**Министерство образования, науки и молодежной политики Нижегородской области**

Государственное бюджетное профессиональное образовательное учреждение

″Нижегородский техникум отраслевых технологий″

Учебное пособие

**«Создание web-страниц**

**с помощью языка HTML»**

Данное учебное пособие рассчитано на студентов 1-3 курсов. В пособии рассматриваются вопросы созданияweb-страниц с помощью языка HTML.

Отдельным разделом представлен компьютерный практикум. В результате выполнения работ студенты приобретут практические навыки по созданию web-страниц.

# Основные понятия и пояснения

Одной из самых распространенных служб интернета (наряду с электронной почтой) является World WideWeb (WWW, Web)- самая мощная информационная система, связывающая между собой миллионы документов (Web-страниц), размещенных на Web-серверах по всему миру. В расширенной этой системы могут принимать участие практически все пользователи, имеющие доступ в Интернет.

Web-страница – это документ (гипертекстовый документ), размеченный с помощью специальных элементов(тегов) языка HTML. Основное отличие этих гипертекстовых документов от хорошо известных нам текстовых состоит в том, что они могут включить ссылки на другие аналогичные документы. Любой пользователь получает доступ к WWW-системе (к любой Web-странице) с помощью Web-браузера (программы-клиента). Web-браузер осуществляет http-запрос к серверу, отображает Web-страницу на экране компьютера пользователя.

Для задания местоположения Web-страниц в World Wide Web использует URL (Uniform Resource Locator). Каждая страница имеет свой уникальный адрес.

Web-сайт – это набор Web-страниц, относящихся к одной фирме или одному пользователю.

**Основные элементы Web-страницы**

* заголовок страницы (может быть);
* элементы навигации (меню) для перехода со страницы на страницу (каждый пункт меню также может быть отдельным рисунком);
* рисунки или фотографии;
* текстовые блоки

Каждая страница представляется отдельным файлом с расширение **.html**. Каждый рисунок представляется файлом с расширение .jpg, .png или .gif.

Специальные программы – браузеры служат для интерпретации html-тегов и отображения содержимого Web-страниц. На экран html-теги не выводятся, они только указывают браузеру, как отображать содержимое документа.

Всегда можно посмотреть html-код любой страницы в браузере. Для этого в верхнем меню браузера найдите пункт View («Вид»)/Source («В виде HTML») для Internet Explorer.

# Язык HTML. Общие сведения

Что такое HTML – Hyper Text Markup Language?

Это язык гипертекстовой разметки, разработанный специально для создания Web-документов. Отметим два важных момента:

1. HTML не является языком программирования» В нем нет логических последовательностей, это именно язык разметки документов (текстов).
2. HTML определяет логическую структуру документа. Стандарт HTML, как и другие стандарты для Web, разработаны под руководством консорциума World Wide Web (World Wide Web Consortium, W3C).

Разметка документа осуществляется с помощью тегов (англ. Tag – отметка).

Синтаксис HTML

Теги html бывают двух типов – контейнерные и одиночные – и – заключаются в угловые скобки <Имя-тега>

**Контейнерные теги** состоят из пары – открывающий и закрывающий тег. Перед именем закрывающего тега необходимо ставить косую черту «/» (прямой слэш). <Имя\_тега> – содержимое, обрабатываемое данным тегом </Имя\_тега>.

Закрывающий тег завершает действие открывающего. Большинство тегов в HTML – контейнерные.

**Одиночные теги** *<Имя\_тега>*

Одиночных тегов не так много. Как контейнерных. Одиночный тег состоит только из открывающего и не требует закрывающего. То, что тег одиночный, как правило, понятно по смыслу.

**Вложенные теги**

В html-теги можно помещать другие теги.

***Следите за вложенностью тегов!*** Тег, открытый первым, закрывают последним. Тег, открытый вторым, закрывают вторым с конца и т.д.

*Например*:<Имя\_тега1><Имя\_тега2> …</Имя\_тега2></Имя\_тега1>

**Теги, атрибуты, значения**

<Тег Имя\_атрибута1= «значение» Имя атрибута2= «значение» > … </Тег>

Тег может иметь атрибуты и значения атрибутов. Атрибуты добавляют в тег для расширения или модификации его действий. Наборы допустимых атрибутов для тега описаны в спецификации языка HTML.

Правила записи атрибутов и значений:

* после имени тега через пробел могут следовать атрибуты;
* атрибуты отделяют друг от друга пробелами;
* порядок следования атрибутов произволен;
* атрибуты не нужно повторно описывать в закрывающем теге;
* значения атрибутов записывают в кавычках «» после знака равенства;
* названия тегов не должны содержать пробелов.

**Регистр написания тегов, атрибутов, значений атрибутов**

В языке HTML безразлично, в каком регистре вы записываете названия тегов и их атрибуты. Чувствительными к регистру являются только некоторые значения атрибутов. Поэтому сразу возьмите за правило писать значения атрибутов в нижнем регистре. Спецификация HTML позволяет опускать кавычки: если значение атрибута состоит из одного слова (букв a-z) или числа (цифр 0-9). Если значение содержит несколько слов, разделенных запятыми или пробелами, либо специальные символы, то кавычки обязательны.

# Базовая структура документа

Любой html-документ (страница) заключается между тегами <HTML> и </HTML>.

Html-документ состоит из двух частей: заголовок – HEAD и тело – BODY. В заголовке содержится информация о документе – название, мета-информация и т.т. В теле находится само содержимое документа – то, что выводится в окне браузера: текст, картинки, таблицы и т.д.

И так, мы с Вами поняли, что HTML — это краеугольный камень, благодаря которому любой браузер пользователя отобразит страницу. Но как выглядит HTML? Давайте рассмотрим простой код, из которого состоит практически любая информационная страница:

<!DOCTYPE html>

<HTML>

<HEAD>

<TITLE>Название документа</TITLE>

</HEAD>

<BODY>

Содержимое документа

</BODY>

</HTML>

А теперь детально разберем из чего состоит любая HTML страница:

**DOCTYPE** это декларация, которая определяет тип документа. Декларация не является тегом HTML это только инструкция для веб-браузера о том, какая версия HTML используется в документе. Декларация !DOCTYPE помогает браузеру отобразить веб-страницу правильно, для этого браузер должен знать, как тип, так и версию документа.

Декларация <!DOCTYPE html> сообщает, что в этом документе используется пятая версия *языка гипертекстовой разметки* – **HTML 5**.

***Обращаю Ваше внимание* на то, что декларация должна указываться самой первой в вашем HTML документе, перед тегом <HTML>. Декларация <!DOCTYPE> не чувствительна к регистру, но записывать её принято в верхнем регистре.**

Текст между тегами [**<HTML>**](https://basicweb.ru/html/tag_html.php) **и </HTML>** сообщает браузеру, что необходимо прочитать документ как программный код, написанный на языке гипертекстовой разметки. Является корневым элементом HTML документа и все остальные элементы должны быть его потомками (вложены в него).

Тег <**TITLE**> (контейнерный </**TITLE**>) определяет название документа. Это единственный обязательный элемент заголовочной части документа. Вы можете его увидеть в самом верху окна браузера. Название документа должно содержать важные ключевые слова, чтобы поисковые системы могли включить вашу страницу в результаты поиска (по определенным запросам пользователей).

Тело html-документа (<**BODY**> </**BODY**>) описывает **видимое содержимое страницы!**

В теле документа размещается информация, которую вы видите в окне браузера – абзацы текста, списки, графические изображения, гипертекстовые ссылки, таблицы…

Html-документ сохраняются с расширением html (имя файла.html)

# Форматирование текста

Для продвижения ресурса в интернете большое значение имеет его наполнение. Кроме наличия качественного и полезного контента, важно и его оформление. Оно должно быть не только правильным, но и красивым. Текст в html по праву является основным способом подачи информации. Html имеет множество разных инструментов для отображения текста. Рассмотрим основные теги и их влияние на SEO-продвижение веб-страницы.

Используя теги, вы делаете дизайн текста и помогаете поисковым роботам понять, определить его назначение и важность.

Правильно оформленный текст в html должен иметь заголовки и подзаголовки. Важные фрагменты должны выделяться жирным или курсивом. Для облегчения восприятия цитаты выделяются кавычками. Кроме этого, существует еще очень много инструментов, которые помогают оформить текст и сделать его более привлекательным как для пользователей, так и поисковиков. И, для всего этого используются разные теги.

**Заголовки (<H#> ... </H#>)**

Эти теги используются для задания заголовков в документе. Существует (определено в спецификации HTML) шесть уровней заголовков: <H1>, <H2>, <H3>, <H4>, <H5>, <H6> (в порядке убывания важности).

Теги заголовков — контейнерного типа. По умолчанию браузеры выводят заголовки жирным начертанием и некоторые — шрифтом большего размера.

*Например:* <H1> Это моя первая страница </H1>

**Выравнивание заголовков**

Заголовки можно выравнивать по левому краю (по умолчанию), по правому и по центру. Для этого используется атрибут ALIGN (**left/center/right**).

*Например*: <H1 ALIGN=center> Это моя первая страница </H1>

**Абзацы (параграфы) <P>...</P>**

Абзац — еще один элемент, определяющий структуру документа, тег контейнерного типа. Используется для разметки абзацев. Встретив этот тег, браузер пропускает строку и начинает текст с новой строки.

Помните, что текст, разбитый на абзацы, воспринимается лучше, нежели сплошной.

**Выравнивание абзацев**

Текст абзацев можно выравнивать по левому краю (по умолчанию), по правому, по центру и по ширине. Для этого используется атрибут ALIGN (**left/center/right**).

*Например*: <Р ALIGN=right>

**Выравнивание текста** <TEXT-ALIGN>

Для выравнивания теста можно использовать атрибут ALIGN. Но он не всегда удобен, поэтому вместо него лучше использовать свойство text-align со значениями left (выравнивание слева), right (выравнивание справа), center (выравнивание по центру), justify (выравнивание по ширине).

CSS text-aling: jusitify

*Например*: <div style="text-aling: jusitify">Текст</div>

**Красная строка**

Если нужно, чтобы абзац начинался с красной строки, то используют один или несколько раз элемент «**&nbsp;**».

Для отступа красной строки во всех абзацах текста внутри тега <HEAD> набирают блок:

<style>

p {

text-indent: 25px; /\* Отступ первой строки в пикселах \*/

}

</style>

**Конец строки <BR>**

Используется для принудительного завершения строки — перехода на новую строку. Вставка подряд нескольких тегов <BR> (одиночных) приводит к пропуску нескольких строк.

**•** использование <BR> внутри абзаца <P> осуществляет перевод текста внутри абзаца на новую строку без пропуска дополнительной строки;

**•** использование подряд нескольких тегов <BR> осуществили пропуск трех строк.

**Стихотворения <PRE>**

Рассмотрим тег <PRE> (*HTML Preformatted Text*), он определяет предварительно отформатированный текст. Текст, который был помещён внутрь тега, по умолчанию будет отображен со шрифтом фиксированной ширины и сохранит как пробелы, так и разрывы строк.

*Например*:

<p>А.С. Пушкин</p>

<pre> Я помню чудное мгновенье: <!-- добавим несколько пробелов для наглядности -->

Передо мной явилась ты,

Как мимолетное виденье,

Как гений чистой красоты.

</pre>

**Управление стилем шрифта <FONT>**

Служит для задания размера, цвета и начертания шрифта, должен иметь хотя бы один из атрибутов SIZE= размер (от 1 до 7), COLOR=цвет, FASE=гарнитура шрифта – лучше не использовать. <BASEFONT SIZE=значение> - размер основного шрифта документа

*Например*: <FONT COLOR=RED> Этот текст красного цвета</FONT>

**Дескрипторы физических стилей**

Физические стили – абсолютное форматирование, т.е. текст, выделенный физическим стилем, во всех браузерах выглядит одинаково.

|  |  |
| --- | --- |
| **Дескриптор** | **Стиль** |
| **<B>** | Полужирный шрифт |
| **<I>** | Курсив |
| **<U>** | Подчеркнутый |
| **<SUB>** | Подстрочный |
| **<SUP>** | Надстрочный |
| **<STRIKE>** | Зачеркнутый |
| **<TT>** | Моноширинный шрифт |

*Например:* <B> **Полужирный шрифт**</B>

**Дескрипторы логических стилей**

Логические стили указывают на характер текста, браузер сам решает как форматировать по отношению к остальной части текста. В различных браузерах они будут отражаться по разному.

|  |  |
| --- | --- |
| **Дескриптор** | **Стиль** |
| **<EM>** | Выделение текста |
| **<STRONG>** | Более сильное выделение |
| **<CITE>** | Текст в виде цитаты |
| **<DFN>** | Определение |

**Цвет в HTML-документе**

При необходимости можно указать цвета текста, ссылок, заголовков и т.д. Для этого или используются таблицы стилей (CSS), или цвет непосредственно задается внутри HTML-тегов:

<FONT COLOR=“цвет“>

<BODY TEXT=“цвет“ ...>

Можно задавать цвет его названием:

<FONT COLOR=“red“>Текст будет красным</FONT>

или используя цифровой код в шестнадцатеричной системе счисления (на базе 16 от 00 до FF), требует шести символов для описания цвета:

<FONT COLOR=“#FF0000“>Текст будет красным</FONT>

Перед кодом ставится #; первые два символа(FF) — это составляющая красного цвета, вторая пара (00) — составляющая зеленого и третья пара (00) —синего.

*Примеры:* #000000 — черный, #FFFFFF — белый,

#808080 — серый.

**Атрибуты тега <BODY>**

В этом теге можно задать: цвет фона страницы BGCOLOR, цвет текста TEXT, цвета ссылок — непосещенных LINK, посещенных VLINK, активных (в момент нажатия на ссылку) ALINK.

**Создание горизонтальной линии <HR>**

Одиночный тег может иметь атрибуты SIZE = толщина линии в пикселях, WIDTH=ширина в % или пикселях, ALIGN=left/center/right, NOSHADE – без эффектов

# Создание списков

Списком называется взаимосвязанный набор отдельных фраз или предложений, которые начинаются с маркера или цифры. Списки предоставляют возможность упорядочить и систематизировать разные данные и представить их в наглядном и удобном для пользователя виде.

В языке HTML используются три вида списков:

**•** маркированные списки (неупорядоченные) <UL>;

**•** нумерованные списки (упорядоченные) <OL>;

**•** списки определений <DL>.

**Маркированные списки** (<**UL**>...</**UL**>)

Вводятся тегом <UL> (контейнерного типа). Между открывающим <UL> и закрывающим </UL> располагаются элементы списка. Каждый элемент списка вводится тегом <LI> (контейнерного типа).

Можно принудительно задать тип маркера, используя атрибут TYPE. Для всего списка — <UL TYPE=disk>, для элемента списка — <LI TYPE= circle>.

Типы маркера (значения атрибута TYPE): disk, circle, square.

По умолчанию, маркированные списки отображаются на веб-странице в виде списка, начинающегося с маленького чёрного круга.

|  |  |
| --- | --- |
| *Например*:  <ul>  <li>Светлое</li>  <li>Тёмное</li>  <li>Пятница</li>  </ul> | *Выглядеть на странице это будет соответственно так:*   * Светлое * Тёмное * Пятница |

Для изменения типа маркера (стиля) вы можете воспользоваться свойством CSS list-style-type, чтобы определить стиль маркера. Возможные значения свойства:

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| list-style-type:none | Убирает маркер. |
| list-style-type:disc | Маленький черный круг. **Это значение по умолчанию**. |
| list-style-type:circle | Круг пустой внутри. |
| list-style-type:square | Маркер в виде квадрата. |

**Нумерованные списки** (<**OL**>...</**OL**>)

Нумерованный (упорядоченный) список предназначен для элементов, которые следуют в определённом порядке.

Вводятся тегом <OL> (контейнерного типа). Между открывающим <OL> и закрывающим </OL> располагаются элементы списка. Каждый элемент списка вводится тегом <LI> (контейнерного типа).

*Например:*

|  |  |
| --- | --- |
| <ol>  <li>Первый пункт</li>  <li>Второй пункт</li>  <li>Третий пункт</li>  </оl> | *Выглядеть будет так:*   1. Первый пункт 2. Второй пункт 3. Третий пункт |

Как и в случае с нумерованным списком, можно принудительно задать тип нумерации, используя атрибут TYPE. Для всего списка — <OL TYPE=А>, для элемента списка — <LI TYPE=i>.

Типы нумерации (значения атрибута TYPE): 1 — арабские цифры (по умолчанию), a — строчные буквы, A — заглавные буквы, i — римские цифры строчными буквами, I — римские цифры заглавными буквами.

По умолчанию браузер начинает нумерацию с первого значения 1, 2, 3... или a, b, c... Можно определить, с какого значения начать, используя атрибут START.

*Например*: <OL START =4 TYPE=A> .....</OL>

**Списки определений** <**DL**>...</ **DL** >

Они представляют собой перечень терминов и определений к этим терминам. Списки описаний используются для формирования пар типа «имя/значение» / «вопрос/ответ» и т.п.

Списки определений вводятся тегом <DL> и завершаются </DL>. Каждый термин вводится тегом <DT> (закрывающий тег можно не указывать). Определение к термину вводится тегом <DD> (закрывающий тег можно не указывать).

|  |  |
| --- | --- |
| *Например:*  <dl>  <dt>Жамевю</dt>  <dd>состояние, противоположное дежавю, внезапно наступающее ощущение того, что хорошо знакомое место или человек кажутся совершенно неизвестными или необычным.</dd>  <dt>Фасцинация</dt>  <dd>повышение эффективности воспринимаемого материала через использование сопутствующих (фоновых) воздействий.</dd>  </dl> | *Результат:*  Жамевю  состояние, противоположное дежавю, внезапно наступающее ощущение того, что хорошо знакомое место или человек кажутся совершенно неизвестными или необычным.  Фасцинация  повышение эффективности воспринимаемого материала через использование сопутствующих (фоновых) воздействий. |

**Маркированные** (неупорядоченные), **нумерованные** (упорядоченные) и списки описаний внутри себя могут содержать элементы нового списка (внутри элементов списка / значений списка), другие элементы HTML, такие как, изображения, ссылки и т.д. Списки можно размещать горизонтально, формировать из них меню навигации и так далее.

# Работа с рисунками

В Web используются следующие форматы изображений:

* формат GIF, поддерживает 256 цветов, прозрачность (позволяет любой цвет в палитре сделать прозрачным), анимацию (сохранение в одном файле нескольких
* формат JPEG, поддерживает до 16,7 млн.цветов. Используется для сохранения фотографических изображений.
* Формат PNG, поддерживает 24-битовые цвета RGB, соответствующие картинки и оттенки серого. Изображения отличаются небольшим весом при хорошем качестве.

Какие ограничения существуют при размещении изображений?

* прежде всего размер, так как большие изображения сильно замедляют время загрузки страницы;
* формат изображения. Размещайте на свои страницы только те форматы изображений, которые поддерживаются браузеров. В настоящий момент – это GIF, PNG и JPEG.

Любую HTML страницу тяжело представить без изображений, чтобы добавить изображение на страницу, то необходимо определить тег <**IMG**>.

Очевидно, что тег <IMG> бесполезен сам по себе, так как он не дает возможности уточнить, какое изображение мы хотим использовать. В этом случае не обойтись без специальных атрибутов, которые уточняют или корректируют элемент (выступают инструкцией для элемента).

Давайте рассмотрим основные атрибуты тега <IMG>, которые вы будете применять в повседневной работе с изображениями:

* **исходный файл** (**SRC**) – местонахождение файла (если указано только наименование файла, а не путь, то браузер ожидает найти файл в той же папке, где находится и веб-страница).
* **альтернативный текст** (**ALT**) - текст, который пользователь сможет увидеть, если элемент по каким-либо причинам не удается отобразить, также служит для интерпретации приложениями, читающими с экрана.
* **размер** (**WIDTH** и **HEIGHT**) - ширина и высота. Рекомендуется всегда указывать ширину и высоту изображения. Если ширина и высота не указаны, то браузер не резервирует место под изображение и при медленной загрузке страница будет «прыгать».
* **всплывающая подсказка** (**TITLE**) – при наведении на картинку, значение атрибута будет отображаться в виде всплывающей подсказки.

*Например:* <IMG SRC=”logo.gif” WIDTH=100 HEIGHT=100 >

Элемент <IMG> имеет два обязательных атрибута: SRC, который определяет *URL* адрес изображения (допускается использовать как абсолютный, так и относительный адрес) и ALT, который указывает альтернативный текст для изображения (например, если изображение не было загружено).

*Рассмотрим пример, в котором добавим изображение на страницу:*

<img src = "10.jpg" alt = "ничоси" width = "200" height = "200" title = "ничоси">

В этом примере для нашего изображения установили следующие значения:

* атрибутом SRC указали наименование файла и путь к файлу (изображение находится в той же папке где и веб-страница). **Обязательный атрибут**
* атрибутом ALT указали альтернативный текст, который пользователь сможет увидеть, если элемент по каким-либо причинам не удается отобразить. **Обязательный атрибут**
* задали ширину (WIDTH) и высоту (HEIGHT) изображения равной 200 пикселей
* атрибутом TITLE указали текст, который будет отображаться при наведении на картинку мышью в виде всплывающей подсказки.

|  |  |
| --- | --- |
| ***Ширина рисунка***  WIDTH=300, WIDTH=50% | ***Высота рисунка***  HEIGHT=150, HEIGHT=50% |

Если Вы не указываете размеры изображения, то оно добавляется на страницу с той шириной и высотой, которое имеет. Если указываете только один параметр, то второе значение будет масштабироваться браузером автоматически. Вы можете указывать значения меньшие или большие чем изображение, но учтите тот факт, что оно может потерять качество при масштабировании, либо значительно исказиться.

**Выравнивание рисунков**

Изображения выравниваются относительно окружающего их текста при помощи атрибутов **ALIGN**, **VSPACE** и **HSPACE.**

ALIGN=left/center/right – слева/по центру/справа в строке

ALIGN=top/middle/bottom – по верхнему краю/по центру/нижнему краю в строке

Определение пустого пространства над и под плавающим изображением (впикселях*)*

VSPACE=10

Определение пустого пространства справа и слева от рисунка (в пикселях)

HSPACE=20

**Увеличение рисунка по щелчку мыши по нему, и возврат в исходное состояние:**

<IMG SRC="имя файла" ALIGN=RIGHT/LEFT

style=" width:’значение ширины маленького изображения px" onclick="this.style.width=' значение ширины большого изображения px'" onmouseout="this.style.width=' значение ширины маленького изображения px'" alt="увеличение картинки на сайте"/>

**Увеличение рисунка при наведении указателя мыши по нему, и возврат в исходное состояние**

<IMG class="aligncenter" SRC="имя файла" style="width:' значение ширины маленького изображения px;" onmouseover="this.style.width=' значение ширины большого изображения px'" onmouseout="this.style.width=' значение ширины маленького изображения px'" alt="увеличение изображения на сайте"/>

# Работа с таблицами

Таблицы помогают нам систематизировать определённые данные в строки и столбцы. Они могут содержать в себе различную информацию, например, такую как текст, изображения, нумерованные (упорядоченные) и маркированные (неупорядоченные) списки и тому подобное.

При создании html-страниц таблицы используются для компоновки элементов дизайна страницы – текста, графики.

Чтобы понять, как составить таблицу, давайте рассмотрим ее простейший вариант:

Основное содержимое таблицы заключается в тег <**TABLE**>..</**TABLE**>.

Каждая строка таблицы помещается в элемент [<**TR**>...</**TR**>](https://basicweb.ru/html/tag_tr.php).

Ячейка заголовка таблицы помещается в элемент [<**TH**>...</**TH**>](https://basicweb.ru/html/tag_th.php) (при этом содержимое отображается жирным начертанием и выравнивается по центру).

Каждая ячейка данных таблицы помещается в элемент [<**TD**>...</**TD**>](https://basicweb.ru/html/tag_td.php).

Наименование таблицы, если оно требуется размещается внутри элемента [<**CAPTION**>...</**CAPTION**>](https://basicweb.ru/html/tag_caption.php) (необязательный элемент таблицы).

**Обращаю Ваше внимание**, что если Вы планируете использовать данный элемент в своей таблице, то он должен следовать в документе сразу за элементом [<TABLE>..</TABLE>](https://basicweb.ru/html/tag_table.php).

Внутри ячейки можно размещать текст, картинки, другие таблицы и т.д.

***Внешний вид таблицы***

Размер рамки таблицы <TABLE BORDER=размер>

По умолчанию браузеры выводят таблицы без рамки (границ)

Атрибут BORDER задает размер рамки в таблице без рамки (границ)

***Расстояние между ячейками*** <TABLE CELLSPACING=размер>

Атрибут CELLSPACING задает расстояние между ячейками (в пикселях) в таблице.

Расстояние между границей ячейки и ее содержимым <TABLE CELLSPACING =размер>.

Атрибут CELLPADDING задает расстояние между границей (рамкой) ячейки и ее содержимым (в пикселях). Этот атрибут тега <TABLE> определяет данное расстояние для всех ячеек в таблице.

***Задание размеров таблицы*** <TABLE WIDTH= размер HEIGHT=размер>

Атрибут WIDTH задает ширину таблицы, HEIGHT – ее высоту.

Размеры могут быть заданы в пикселях (абсолютные значения) и в процентах от доступной ширины экрана (относительные значения).

Чаще высота таблицы не указывается. Она изменяется в зависимости от содержимого ячеек.

<TABLE WIDTH= 100%> – данная таблица занимает всю ширину экрана браузера.

***Задание ширины ячеек***

<TD WIDTH= размер HEIGHT=размер>

WIDTH задает ширину ячейки, HEIGHT – высоту. Эти размеры также могут быть заданы в пикселях и в процентах.

***Выравнивание таблицы*** – расположение таблицы на странице.

<TABLE ALIGN= значение>, возможные значения: left/center/right

Значения left/right определяют выравнивание таблицы влево и вправо на странице с соответственным обтекание текста. Значение center выравнивает таблицу по центру страницы. В этом случае текст не обтекает таблицу.

***Выравнивание содержимого ячеек и строк***

<TD ALIGN= значение VALIGN= значение>.

Атрибут ALIGN выравнивает содержимое ячеек по горизонтали и может иметь значения: left/center/right.

Атрибут VALIGN выравнивает содержимое ячеек по вертикали и может иметь значения: top/middle/bottom – по верхнему краю/по центру/нижнему краю в ячейке.

По умолчанию горизонтальное выравнивание – ALIGN= left, вертикальное – VALIGN= middle.

Если задать выравнивания по горизонтали и вертикали для строки таблицы:

<TR ALIGN=center VALIGN=top>, то содержимое всех ячеек в этом ряду будет выровнено по центру и прижато к верхней границе ячейки. Таким образом, не нужно указывать эти атрибуты у каждой ячейки.

***Задание заднего фона таблицы, рядов и ячеек***

* <TABLE BGCOLOR=цвет BACKGROUND= «путь к картинке»> – задание фона для всей таблицы, определяет задний фон всех ячеек таблицы.
* <TR BGCOLOR=цвет > – задание фона для строки таблицы, определяет задний фон всех ячеек таблицы.
* < TD BGCOLOR=цвет > – задание фона для ячейки таблицы.

Цвета могут быть заданы шестнадцатиричным RGB-значениями или стандартными названиями.

Обратите внимание, что цвет, заданный в ячейке перекрывает цвет, установленный в строке, который в свою очередь перекрывает цвет, установленный в таблице.

<BORDRECOLOR=цвет> – цвет границы таблицы

*Например*:

<table border = "1"> <!--начало содержимого таблицы-->

<caption>Элементарная таблица</caption> <!-- наименование таблицы-->

<tr> <!--описываем первую строку-->

<th>Ячейка заголовка 1</th><th>Ячейка заголовка 2</th><th>Ячейка заголовка 3</th>

</tr>

<tr> <!--описываем вторую строку-->

<td>Ячейка данных 1 Строка 2</td><td>Ячейка данных 2 Строка 2</td><td>Ячейка данных 3 Строка 2</td>

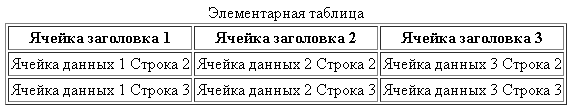
</tr>

<tr> <!-- описываем третью строку-->

<td>Ячейка данных 1 Строка 3</td><td>Ячейка данных 2 Строка 3</td><td>Ячейка данных 3 Строка 3</td>

</tr>

</table> <!-- конец таблицы-->



***Перекрывание строк и колонок (объединение)***

<TD ROWSPAN=число строк COLSPAN=число колонок>

Атрибут ROWSPAN описывает число строк, перекрываемых ячейкой. Атрибут COLSPAN описывает число колонок, перекрываемых ячейкой.

*Например*:

<table border = "1"> <!-- начало содержимого таблицы-->

<caption>Элементарная таблица</caption> <!--наименование таблицы-->

<tr> <!--описываем первую строку-->

<th colspan = "2">Ячейка заголовка 1</th><!--растягиваем на 2 ячейки-->

<th>Ячейка заголовка 2</th>

</tr>

<tr><!--описываем вторую строку-->

<td rowspan = "2">Ячейка данных 1 Строка 2</td><td>Ячейка данных 2 Строка 2</td><td>Ячейка данных 2 Строка 3</td>

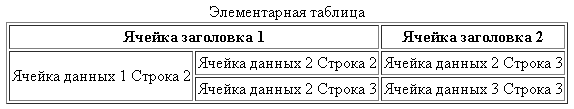
</tr>

<tr> <!--описываем третью строку (первая ячейка занята предыдущей строкой) -->

<td>Ячейка данных 2 Строка 3</td><td>Ячейка данных 3 Строка 3</td>

</tr>

</table> <!--конец таблицы-->



# Гиперссылки

С помощью ссылки устанавливается связь с другим (или с тем же) документом. Ссылкой может быть фрагмент теста или изображение. По умолчанию текстовая ссылка подчеркнута. При подведении курсора мыши к гипертекстовой ссылке он превращается в раскрытую ладонь. При нажатии на ссылку браузер обращается к серверу, запрашивая документ, который указан в HTML-теге ссылки.

Тег <**A**>…</**A**> используется для создания гипертекстовой ссылки.

Некоторые теги **используются только совместно с атрибутами** (бесполезны сами по себе) и это снова тот случай. Атрибут href (аббревиатура от *hyper reference*) - это основной атрибут тега <A>, указывающий браузеру, **URL-адрес страницы**, куда ведет ссылка. **Единый указатель ресурса** (англ. *Uniform Resource Locator*, URL) — единообразный локатор (определитель местонахождения) ресурса.

*Например*:

|  |  |
| --- | --- |
| <a href = "http://www.yandex.ru">Найдется всё</a> | [Найдется всё](http://www.yandex.ru) |

В приведенном примере используется абсолютный адрес пути. Название абсолютный связано с тем, что мы задаем полный веб-адрес страницы в качестве пути к файлу. Ниже приведены примеры указания абсолютных путей:

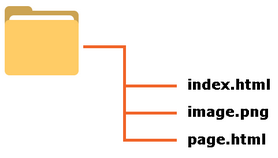
<a href = "https://www.yandex.ru">Содержимое элемента</a> <!-- указываем полный путь (протокол https) -->

<a href = "http://www.yandex.ru">Содержимое элемента</a> <!-- указываем полный путь (протокол http) -->

<a href = "//www.yandex.ru">Содержимое элемента</a> <!-- допускается указывать полный путь без протокола -->

В большинстве случаев, абсолютные пути ссылок применяются для страниц или файлов, которые находятся на других ресурсах (располагаются не на том же сервере, что и страница с которой ссылаются).

Давайте рассмотрим аналогичный пример, но также укажем ссылку на html документ, расположенный в той же папке, что и сам файл:



<!DOCTYPE html>

<html>

<head>

<title>Пример подключения файлов к HTML документу из той же папки</title>

</head>

<body>

<img src = "image.png" alt = "someimage info"> <!-- размещаем изображение -->

<a href = "page.html">Содержимое элемента</a> <!-- ссылаемся на страницу -->

</body>

</html>

**Внутренние ссылки** – это ссылки на объекты в пределах одного документа. В то место, куда указывает ссылка необходимо поместить **якорь** – тег <A NAME=”ИМЯ ЯКОРЯ”>…</A>

Тег внутренней ссылки <A HREF =”#ИМЯ ЯКОРЯ”>…</A>

Атрибут **TARGET** используется для того, чтобы указать браузеру, где необходимо открыть web-страницу.

Следующий пример открывает документ в новом окне браузера:

<a href="http://yandex.ru" target="\_blank">Поиск </a>

Ниже представлены все значения этого атрибута:

|  |  |
| --- | --- |
| **Атрибут** | **Значение** |
| \_blank | Загружает страницу в новое окно браузера. |
| \_self | Загружает страницу в текущее окно. **Это значение по умолчанию.** |

Для того, чтобы использовать изображение в качестве ссылки, вам необходимо поместить изображение между тегами <A> </A>:

<a href = "http://yandex.ru">

<img src="10.jpg" alt="Поиск в Яндекс"> <!-- размещаем изображение в качестве содержимого гиперссылки -->

</a>

Для перехода по ссылкам внутри страницы:

|  |  |
| --- | --- |
| Ссылка для перехода:  <A HREF=#1.html> Описание</A> | *Якорь*:  <A Name=1> Описание</A> |

Для перехода по ссылкам между страницами:

|  |  |
| --- | --- |
| На странице откуда переходим:  <A HREF=2.html> Номера и цены</A> | На странице куда переходим:  <A Name=1> Номера и цены</A> |