Язык разметки HTML

# Язык HTML

Язык HTML (HyperText Markup Language - гипертекстовый язык разметки) был разработан Тимом Бернерс-Ли во время его работы в CERN. Язык был придуман, как одна из компонент технологии WWW.

На сегодняшний день актуальной является версия HTML5.

# Структура HTML-документа

Пример документа HTML.

<!DOCTYPE HTML .” .dtd”> - ссылка на DTD с правилами описания тэгов (конструкция из SGML), обычно ее не используют

<HTML>

<HEAD>

<TITLE>Наш первый документ на языке HTML</TITLE>

</HEAD>

<BODY>

<P>Пример!

</BODY>

</HTML>

В HTML5 используется конструкция

<!DOCTYPE html>

Это признак использования HTML5.

Документ HTML состоит из раздела заголовка документа (между тэгами <HEAD> и </HEAD>) и тела документа (между тэгами <BODY> и </BODY>). Название документа (между тэгами <TITLE> и </TITLE >) отображается в заголовке окна броузера, а содержимое документа (тело документа) отображается в окне броузера. В этом примере тело документа состоит только из одного параграфа текста, (тег <P>).

|  |
| --- |
| Наш первый документ на языке HTML |
| Пример! |

В разделе заголовка документа может также располагаться невизуальная информация – список ключевых слов для поисковых машин, программы (сценарии) на Java Script или Visual Basic Script и др.

Данный документ может быть представлен в виде дерева.

|  |  |  |
| --- | --- | --- |
| HTML |  |  |
| |---- | HEAD |  |
| | | |---- | TITLE |
| | |  |  |
| |---- | BODY |  |
| | | |---- | P |

# Основные типы конструкций HTML (относится ко всем тэговым языкам – SGML, XML)

HTML является тэговым языком – содержит тэги, элементы, атрибуты.

**3.1. Элементы**

Элементы – основная конструкция документов HTML.

Каждое объявление элемента обычно включает три части: начальный тэг, содержимое и конечный тэг.

Имя элемента отображается в начальном тэге (пишется <имя-элемента>) и в конечном тэге (пишется </имя-элемента>), перед именем элемента в конечном тэге ставится прямой слеш. Например, начальные и конечные тэги элемента UL определяют маркированный список:

<UL>

<LI><P>...элемент списка 1...

<LI><P>...элемент списка 2...

</UL>

(LI – элемент списка)

Содержимым элемента могут быть текст, другие элементы, комментарии и т.д.

Некоторые типы элементов HTML позволяют опускать конечные тэги (например, типы элементов P и LI). Несколько типов элементов также позволяют опускать начальные тэги; например, HTML HEAD и BODY.

Некоторые типы элементов HTML не имеют содержимого. Например, элемент перехода на следующую строку <BR> не имеет содержимого; его роль - прерывание строки текста. Такие пустые элементы никогда не имеют конечных тэгов.

***Элементы и тэги - это не синонимы.*** Элемент определяется тэгами (начала или конца). Например, элемент HEAD всегда присутствует, даже если начальный и конечный тэги HEAD отсутствуют.

**3.2. Атрибуты**

С элементами могут быть связаны свойства, называемые атрибутами, которые могут иметь значения.

**Пример использования атрибутов.**

<H1 id="section1">заголовок</H1>

Пары атрибут/значение помещаются перед закрывающей скобкой ">" начального тэга элемента. В начальном тэге элемента может быть любое число атрибутов, разделенных пробелами. Они могут указываться в любом порядке.

В данном примере для элемента H1 установлен атрибут id.

По умолчанию необходимо, чтобы все значения атрибутов были разделены с помощью двойных или одинарных кавычек. Одинарные кавычки могут включаться в значение атрибута, если значение отделяется двойными кавычками, и наоборот.

В HTML использование кавычек необязательно, хотя и рекомендутся.

Некоторые атрибуты играют роль логических переменных (например, атрибут selected для элемента OPTION). Их наличие в начальном тэге элемента подразумевает, что значением атрибута является "истина". Их отсутствие означает "ложь".

В HTML логические атрибуты могут быть в минимизированной форме - в начальном тэге элемента находится только **значение** атрибута. Таким образом, selected можно установить, написав:

<OPTION selected>

**3.3. Ссылки на символы**

Ссылки на символы - это имена символов, которые могут быть включены в документ HTML. Они удобны для обращения к редко используемым символам или к символам, которые трудно или невозможно вводить в средствах разработки документов. Ссылки на символы начинаются со знака "&" и заканчиваются точкой с запятой (;).

**Примеры:**

1. "&lt;" представляет знак <.
2. "&gt;" представляет знак >.
3. "&quot;" представляет знак ".
4. "&#229;" (десятичное число) представляет букву "a" с кружком сверху.
5. "&#xE5;" (шестнадцатеричное число) представляет букву "a" с кружком сверху.

3.4. Комментарии

Комментарии в HTML имеют следующий синтаксис:

<!-- это комментарий -->

<!-- это тоже комментарий,

он занимает несколько строк -->

# Основные элементы языка HTML

**Элемент HTML**

Корневой элемент документа

Начальный тег: ***необязателен***, Конечный тег: ***необязателен***

**Элемент HEAD**

Секция заголовка документа

Начальный тег: ***необязателен***, Конечный тег: ***необязателен***

3. Элемент TITLE

название документа (отображается в окне броузера)

Начальный тег: ***обязателен***, Конечный тег: ***обязателен***

4. Элемент BODY

Определение тела документа

Начальный тег: ***необязателен***, Конечный тег: ***необязателен***

Атрибуты:

background = [uri](file:///D:\root\yura\institute\повышение_квалификации\2017_10_30\черновики\html\types.html#type-uri)

Значение этого атрибута - URI, указывающий на изображение. Это изображение является фоном страницы.

URI (URL, URN) *(Universal Resource Identifier) - универсальный идентификатор ресурсов* (вторая компонента Web, придуманная Бернерсом-Ли). Адрес какого-либо ресурса в Интернет.

В простейшем виде URI – имя файла.

text = [[*цвет*](file:///D:\root\yura\institute\повышение_квалификации\2017_10_30\черновики\html\types.html#type-color)](file:///D:\root\yura\institute\повышение_квалификации\2017_10_30\черновики\html\types.html#type-color)

Этот атрибут устанавливает цвет текста.

link = [*цвет*](file:///D:\root\yura\institute\повышение_квалификации\2017_10_30\черновики\html\types.html#type-color)

Этот атрибут устанавливает цвет текста гипертекстовых ссылок, по которым не было переходов.

vlink = [*цвет*](file:///D:\root\yura\institute\повышение_квалификации\2017_10_30\черновики\html\types.html#type-color)

Этот атрибут устанавливает цвет текста ссылок, по которым были переходы.

alink = [*цвет*](file:///D:\root\yura\institute\повышение_квалификации\2017_10_30\черновики\html\types.html#type-color)

Этот атрибут устанавливает цвет текста текущей выбранной ссылки.

bgcolor = [*цвет*](file:///D:\root\yura\institute\повышение_квалификации\2017_10_30\черновики\html\types.html#type-color)

Этот атрибут устанавливает цвет фона тела документа.

Эти атрибуты отменены в HTML5.

**Как задается цвет?**

Цвет можно задавать строковой константой или 16-ным значением.

(White = "#FFFFFF" , Red = "#FF0000" , Green = "#00FF00" Blue = "#0000FF")

Кодирование цвета: #RRGGBB – 3 байта, каждый принимает значение от 00 до FF, где RR – интенсивность красного цвета, GG – интенсивность зеленого цвета, BB – интенсивность синего цвета.

Пример:

<HTML>

<HEAD>

<TITLE>пример</TITLE>

</HEAD>

<BODY bgcolor="white" text="black" link="red" alink="fuchsia" vlink="maroon"> ... тело документа... </BODY> </HTML>

В данном примере устанавливается белый цвет фона, черный цвет текста и красный цвет непосещенных гиперссылок, цвет фуксии (похож на фиолетовый) при активизации ссылок и коричневый для ссылок, по которым были переходы.

**Объявление заголовков: Элементы H1, H2, H3, H4, H5, H6**

Начальный тег: ***обязателен***, Конечный тег: ***обязателен***

В языке HTML существует шесть уровней заголовков: H1 - наиболее важный - и H6 - наименее важный. Обычно браузеры отображают более важные заголовки более крупным шрифтом.

Пример:

<BODY>

<H1>Заголовок 1-го уровня</H1>

<H2>Заголовок 2-го уровня</H2>

<P>Текст</P>

</BODY>

Атрибуты:

align = left|center|right|justify

Этот атрибут задает горизонтальное выравнивание элемента. Возможные значения:

1. left: строки текста выравниваются по левому краю.
2. center: строки текста выравниваются по центру.
3. right: строки текста выравниваются по правому краю.
4. justify: строки текста выравниваются по обоим краям.

Пример:

<H1 align="center"> Заголовок 1-го уровня по центру </H1>

**Параграфы элемент P**

Начальный тег: ***обязателен***, Конечный тег: ***необязателен***

Элемент P представляет параграф текста.

Атрибуты: align.

Примеры:

<P align=center>Центрированный текст.

<P align="right">Правый край.</P>

**Элемент BR**

Принудительный переход на новую строку

Начальный тег: ***обязателен***, Конечный тег: ***запрещен***

Задает точку переноса, а не элемент, поэтому не может иметь содержимого.

**Элемент HR**

Горизонтальный разделитель (линия)

Начальный тег: ***обязателен***, Конечный тег: ***запрещен***

Атрибуты:

1. noshade (логический) – отсутствие тени (без углубления, как обычная линия)

2. size = высота в пикселах

3. width = ширина в пикселах или процентах

4. align = left|center|right

Примеры:

<HR>

<HR width="50%" align="center">

<HR noshade size="50" width="50" align="left"> - серый квадрат слева 50x50 пикселов

В HTML5 атрибуты эти атрибуты не поддерживаются, считается что верстку необходимо делать с помощью стилей.

**Вставка звука и видео на страницу**

Оба способа не определены в стандарте и являются расширениями броузера.

**Элемент EMBED**

Элемент Embed отображается визуально в виде панели для проигрывания звука или видео.

Атрибуты:

1. src = URI звукового файла или видео (AVI)

2. width = ширина (px или %)

3. height = высота

4. hidden = true | false (не отображать визуально)

5. autostart = true | false (автозапуск)

6. loop = true | false (проигрывать в цикле)

С помощью EMBED в HTML5 вcтавляются только плагины. Атрибуты:

src

type

height

width

**Элемент audio (новый, появился в HTML 5)**

Внутри элемента можно писать текст, который отображается, если браузер не поддерживает этот элемент.

Не поддерживается Firefox, Google Chrome поддерживает.

Атрибуты::

1. src = URI файла

2. autoplay="autoplay" - автоматическое проигрывание при загрузке документа

3. controls="controls" - отображение кнопок запуска, остановки и т.д.

4. loop = "loop" - проигрывание в цикле

<audio src="1.mp3" controls="controls" loop="loop" >  
Браузер не поддерживает элемент audio  
</audio>

Атрибут controls (и другие) – это логические атрибуты (то есть они или присутствуют или нет). Но в HTML5 они записываются в форме имя=значение.

**Элемент video (новый, появился в HTML 5)**

Элемент VIDEO используется для вставки видеофрагментов.

Атрибуты такие же как в элементе AUDIO, есть еще атрибуты для задания ширины и высоты: width = ширина (в пикселах или %) height = высота (в пикселах или %)

<video src="1.wmv" controls="controls">  
Браузер не поддерживает элемент video  
</video>

# Элементы отображения шрифтов и разметки текста

Элементы управления шрифтами: FONT и BASEFONT – **НЕ ПОДДЕРЖИВАЮТСЯ В HTML5,** но могут встречаться в существующих сайтах.

Элемент FONT изменяет размер и цвет шрифта для содержимого элемента.

Начальный тег: ***обязателен***, Конечный тег: не ***обязателен***

Атрибуты:

size = число

Устанавливает размер шрифта. Возможные значения:

1. Абсолютная шкала. Целое число от 1 до 7. Устанавливает фиксированный размер шрифта, представление которого зависит от броузера.
2. Относительная шкала. Значение "+1" означает на один размер больше по сравнению с базовым. Значение "-3" означает шрифт на три размера меньше. Все размеры относятся к шкале от 1 до 7. По умолчанию, базовый размер шрифта = 3. Базовый размер шрифта можно задать с помощью элемента <BASEFONT>.

color = [цвет](../types.html#type-color) текста

face = строка названий шрифтов

Определяет список разделенных запятыми названий шрифтов (в порядке приоритета).

Пример:

<FONT SIZE=”5” COLOR=”GREEN” FACE="ARIAL"><P>ARIAL</P></FONT>

<FONT SIZE=”-2” COLOR=”BROWN” FACE="TIMES,ARIAL"><P>TIMES</P></FONT>

Элемент [BASEFONT](../present/graphics.html#edef-BASEFONT) устанавливает базовые параметры шрифта (размер, цвет, вид шрифта). Элемент [BASEFONT](../present/graphics.html#edef-BASEFONT) может быть указан несколько раз на странице.

Начальный тег: ***обязателен***, Конечный тег: не обязателен.

Если конечный тэг не указан, то действие элементов FONT или BASEFONT распространятется до конца документа.

Атрибуты у элемента <BASEFONT> такие же, как у элемента <FONT>, за исключением того, что в атрибуте SIZE можно указывать только значения по абсолютной шкале.

Пример:

<BASEFONT SIZE="5" COLOR="GREEN" FACE="Times,Arial">

<P>Размер 5</P>

<FONT SIZE=-1><P>Размер 4</P></FONT>

<FONT SIZE=-2><P>Размер 3</P>

**Элементы разметки текста**

**TT -** Представляется как моноширинный шрифт (шрифт пишущей машинки). (ВИДЫ шрифтов: моноширинные, пропорциональные; с засечками, без засечек).

**I -** Представляет курсивом.

**B -** Представляется полужирным шрифтом.

**BIG -** Представляется "крупным" шрифтом. (рекурсивное действие)

**SMALL -** Представляется "малым" шрифтом. (рекурсивное действие)

**STRIKE** и **S -** Представляется перечеркнутым шрифтом.

**U -** Представляется подчеркнутым шрифтом. (лучше использовать осторожно, чтобы пользователь не воспринял как гиперссылку)

Пример:

<P><b>полужирный</b>, <i>курсив</i>, <b><i>полужирный курсив</i></b>, <tt>моноширинный</tt>, <big>большой</big> и <small>малый</small> текст.

Могут употребляться внутри большинства элементов, например внутри параграфа или заголовка.

**P** – Параграф текста

**EM –** Выделение (обычно курсив).

**STRONG -** Более сильное выделение (обычно полужирный шрифт).

**DFN –** определение термина.

**CODE –** Текст программы.

**SAMP -** пример

**KBD -** Текст, который должен ввести пользователь.

**VAR -** переменная

**ABBR -** аббревиатура (например, WWW, HTTP, URI, и т.д.).

**ACRONYM -** Акроним (например, радар и т.д.).

**ADDRESS** – Информация об адресе

**CITE -** Цитата. (как правило отображается курсивом)

**BLOCKQUOTE –** длинная цитата (более одного параграфа). Как правило, отображается с отступом.

**Q -** короткая цитата (используется внутри параграфа) По стандарту должна отображаться в кавычках, но это обеспечивается не всеми броузерами.

**SUB -** Нижний индекс

**SUP -** Верхний индекс

**PRE** - Форматированный текст. Отображается моноширинным шрифтом. Внутри элемента PRE учитываются все пробелы и символы перехода на следующую строку. При отображении большинства других элементов эти символы не учитываются.

Начальный тег: ***обязателен***, Конечный тег: ***обязателен***

Данные элементы представляют собой элементы логической разметки текста. Их использование необязательно, но если мы используем, например, элемент CODE, броузер будет стараться показать его содержимое так, что это будет в наибольшей степени похоже на текст программы. Позволяют быстро достигнуть желаемого результата при создании простых документов.

Атрибуты:

title = текст (отображается в виде всплывающей посдказки). Атрибут title присутствует у большинства элементов HTML, в том числе у рассмотренных элементов P и H1..H6.

Примеры:

<P>Текст с <U title="подсказка">всплывающей подсказкой</U></P>

<P>Текст с <FONT title="подсказка">всплывающей подсказкой</FONT></P>

H<sub>2</sub>O

E = mc<sup>2</sup>

<P>

1-я строка без отступа.

Тоже 1-я строка.

</P>

<P>

&nbsp;&nbsp;&nbsp;1-я строка с отступом.

Тоже 1-я строка.

</P>

<PRE>

1-я строка с отступом.

2-я строка.

</PRE>

# Элементы создания списков

**Упорядоченные и неупорядоченные списки.**

UL – неупорядоченный (маркированный) список

Начальный тег: ***обязателен***, Конечный тег: ***обязателен***

OL – упорядоченный (нумерованный) список

Начальный тег: ***обязателен***, Конечный тег: ***обязателен***

LI (list item) – элемент маркированного или нумерованного списков

Начальный тег: ***обязателен***, Конечный тег: не***обязателен***

Упорядоченные и неупорядоченные списки генерируются одинаково за исключением того, что броузер нумерует элементы упорядоченных списков. Эти номера могут представляться несколькими способами. Элементы неупорядоченного списка помечаются маркером.

Оба эти типа списков состоят из последовательностей элементов списков LI.

Атрибуты:

type = строка

Устанавливает стиль элемента списка.

Для элемента [UL](../struct/lists.html#edef-UL) возможными значениями атрибута [type](../struct/lists.html#adef-type-LI) являются disc, square и circle. Значение, используемое по умолчанию, зависит от уровня вложенности текущего списка. Эти значения не учитывают регистр. Представление каждого значения зависит от броузера. Как правило, броузер отображает "disc" в виде небольшого заполненного кружка, "circle" - в виде окружности, а "square" в виде небольшого квадрата.

Для элемента [OL](../struct/lists.html#edef-OL) возможные значения атрибута [type](../struct/lists.html#adef-type-LI) следующие (учитывается регистр):

1 - арабские цифры

a - буквы нижнего регистра

A – буквы верхнего регистра

i – римские цифры в нижнем регистре

I – римские цифры в верхнем регистре

start = [*число*](../types.html#type-number)

Только для [OL](../struct/lists.html#edef-OL). Задает начальный номер первого элемента в упорядоченном списке.

value = [*число*](../types.html#type-number)

Только для [LI](../struct/lists.html#edef-LI). Устанавливает номер текущего элемента списка. Используется в том случае, когда необходимо указать номера элементов списка, отличающиеся от автоматической нумерации.

compact

Список становится более компактным по вертикали, если это возможно.

Пример:

<OL type="1">

<LI><P>Пункт 1

<UL>

<LI>1.1

<LI>1.2

</UL>

<LI><P>Пункт 2

<OL type="I">

<LI>2.1

<LI>2.2

</OL>

</LI>

</OL>

Результат:

1. Пункт 1

* 1.1
* 1.2

2. Пункт 2

I. 2.1

II. 2.2

<ol>

<li value="30">N 30.

<li value="40">N 40.

<li>N 41.

</ol>

Результат:

30. N 30

40. N 40

41. N 41

**Список определений ( элементы DL, DT и DD)**

Отображается отступами.

DL - список определений

Начальный тег: ***обязателен***, Конечный тег: ***обязателен***

DT - определяемый термин

DD - определение

Начальные тег: ***обязателен***, Конечный тег: ***не обязателен***

Пример:

<DL>

<DT>Термин 1 <DD>Определение 1

<DT>Термин 2 <DD>Определение 2

</DL>

( Результат:

Термин 1

Определение 1

Термин 2

Определение 2

)

# Гипертекстовые ссылки

Гипертекстовая ссылка состоит из двух якорей (anchors). Ссылка начинается в "исходном" якоре (источнике) и указывает на "целевой" якорь, который может быть любым ресурсом Интернет (например, изображением, звуковым файлом, программой, документом HTML, элементом в документе HTML и т.д.).

Гипертекстовая ссылка задается с помощью элемента <A>.

Если для элемента [A](../struct/links.html#edef-A) установлен атрибут [href](../struct/links.html#adef-href), этот элемент определяет исходный якорь.

Пример:

<A href="http://iu5.bmstu.ru">Сайт ИУ-5</A>

Конечным якорем является главная страница сайта ИУ-5, его задает атрибут href.

Как правило, броузер подчеркивает содержимое элемента <A> и выделяет его синим цветом, определяя как гипертекстовую ссылку.

Задание конечных якорей гиперссылки:

1. Конечный якорь задается как URL, указывающий на ресурс Интернет.

<A href="picture.gif" title=”изображение”>ссылка на изображение</A>

2. Конечный якорь задается в атрибуте “name” элемента <A>.

<A href="#ancor1">начальный якорь</A>

...

<A name="ancor1">конечный якорь</A>

3. Конечный якорь задается в атрибуте ID любого элемента HTML.

Эта возможность появилась в последних версиях спецификации HTML.

<A href="#id\_h1">начальный якорь</A>

...

<H1 id="id\_h1">Заголовок</H1>

Можно устанавливать атрибуты [name](../struct/links.html#adef-name-A) и [href](../struct/links.html#adef-href) одновременно в одном элементе [A](../struct/links.html#edef-A).

Каким образом задается URL в атрибуте href:

1. С помощью абсолютного URL: http://www.mycompany.com/one.html#anchor-one

2. С помощью относительного URL: one.html#anchor-one или ../two/two.html#anchor-two

(если документ располагается на том же сайте)

3. Если якорь определен в текущем документе: #anchor-one

По возможности надо стараться использовать относительные ссылки.

Имена якорей должны быть уникальны, регистр не учитывается.

Ссылки и якоря не могут быть вложенными; элемент А не должен содержать других элементов А.

# Создание таблиц

Пример таблицы:

|  |  |  |
| --- | --- | --- |
| Языки разметки | | |
|  | **Языки разметки** | |
| **HTML** | **XML** |
| **Год создания** | 1990 | 1998 |

<TABLE border=1>

<CAPTION>Языки разметки</CAPTION>

<TR><TH rowspan=2><TH colspan=2>Языки разметки

<TR><TH>HTML</TH><TH>XML</TH></TR>

<TR><TH>Год создания<TD>1990<TD>1998

</TABLE>

Элементы, используемые для создания таблиц:

**Элемент TABLE - таблица**

*Начальный тег:* ***обязателен****, конечный тег:* ***обязателен***

Атрибуты:

1. align = left|center|right . Горизонтальное выравнивание таблицы в документе.

2. width = ширина таблицы (в пикселах или %). Если значение атрибута меньше минимально возможной ширины таблицы, то таблица отображается целиком с минимально возможной шириной.

3. bgcolor = цвет фона

4. border = ширина внешней границы таблицы в пикселах

5. frame = void|above|below|hsides|vsides|lhs|rhs|box|border

Атрибут определяет видимость внешних границ таблицы.

1. void: Границ нет. По умолчанию, если не задан border.
2. above: Только верхняя граница.
3. below: Только нижняя граница.
4. hsides: Только верхняя и нижняя границы.
5. vsides: Только левая и правая границы.
6. lhs: Только левая граница.
7. rhs: Только правая граница.
8. box, border: Все четыре границы. По умолчанию, если border>0

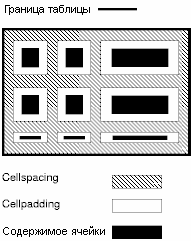
6. rules = none|groups|rows|cols|all

Атрибут определяет видимость внутренних границ между ячейками таблицы.

1. none: Нет границ. По умолчанию, если не задан border.
2. rows: Границы отображаются только между строками.
3. cols: Границы отображаются только между столбцами.
4. all: Отображаются все внутренние границы. По умолчанию, если border>0

7. cellspacing = толщина внутренней границы (в пикселах)

8. cellpadding = отступ между внутренней границей и текстом ячейки (в пикселах)



**Элемент CAPTION – подпись таблицы (необязательный).**

*Начальный тег:* ***обязателен****, конечный тег:* ***обязателен***

Атрибуты:

align = top|bottom|left|right

Расположение подписи сверху по центру (по умолчанию), снизу по центру, сверху слева или сверху справа от таблицы.

Элемент CAPTION должен располагаться только непосредственно после начального тега [TABLE](../struct/tables.html). Элемент [TABLE](../struct/tables.html) может включать только один элемент [CAPTION](../struct/tables.html).

**Элемент TR – строка таблицы.**

*Начальный тег:* ***обязателен****, Конечный тег:* ***не обязателен***

Атрибуты:

1. align = left|center|right

Горизонтальное выравнивание в ячейках строки.

2. valign = top|middle|bottom

Вертикальное выравнивание в ячейках строки.

1. top: Выравнивание по верхнему краю ячейки.
2. middle: Выравнивание по середине ячейки. (По умолчанию.)
3. bottom: Выравнивание по нижнему краю ячейки.

3. height = высота строки в пикселах (или в % от высоты таблицы)

**Ячейки таблицы: элементы TH и TD**

*Начальный тег:* ***обязателен****, Конечный тег:* ***не обязателен***

Могут использоваться только внутри элемента TR. TD - ячейка данных таблицы, TH - ячейка заголовка таблицы. TD как правило отображается обычным шрифтом с выравниванием влево, а TH – полужирным с выравниванием по центру.

Атрибуты:

1. rowspan = число объединяемых ячеек по вертикали

2. colspan = число объединяемых ячеек по горизонтали

Атрибуты rowspan и colspan могут быть использованы совместно для объединения ячеек по горизонтали и по вертикали.

Пример: <TD rowspan=2 colspan=2>Четыре ячейки (2х2) </TD>

3. nowrap (логический). Запрет переноса слов внутри ячейки. Текст ячейки располагается в одну строку.

4. width = ширина ячейки в пикселах (или % от ширины таблицы)

5. height = высота ячейки в пикселах

Также могут быть заданы атрибуты [bgcolor](../present/graphics.html) [align](../struct/tables.html), [valign](../struct/tables.html), которые перекрывают соответствующие атрибуты элементов TABLE и TR.

# Изображения в документах HTML

Для вставки изображений в документ HTML используется элемент IMG.

*Начальный тег:* ***обязателен****, Конечный тег:* ***запрещен***

Атрибуты элемента IMG:

1. src = [uri](../types.html)

Атрибут задает адрес (URI) изображения. Как правило, является ссылкой на файл в формате GIF или JPEG.

2. alt = строка

Альтернативный текст, который выводит броузер, если он не может найти или показать изображение. Пользователь может использовать текстовый броузер (например LYNX). Во многих графических броузерах (например в Netscape) также существует возможность явного отключения вывода изображений (для ускорения загрузки страниц).

Атрибуты src и alt являются обязательными.

3. width = ширина (в пикселах или %)

4. height = высота (в пикселах или %)

Если указаны атрибуты [width](../struct/objects.html) и [height](../struct/objects.html), то они переопределяют исходный размер изображения. Броузер масштабирует изображение так, чтобы оно соответствовало заданной ширине и высоте.

Если ширина или высота заданы в процентах, то процент вычисляется не от исходного размера изображения, а от размера окна броузера.

Если задается только один параметр, то второй, как правило, устанавливается так, чтобы сохранились пропорции исходного изображения. (IE)

5. border = ширина границы в пикселах

6. hspace = горизонтальный отступ справа и слева (в пикселах или %)

7. vspace = вертикальный отступ сверху и снизу (в пикселах или %)

8. align = bottom | middle | top | left | right

Выравнивание изображения в тексте.

1. bottom: Изображение выравнивается по вертикали по нижней границе текущей строки текста. (По умолчанию).

|--|

111111111|--|1111111111

1. middle: Изображение выравнивается по вертикали по середине текущей строки текста.

|--|

111111111|--|1111111111

|--|

1. top: Изображение выравнивается по вертикали по верхней границе текущей строки текста.

111111111|--|1111111111

|--|

1. left или right: Изображение перемещается к левой или правой границе окна броузера. Текст может обтекать изображение.

111111

|--|11

|--|11

111111

9. usemap = имя навигационной карты

Связывает изображение с клиентской навигационной картой, задаваемой с помощью элемента [MAP](../struct/objects.html). Значение атрибута usemap должно совпадать со значением атрибута [name](../struct/objects.html) элемента [MAP](../struct/objects.html).

10. ismap (логический). Задает использование серверной навигационной карты.

Примеры:

<IMG src=1.gif alt=картинка>

<IMG src=1.gif alt=картинка width=100 height=50>

<IMG src=1.gif alt=картинка border=10>

<P>Текст</P>

<!-- 100 px -->

<IMG src=1.gif alt=картинка vspace=100>

<!-- 100 px -->

<P>Текст</P>

<P>Текст<IMG src=1.gif hspace=100>Текст</P>

|--|

Текст |--| Текст

<P>Текст<IMG src=1.gif align=middle>Текст</P>

|--|

Текст|--|Текст

|--|

<P>Текст<IMG src=1.gif align=middle hspace=100>Текст</P>

|--|

Текст |--| Текст

|--|

В HTML5 появился новый элемент CANVAS с атрибутами:

* id=id элемента (для удобной ссылки в JavaScript)
* width = ширина (в пикселах или %)
* height = высота (в пикселах или %)

В этот элемент нельзя загрузить изображение, на нем можно только рисовать с помощью JavaScript.

# Навигационные карты

Навигационные карты позволяют определять области изображения и назначать каждой области определенный URI (конечный якорь гиперссылки).

Пример:

----------------

| URI 1 | URI 2|

----------------

| URI 3 | URI 4|

----------------

Существует два типа навигационных карт:

1. *Клиентская.* Координаты точки интерпретируются броузером, который выбирает соответствующую гиперссылку.
2. *Серверная.* Координаты точки передаются сценарию на сервере, который интерпретирует координаты и выполняет соответствующие действия.

Как правило, предпочтение отдается клиентским навигационным картам. Серверные карты используются только в случае очень сложных карт (когда трудно задать все области) или в том случае, когда необходимо произвести какие-либо действия на стороне сервера (например, обращение к БД).

**Клиентские навигационные карты**

Задаются с помощью элемента MAP и вложенных элементов AREA.

Элемент MAP определяет клиентскую навигационную карту.

*Начальный тег:* ***обязателен****, Конечный тег:* ***обязателен***

*Атрибуты:*

name = имя навигационной карты (обязательный)

Элемент AREA определяет область клиентской навигационной карты

*Начальный тег:* ***обязателен****, Конечный тег:* ***запрещен***

*Атрибуты:*

1. shape = default|rect|circle|poly

Определяет форму области. Возможные значения:

1. default: Задает всю область.
2. rect: Определяет прямоугольную область.
3. circle: Определяет круглую область.
4. poly: Определяет многоугольную область.

2. coords = *координаты* точек

Число и порядок значений зависят от атрибута shape. Координаты задаются относительно верхнего левого угла.

1. rect: x левой границы, y верхней границы, x правой границы, y нижней границы. (X1,Y1,X2,Y2)
2. circle: x центра, y центра, радиус. (Xc,Yc,R)
3. poly: x1, y1, x2, y2, ..., xN, yN.

Первая и последняя точка многоугольника автоматически замыкаются.

3. href = URI

Конечный якорь гиперссылки.

4. nohref (логический)

Обозначает, что с этой областью гиперссылка не связана. (Не используется совместно с href).

Возможно также использование атрибутов alt и title. Содержимое обоих атрибутов отображается как всплывающая подсказка. Атрибут alt может использоваться текстовыми броузерами для создания текстовых ссылок вместо графической навигационной карты.

Если несколько областей AREA перекрываются, то приоритет имеет первый элемент AREA.

Возможно использование одной навигационной карты с различными изображениями.

<IMG src=image.gif height=200 width=200 usemap=#map1>

<MAP name="map1">

<AREA href="1.html" shape="rect" coords="0,0,50,40" alt="Прямоугольник 0,0 - 50,40">

<AREA href="2.html" shape="circle" coords="150,50,50" alt="Окружность с центром 150,50 и радиусом 50">

<AREA href="3.html" shape="poly" coords="100,100,150,100,190,190" alt="Треугольник 100,100 - 150,100 - 190,190">

<AREA href="4.html" shape="default" coords="0,0,200,200" alt="Основная часть изображения">

</MAP>

<!-- Гиперссылка в виде кольца -->

<IMG src=image.gif height=200 width=200 usemap=#map1>

<MAP name="map1">

<AREA nohref shape="circle" coords="100,100,50" alt="Нет гиперссылки">

<AREA href="1.html" shape="circle" coords="100,100,100" alt="Гиперссылка">

</MAP>

**Серверные навигационные карты.**

Серверные навигационные карты, как правило, используются в случае сложных карт.

Чтобы создать серверную навигационную карту, необходимо включить элементов [IMG](../struct/objects.html) в элемент [A](../struct/links.html). Для элемента [IMG](../struct/objects.html) должен быть установлен логический атрибут ismap.

Когда пользователь активизирует гиперссылку, щелкнув на изображении, координаты точки передаются на веб-сервер.

Пример:

<A href="http://iu5.bmstu.ru/script.asp">

<IMG src=image.gif height=400 width=400 ismap>

</A>

Координаты передаются на сервер следующим образом. Броузер получает адрес URI из атрибута [href](../struct/links.html) элемента [A](../struct/links.html) и добавляет ’?’, за которым следуют координаты x и y, разделенные запятой.

Если пользователь производит щелчок по точке с координатами x=333, y=333, то происходит обращение по следующему URI: **http://iu5.bmstu.ru/script.asp?333,333**

Обработка координат целиком возлагается на серверный сценарий.

# Группировка элементов:

**Элементы DIV и SPAN**

*Атрибуты:* [*align*](../present/graphics.html)*,* [*id*](../struct/global.html)*,* [*title*](../struct/global.html)

Контейнерные элементы [DIV](../struct/global.html) и [SPAN](../struct/global.html) обеспечивают общий механизм добавления структуры в документы HTML. Определяют информацию уровня блока ([DIV](../struct/global.html)) или информацию уровня элемента блока ([SPAN](../struct/global.html)). Предполагается, что элементы SPAN могут быть вложены в DIV, хотя это не обязательно. DIV также может быть вложен в DIV.

Предполагается, что DIV должен быть отдельным блоком, а SPAN является inline-элементом.

Сами элементы визуально не отображаются (отображается только их содержимое). Элементы часто используются совместно с [таблицами стилей](../present/styles.html), сценариями на JavaScript, могут использоваться как конечный якорь гиперссылки с атрибутом id.

Пример:

<DIV align=center title=блок>

<P>1

<DIV align=right title="вложенный блок">2</DIV>

<P>3

</DIV>

**Элементы details и summary**

В HTML5 добавлены элементы, которые позволяют создавать раскрывающиеся разделы.

Элемент details создает раскрывающийся элемент, в элементе summary отображается часть, которая всегда видима.

Пример:

<details>

<summary>Видимый заголовок элемента</summary>

<div>Скрытое содержимое</div>

</details>

# Семантическая разметка

В HTML5 добавлены элементы, которые позволяют создавать семантическую (логическую) разметку документа.

* main – основное содержимое веб-страницы;
* article – законченный фрагмент документа (статья);
* section – секция документа;
* nav – содержит элементы навигации для сайта;
* header – элемент логического заголовка;
* footer – нижний заголовок;
* address – контактная информация;
* aside – боковой блок, вспомогательная информация.

Пример:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Элемент details в HTML5</title>

</head>

<body>

<header>

<h1>Заголовок сайта</h1>

</header>

<main>

<nav>

<ul>

<li><a href="...">Главная</a></li>

<li><a href="...">Контакты</a></li>

</ul>

</nav>

<div>Основная часть документа</div>

<article>

<div>Статья 1</div>

<section>Секция 1</section>

<address>Здесь должен быть адрес</address>

</article>

<aside style="float:right; width:200px;">Это должно быть где-то сбоку</aside>

</main>

<footer>

<h1>Нижний заголовок сайта</h1>

</footer>

</body>

</html>

Семантические тэги не имеют визуального поведения по умолчанию. Используются только для группировки элементов.

# Фреймы

Под фреймом (кадром) понимается фрагмент окна броузера, который выделяется как отдельное окно. С помощью фреймов возможно создание многооконных HTML-документов. В каждом фрейме отображается отдельный документ HTML. С использованием фреймов возможно в одном окне броузера просматривать несколько HTML-документов, каждый документ в своем фрейме.

Пример: index.html

-----------------------------------------

| Заголовок (header.html) |

-----------------------------------------

| Меню (left) | Основное текст (right) |

| (menu.html) | (about.html) |

| | |

| | |

-----------------------------------------

| Нижний заголовок (footer.html) |

-----------------------------------------

Файл index.html (Как правило при создании сайта начальный файл называется index.html или default.html)

<HTML>

<HEAD>

<TITLE>Простой документ с фреймами</TITLE>

</HEAD>

<FRAMESET rows="50px,80%,\*" border=5 bordercolor=green>

***<!--***

***(rows="50px,80%,\*")***

***rows="100px,100px,100px" - при масштабировании сохраняются пропорции***

***rows="30%,50%,\*"***

***rows="30%,50%,20%***

***rows="3\*,5\*,2\*" - в долях***

***rows="1\*,200px,3\*" - Второй фрейм занимает 200px, первый фрейм получает 25% оставшегося пространства, а третий - 75%.***

***rows="1\*,50%,3\*" - Второй фрейм занимает 50% высоты окна броузера, первый фрейм получает 25% оставшегося пространства, а третий - 75%.***

***rows="\*,20px,50%,20px,3\*" - Третий фрейм занимает 50% высоты окна броузера, второй и четвертый - 20px, первый фрейм получает 25% оставшегося пространства, а пятый - 75%. (\* = 1\*, если \* встречается 1 раз, то обозначает все оставшееся пространство)***

***-->***

<FRAME src="header.html" scrolling="no">

<FRAMESET cols="20%,\*">

***<!--***

***(cols="20%,\*")***

***cols="20%,80%"***

***cols="1\*,4\*"***

***-->***

<FRAME **(noresize)** name="left" src="menu.html">

<FRAME name="right" src="about.html">

</FRAMESET>

<FRAME src="footer.html" scrolling="no">

<NOFRAMES>

<P>Версия документа без фреймов

<P>В этом документе содержатся фреймы, но ваш броузер их не поддерживает.</P>

</NOFRAMES>

</FRAMESET>

</HTML>

Документ HTML, в котором описывается компоновка фреймов называется **документом с фреймами** и выглядит не так, как обычный документ HTML. Стандартный документ имеет один раздел [HEAD](../struct/global.html#edef-HEAD) и один раздел [BODY](../struct/global.html#edef-BODY). Документ с фреймами имеет раздел [HEAD](../struct/global.html#edef-HEAD) и раздел [FRAMESET](../present/frames.html#edef-FRAMESET), который заменяет раздел [BODY](../struct/global.html#edef-BODY).

В разделе [FRAMESET](../present/frames.html#edef-FRAMESET) задается расположение фреймов в окне броузера. Обычные визуальные элементы не должны присутствовать в документе с описанием фреймов.

Иерархия элементов:

FRAMESET

|

|-FRAMESET, FRAME, NOFRAMES

Элемент FRAMESET определяет набор фреймов, FRAME – содержимое и вид одного фрейма, NOFRAMES – задает альтернативное содержимое документа для броузеров без поддержки фреймов.

Элемент FRAMESET

Начальный тег: ***обязателен***, конечный тег: ***обязателен***

Определяет набор фреймов.

Атрибуты:

1. rows = список длин, определяющих высоту горизонтальных фреймов

(Разделенный запятыми список пикселов, процентов и относительных длин. По умолчанию используется 100%, что обозначат одну строку.)

2. cols = список длин, определяющих ширину вертикальных фреймов

(Разделенный запятыми список пикселов, процентов и относительных длин. По умолчанию используется 100%, что означат один столбец.)

(Пикселы – фиксированный размер, % - % от окна броузера, относительная длина (\*) – доля от оставшегося пространства. Приоритет: 1.фиксированная ширина в PX, 2.%, 3.оставшаяся часть окна делится в долях(\*).)

3. frameborder = 1|0 (IE: YES | NO)

Определяет видимость границ фреймов для данного FRAMESET. Возможные значения:

1. 1,YES: Границы отображаются. (По умолчанию).
2. 0,NO: Границы не отображаются. Однако, они могут все равно отображаться, если заданы в соседних фреймах.

4. border = толщина границы между фреймами в PX. Если border = 0, граница не отображается.

5. framespacing = дополнительное расстояние между фреймами в PX. Если явно задается framespacing= 0, то отображается тонкая граница.

Фактически, и тот и другой атрибут определяет толщину границы. Если задать оба атрибута, то отображается толщина, указанная во framespacing. (не складываются).

6. bordercolor=цвет границы

Элементы FRAMESET могут быть вложены друг в друга. Число уровней вложенности не ограничено.

Элемент FRAME

Начальный тег: ***обязателен***, конечный тег: **не используется**

Определяет содержимое и вид одного фрейма.

Атрибуты:

1. name = строка

Имя фрейма. Может использоваться в качестве цели в гиперссылках (<A target=”цель”>).

2. src = [*uri*](../types.html#type-uri)

Начальное содержимое фрейма, например документ HTML. Может быть документом, содержащим фреймы. (без рекурсии)

3. noresize (логический)

Запрещает изменение размеров фрейма. (границы неперемещаемы).

4. scrolling = auto|yes|no

Задает информацию о полосах прокрутки (scrollbar’ах) фрейма. Возможные значения:

1. auto: Полосы прокрутки (scrollbar’ы) появляются по мере необходимости. (По умолчанию).
2. yes: Всегда появляются полосы прокрутки. (В IE: всегда появляется вертикальная полоса прокрутки, горизонтальная появляется по мере необходимости.)
3. no: Полосы прокрутки не появляются.

5. marginwidth = [*пикселы*](../types.html#type-pixels)

Определяет ширину левого и правого отступа между границей фрейма и содержимым фрейма (текстом).

6. marginheight = [*пикселы*](../types.html#type-pixels)

Определяет ширину верхнего и нижнего отступа между границей фрейма и содержимым фрейма.

Возможно использование атрибутов frameborder и bordercolor, которые перекрывают соответствующие значения элемента FRAMESET.

Элемент NOFRAMES

Начальный тег: ***обязателен***, конечный тег: ***обязателен.***

Задает альтернативное содержимое документа для броузеров без поддержки фреймов.

Пример. Совместное использование атрибутов cols и rows.

|  |  |  |
| --- | --- | --- |
| 11 | 12 | 13 |
| 21 | 22 | 23 |

<!-- 2 строки Х 3 столбца -->

<FRAMESET rows="50%,\*" cols="33%,33%,\*">

<!-- Фреймы создаются построчно (строка,столбец) -->

<FRAME src="11.html">

<FRAME src="12.html">

<FRAME src="13.html">

<FRAME src="21.html">

<FRAME src="22.html">

<FRAME src="23.html">

</FRAMESET>

**Задание целевого фрейма. Атрибут target.**

В элементах A и AREA возможно указание целевого фрейма гиперссылки с помощью атрибута target (в начальном якоре можно указать фрейм, в котором будет открываться содержимое гиперссылки).

Атрибут target:

target = имя фрейма (<FRAME name=”имя фрейма”>) | зарезервированное имя фрейма

зарезервированное имя фрейма = **\_blank | \_self | \_parent | \_top**

**\_blank** - целевым фреймом является новое окно броузера.

**\_self** – целевым фреймом является текущий фрейм (по умолчанию) *(элемент BASE)*

**\_top** – целевым фреймом является полное окно броузера (все остальные фреймы закрываются)

**\_parent** – целевым фреймом является родительский фрейм

Броузер пытается загрузить документ в родительский фрейм. Как правило аналогично **\_top. (Если загрузить в один из фреймов (right) документ, содержащий фреймы, то ссылка с атрибутом target=\_parent в любом из этих фреймов приведет к открытию документа во фрейме right).**

Пример файла menu.html

<P>

<A href="1.html" target="right">Ссылка открывается в правом фрейме</A>

<P>

<A href="1.html#Anchor1" target="right">Ссылка на фрагмент документа открывается в правом фрейме</A>

<P>

<A href="1.html" target="\_blank">Ссылка открывается в новом окне</A>

<P>

<A href="1.html" target="\_top">Ссылка открывается в полном окне броузера</A>

**Встроенные (плавающие) фреймы. Элемент IFRAME.**

В HTML5 поддерживаются только плавающие фреймы.

(IFRAME – in-line floating frames)

Элемент [IFRAME](../present/frames.html#edef-IFRAME) позволяет осуществлять вставку фрейма в тело HTML-документа.

Пример:

<P>

Текст

<IFRAME name=iframe1 src="1.html" width="200" height="200" scrolling="no" align=middle>

Ваш броузер не поддерживает фреймы

</IFRAME>

Текст

</P>

<P>

<A href="2.html" target=iframe1>гиперссылка</A>

Встраиваемая информация определяется атрибутом [src](../interact/forms.html#adef-src), а содержимое элемента [IFRAME](../present/frames.html#edef-IFRAME), отображается только если броузер не поддерживает фреймы (аналогично содержимому элемента NOFRAMES).

Атрибуты:

1. width = ширина встроенного фрейма в PX или %.

2. height = высота встроенного фрейма в PX или %.

3. ALIGN = TOP | MIDDLE | BOTTOM | LEFT | RIGHT

Задает выравнивание фрейма в тексте (как в случае с изображениями).

Атрибуты HSPACE и VSPACE задают горизонтальный и вертикальный внешние отступы (как в изображениях).

Атрибуты NAME, SRC, FRAMEBORDER, SCROLLING, MARGINHEIGHT, MARGINWIDTH используются так же, как и в обычных фреймах.

Атрибуты BORDER, BORDERCOLOR, FRAMESPACING определены спецификацией, но, похоже, что IE5.0 их игнорирует.

Изменять размер встроенных фреймов нельзя, поэтому у них нет атрибута [noresize](../present/frames.html#adef-noresize).

# Элементы секции HEAD

TITLE – определение заголовка документа

BASE – задание базового URI

META – определение метаданных

SCRIPT – создание сценариев на JavaScript и VBScript

STYLE – определение встроенной таблицы стилей

LINK – определение вспомогательных связей между документами (ссылка на внешние таблицы стилей)

**Элемент BASE**

Начальный тэг: ***обязателен***, Конечный тэг: ***запрещен***

Задание базового URI для гиперссылок.

Атрибуты:

href = URI

Базовый URI для относительных гиперссылок.

target = имя фрейма по умолчанию

Имя фрейма по умолчанию, в котором открываются гиперссылки.

Пример:

<HEAD>

<BASE href="http://iu5.bmstu.ru" target="\_blank">

</HEAD>

<BODY>

<A href="one/one.html#Anchor1">Относительная гиперссылка</A>

<P>Будет преобразована в http://iu5.bmstu.ru/one/one.html и откроется в новом окне </P>

</BODY>

Элемент META

Начальный тэг: ***обязателен***, Конечный тэг: ***запрещен***

Определение метаданных.

Атрибуты:

name = имя свойства

content = значение свойства

http-equiv = Имя заголовка ответа HTTP. (Может использоваться вместо атрибута name.)

Примеры:

<META name="Author" content="Имя создателя документа">

<META name="keywords" content="поисковый образ документа – список слов для поисковых машин через запятую">

Задание поля Content-Type заголовка HTTP-протокола

<META http-equiv="Content-Type"content="text/html; charset=Windows-1251">

Автоматическое обновление содержимого документа в броузере

<META HTTP-EQUIV="Refresh" CONTENT="10(время в секундах); URL=http://www.abc.com">

Если в течении заданного времени не будет предпринято никаких действий, то автоматически будет загружен документ по заданному адресу.