27.10.2021 МДК0301 Тема Введение в HTML

**Тема лекции: HTML. Таблицы. Фреймы. Мультимедийные возможности. Графика. Формы.**

**Таблицы**

Теги *HTML* для создания таблиц первоначально предназначались для представления строк и столбцов табулированных данных. Однако дизайнеры научились с их помощью управлять разметкой *Web*-страниц: создавать столбцы текста, задавать интервалы между элементами и изменять внешний вид текста способами, недоступными другим тегам форматирования *HTML*.

Таблицы в языке *HTML* всегда имеют прямоугольный вид и состоят из строк, которые в свою *очередь* состоят из ячеек. Все языковые конструкции, описывающие компоненты создаваемой таблицы, заключаются между тегами <TABLE> и </TABLE>.

Заполнение таблицы происходит построчно; для обозначения строки используется пара тегов <TR>... </TR>. Строка состоит из ячеек, для задания которых используют либо теги <TH>... </TH>, если эти ячейки содержат *заголовки* столбцов, либо теги <TD>... </TD>. *Заголовки* выводятся полужирным шрифтом и располагаются по центру ячейки. Данные имеют обычный *шрифт* и выравниваются по левой стороне ячейки. Для задания заголовка всей таблицы используются теги <CAPTION> и </CAPTION>.

**Пример**

<TABLE>

<CAPTION>Заголовок таблицы</CAPTION>

<TR><TH>Заголовок 1</TH><TH>Заголовок 2</TH></TR>

<TR><TD>Ячейка 1</TD><TD>Ячейка 2</TD></TR>

<TR><TD>Ячейка 3</TD><TD>Ячейка 4</TD></TR>

</TABLE>



Наличие в ячейках данных не обязательно. Создать пустую ячейку можно двумя способами: ничем не заполнять ее *контейнер* ( <TD></TD> ), либо поместить в нее символ неразрывного пробела, задаваемого специальной последовательностью символов - &nbsp; (т. е. создать ячейку вида <TD>& nbsp; </TD> ).

Нет надобности отдельно создавать пустые ячейки, если планируется, что все оставшиеся в строке ячейки не будут заполнены. Так как *тег* <TR> сигнализирует о начале новой строки, пустые ячейки будут добавлены браузером автоматически.

**Основные атрибуты тега <TABLE>**

Назначение основных атрибутов тега <TABLE> и значения, которые они могут принимать перечислены в таблице.

|  |  |
| --- | --- |
| **Атрибут** | **Назначение** |
| BORDER=n | Определяет ширину рамки таблицы (в пикселах), например, BORDER=1 ; значение, равное нулю, означает отсутствие рамки |
| WIDTH=n | Определяет ширину всей таблицы в пикселах, либо в процентах от ширины окна браузера |
| HEIGHT=n | Определяет высоту всей таблицы в пикселах, либо в процентах от высоты окна браузера |
| ALIGN | Задает горизонтальное выравнивание таблицы в окне браузера; имеет значения left, center и right (по умолчанию - left) |
| CELLPADDING=n | Добавляет свободное пространство между данными внутри ячейки и ее границами; по умолчанию значение равно 2 |
| CELLSPACING=n | Добавляет свободное пространство между ячейками внутри всей таблицы; по умолчанию значение равно 2 |
| HSPACE=n | Задает области свободного пространства указанной ширины (в пикселах) слева и справа от таблицы |
| VSPACE=n | Задает области свободного пространства заданной высоты (в пикселах) сверху и снизу от таблицы |
| BGCOLOR=цвет | Устанавливает цвет фона всей таблицы |
| BACKGROUND=URL | Указывает фоновое изображение для таблицы, где URL - адрес источника (имя файла с изображением) |

**Пример**

Изменим содержимое документа, созданного в предыдущем примере, добавив атрибуты BORDER и ALIGN в тег <TABLE>:

<TABLE BORDER=1 ALIGN=center>

Теперь ячейки таблицы будут обрамлены рамкой, a таблица выровнена по центру окна браузера.



**Выравнивание данных в ячейках**

При помощи атрибутов ALIGN и VALIGN можно по-разному размещать данные относительно границ ячейки. Эти атрибуты используются совместно с тегами <CAPTION>, <TR>, <TH> и <TD> в самых различных комбинациях. Ниже приведены значения атрибутов для перечисленных элементов.

|  |  |
| --- | --- |
| **Тег** | **Назначение атрибута** |
| <TR> | Атрибут ALIGN может принимать значения left, center и right (по умолчанию - left для данных и center для заголовков); он определяет горизонтальное выравнивание данных в ячейках и действует на всю строку, если не отменяется тем же атрибутом в отдельной ячейке.Атрибут VALIGN может иметь значения top, bottom, middle и *baseline* (по умолчанию - middle); он регулирует положение данных относительно верхней и нижней границ ячейки и влияет на всю строку, если не отменяется таким же атрибутом в отдельной ячейке. *baseline* применяется ко всем элементам строки и выравнивает их по базовой линии. |
| <TH> | Атрибут ALIGN может принимать значения left, center и right (по умолчанию - center).Атрибут VALIGN может иметь значения top, bottom и middle (по умолчанию - middle). |
| <TD> | Атрибут ALIGN может принимать значения left, center и right (по умолчанию - left.Атрибут VALIGN может иметь значения top, bottom и middle (по умолчанию - middle) |
| <CAPTION> | Атрибут ALIGN может иметь значения top и bottom (по умолчанию - top); размещает заголовок таблицы сверху или снизу |

**Объединение ячеек**

Смежные ячейки таблицы могут объединяться. Например, в таблице из нескольких столбцов все ячейки первой строки можно объединить и поместить в этой строке красивый заголовок таблицы. Возможно также объединение нескольких строк или создание пустой прямоугольной области.

Для соединения двух смежных ячеек в одном столбце нужно использовать атрибут ROWSPAN тега <TH> или <TD>, например,

<TD ROWSPAN=2>

Для объединения двух смежных ячеек в одной строке нужно использовать атрибут COLSPAN тех же тегов, например,

<TD COLSPAN=2>

**Пример**

В следующей таблице используется объединение столбцов и строк.

<HTML>

<BODY>

<TABLE BORDER=1 ALIGN=center>

<TR><TH COLSPAN=3>Заголовок на 3 столбца</TH></TR>

<TR>

<TH>Заголовок на 1 строку</TH>

<TD>Ячейка 1</TD>

<TD>Ячейка 2</TD>

</TR>

<TR>

<TH ROWSPAN=3>Заголовок на 3 строки</TH>

<TD>Ячейка 3</TD>

<TD>Ячейка 4</TD>

</TR>

<TR><TD>Ячейка 5</TD><TD>Ячейка 6</TD></TR>

<TR><TD>Ячейка 7</TD><TD>Ячейка 8</TD></TR>

</TABLE>

</BODY>

</HTML>

Таблица будет иметь следующий вид.



Если вы хотите создать таблицу с объединениями столбцов и в то же время точно контролировать ширину каждого столбца, необходимо задать ширину по крайней мере одной ячейки в каждом столбце. Для полной уверенности найдите время и задайте ширину каждой ячейки в таблице. Когда объединения столбцов пересекаются, очень легко получить непредсказуемый результат.

**Цвет в таблицах**

В HTML не предусмотрено специальных средств раскрашивания таблиц. Вы можете изменить цвет фона ячейки при помощи атрибута BGCOLOR перед размещением в ней текста или изображения, а также использовать атрибут BORDERCOLOR для изменения цвета рамки ячейки. Теги <TABLE>, <TD>, <TH> и <TR> также допускают использование в них указанных атрибутов. Таким образом, вы можете изменить цвет всей таблицы, отдельной ячейки или строки таблицы.

Значения цветов, установленные на уровне ячейки, будут перекрывать значения, установленные на уровне строки, которые в свою очередь, будут перекрывать значения, заданные на уровне всей таблицы.

**Пример**

Создадим таблицу, удовлетворяющую следующим требованиям:

* цвет таблицы - белый;
* вторая строка светло-серого цвета, в ячейке 2 - цвет "teal"
* цвет надписи в ячейке 5 - красный;
* первый столбец составляет 20% от ширины таблицы, два другие - по 40%;
* ячейка 3 центрирована, а ячейка 4 - выровнена вправо;
* ячейки 6 и 8 объединены в одну, центрированы и выровнены по нижнему краю ячейки;
* поля внутри ячеек - 10 пикселей.



<HTML>

<BODY>

<TABLE BORDER=8 WIDTH=90% BGCOLOR="white"

CELLPADDING=10 ALIGN=center>

<TR>

<TH WIDTH=20%>Простой заголовок</TH>

<TH WIDTH=80% COLSPAN=2>Заголовок на 2 столбца</TH>

</TR>

<TR BGCOLOR="#CCCCCC">

<TH WIDTH=20%>Заголовок на 1 строку</TH>

<TD WIDTH=40%>Ячейка 1</TD>

<TD WIDTH=40% BGCOLOR="teal">Ячейка 2</TD>

</TR>

<TR>

<TH WIDTH=20% ROWSPAN=3>Заголовок на 3 строки</TH>

<TD ALIGN=center>Ячейка 3</TD>

<TD ALIGN=right>Ячейка 4</TD>

</TR>

<TR>

<TD><FONT COLOR="red">Ячейка 5</FONT></TD>

<TD ROWSPAN=2 ALIGN=center VALIGN=bottom>Ячейка 6</TD>

</TR>

<TR>

<TD>Ячейка 7</TD>

</TR>

</TABLE>

</BODY>

</HTML>

**Фреймы**

Один из способов выдать сразу несколько файлов *HTML* на экран пользователя - это открыть несколько окон браузера. Другой *путь* состоит в том, чтобы разбить окно на несколько разделов. Эти *разделы* называются **фреймами** или кадрами. В каждом фрейме показывается свой *HTML*-документ. Каждый *фрейм* может иметь свои полосы прокрутки, ссылки, графические изображения и т. д. Фреймы могут функционировать независимо или влиять друг на друга, используя ссылки, указывающие на другие фреймы.

**Контейнер <FRAMESET>**

Web-страница, которая разделена на фреймы, называется документом группы фреймов. Документы группы фреймов содержат стандартный заголовок, задаваемый тегом HEAD, но в отличии от стандартных HTML-документов, они не содержат тега BODY. Вместо него используется контейнер (т. е. парный тег) <FRAMESET>, который применяется для определения строк и столбцов отдельных фреймов, каждый из которых обозначается тегом <FRAME>.

Если включить контейнер BODY в документ, где используется контейнер FRAMESET, то кадры будут проигнорированы программой просмотра, и информация, содержащаяся в документах, задаваемых тегами <FRAME>, не будет выведена. Будет показана только информация, содержащаяся в контейнере BODY.

Внутри контейнера <FRAMESET> ... </FRAMESET> могут располагаться только теги <FRAME> или другие контейнеры FRAMESET.

**Определение параметров кадров**

Тег <FRAMESET> имеет два главных атрибута: ROWS и COLS, задающих разбиение на строки и столбцы соответственно. Ниже приведен вид контейнера FRAMESET:

<FRAMESET ROWS="список\_значений" COLS="список\_значений">

...

</FRAMESET>

Можно определить любое число рядов и столбцов; необходимым условием является указание хотя бы одного из атрибутов ROWS или COLS.

Кадр не может быть единственным: если вы определили единственный ряд и единственный столбец, то программа просмотра проигнорирует контейнер FRAMESET, и экран останется пустым. Если определены по крайней мере два ряда или два столбца, другой атрибут может быть опущен (ему по умолчанию будет присвоено значение, равное 100%).

Значение атрибута ROWS или COLS представляет собой строку, содержащую список значений в пикселах, процентах или относительных единицах, разделенных запятыми. Количество рядов или столбцов кадров определяется числом этих значений.

**Пример**

Для задания сетки кадров из трех рядов, высота которых 100, 240 и 140 пикселов соответственно, используйте тег

<FRAMESET ROWS="100,240,140">

Задание высоты ряда в пикселах, однако, является плохим стилем, так как при этом не учитывается тот факт, что окна браузеров могут иметь самую разную величину. В абсолютных единицах стоит указывать размеры кадра лишь для размещения небольших изображений, в остальных же случаях лучше пользоваться относительными величинами.

**Пример**

Тег <FRAMESET ROWS="25%, 50%, 25%>" задаст три кадра, размером по 25%, 50% и 25% от высоты окна браузера.



Относительно точности указания размеров фреймов в процентах можно не беспокоиться: если сумма значений не равна 100%, то масштаб кадров будет пропорционально изменен.

Задание параметров кадров в относительных единицах выглядит примерно так:

<FRAMESET COLS="\*,2\*,3\*">

Символ \* обозначает пропорциональное деление окна программы просмотра. В данном примере окно будет разделено на три вертикальных кадра, первый из которых будет иметь ширину в 1/6, второй - в 2/6 (или 1/3) и третий - в 3/6 (или 1/2) от ширины окна браузера. Единица при указании относительных значений может быть опущена.



Указание значений атрибутов ROWS и COLS может быть и смешанным, включающим любое сочетание *абсолютных размеров*, процентных отношений и относительных значений, например,

<FRAMESET COLS="100,25%,\*,2\*">

Здесь первому кадру присвоено абсолютное значение в 100 пикселов по ширине, второму - 25% от ширины окна. Оставшееся пространство делится между третьим и четвертым кадрами в пропорции 1 к 2.



Приоритеты в указаниях значений атрибутов таковы: в первую очередь (слева направо) отводится место для кадра с абсолютным значением, затем - для кадра со значением в процентах, и в последнюю очередь - для кадров с относительными величинами.

Если вы пользуетесь *абсолютными величинами* в атрибутах ROWS и COLS, не делайте такие кадры большими - ведь они должны поместиться в окно браузера любого размера. Совместно с такими кадрами для лучшей балансировки рекомендуется использовать хотя бы один кадр, определенный в процентах или в относительных величинах.

При определении обоих атрибутов получается сетка кадров, например, тег

<FRAMESET ROWS="\*,2\*,\*" COLS="2\*,\*">

задает сетку из трех рядов и двух столбцов. В данном примере первый и последний ряды занимают по 1/4, второй ряд - половину от высоты окна. Первый столбец занимает 2/3, а второй - 1/3 ширины окна браузера.

**Тег <FRAME>**

Тег <FRAME> определяет отдельный кадр. Он должен располагаться внутри контейнера FRAMESET.

**Пример**

<FRAMESET ROWS="\*,2\*">

 <FRAME>

 <FRAME>

</FRAMESET>

Заметьте, что этот тег не является контейнером и, в отличие от тега <FRAMESET>, не имеет закрывающего тега.

Число тегов <FRAME> **обязательно** должно быть равно числу кадров, определенных в теге <FRAMESET>. В нашем примере определено два кадра, поэтому контейнер содержит соответственное количество тегов <FRAME>. Пока кадры ничем не заполнены.

<FRAME SRC="URL" NAME="имя\_окна"

 SCROLLING=yes|no|auto MARGINWIDTH="значение"

 MARGINHEIGHT="значение" NORESIZE>

Использовать все атрибуты необязательно. Чаще всего вы будете пользоваться только одним атрибутом - SRC. Строка

<FRAME SRC="URL">

определяет URL-адрес содержимого кадра. Это обычно файл HTML-документа, расположенный в том же каталоге, что и документ, содержащий контейнер FRAMESET, например,

<FRAME SRC="first.html">

Этот документ должен быть полноценным HTML-документом, т. е. содержать все обязательные части (контейнеры HTML, HEAD, BODY и т. д.).

В случае, если программа просмотра не сможет найти указанный файл, кадр не будет построен, и браузер выведет сообщение об ошибке. Если же в теге вовсе не указан атрибут SRC, кадр будет создан и оставлен пустым.

Текст, заголовки, графика и другие элементы не могут напрямую включаться в документ с кадрами. Все они должны вводиться только с помощью указания URL-адреса элемента. Если контейнер FRAMESET содержит "инородное тело", оно будет выведено, а кадры полностью проигнорированы.

Атрибут MARGINWIDTH=n задает размещение слева и справа от содержимого кадра областей свободного пространства высотой по n пикселов, а MARGINHEIGHT=n, соответственно, сверху и снизу.

Значения этих атрибутов всегда должны указываться в абсолютных значениях (пикселах). Так, тег

<FRAME MARGINHEIGHT="5" MARGINWIDTH="7">

создаст внутреннюю рамку на верхней и нижней границах кадра шириной в 5 пикселов, а на левой и правой границе - шириной в 7 пикселов. Внутри этой рамки данные выводиться не будут. Эти атрибуты не имеют ничего общего с рамкой кадра, определяемой браузером, либо задаваемой при помощи атрибута BORDER.

К построенным вами кадрам автоматически добавляются полосы прокрутки, если содержание кадра больше его размера. Иногда это может нарушить эстетику страницы, поэтому в HTML предусмотрен атрибут SCROLLING тега <FRAME>, имеющий следующий формат:

<FRAME SCROLLING="yes|no|auto">

Этот атрибут может принимать одно из трех значений: yes, no и auto. Последнее значение подразумевается по умолчанию, т. е. когда атрибут не определен. Если указано значение yes, полоса прокрутки появится в любом случае, значение no запрещает появление полосы. Определение атрибута SCROLLING, например, может быть следующим:

<FRAME SCROLLING=yes>

По умолчанию размеры кадров могут легко изменяться читателями, однако понятно, что это может сильно нарушить авторский замысел. Поэтому вы, скорее всего, захотите использовать атрибут NORESIZE тега <FRAME>, запрещающий возможность "перекраивания" вашей страницы:

<FRAME NORESIZE>

Этот атрибут не имеет значений. Указав его в одном кадре, вы тем самым запретите изменять размеры и всех смежных кадров. Как правило, этого бывает достаточно, чтобы "закрепить" рамки всех кадров страницы на месте.

Когда вы захватываете рамку кадра мышью, то указатель становится двунаправленной стрелкой, если перемещение рамки возможно. В противном случае, т. е. когда использован атрибут NORESIZE, двунаправленная стрелка не появляется.

Для определения рамки кадра в HTML существуют три атрибута: BORDER, FRAMEBORDER и BORDERCOLOR. Первый из этих атрибутов используется только с тегом <FRAMESET> и устанавливает ширину всех рамок для всех кадров контейнера FRAMESET. Эта величина указывается в пикселах, например,

<FRAMESET BORDER="10">

Если этот атрибут нулевой, то все кадры контейнера будут без рамок. По умолчанию атрибут BORDER имеет значение 5.

Атрибут FRAMEBORDER используется с тегами <FRAMESET> и <FRAME> и может принимать два значения: yes или no. В случае yes рамка имеет трехмерную форму. Если FRAMEBORDER="no", рамка невидима, т. е. она имеет цвет фона окна браузера, устанавливаемого по умолчанию.

По умолчанию атрибут FRAMEBORDER имеет значение yes, т. е. предусматривает наличие трехмерной рамки. Рамка кадра будет невидимой, если значение FRAMEBORDER="no" установлено для всех кадров, смежных с ним.

Атрибут BORDERCOLOR может использоваться с тегами <FRAMESET> и <FRAME>. Ему может быть присвоено стандартное имя цвета или RGB-значение.

**Пример**

FRAMESET BORDERCOLOR="red" ROWS="\*,\*">

 <FRAME SRC="first.html" BORDERCOLOR="#FF00FF">

 <FRAME SRC="first.html">

</FRAMESET>

Здесь атрибут BORDERCOLOR тега <FRAMESET> ; устанавливает красный цвет рамок ("red"), однако такой же атрибут тега <FRAME> отменяет это значение и определяет цвет рамки первого кадра как фиолетовый. В результате второй кадр, в котором цветовой атрибут не определен, будет иметь часть рамки фиолетовой (на стороне, смежной с первым кадром), а остальную часть рамки - красной.

Если же в двух смежных кадрах определены свои собственные атрибуты BORDERCOLOR, то ни один из них не будет иметь силы. Цвет их рамок будет определяться соответствующим атрибутом контейнера FRAMESET.

**Мультимедийные возможности HTML**

*В Браузеры встроены* возможности подключать дополнительные программные модули для мультимедийных приложений. П*оддержка* технологий LiveAudio (для звуковых файлов форматов *WAV*, AU, *AIFF* и *MIDI*), Live3D (*VRML*), LiveVideo (видео-файлы АVI) и *QuickTime* (файлы формата *MOV*, включающие текст и звук *MIDI*). Все это может быть интегрировано в вашу страницу при помощи тега <EMBED>и<NOEMBED>.

Для других браузеров указывается элемент <NOEMBED>, где между начальным и конечным тегами указывается *альтернативный текст*. Отметим некоторые параметры тега <EMBED>.

|  |  |
| --- | --- |
| **Атрибут** | **Назначение** |
| SRC | Полный путь или имя файла вставляемого объекта. Параметр **обязателен** |
| PALETTE | Тип палитры для просматриваемого объекта. Имеет значения: **foreground**, **background** |
| HIDDEN | Тип *отображения объекта* на экране. Значения: **false** - объект не отображается (по умолчанию), **true** - объект отображается |
| TYPE | Тип файла объекта |

Другие атрибуты этого тега аналогичны атрибутам тега <IMG> (собственно, и вставка объекта в *HTML*-документ происходит похожим образом), поэтому здесь не будут описаны такие параметры элемента, как NAME, ALT, BORDER, HEIGHT, WIDTH, ALIGN, HSPACE, VSPACE.

**Пример**

Строка <EMBED SRC=*nature*.*wav*> позволит использовать в качестве фонового звука музыкальный *файл* **nature.wav**, а для вложения файла **js.pdf** в формате **PDF** (Adobe *Portable Document* *Format*) нужно добавить строку

<EMBED SRC=js.pdf WIDTH=500 BORDER=0 ALIGN=left>

**Графика(изображения)**

Файл HTML и графический файл – это два совершенно самостоятельных файла.

При использовании графики загрузка страницы замедляется, поэтому необходимо следить, чтобы размер графических файлов был как можно меньше. Наиболее популярные форматы графических файлов, используемых в Web – дизайне.

Графика в Web – страницах используется:

1. Для пояснения содержания документа (иллюстрация);

2. Для организации значков и кнопок для пользовательской навигации;

3. Для организации размеченных карт.

**Форматы графических файлов**

Графический формат GIF

GIF (Graphic Interchange Format). Для кодирования цвета в нем используется 8 бит, т.е. 256 различных цветов и столько же градаций серого. Наборы цветов (палитры) могут быть различными. Один GIF – файл может содержать несколько изображений, позволяющих создавать движущиеся или изменяющие образы. Само по себе 8-битовое кодирование цвета предполагает, что размер графического файла должен быть относительно небольшим. Но, кроме того, для уменьшения размера файла используется еще и сжатие изображения. Существует две разновидности GIF – файла: сжатые и обычные, в которых сжатие отсутствует. Дополнительное преимущество заключается в том, что хранение информации в файле может быть организовано таким образом, чтобы при выводе рисунка происходило чередование строк. Т.е. в начале будут выводиться строки с номерами 1, 5, 9 и т.д., затем с номерами 2, 6, 10 и так далее, до тех пор пока весь рисунок не будет отображен.другое преимущество GIF – возможность сделать часть изображения прозрачной. Прозрачным может стать только один цвет. Так, создание прозрачного фона позволяет более естественно вписать рисунок в документ и избежать появления прямоугольника, обозначающего границы изображения. \*

Файл с расширением GIF, следует применять, если:

1. Необходимо сделать изображение с прозрачным фоном, т.е. чтобы через фон изображения просвечивал фон самой страницы. Изображение при этом выглядит более естественно.
2. Необходимо использовать чередование строк. Тогда загрузка иллюстрации выполняется по частям. Включенная опция чередования строчности дает возможность до полной загрузки иллюстрации понять, что она собой представляет и принять решение, стоит ли продолжать прием.
3. Файл имеет небольшой размер (для кнопки или окна), эксперименты показали, что gif занимает меньше места, чем jpg.

Графический формат JPG

Вторым подходящим для Интернета графическим форматом является JPEG (JPG), названный так в честь своего разработчика – Join Photographic Experts Group. Этот формат обеспечивает 24-битовое кодирование цвета и лучше подходит для хранения таких изображений, как фотографии. Недостатком формата является возможность искажения цвета в результате сжатия данных.\*

Файл с расширением JPG, следует применять, если:

1. Используется 24-битовое изображение фотографического качества.
2. Используется большое полноцветное изображение, и эксперимент показал, что размер файла меньше, чем размер gif – файла.

Графический формат PNG (Portable Network Graphics):

Создан с целью замены формата GIF. В отличие от своего прототипа новый формат позволяет использовать как 8-битовое, так и 24-битовое кодирование цвета. Кроме того, алгоритм создания прозрачности усовершенствован. С помощью альфа – канал прозрачность может быть задан для участков изображения, содержащих разные цвета. Для 24-битовых форматов прозрачность может быть неполной, т.е. разработчик устанавливает ее величину в процентах (от 0 до 1000).

**Встраивание графики в Web – страницу**

Для встраивания используется тег <IMG>. Этот тег указывает браузеру в каком месте и какое изображение нужно воспроизвести. Данный тег является непарным.

Атрибуты:

* Src=”\_\_” – указывает URL – адрес загружаемой иллюстрации Этот атрибут задается для любого рисунка.

Например: <IMG Src = “mouse.gif”>

Если URL адрес не содержит пути поиска файла, значит графический файл находится в том же каталоге, где HTML – документы.

Например: <IMG Src = “C:\PICTURE\mouse.gif”>

Если файл находится на вашем сервере, URL – адрес удаленного поиска:

Например: <IMG Src = “HTTP://WWW.anysite.com\mouse.gif”>

* Alt=”\_\_\_\_” – всплывающая подсказка, используется в браузере для ускорения загрузки Web – страницы может быть отключен режим графики. Текст, который задан в атрибуте Alt появляется на экране в ограниченном прямоугольнике. Если режим графики отключен, или страница грузится долго, тогда в этом случае не дожидаясь конца загрузки можно оценить содержание иллюстрации и если это например, ссылка, то перейти дальше.

Например:

<IMG Src = “searchmouse.gif” Alt=”Ознакомьтесь с нашим сайтом ”>

* Width=n и Height=m – размер иллюстрации. Браузер отводит рамку для картинки заданного размера, а затем загружает текст, и пока шрузится графика, можно начать читать текст.

Если

* Color=”цвет” –цвет линии.
* Noshade – выключение теней, подавляет иллюзию трехмерности.

**Формы в HTML-документах**

Формы *HTML* позволяют получать информацию от читателей. До сих пор мы обсуждали только способы вывода данных, теперь речь пойдет об обратном действии. Формы дают возможность запрашивать информацию в виде свободного текста, получать ответы типа "да/нет" или делать выбор из нескольких опций.

Вы можете использовать формы с различными целями. Простейшим примером является *размещение* формы, куда читатели, посетившие *сайт*, смогут записать свои отзывы. Круг применения форм *HTML* ограничивается только вашей фантазией.

**Тегом <FORM> начинается каждая форма. В нем нужно определить два атрибута, указывающих используемый скрипт и метод посылки данных.**

|  |  |
| --- | --- |
| **Атрибут** | **Назначение** |
| ACTION | Определяет URL, который примет и обработает данные формы. Если этот атрибут не определен, данные отправляются по адресу страницы, на которой помещена форма |
| METHOD | Указывает форме, как послать информацию соответствующей программе обработки (скрипту). Обычно он получает значение **post**, тогда информация формы посылается отдельно от URL. Значению **get** соответствует посылка вместе с URL |

**Пример**

<FORM METHOD="post" ACTION="/cgi-bin/comment\_script">

...

</FORM>

В этом примере дано указание браузеру отправить заполненную форму для обработки скриптом **comment\_script**, расположенным в каталоге **cgi-bin** вашего сервера, и использовать метод посылки **post**.

На странице можно расположить любое число форм, однако, нужно следить за тем, чтобы не поместить одну форму в другую.

**Работа с тегами форм**

В HTML существует три тега для создания различного типа полей в форме: <*TEXTAREA*>, <SELECT> и <INPUT>. Любое их количество может быть размещено в контейнере между тегами <FORM> и </FORM>. Ниже дано их краткое описание, а подробнее они будут рассмотрены чуть позже.

|  |  |
| --- | --- |
| **Тег** | **Назначение** |
| <*TEXTAREA*> | Определяет поле, в которое пользователь вводит многострочную текстовую информацию |
| <SELECT> | Позволяет пользователю сделать выбор в окне с полосой прокрутки, либо в раскрывающемся меню |
| <INPUT> | Обеспечивает некоторые другие виды ввода информации: ввод одной строки текста, установку и сброс флажков (*check boxes*), выбор переключателя (*radio buttons*) и нажатие кнопки для отправки данных или *очистки формы* |

**Тег <TEXTAREA>**

Этот тег предназначен для построения поля с целью ввода многострочный текстовой информации. В контейнере *TEXTAREA* допускается размещать любой текст, который будет выведен в поле ввода по умолчанию. Перечислим атрибуты этого тега.

|  |  |
| --- | --- |
| **Атрибут** | **Назначение** |
| NAME | Определяет название поля |
| ROWS | Устанавливает высоту поля, т. е. число строк в нем |
| COLS | Устанавливает ширину поля, т. е. длину строки |

**Пример**

<HTML><BODY>

<FORM>

<TEXTAREA> Какой-то текст </TEXTAREA>

</FORM>

</BODY></HTML>



При помощи атрибутов ROWS и COLS можно задать поле любого размера. Хотя эти атрибуты не являются обязательными, они не имеют определенных значений по умолчанию (для каждого браузера эти значения различны), поэтому лучше их всегда указывать явно.

**Тег <SELECT>**

Этот тег используется для создания всплывающего меню или списка опций с полосой прокрутки. Список опций и пункты меню располагаются внутри контейнера SELECT. Аналогично тегу <*TEXTAREA*>, <SELECT> требует обязательного определения имени в атрибуте NAME. Количество опций указывается в атрибуте SIZE. Ниже перечислены атрибуты тега <SELECT>.

|  |  |
| --- | --- |
| **Атрибут** | **Назначение** |
| NAME | Определяет название информации |
| SIZE | Определяет вертикальный размер окна для опций выбора. Если атрибут опущен или его значение равно 1, выводится всплывающий список опций. Если указано число больше единицы, то опции выводятся в окне с полосой прокрутки. Если значение атрибута больше, чем фактическое количество элементов списка, добавляются пустые строки. При их выборе пользователем возвращаются пустые поля |
| MULTIPLE | Позволяет выбирать сразу нескольких опций |

Список опций включается в контейнер <SELECT> ; при помощи тега <OPTION>. Этот тег имеет два атрибута.

|  |  |
| --- | --- |
| **Атрибут** | **Назначение** |
| VALUE | Указывает значение, возвращаемое программе обработки (скрипту), в случае выбора опции пользователем |
| SELECTED | Указывает на опцию, выбранную по умолчанию |

**Пример**

Ниже приведено содержимое двух HTML-документов, содержащих похожие формы. Во втором документе предварительно выбран один элемент из списка (добавлен атрибут SELECTED ).

<HTML>

<BODY>

 Выберите время года:

<FORM>

<SELECT NAME=year>

<OPTION SELECTED VALUE="winter"> зима

<OPTION VALUE="spring"> весна

<OPTION VALUE="summer"> лето

<OPTION VALUE="autumn"> осень

</SELECT>

</FORM>

</BODY>

</HTML>



<HTML>

<BODY>

<FORM>

 Выберите время года:

<SELECT MULTIPLE NAME="year">

<OPTION SELECTED VALUE="winter"> зима

<OPTION VALUE="spring"> весна

<OPTION VALUE="summer"> лето

<OPTION VALUE="autumn"> осень

</SELECT>

</FORM>

</BODY>

</HTML>



**Тег <INPUT>**

Тег <INPUT>, в отличие от <*TEXTAREA*> и <SELECT>, является одиночным тегом. Он предназначен для сбора информации различными способами, включая текстовые поля, поля для ввода пароля, переключатели, флажки, кнопки для отправки данных (Submit) и для *очистки формы* (Reset, Clear).

Тег <INPUT> располагает следующими атрибутами.

|  |  |
| --- | --- |
| **Атрибут** | **Назначение** |
| SIZE | Указывает размер поля ввода в символах |
| MAXLENGTH | Определяет максимально возможное число символов, вводимых в поле |
| VALUE | Для текстового поля определяет текст, выводимый по умолчанию. Для флажков и переключателей указывает значение, возвращаемое программе обработки. Для кнопок отправки и *очистки формы* определяет надпись на кнопке |
| CHECKED | Устанавливает флажок или переключатель во включенное состояние по умолчанию. С другими типами тегов <INPUT> не употребляется |
| TYPE | Устанавливает тип поля ввода |

**Тип поля ввода, атрибут TYPE**

Атрибут TYPE тега <INPUT> может принимать следующие значения.

|  |  |
| --- | --- |
| **Атрибут** | **Назначение** |
| TEXT | Является значением по умолчанию и предполагает создание одной строки для ввода данных. Для этого типа поля ввода употребляются атрибуты NAME (обязательный), SIZE, VALUE и MAXLENGTH |
| PASSWORD | Позволяет заменять вводимые символы пароля звездочками. Для этого типа поля ввода используются атрибуты NAME (обязательный), SIZE, MAXLENGTH и VALUE |
| CHECKBOX | Позволяет вывести поле для установки флажка в виде маленького квадратика, в котором может быть произведена отметка опции "галочкой". Может использоваться совместно с атрибутами NAME (обязательный), VALUE и CHECKED (определяет установленный по умолчанию флажок). Флажки обычно употребляются, когда можно выбрать сразу несколько опций из числа предложенных.Нужно быть очень осторожным в использовании флажков и переключателей, если цвет фона страницы определяется не документом, а пользователем при помощи установок программы просмотра. Не допускайте, чтобы опции сливались с фоном страницы |
| RADIO | Позволяет выбрать только одну из представленного числа опций. Переключатели можно группировать, задавая одно и то же значение атрибута NAME (обязательный). Так же используются атрибуты VALUE и CHECKED |
| RESET | Позволяет создать кнопку для *очистки формы*. Атрибут VALUE может быть использован здесь для наименования этой кнопки (по умолчанию кнопка имеет надпись Reset) |
| SUBMIT | Используется для создания кнопки, по нажатию которой введенные данные отправляются на сервер для обработки программой-скриптом. В атрибуте VALUE может быть указано название для этой кнопки (по умолчанию - **Submit Query** ) |

**Пример**

В следующей форме используется значение TEXT.

<HTML>

<BODY>

<FORM>

Введите номер телефона:

<INPUT TYPE="TEXT" NAME="phone"

 SIZE="15" MAXLENGTH="12">

</FORM>

</BODY>

</HTML>



**Пример**

Использование значения PASSWORD.

<HTML>

<BODY>

<FORM>

Введите секретное слово:

<INPUT TYPE="password" NAME="secret\_word"

SIZE="30" MAXLENGTH="30">

</FORM>

</BODY>

</HTML>



**Пример**

Использование значения CHECKBOX

<HTML>

<BODY>

<FORM>

<INPUT TYPE="checkbox" NAME="checkbox1"

VALUE="checkbox\_value1"> Вариант

<BR>

<INPUT TYPE="checkbox" NAME="checkbox2"

 VALUE="checkbox\_value2" CHECKED>

 Предварительно выбранный вариант

</FORM>

</BODY>

</HTML>



**Пример**

В этом примере две формы расположены в соседних ячейках таблицы.

<HTML>

<BODY>

<TABLE ALIGN=center BORDER CELLSPACING=10>

<TR>

<TD>

Форма 1:

<FORM>

<INPUT TYPE="radio" NAME="choice"

 VALUE="choice1"> yes.

<INPUT TYPE="radio" NAME="choice"

 VALUE="choice2"> no.

</FORM>

</TD>

<TD>

Форма 2:

<FORM>

<INPUT TYPE="radio" NAME="choice"

 VALUE="choice1" CHECKED> yes.

<INPUT TYPE="radio" NAME="choice"

 VALUE="choice2"> no.

</FORM>

</TD>

</TR>

</TABLE>

</BODY>

</HTML>



**Пример**

В следующей форме используется значение RESET.

<HTML>

<BODY>

<FORM>

<INPUT TYPE="reset">

<BR>

<INPUT TYPE="reset" VALUE="Очистить форму!">

</FORM>

</BODY>

</HTML>



**Пример**

Использование значения SUBMIT.

<HTML>

<BODY>

<FORM>

<TEXTAREA> Какой-то текст </TEXTAREA>

<BR>

<INPUT TYPE="submit" VALUE="Отослать данные!">

</FORM>

</BODY>

</HTML>



**Нестандартное использование элементов форм**

Многие теги HTML используются в целях, для которых они изначально не предназначались, например, таблицы - для разметки страниц. Поэтому смело экспериментируйте и с элементами форм.

Наиболее гибкими являются меню выбора. Всплывающее меню может выполнять функцию информационной полосы, включенной в текстовый поток. Например, можно использовать меню выбора для размещения небольших рассказов (помещая фрагменты текста в элементы OPTION. Применение этих возможностей заставляет пользователя активнее взаимодействовать с содержимым вашей страницы.





Ниже приведен HTML-документ, в котором раскрывающийся список используется для размещения небольшого рассказа об авторе.

<HTML>

<BODY>

Немного о себе:

<FORM>

<SELECT NAME="about">

<OPTION SELECTED VALUE="line0"> Автобиография

<OPTION VALUE="line1">

<OPTION VALUE="line2"> Меня зовут Михаил.

<OPTION VALUE="line3"> Я студент первого курса МГИУ.

<OPTION VALUE="line4"> Мои увлечения: футбол, теннис.

<OPTION VALUE="line5"> Я жду от вас писем по адресу:

<OPTION VALUE="line6"> petrov@mail.msiu.ru

</SELECT>

</FORM>

</BODY>

</HTML>

**Вопросы к домашнему заданию:**

1. **Основные теги используемые при создании Таблиц?**
2. **Что такое Фреймы?**
3. **Для чего используется Графика?**
4. **Для чего нужны Формы?**

Присылать **kuzn117@yandex.ru Срок 28.10.2021**