковые и видеофайлы - практически любую электронную информацию.

Текст на языке *HTML* интерпретируется браузером и отображается в виде документа, в удобной для человека форме.

С помощью *HTML* должным образом форматируют Web-страницы и устанавливают между ними связь.

При этом не имеет значения, на какой из компьютерных платформ созданы файлы, которые использованы на

Появление гипертекста

В середине 1960-х Тед Нельсон создал слово "**гипертекст**" для описания системы непоследовательных ссылок внутри текста.

Идея состояла в том, чтобы перемещаться по текстовым ссылкам, не читая при этом материал в линейной последовательности.

Фрагмент информации в одном месте будет вести к родственной информации в другом месте через цепочку ссылок, чтобы собрать сведения из различных источников, разбросанных по множеству документов.

Появление гипертекста

И только пятнадцать лет спустя Тим Бернерс-Ли, работавший консультантом в Европейском центре ядерных исследований (CERN), написал программу с названием "Enquire-Within-Upon-Everything" ("Задай любой вопрос"), которая позволяла создавать ссылки между произвольными узлами текста в документе.

Каждый узел имел заглавие-идентификатор и список двунаправленных ссылок, поэтому читатели могли перемещаться из одного раздела документа в другой, активируя текстовые ссылки.

В 1990 Бернерс-Ли начал работу над гипертекстовым "**браузером**". Он придумал термин "WorldWideWeb" ("Всемирная паутина") для названия программы и "World Wide Web" - для названия проекта.

Проект WWW был первоначально разработан для создания распределенной гипермедийной системы, которая была легко доступна из любого настольного компьютера, и для согласования физических исследований, разбросанных по всему миру.

Web содержала стандартные форматы для текста, графики, звука и видео, которые легко индексировались и были доступны для поиска всем сетевым машинам.

Были предложены стандарты для:

Единообразного локатора ресурса (URL - Uniform Resource Locator).

Он является схемой адресации Web.

Протокола передачи гипертекста (HTTP - HyperText Transfer Protocol).

Он является множеством сетевых правил для передачи Web-страниц.

Языка разметки гипертекста (HTML - HyperText Markup Language).

Внешний вид HTML документа (например, заголовок, тело документа и колонтитул) определяется специальными его элементами, которые должным образом интерпретирует Web-браузер. Эти элементы стандартизированы *Консорциумом World Wide Web* (World Wide Web Consortium — W3C) и называются *дескрипторами (или тегами)*.

Дескрипторы (теги) — это элементы, которыми определяются внешний вид и "поведение" Web-страницы.

Как правило, внешний вид и расположение элементов Web-страницы (текст, изображение и т.д.) определяются парой дескрипторов, между которыми находятся эти

простейший текстовый редактор. Но существуют и
Размещение файлов Web-страниц
специализированные средства. Среди них Smart Web Builder -
html-редактор с удобным интерфейсом и широкими
возможностями

(<http://www.softportal.com/software-4614-smart-web-builder.html>).

Чтобы другие пользователи Internet могли по оценить ваши Web-страницы, файлы этих страниц необходимо разместить на Web-сервере. Место на нем, и подключение к Internet обеспечивают Internet-провайдеры (*JSP — Internet Service Provider*). Есть и другие организации, которые могут предоставить услуги по хранению Ваших Web-страниц на сервере. Они называются *Web Presence Provider (WPP)*.

Основные элементы HTML

HTML и *XHTML* — это языки гипертекстовой разметки, которыми определяется структура текстовых элементов документа;

HTML — язык, не зависящий от типа компьютерной платформы. Независимо от компьютерной платформы — будь то PC, Macintosh или UNIX, — если воспользоваться только Web-браузером, для посетителей вашего Web-узла страницы будут отображаться одинаково;

XHTML — это последняя и более структурированная версия *HTML*.

Основные элементы HTML

HTML — это язык макетирования документов и описания гиперссылок. Однако он не соответствует строгому стандарту *XML*. Поэтому W3C предложил язык *XHTML*, переработав HTML в соответствии с требованиями *XML*. XHTML поддерживает все возможности HTML, применяя более жесткие правила XML. Создавать XHTML-документы ничуть не сложнее чем писать обычную HTML-разметку, а перспективы использования таких документов значительно шире, поскольку к ним оказывается применим весь инструментарий XML.

Поэтому в последнее время наибольшее внимание уделяется именно XHTML, как приоритетному инструменту разметки

Инструментарий

Текстовый редактор

HTML-документ можно создавать в любом текстовом редакторе, хоть Блокноте, но для этой цели подойдет не всякая программа. Нужна такая, чтобы поддерживала следующие возможности:

- подсветка синтаксиса — выделение тегов, текста, ключевых слов и параметров разными цветами. Это облегчает поиск нужного элемента, ускоряет работу разработчика и снижает возникновение ошибок;
- работа с вкладками. Сайт представляет собой набор файлов, которые приходится править по отдельности, для чего нужен редактор, умеющий одновременно работать сразу с несколькими документами. При этом файлы удобно открывать в отдельных вкладках, чтобы быстро переходить к нужному документу;
- проверка текущего документа на ошибки

Инструментарий

Браузер

Браузер это программа, предназначенная для просмотра веб-страниц. Вначале подойдет любой браузер, но с повышением опыта и знаний потребуется завести несколько, чтобы проверять правильность отображения сайта в разных браузерах.

Каждый браузер имеет свои уникальные особенности, поэтому для проверки универсальности кода требуется просматривать и корректировать код с их учетом.

На сегодняшний день наибольшей популярностью пользуются три браузера: Firefox, Internet Explorer и Opera.

Графический редактор

Графический редактор необходим для обработки изображений и их подготовки для публикации на веб-странице. Самой популярной программой такого рода является Photoshop. Но зачастую лучше воспользоваться чем-нибудь более простым. Так, программа Paint.Net позволяет сделать все необходимые манипуляции с изображениями, вдобавок бесплатна для использования.

Теги

Чтобы браузер при отображении документа понимал, что имеет дело не с простым текстом, а с элементом форматирования и применяются теги. Общий синтаксис написания тегов следующий.

```
<тег параметр1="значение" параметр2="значение">
```

```
<тег параметр1="значение" параметр2="значение">...</тег>
```

Видно, что теги бывают двух типов — одиночные и парные (контейнеры). Одиночный тег используется самостоятельно, а парный может включать внутри себя другие теги или текст. Теги имеют различные параметры. Они , разделяются между собой пробелом. Бывают теги без дополнительных параметров.

Условно их можно подразделить на **обязательные**, они непременно должны присутствовать, и **необязательные**, их добавление зависит от цели применения тега.

Теги

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset= windows
-1251">
<title>Пример текста</title>
</head>
<body>
<p>Здесь используется одиночный тег <meta>.
    Парных тегов несколько: <html>, <head>, <title>, <body> и
<p>.
</p> </body> </html>
```

Парные теги, еще называют **контейнерами**.

Правила применения тегов

Парные теги состоят из двух частей — открывающий и закрывающий тег.

Открывающий тег обозначается как и одиночный — `<тег>`.

В закрывающем используется слэш — `</тег>`.

Можно вкладывать в контейнер другие теги, но следует соблюдать их порядок.

Согласно спецификации HTML все значения параметров тегов следует указывать в двойных ("пример") или одинарных кавычках ('пример').

Отсутствие кавычек не приведет к ошибкам, браузеры во многих случаях достаточно корректно обрабатывают код и без кавычек.

Любые теги, и их параметры нечувствительны к регистру. Можно писать — `
`, `
` или `
`.

Рекомендуется придерживаться выбранной формы записи на протяжении всех страниц сайта.

Значения параметров тегов

Если для тега не добавлен какой-либо допустимый параметр, это означает, что в этом случае браузер будет подставлять значение, заданное по умолчанию.

Если вы ожидали получить иной результат на веб-странице, проверьте, м.б. вы не указали значения некоторых параметров. Иногда используются некоторые параметры у тегов, без присваивания им никакого значения.

Порядок параметров в теге не имеет значения и на результат отображения элемента не влияет.

Теги вида

`` и

``

по своему действию равны.

Каждый параметр тега относится к определенному типу (*текст, число, путь к файлу* и др.). Это нужно учитывать при написании параметра..

Структура html-документа



```
<html>
<head>
<title>Назв.</title>
</head>
<body>
Текст документа!
<address>
Автор документа
</address>
</body>
</html>
```

Структура html-документа

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html;  
charset=windows-1251">  
<title>Шаблон базового документа</title>  
</head>  
<BODY Background="UTSbanner.gif" Bgcolor="Gray",  
Text="Black", Link="Blue", Alink="Green", Vlink="Red">  
Здесь будет основной текст страницы!  
<address>  
Текст набрал Ю.Молородов<Br> 22 апреля 2010 г.  
</address>  
</body>  
</html>
```

Структура html-документа

Элемент `<!DOCTYPE>` используется для указания типа текущего документа — DTD (document type definition, описание типа документа).

Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую Веб-страницу.

HTML существует в нескольких версиях.

Также имеется XHTML (EXtensible HyperText Markup Language, расширенный язык разметки гипертекста), похожий на HTML, но различающийся с ним по синтаксису.

Чтобы браузер «не путался» и понимал, согласно какому стандарту отображать Веб-страницу и необходимо в первой строке кода задавать `<!DOCTYPE>`.

Структура html-документа

Замечание

Часто можно встретить код HTML вообще без использования `<!DOCTYPE>`.

Но Веб-страница все равно будет показана.

Однако может получиться, что один и тот же документ отображается в браузере по-разному при использовании `<!DOCTYPE>` и без него.

Кроме того, браузеры могут по-своему показывать такие документы, в итоге страница может отображаться совсем не так, как это хотел разработчик.

Чтобы таких ситуаций не произошло, добавляйте `<!DOCTYPE>` в начало документа.

Структура html-документа

В *XHTML* предусмотрены три разновидности дескриптора `<!DOCTYPE>`: *strict*, *transitional* и *frameset*. В любом случае декларировать этот дескриптор надлежит только в верхней части кода Web-страницы.

- **Strict.** Эту разновидность дескриптора следует использовать только в том случае, если ваши страницы предназначены для просмотра с помощью Web-браузеров последних версий, которые должным образом поддерживают таблицу стилей. Код дескриптора такого типа выглядит следующим образом:

```
<!DOCUMENT html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "DTD/xhtml11-strict.dtd">
```

Структура *html*-документа

- **Transitional.** Если вы не уверены, что доступ к вашим Web-страницам будет осуществляться только с помощью браузеров последних версий, используйте разновидность transitional дескриптора `<!DOCTYPE>`.
`<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"DTD/xhtml1-transitional.dtd">`
- **Frameset.** Тип frameset следует использовать для работы с фреймами.
`< DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"DTD/xhtml1-frameset.dtd">`
Обратите внимание, что дескриптор `<! DOCUMENTS— . . . >` набран прописными буквами.
Прочие дескрипторы Web-страницы должны быть набраны строчными буквами

Простейший html-документ

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251"> <
title>Пример веб-страницы</title>
</head>
<body>
<h1>Заголовок</h1>
<!-- Комментарий --> <p>Первый абзац.</p> <p>Второй
абзац.</p>
<address>
Текст набрал Ю.Молородов<Br> 22 апреля 2010 г.
</address>
</body>
</html>
```

Типы тегов (дискрипторов)

Каждый тег HTML принадлежит к определенной группе тегов: 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;
- встроенные элементы;
- универсальные элементы;
- списки;
- таблицы.

Теги верхнего уровня

С их помощью формируется структура Веб-страницы и определяются раздел заголовка и тело документа.

<HTML>

Тег `<html>` является контейнером, который заключает в себе всё содержимое веб-страницы, включая теги `<head>` и `<body>`.

Открывающий и закрывающий теги `<html>` в документе необязательны, но хороший стиль диктует непременно их использование.

<HEAD>

Тег `<head>` предназначен для хранения других элементов, цель которых — помочь браузеру в работе с данными.

Внутри контейнера `<head>` находятся метатеги, которые используются для хранения информации, предназначенной для браузеров и поисковых систем. Так, поисковые системы обращаются к метатегам чтобы найти описание сайта, ключевые слова и пр.

Теги верхнего уровня

<BODY>

Тег <body> предназначен для хранения содержания Веб-страницы, отображаемого в окне браузера.

Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера <body>.

К такой информации относятся текст, изображения, таблицы, списки и др.

<html>
 <head>
 ...
 </head>
 <body>
 ...
 </body>
</html>

Здесь контейнер <html> определяет «каркас» всей Веб-страницы. Вначале задается тег <head>, затем идет контейнер <body>, в нем хранится содержательная часть документа, которая и отображается в браузере. Теги <html> и <body> хотя и не относятся к обязательным тегам (их можно не размещать в коде), но стоит добавлять всегда. Это позволяет получить четкую и понятную структуру документа.

Теги заголовка документа

К ним относятся элементы, которые располагаются в контейнере `<head>`. Эти теги не отображаются в окне браузера, за исключением тега `<title>`, который определяет название Веб-страницы.

<TITLE> Используется для отображения строки текста в левом верхнем углу окна браузера. Она сообщает пользователю название сайта и другую информацию разработчика.

<МЕТА> Метатеги используются для хранения информации, предназначенной для браузеров и поисковых систем.

Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных. Тег `<meta>` всего один, он у него масса

параметров, поэтому к нему и применяется множественное

Блочные элементы

Блочные элементы занимают всю доступную ширину, высота элемента определяется его содержимым, и он всегда начинается с новой строки.

<BLOCKQUOTE> Предназначен для выделения длинных цитат внутри документа. Текст, обозначенный этим тегом, отображается как выровненный блок с отступами слева и справа (примерно по 40 пикселей), а также с пустым пространством сверху и снизу.

<DIV> Тег `<div>` относится к универсальным блочным контейнерам и применяется в тех случаях, где нужны блочные элементы без дополнительных свойств. С помощью тега `<div>` можно выравнивать текст внутри этого контейнера с помощью параметра `align`.

<H1>...<H6> Эта группа тегов определяет текстовые заголовки разного уровня, которые показывают относительную важность секции, расположенной после заголовка.

Блочные элементы

<HR>

Рисует горизонтальную линию, которая по своему виду зависит от используемых параметров, а также браузера. Линия всегда начинается с новой строки, а после нее все элементы отображаются на следующей строке.

<P>

Определяет параграф (абзац) текста.

<PRE>

Задаёт блок предварительно форматированного текста. Такой текст отображается обычно моноширинным шрифтом и со всеми пробелами между словами. В HTML любое количество пробелов идущих в коде подряд на Веб-странице показывается как один.

Тег `<pre>` позволяет обойти эту особенность и отображать текст как требуется разработчику.

Встроенные элементы

Это элементы Веб-страницы, которые являются непосредственной частью другого элемента, например, текстового абзаца. Они используются для изменения вида текста или его логического выделения.

<A> Тег `<a>` является важным элементом HTML. Он предназначен для создания ссылок. В зависимости от присутствия параметров *name* или *href* тег `<a>` устанавливает ссылку или якорь.

**** Определяет жирное начертание шрифта.

<BIG> Тег `<big>` увеличивает размер шрифта на единицу по сравнению с обычным текстом. В HTML размер шрифта измеряется в условных единицах от 1 до 7. Средний размер текста, используемый по умолчанию, принят 3. Т.о., добавление тега `<big>` увеличивает текст на одну условную единицу.

**
** Тег `
` устанавливает перевод строки в том месте, где он находится. Тег `
` не добавляет пустой отступ перед строкой, в отличие от тега параграфа `<p>`.

Встроенные элементы

**** Тег предназначен для акцентирования текста, отображая его курсивом.

<I> Устанавливает курсивное начертание шрифта.

**** Предназначен для отображения на Веб-странице изображений в графическом формате GIF, JPEG или PNG. Если нужно, то рисунок можно сделать ссылкой на другой файл, поместив тег в контейнер <a>. При этом вокруг изображения отображается рамка, которую можно убрать, добавив параметр border="0" в тег .

<SMALL> Он уменьшает размер шрифта на единицу по сравнению с обычным текстом. Похож на тег <big>, но действует с точностью до наоборот.

**** Предназначен для определения встроенного элемента внутри документа.

**** Используется для акцентирования текста. Браузеры отображают такой текст жирным начертанием.

Встроенные элементы

<SUB> Отображает шрифт в виде нижнего индекса. Текст при этом располагается ниже базовой линии остальных символов строки и уменьшенного размера — H_2O .

<SUP> Отображает шрифт в виде верхнего индекса. По своему действию похож на `<sub>`, но текст отображается выше базовой линии текста — m^2 .

Есть разница между блочными и встроенными элементами:

- Встроенные элементы могут содержать только данные или другие встроенные элементы. В блочные можно вкладывать другие блочные элементы, встроенные элементы, и данные. Т.е., встроенные элементы не могут хранить блочные элементы.
- Б.Э. занимают всю доступную ширину окна браузера. Ширина встроенных элементов равна их содержимому плюс значения отступов, полей и границ.
- Блочные элементы всегда начинаются с новой строки

Универсальные элементы

Их особенность в том, что они в зависимости от контекста могут использоваться как блочные или встроенные элементы.

Тег используется для выделения текста, который был удален в новой версии документа. Такое форматирование позволяет отследить, какие изменения в тексте документа были сделаны. Браузеры обычно помечают текст в контейнере как перечеркнутый.

<INS>

Тег <ins> предназначен для акцентирования вновь добавленного текста и обычно применяется наряду с тегом . Браузеры помечают содержимое контейнера <ins> подчеркиванием текста.

Теги списков

Это взаимосвязанный набор отдельных фраз или предложений, начинающиеся с маркера или цифры. Они предоставляют возможность упорядочить и систематизировать разные данные и представить их в наглядном и удобном для пользователя виде.

**** устанавливает нумерованный список. Каждый элемент списка начинается с числа или буквы и увеличивается по нарастающей.

**** устанавливает маркированный список.

<DD>, <DT>, <DL> Тройка элементов предназначена для создания списка определений. Каждый такой список начинается с контейнера **<dl>**, куда входит тег **<dt>** создающий термин и тег **<dd>** задающий определение этого термина. Закрывающий тег **</dd>** не обязателен, т.к. следующий тег сообщает о завершении предыдущего элемента. Тем не менее, лучше закрывать все теги.

Теги таблиц

Таблица состоит из строк и столбцов ячеек. Они могут содержать текст и рисунки. Их используются для упорядочения и представления табличных данных.

<TABLE> Служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов `<tr>` и `<td>`.

<TD> предназначен для создания одной ячейки таблицы. Тег `<td>` должен размещаться внутри контейнера `<tr>`, который в свою очередь располагается внутри тега `<table>`.

<TH> предназначен для создания одной ячейки таблицы, которая обозначается как заголовочная. Текст в ячейке отображается браузером жирным шрифтом и выравнивается по центру.

<TR> Тег `<tr>` служит контейнером для создания строки таблицы.

Фреймы

Они разделяют окно браузера на отдельные области, расположенные вплотную друг к другу. В каждую из таких областей загружается самостоятельная Веб-страница определяемая с помощью тега `<frame>`. С их помощью Веб-страница делится на несколько документов. Они содержат навигацию по сайту и его контент. Механизм фреймов позволяет открывать документ в одном фрейме, по ссылке, нажатой в другом фрейме. Можно использовать вложенную структуру элементов. Это позволяет дробить фреймы на более мелкие области.

<FRAME> определяет свойства отдельного фрейма, на которые делится окно браузера.

<FRAMESET> заменяет собой элемент `<body>` на веб-странице и формирует структуру фреймов.

<IFRAME> создает плавающий фрейм, который находится внутри обычного документа. Позволяет загружать в область заданных размеров любые другие независимые документы.

Значения параметров у тегов. Цвет.

Параметры тегов расширяют возможности тегов и позволяют управлять настройками отображения элементов Веб-страницы. Количество параметров велико, но их значения можно сгруппировать по разным типам: задающих цвет, размер, адрес и др.

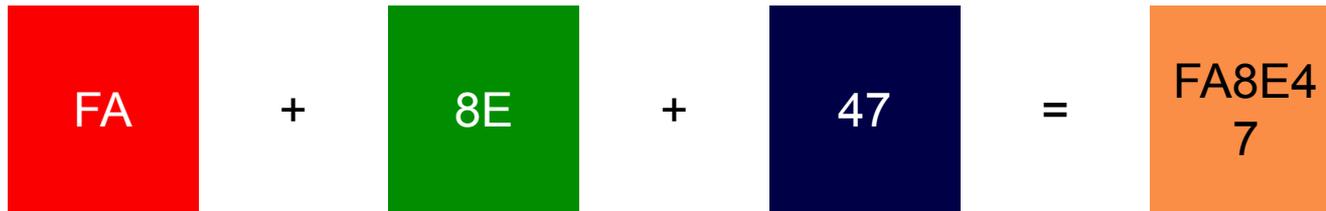
Цвет задается: с помощью шестнадцатеричного кода и по названию некоторых цветов. Чаще используется способ, основанный на шестнадцатеричной системе исчисления. Он более универсальный.

Шестнадцатеричные цвета. Она в отличие от десятичной системы, базируется на числе 16. Цифры будут следующие: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Здесь числа от 10 до 15 заменены латинскими буквами. Числа больше 15 в ней образуются объединением двух чисел в одно. Так 255 в десятичной системе соответствует число FF в шестнадцатеричной.

Значения параметров у тегов. Цвет

Чтобы не возникало путаницы в определении системы счисления, перед шестнадцатеричным числом ставится символ решетки #, например #aa69cc. Регистр не важен. Поэтому #F0F0F0 = #f0f0f0. Типичный цвет, используемый в HTML: <body bgcolor="#fa8e47">. Цвет фона Веб-страницы задан как #FA8E47. Первые две цифры (FA) определяют красную составляющую цвета, цифры с третьей по четвертую (8E) — зеленую, а последние две цифры (47) — синюю. В итоге получится такой цвет.



Каждый из трех цветов — красный, зеленый и синий — может принимать значения от 00 до FF.

В итоге можно получить 256 оттенков.

Значения параметров у тегов. Цвет

Т.о. общее количество цветов может быть $256 \times 256 \times 256 = 16.777.216$ комбинаций.

Цветовая модель, основанная на красной, зеленой и синей составляющей получила название **RGB** (red, green, blue).

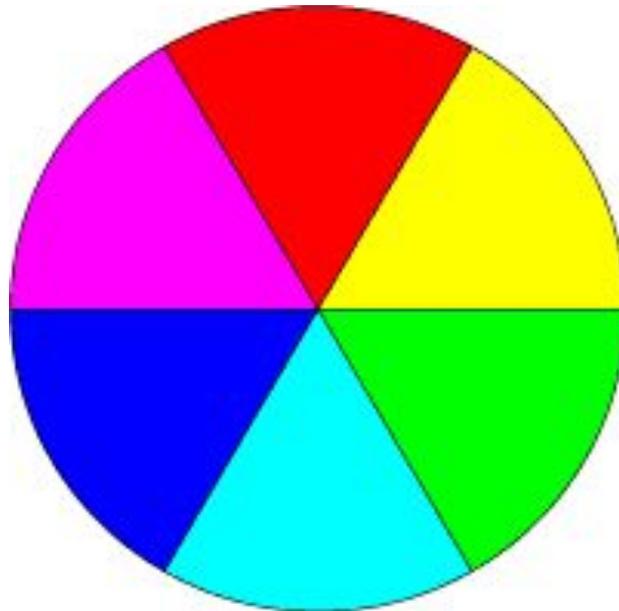
Эта модель аддитивная. При сложении всех трех компонент получается белый цвет.

Некоторые правила.

- Если значения компонент цвета одинаковы (например: **#D6D6D6**), то получим серый оттенок.
- Чем больше число, тем светлее цвет, значения при этом меняются от **#000000** (черный) до **#FFFFFF** (белый).
- Цвет со значением **#FF0000** самый красный из возможных красных оттенков.
- Аналогично - зеленый цвет (**#00FF00**) и синий (**#0000FF**).
- Желтый цвет (**#FFFF00**) получается смешением красного с зеленым.

Значения параметров у тегов. Цвет

Это хорошо видно на цветовом круге. Здесь представлены основные цвета **RGB** и комплиментарные или дополнительные. К ним относятся желтый, голубой и фиолетовый (пурпурный). Любой цвет можно получить смешением близлежащих к нему цветов. Голубой (#00FFFF) получается за счет объединения синего и зеленого цвета.



Цветовой круг

Значения параметров у тегов. Размер.

В **HTML** размеры элементов или расстояния между ними задаются в пикселах или процентах.

Пиксел — это элементарная точка на экране монитора. Она является относительной единицей измерения. Ее величина зависит от установленного экранного разрешения и размера монитора.

Так популярное разрешение монитора - 1024x768 пикселей. Картинка с такими размерами будет занимать всю область экрана. Увеличив разрешение монитора до 1280x1024, мы, тем самым, уменьшим размеры изображения на экране.

При использовании пикселей в качестве значений пишется только число без указания единиц, например: `width="380"`.

Значения параметров у тегов. Адрес.

Адресом называется путь к документу, например, к графическому файлу. Адрес нужен тогда, когда делается ссылка на Веб-страницу или загружается определенный файл.

Так в теге адрес используется в качестве аргумента параметра src. Он задает путь к файлу с изображением.

Синонимом адреса выступает URL (Universal Resource Locator, универсальный указатель ресурсов).

Различают **абсолютные** и **относительные** адреса.

Абсолютные адреса работают везде, где задан URL. Они начинаются **всегда** с указания протокола передачи данных.

Для Веб-страниц они начинаются с ключевого слова http://.

HTTP (HyperText Transfer Protocol, протокол передачи гипертекста).

```
<body>
```

```
  <a href="http://htmlbook.ru/html/body.html">Описание тега BODY</a>
```

```
</body>
```

Значения параметров у тегов. Адрес.

Относительные адреса указываются от корня сайта или текущего документа. Например, код `` означает загрузить графический файл с именем `pic.gif`. Он расположен в той же папке, что и сама веб-страница.

Адрес указывает обычно на файл `index.html`, который находится в корне сайта.

`/images/pic.gif`. Слэш (символ `/`) перед адресом говорит о том, что адресация начинается от корня сайта. Ссылка ведет на рисунок `pic.gif`, который находится в папке `images`. А та в свою очередь размещена в корне сайта.

`../help/me.html`

Двоеточие перед именем указывает браузеру перейти на уровень выше в списке каталогов сайта и там «поискать» файл `me.html`.

Особенности текста в HTML

Любое количество пробелов идущих подряд, в браузере отображается как один.

Сколько бы ни было пробелов между словами, это не повлияет на конечный вид текста. Это правило относится к символам табуляции и переносу текста.

Исключением из этого правила является тег `<pre>`, внутри него любое число пробелов отображается именно так, как оно указано в коде.

Нет расстановки переносов в тексте.

HTML не поддерживает расстановку переносов в словах как это делают текстовые редакторы. Это несущественно, пока не используется выравнивание текста по ширине.

В этом случае блок текста выравнивается по левому и правому краю. Короткие строки при этом растягиваются за счет автоматического добавления пробелов между словами.

Особенности текста в HTML

Текст занимает ширину окна браузера.

Если вы просто напишите одну длинную строку в коде HTML, то в браузере она будет отформатирована, чтобы текст поместился по ширине в окно.

Переносы текста будут добавлены автоматически в местах пробела или дефиса.

Что произойдет, если в тексте нет ни того, ни другого символа?

Браузер не сможет создать переносы и отобразит текст одной строкой.

Если она шире окна браузера, то появится горизонтальная полоса прокрутки.

Особенности текста в HTML. Абзацы.

Как правило, блоки текста разделяют между собой абзацами (параграфами). По умолчанию между ними существует небольшой вертикальный отступ, называемый **отбивкой**.

Синтаксис создания абзацев:

```
<p>Абзац 1</p>
```

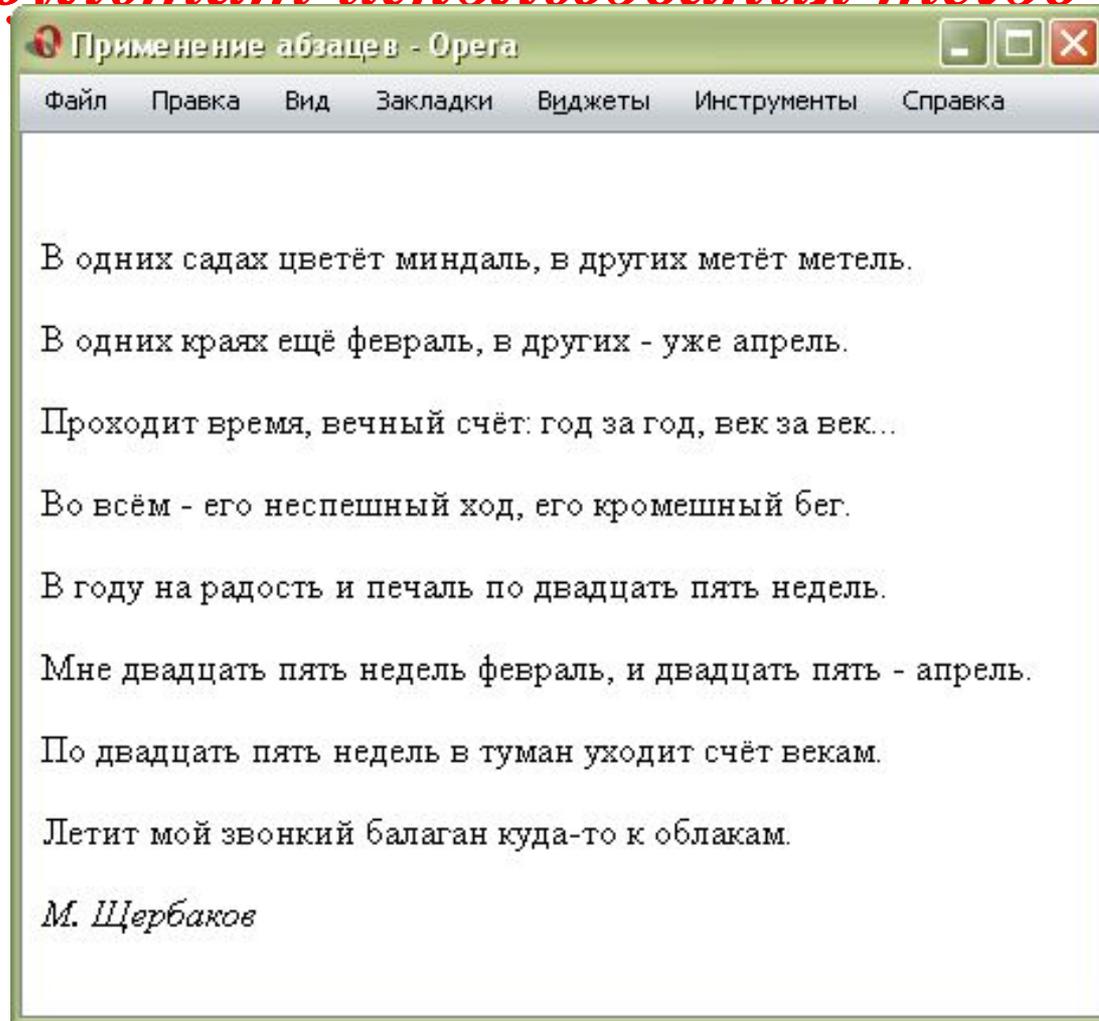
```
<p>Абзац 2</p>
```

Каждый абзац начинается с тега `<p>` и должен иметь необязательный закрывающий тег `</p>`.

Между абзацами возникают большие отступы. От них можно избавиться, если в местах переноса строк добавлять тег `
`.

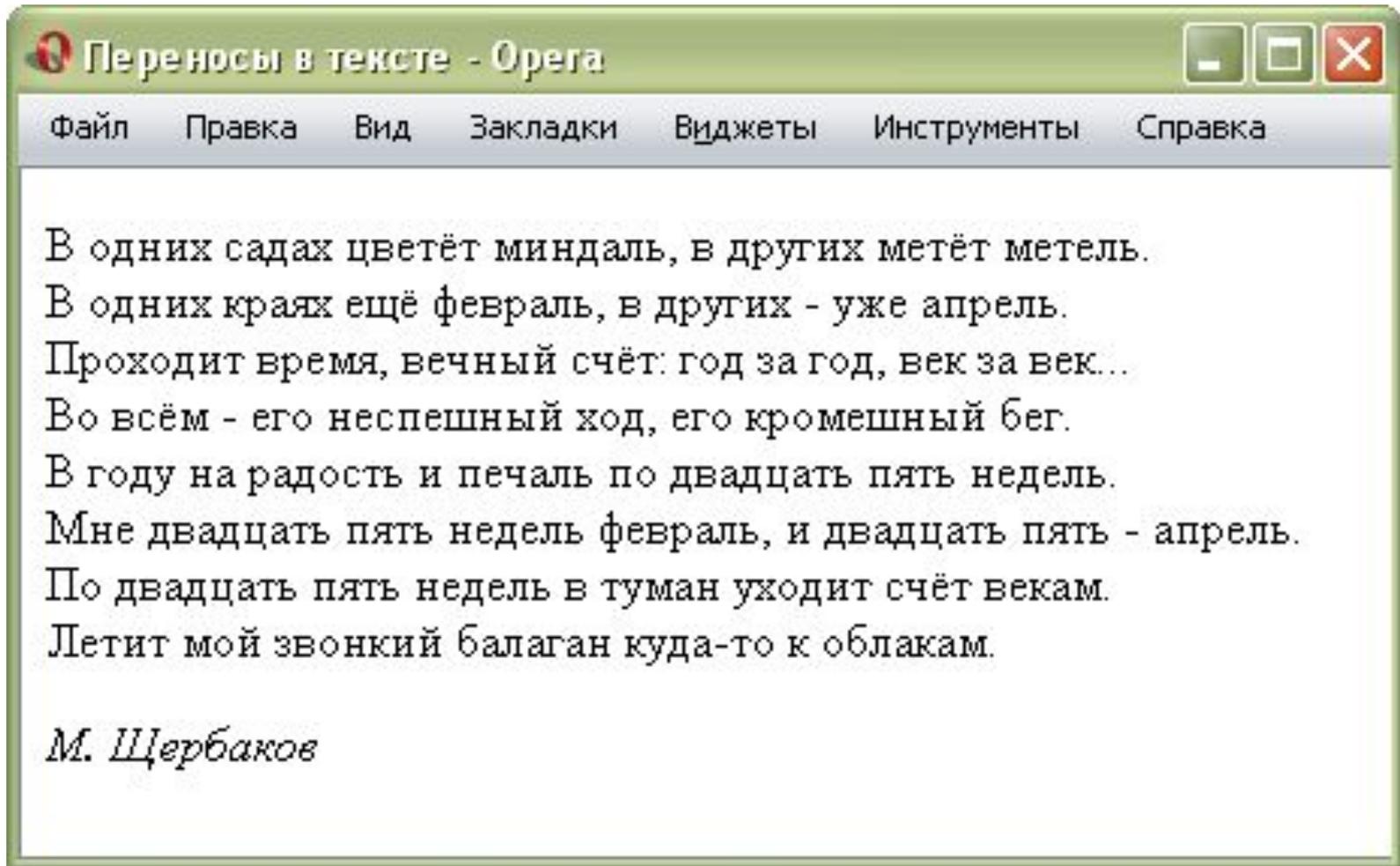
В отличие от абзаца, тег переноса строки `
` не создает вертикальных отступов между строками.

Результат использования тегов <p>



Отступы на Веб-странице при использовании абзацев

Результат использования тегов `
`



Вид текста с учетом переносов

Значения параметров у тегов. Заголовки.

Заголовки выполняют важную функцию на Веб-странице.

1. Они показывают важность раздела, к которому относятся. Чем больше заголовков и его «вес» , тем более он значимый. В газетах и журналах передовицы набраны крупным шрифтом, тем самым, привлекая к ним внимание и говоря: «вот это надо читать обязательно» .
2. С помощью различных заголовков легко регулировать размер текста. Чем выше уровень заголовка, тем больше размер шрифта. Самым верхним уровнем является уровень 1 (<h1>), а самым нижним — уровень 6 (<h6>).
3. Поисковики добавляют рейтинг тексту, если он находится внутри тега заголовка. Это важно для раскрутки сайта и для его занятия первых строк выдачи результата в поисковой системе по ключевым словам.

Чаще применяют заголовки с первого по третий уровень.

Выравнивание текста

Выравнивание текста определяет его внешний вид и ориентацию краев абзаца.

Может выполняться по левому краю, правому краю, по центру или по ширине. Наиболее распространенный вариант — выравнивание по левому краю, когда слева текст сдвигается до края, а правый остается неровным.

Выравнивание по правому краю и по центру чаще используется в заголовках и подзаголовках.

При выравнивании по ширине, в тексте между словами могут появиться большие интервалы.

Для установки выравнивания текста используется тег параграфа `<p>` с параметром `align`.

Выравнивание текста

Блок текста можно выравнивать с помощью тега `<div>` с параметром `align`.

Он может принимать следующие значения:

`left` — выравнивание по левому краю, задается по умолчанию;

`right` — выравнивание по правому краю;

`center` — выравнивание по центру;

`justify` — выравнивание по ширине (одновременно по правому и левому краю).

Этот аргумент работает только для текста, длина которого более, чем одна строка.

Аргумент `align` можно применять как для текста, так и для заголовков.

Значения параметров тегов Начертание.

Жирное начертание.

Насыщенностью называют увеличение толщины линий шрифта и соответственно контраста. Обычно различают четыре вида насыщенности: светлое начертание, нормальное, полужирное и жирное. Однако с помощью HTML можно установить только нормальное и жирное начертание. Для установки текста жирного начертания применяется два тега: `` и ``.

``Жирное начертание шрифта``

``Сильное выделение текста``

Курсивное начертание.

Курсивный шрифт представляет собой не просто наклон отдельных символов, для некоторых шрифтов это полная переделка под новый стиль, имитирующий рукописный.

Курсив для текста определяют два тега: `<i>` и ``.

`<i>`Курсивное начертание шрифта`</i>`.

Замечания

Теги `` и ``, также как `<i>` и `` хотя и похожи по своему действию, являются не совсем эквивалентными и заменяемыми. Тег `` — является тегом физической разметки и устанавливает жирное начертание текста.

Тег `` — тегом логической разметки и выделяет помеченный текст.

Разделение тегов на **логическое** и **физическое** форматирование изначально предназначалось, чтобы сделать HTML универсальным, т.е. не зависящим от устройства вывода информации.

Теоретически, если воспользоваться, например, речевым браузером, то текст, оформленный с помощью тегов `` и ``, будет отмечен по-разному.

Но получилось так, что в популярных браузерах результат использования этих тегов равнозначен..

Лекция окончена!

Благодарю за внимание!

Ваши вопросы.

Молородов Юрий Иванович
yuto@ict.nsc.ru

ftp://pine.ict.nsc.ru/pub/tmp/SibGUTI_Lec