

CSS-таблица

У таблиц в истории развития HTML непростая судьба. Они, можно сказать, оказались не в то время, не в том месте: таблицы долгое время использовали не по назначению для разметки каркаса страниц. Но так как таблица — это довольно «тяжёлый» инструмент и предназначена для других целей, табличной разметкой страниц было сложно управлять. Ситуация осложнялась тем, что на заре эпохи интернета, браузеры могли отображать таблицы совсем по-разному, что также усложняло контроль разметки.

С тех пор за таблицами закрепились нехорошая репутация. Некоторые разработчики до сих пор боятся использовать таблицы в разметке, даже там, где они вполне уместны. Самое время избавляться от этой пагубной привычки. Тем более, что сейчас таблицы можно использовать не только в HTML, но и в CSS.

Чтобы из любого элемента сделать таблицу в CSS, нужно его свойству `display` присвоить значение `table`. Всё просто:

```
.this-is-table {  
    display: table;  
}
```

В этом случае блок `.this-is-table` станет отображаться равнозначно тегу `<table>`.

Ряды и ячейки таблицы

Как вы уже поняли, мы можем одновременно и иметь семантическую разметку, и отображать её таблицей, если захотим. А в случае необходимости, можно просто отменить `display: table`, назначив другое подходящее значение `display`: например, `block`. Это ключевое преимущество перед HTML-таблицами, у которых изменить отображение на «нетабличное» нельзя.

Давайте продолжим разбираться с таблицей и полками. У HTML-таблиц есть ряды `<tr>` и ячейки `<td>`. Их аналоги в мире CSS-таблиц — значения свойства `display: table-row` и `table-cell`.

```
.this-is-table-row {
    display: table-row;
}

.this-is-table-cell {
    display: table-cell;
}
```

```
}
```

Ячейки включаются в ряды, ряды в саму таблицу.

Заголовок таблицы

Следующий элемент мира HTML-таблиц — табличный заголовок `<caption>`. Его представление в CSS:

```
.this-is-caption {  
  
  display: table-caption;  
  
}
```

В HTML-таблице тег заголовка располагается внутри неё, в CSS-таблице это так же.

Также к заголовкам CSS-таблиц применимо свойство для вертикального позиционирования заголовка таблицы `caption-side`.

Группировка верхних рядов и ячеек

В HTML-таблицах можно группировать табличные ряды «шапки», основного содержания и «подвала».

Для группировки верхних рядов используется тег `<thead>`. В него включаются один или несколько рядов. Можно использовать несколько элементов `<thead>` одновременно.

Аналог группировки верхних рядов в CSS — свойство `display: table-header-group`. Отметим, что блок, которому задано это свойство, всегда отображается вверху CSS-таблицы, даже если в разметке он идёт не первым. Это поведение идентично тегу `<thead>`.

Группировка нижних рядов и ячеек

Аналогично «шапке» у таблицы можно сделать «подвал».

В HTML-таблицах это тег `<tfoot>`, а в CSS — свойство `display: table-footer-group`. Как и в случае «шапки», в таблицу можно включать несколько «подвалов» одновременно.

Аналогично «шапке», блок, которому задано свойство `display: table-footer-group`, всегда отображается снизу CSS-таблицы, даже если в разметке он идёт не последним. Это поведение идентично тегу `<tfoot>`.

Группировка основных рядов и ячеек, часть 1

Ряды содержания таблицы в HTML включаются в тег `<tbody>`. В CSS же эту задачу выполняет свойство `display: table-row-group`.

Дополнительные обёртки очень удобны для стилизации элементов таблицы.

Группировка основных рядов и ячеек, часть 2

Так же, как допускается несколько шапок и подвалов таблицы, может быть и несколько групп её содержимого. Это также удобно для стилизации отдельных групп.

Колонка таблицы

В HTML-таблицах для стилизации столбцов используется тег `<col>`. Тег пишется в начале таблицы и не закрывается. Первый `<col>` в разметке влияет на каждую первую ячейку в рядах таблицы, то есть, на первый столбец. Следующий `<col>` будет стилизовать второй столбец и так далее.

В примере ниже ячейкам 1.1 и 1.2 (первый столбец) задаётся ширина `20%`, а ячейкам 2.1 и 2.2 (второй столбец) ширина — `80%`:

```
<table>

  <col width="20%">

  <col width="80%">

  <tr>

    <td>1.1</td> <td>2.1</td>

  </tr>

  <tr>

    <td>1.2</td> <td>2.2</td>

  </tr>

</table>
```

В CSS-таблицах всё так же: внутри таблицы нужно создать пустой тег и задать ему свойство `display: table-column` — это аналог тега `<col>`. Правила для него будут применяться к первому столбцу, следующий элемент с `table-column` стилизует второй столбец и так далее.

Группировка колонок таблицы

Столбцы тоже можно группировать. В HTML-таблицах для этой цели используется тег `<colgroup>`. Теги `<col>` просто включаются в `<colgroup>` и стилизуется уже группа целиком, а не отдельные столбцы.

В примере ниже ячейкам 1.1 и 1.2 (первый столбец) задаётся ширина `20%`, а ячейкам 2.1, 2.2, 3.1 и 3.2 (второй и третий столбец) ширина — `80%`:

```
<table>

  <col width="20%">

  <colgroup width="80%">

    <col>

    <col>

  </colgroup>

  <tr>

    <td>1.1</td> <td>2.1</td> <td>3.1</td>

  </tr>

  <tr>

    <td>1.2</td> <td>2.2</td> <td>3.2</td>

  </tr>

</table>
```

В CSS-таблицах столбцы группируются аналогично. Группа столбцов создаётся свойством `display: table-column-group` (аналог тега `<colgroup>`). Стилизуя группу, мы задаём правила для каждого дочернего столбца.

Строчная таблица

В CSS существуют блочные `block` и строчные `inline` элементы. Есть также и блочно-строчные `inline-block` элементы, сочетающие свойства обоих типов.

Таблицы при этом имеют свой собственный тип `table`, но ведут себя в потоке документа как блоки, то есть занимают отдельную строку. Существует ли возможность сделать таблицу «строчной», чтобы она располагалась в одной строке с другими строчными элементами? Да! И такое поведение таблицы задаётся свойством `display: inline-table`.

Горизонтальное выравнивание таблицы

Лёгкий способ горизонтально отцентровать блок произвольной ширины на странице — присвоить ему отображение таблицей и задать `margin: auto`.

В этом таблица схожа по поведению с обычным блочным элементом с фиксированной шириной, но при этом ширина таблицы может быть динамической.

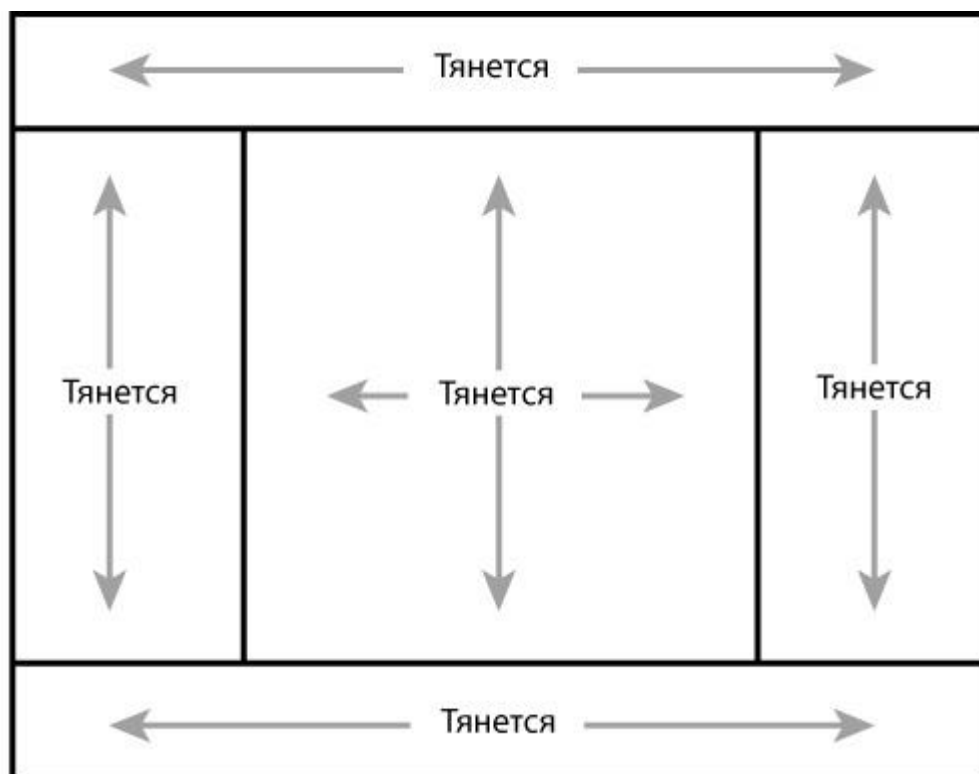
Ячейки и границы таблицы

В CSS-таблицах можно управлять отступами и схлопыванием рамок между ячейками. Для этого, как и у HTML-таблиц, используются свойства `border-collapse` и `border-spacing`.

Свойство `border-collapse` «схлопывает» границы ячеек, а `border-spacing` — определяет расстояние между ячейками внутри таблицы.

Однако, у свойства `border-spacing` есть одна тонкость. Если в нём задано одно значение, то расстояние между ячейками устанавливается одновременно по вертикали и горизонтали. Если значений два, то первое определяет горизонтальное расстояние, а второе — вертикальное.

Holy Grail, часть 1



Итак, теперь у нас есть всё, чтобы заполучить Святой Грааль вёрстки!

Так называемая разметка «Holy Grail» — это вариант разметки страницы с шапкой, тремя равными по высоте колонками (две фиксированные боковые колонки и тянущийся центр) и прилипающим подвалом. Центральная колонка должна тянуться, а боковые — иметь фиксированную ширину.

Давайте по шагам разберём, как получить Святой Грааль.

Начнём с «прилипающего» подвала. Он назван так, потому что:

- «прилипает» к самому низу страницы, когда основное содержимое страницы небольшое и не отталкивает подвал ниже экрана,
- «отлипает», когда высота содержимого становится достаточно большой и влияет на высоту страницы.

Для реализации «прилипающего» подвала воспользуемся CSS-таблицами. Для начала зададим главному контейнеру табличное отображение и 100% ширину и высоту.

Чтобы блок был 100% высоты области просмотра, всем его родительским элементам (в нашем случае `html` и `body`) тоже должна быть задана 100% высота.

Holy Grail, часть 2

Теперь давайте шапку и подвал сделаем рядами нашей табличной разметки. Это даст особый эффект: ряды растянутся и заполнят всю ширину и высоту таблицы, а последний ряд выравнивается по нижней части таблицы — это то, что нам нужно.

Теперь остаётся только задать нужную высоту шапки и подвала. Её можно задать минимально необходимой, например, будет достаточно `1px`. В результате вычисленная высота ряда станет минимальной, но не будет меньше высоты его содержимого.

Таким образом, наш «прилипающий» подвал будет работать как следует.

Holy Grail, часть 3

Теперь разберёмся со столбцами нашей разметки.

Блок `main` также отобразим таблицей, вложенной в `.table-layout`. Дочерние блоки `main` сделаем ячейками таблицы, чтобы расположить их в ряд. И, наконец, растянем `main` на 100% доступной ширины и высоты, и вместе с `main` растянутся ячейки.

Такая разметка `main` очень гибкая: можно без проблем менять ширину боковых столбцов, при этом основная колонка будет занимать всю оставшуюся ширину.