

Таблицы и фреймы

По материалам курса University of
Washington

<http://www.cs.washington.edu/education/courses/cse190m/07sp/index.shtml>

Таблицы

Таблицы следует использовать в том случае, если размещаемый материал имеет табличную природу: расписание, глоссарий,...

Таблицы не следует использовать для размещения материала на странице, для этого лучше использовать позиционирование блоков.

Таблица – это блочный элемент `<table>`, содержащий внутри себя строки – блочные элементы `<tr>`, которые, в свою очередь, содержат внутри себя элементы данных – ячейки (строчные элементы `<td>`).

Пример таблицы: расписание занятий.

```
<table>
  <caption>Расписание занятий</caption>
  <tr><td>Понедельник</td><td>Алгебра</td><td>15:00</td></tr>
  <tr><td>Среда</td><td>Матанализ</td><td>13:30</td></tr>
  <tr><td>Пятница</td><td>Javascript</td><td>17:00</td></tr>
</table>
```

Все, что находится вне ячеек таблицы, размещается перед таблицей.

Например, элемент `<caption>` в примере, приведенном выше, задает заголовок таблицы.

Заголовки и стили

Кроме собственно заголовка таблицы, можно некоторые ячейки помечать как заголовки. По умолчанию они выделяются жирным шрифтом и центрируются.

```
<table>
  <caption>Расписание занятий</caption>
  <tr><th>День</th><th>Предмет</th><th>Время</th></tr>
  <tr><td>Понедельник</td><td>Алгебра</td><td>15:00</td></tr>
  <tr><td>Среда</td><td>Матанализ</td><td>13:30</td></tr>
  <tr><td>Пятница</td><td>Javascript</td><td>17:00</td></tr>
</table>
```

С помощью стилей можно задать вид таблицы и отдельных ее элементов. Например:

```
table { caption-side: bottom; border: 2px solid; }
th { font-size: larger; background-color: yellow; }
td { background-color: #e0e0e0; color: blue; }
caption { font-size: 16pt; font-style: italic; }
```

IE не понимает указание положения заголовка `caption-side`!

[table.html](#)

Коллапс границ

По умолчанию границы ячеек отстоят друг от друга и от границы самой таблицы на некоторое расстояние. Это легко увидеть, если задать границу ячеек и таблицы:

```
table { border: 1px solid black; }  
td { border: 1px solid black; }
```

С помощью задания стиля `border-collapse` можно убрать промежутки между границами:

```
table { border-collapse: collapse; border: 1px solid black; }  
td { border: 1px solid black; }
```

[collapse.html](#)

Стиль, при котором границы таблицы и отдельных ячеек разделены, задается значением `border-collapse: separate;` (принят по умолчанию). При этом можно задавать ширину промежутка между ячейками таблицы с помощью стиля `border-spacing`, например

```
table { border-collapse: separate;  
        border: 1px solid black;  
        border-spacing: 0.2cm 1em; }
```

Колонки

Вообще говоря, таблица состоит из строк (<tr>). Но для задания стилей для отдельных колонок можно указать отдельные элементы для разных колонок и групп колонок в таблице (<col>, <colgroup>).

```
<table>
  <caption>Расписание занятий</caption>
  <col class="firstcol"/>
  <colgroup class="maincolumns"><col/><col/></colgroup>

  <tr><th>День</th><th>Предмет</th><th>Время</th></tr>
  <tr><td>Понедельник</td><td>Алгебра</td><td>15:00</td></tr>
  <tr><td>Среда</td><td>Матанализ</td><td>13:30</td></tr>
  <tr><td>Пятница</td><td>Javascript</td><td>17:00</td></tr>
</table>
```

Чаще всего это используется для задания ширины отдельных колонок, но может также использоваться для выделения колонок цветом, задания специфического шрифта и т.д.

[columns.html](#)

Слияние ячеек

Для некоторых отдельных ячеек можно указать, что данные занимают несколько рядов и/или колонок. Для этого в элементе `<td>` задают атрибуты `rowspan` и `colspan`, например:

```
<table>
  <caption>Расписание занятий</caption>
  <col class="firstcol"/>
  <colgroup class="maincolumns"><col/><col/></colgroup>

  <tr><th>День</th><th>Предмет</th><th>Время</th></tr>
  <tr><td rowspan="2">Понедельник</td>
    <td>Алгебра</td><td>15:00</td></tr>
  <tr><td>Геометрия</td><td>15:00</td></tr>
  <tr><td>Среда</td><td>Матанализ</td><td>13:30</td></tr>
  <tr><td>Пятница</td>
    <td colspan="2">Физкультура - целый день</td></tr>
</table>
```

[spans.html](#)

Фреймы

Страница может быть разбита визуально на несколько областей, в каждую из которых помещается содержание отдельной страницы. Структура разбиения задается в отдельном HTML-файле, а уже в нем имеются ссылки на страницы, формирующие содержимое.

```
<html>
  <head>
    <title>Расписание занятий</title>
  </head>
  <frameset cols="350,*">
    <frame src="contents1.html"/>
    <frame src="contents2.html" name="frame2"/>
  </frameset>
</html>
```

[frames.html](#)

Замечания:

1. Элементом `frameset`'а может быть `frameset`, что позволяет получать различные структуры разбиений.
2. Браузер может не поддерживать фреймы (это очень старые браузеры).
3. Если в страницу включен DTD, то это специальный Frameset DTD.
4. Элемент `<body>` включается только для элемента `<noframes>`.

Управление поведением и видом фреймов

По умолчанию каждый фрейм имеет границу, размер его можно менять динамически и, если содержимое страницы не помещается во фрейм, то к нему автоматически добавляются полосы прокрутки (scrollbars).

Все это можно задавать с помощью атрибутов элемента `<frame>` (не CSS!)

Атрибут	Возможные значения
<code>frameborder</code>	0, 1
<code>noresize</code>	<code>noresize</code>
<code>scrolling</code>	yes, no, auto
<code>marginwidth</code> , <code>marginheight</code>	размер в пикселях

```
<frameset rows="200,*">
  <frame src="frame1.html" frameborder="0" noresize="noresize"/>
  <frameset cols="25%,75%">
    <frame src="frame2.html" name="frame2" scrolling="no"/>
    <frame src="frame3.html" name="frame3" marginwidth="50"/>
  </frameset>
</frameset>
```

[frames-attr.html](#)

Загрузка страниц во фреймы

При переходах по гиперссылкам можно указывать, в какое окно загружать целевую страницу, с помощью атрибута `target`.

<code></code>	Загружает страницу в новое окно
<code></code>	Загружает страницу в текущее окно
<code></code>	Загружает страницу на место главной («вырваться» из фреймов)
<code></code>	Загружает страницу в окно (фрейм) с именем <code>name</code>

[targets.html](#)

Того же эффекта можно добиться, используя метод `open` в программах на *Javascript*.

```
window.open(url, name);
```

где аргумент `url` соответствует атрибуту `href`, а аргумент `name` – атрибуту `target`.

[buttons.html](#)

Особенности программирования

Программы, исполняющиеся в разных фреймах, независимы друг от друга, они используют разный глобальный контекст, объект `window` у них свой.

Тем не менее, можно получить ссылку на контекст другого окна, используя связь «родитель – ребенок».

Если хотим изнутри фрейма получить ссылку на «родительское» окно:

```
window.parent;
```

Если хотим из «родительского» окна получить доступ к документу фрейма:

```
window.document.getElementById(frameId).contentDocument;
```

IE не поддерживает этот W3C стандарт!

Но в нем документ имеет массив фреймов, а у них есть атрибут `document`:

```
window.document.frames[frameId].document;
```

Пример программирования

Напишем функцию, которая работает в одном фрейме, и при этом определяет цвет фона для тела документа в другом фрейме.

```
function findColor(frameName) {  
    var frame = parent.document.getElementById(frameName);  
    if (!frame) return 'white';  
    var color = frame.contentDocument.body.style.backgroundColor;  
    if (!color) color = 'white';  
    return color;  
}
```

Вот как подобная функция используется в примере (реальный пример сделан чуть-чуть сложнее, чем приведенная выше функция, чтобы обеспечить совместимость с IE):

[js-frames.html](#)
