ЯЗЫК HTML (продолжение)

Таблицы, фреймы, блоки, слои

Для управления взаимным расположением элементов Web-страницы применяются таблицы, фреймы, блоки и слои.

Таблицы

С помощью таблиц просто располагать текст и картинки в нужных нам местах.

Таблицы могут быть вложенными. Таблицы с невидимыми границами широко используются для красивого размещения текста и графики.

Пример таблицы:

```
<TABLE WIDTH="100%" BORDER="1">
  <TR>
    <TD>1</TD>
    <TD>2</TD>
    <TD>3</TD>
  </TR>
  <TR>
    <TD>4</TD>
    <TD>5</TD>
    <TD>6</TD>
  </TR>
</TABLE>
```

Вот как будет выглядеть этот код на экране:

1	2	3
4	5	6

Фреймы

Фреймы представляют собой независимые области окна браузера. Фактически экран делится на несколько окон, в каждое из которых можно выводить информацию.

Структура фреймовых HTML документов существенно отличается от обычных.

Структура фреймового HTML документа:

<HTML>
<HEAD>
ЗАГОЛОВОК ДОКУМЕНТА
</HEAD>
<FRAMESET>
описание набора фреймов
</FRAMESET>
</HTML>

Тег <FRAMESET>...</FRAMESET> имеет следующие атрибуты:

ROWS - описание строк фреймовой структуры (проценты высоты окна браузера, пропорции, высота в пикселах);

COLS - описание столбцов фреймовой структуры (проценты ширины окна браузера, пропорции, ширина в пикселах);

FRAMEBORDER - описывает границу фрейма (по умолчанию значение YES- граница есть, NO-без границы);

BORDER - ширина границы фрейма (по умолчанию значение 5);

BORDERCOLOR - цвет сетки фреймовой структуры.

Например:

```
<FRAMESET ROWS="50%, 50%" COLS="50%, 50%" FRAMEBORDER=NO>
</FRAMESET>
```

Каждый отдельный элемент фреймовой структуры (отдельное подокно браузера) описывается при помощи тега <FRAME>, имеющего следующие атрибуты:

SRC – ссылка на загружаемый в фрейм HTML-файл;

NAME - имя фрейма (имя подокна браузера), для доступа к фрейму и обновления его содержимого;

MARGINHEIGHT - ширина верхнего и нижнего свободного поля фрейма в пикселях;

MARGINWIDTH - ширина левого и правого свободного поля фрейма в пикселях;

SCROLLING - полосы прокрутки содержимого фрейма (AUTO/YES/NO, по умолчанию значение - AUTO);

NORESIZE - наличием данного атрибута, пользователю запрещается изменять размеры фрейма при просмотре документа (по умолчанию это возможно при помощи мыши);

FRAMEBORDER - описывает границу фрейма (YES/NO);

BORDERCOLOR – задает цвет границы фрейма.

Возможно использование вложенных фреймовых структур.

Пример:

```
<FRAMESET ROWS="15%, 70%, 15%">
<FRAME SRC="header.htm" NORESIZE SCROLLING=NO>
<FRAMESET COLS="*,*" border="0" frameborder="0">
<FRAME SRC="left.html" name="left">
<FRAME SRC="right.html" name="right">
</FRAMESET><FRAME SRC="footer.htm" NORESIZE SCROLLING=NO>
</FRAMESET>
```

Вышеприведенная фреймовая структура делит экран на три горизонтальные части. Верхняя часть занимает 15% высоты окна браузера и ее содержимым является документ header.htm. Область окна браузера (70% высоты), отведенная под средний фрейм, содержит вложенную структуру фреймов, разделяющих родительский фрейм на два равных столбца, в них выводятся документы left.htm и right.htm. Оставшуюся нижнюю часть окна браузера занимает фрейм, в который выводится документ footer.htm.

Блоки

Блоки создаются при помощи тега <DIV>...</DIV>. Элемент <DIV> является блочным элементом и предназначен для выделения фрагмента документа.

Как правило, вид блока управляется с помощью стилей. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить параметр class или id.

Тег <DIV>...</DIV> имеет ряд параметров:
align - задаёт выравнивание содержимого тега <DIV>. Варианты: center | left | right | justify.
title - добавляет всплывающую подсказку (hint) к содержимому, например title="Новости".
unselectable="on"|"off" - запрещает или разрешает выделять текст в данном блоке. Значение "on" запрещает выделение текста (например, если вы не хотите, чтобы его копировали с вашей страницы).

Самое важное – позиционирование блока в нужном месте страницы.

Для этого тег DIV в обязательном порядке используется вместе с таблицей стилей CSS.

В CSS есть два способа позиционирования блока в нормальном потоке: относительный (relative) и статический (static).

Способ Static используется по умолчанию и выполняет стандартное форматирование, оставляя блок в нормальном потоке .

Способ Relative позволяет сдвинуть положение блока относительно того положения, которое он занимал бы в нормальном потоке.

Например, position: relative; left: 20px; color: green;

Это текст, сдвинутый на 20 пикселей вправо.

Аналогично можно сдвигать блок по вертикали - для этого используется свойство top. Можно также использовать right и bottom, если требуется.

Абсолютное позиционирование.

При абсолютном позиционировании блок изымается из нормального потока и помещается браузером в то место, которое вы указали.

Пример:

position: absolute; top: 10px; right: 10px;
background: green; border: 1px dotted red; width: 15px; height: 15px;

Получаем небольшой (15x15 пикселей) зеленый (background: green) квадратик с красной (red) пунктирной (dotted) границей толщиной 1 пиксел (border:1 px) в правом верхнем углу страницы (отступ от верха на 10 пикселей и от правого края окна на 10 пикселей).

Вложенные блоки. Если у блока-родителя определено свойство position, то вложенный блок позиционируется относительно родителя, а если это свойство не определено, то относительно окна браузера.

Наиболее типичной является ситуация, когда страница разбивается на несколько абсолютно позиционированных блоков, внутри которых используется относительное позиционирование (а так как вложенные блоки позиционируются относительно родителя, то структура страницы сохраняется).

Блочная структура позволяет (порой весьма заметно) сократить размер страницы и очень сильно повышает логичность документа, так как порядок данных в коде соответствует логике документа.

Еще один способ позиционирования: плавающий (float). Плавающие блоки ведут себя очень похоже на картинки с указанным тегом align: они прижимаются к указанному краю контейнера, заставляя "нормальный поток" их обтекать.

Слои

Блок, описываемый тегом DIV, можно воспринимать и как слой. Слои в HTML похожи на слои в Photoshop: их можно представить как прозрачнее пленки, наложенные одна на другую. Слой описывается точно так же, как и блок.

При этом в CSS-описании параметров стиля появляются новые возможности:

background-image:url(путь) - определяет фоновое изображение слоя. В качестве пути можно указывать как полный URL, так и относительный путь

z-index:число –служит для определения порядка расположения слоев на странице. Чем меньше значение свойства z-index, тем позднее слой выводится на экран.

Полный пример тега DIV, определяющего слой, может выглядеть следующим образом:

```
<div style="position:absolute; top:25; left:25; width:200; height:400;
background-image:url(/img/bgimg.gif);">
Содержимое слоя
</div>
```

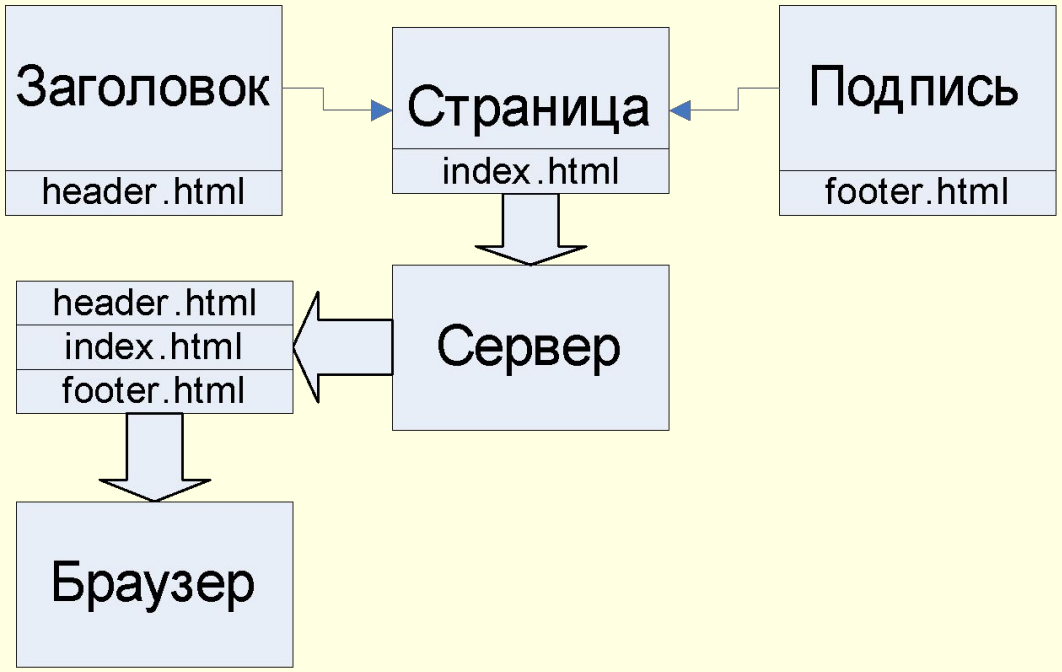
Технология SSI

Иногда в адресах Web-страниц можно видеть расширение не .html, а .shtml. Это указывает на то, что данная страница выполнена с применением технологии SSI (Server Side Includes) и размещена на сервере, поддерживающем SSI.

Основная идея SSI – вынесение повторяющихся фрагментов страниц в отдельные файлы и затем включение их в нужное место HTML-документа специальной SSI-директивой.

Эта технология помогает экономить время при создании сайта и, насколько это возможно, облегчить работу web-дизайнера.

SSI-технология



Все SSI-директивы заключаются в комментарии HTML-кода. Главным отличием директивы от просто комментария является символ "#" сразу после "<!--". То есть все SSI-директивы заключаются в символы

<!--#:[пробел]-->

После символа # пишутся команды SSI:

include - включает текст указываемого файла в данный документ. Положение файла на сервере указывается с помощью следующих атрибутов.
file - необходимо указать путь относительно данного документа.
virtual - можно указывать как относительный (как в случае с file), так и абсолютный путь.

set - устанавливает значение какой-либо SSI-переменной. Переменная объявляется с помощью атрибута var, а её значение задается с помощью атрибута value.
Например:<!--#set var="a" value="variable" -->

echo - выводит на экран значение переменной, указанной в атрибуте var.
Например: <!--#echo var="a" -->

Условные операторы имеют следующую форму написания:

```
<!--#if expr="condition" -->
<!--#elif expr="condition" -->
<!--#else -->
<!--#endif -->
```

где condition - условие для сравнения.

Пример использования условного оператора:

```
<!--#set var="Monday" -->
<!--#if expr="$Monday " -->
<p>Сегодня понедельник.</p>
<!--#else -->
<p>Что угодно, но не понедельник.</p>
<!--#endif -->
```