Урок 4

Псевдоклассы, табличная верстка

Псевдоклассы и псевдоэлементы. Создание таблиц. Объединение ячеек. Вложенные таблицы. Стилевое оформление таблиц.

Псевдоклассы и псевдоэлементы

Псевдоклассы и псевдоэлементы позволяют назначить CSS-стили структурам, существование которых на веб-странице не обязательно. Т.е. стили применяются к частям документа не на основании той информации, которая содержится в его структуре, и даже изучив эту структуру, невозможно понять, что к этим элементам применяются какие-либо стили.

Псевдоклассы

Псевдоклассы позволяют добавлять особые классы к элементам, выбирая объекты, которых нет в структуре веб-страницы либо которые нельзя выбрать с помощью обычных селекторов, например, первая буква или первая строка одного абзаца.

Вы наверняка замечали на многих сайтах: когда вы наводите мышкой на какой-либо пункт меню, он меняет свой вид, у него может изменяться цвет фона, цвет ссылки, даже шрифт или его размер. Это происходит благодаря псевдоклассам. Рассмотрим их синтаксис.

```
Селектор:псевдокласс {
      свойство1: значение1;
}
a:hover {
     color: #ccc;
}
```

После селектора ставится двоеточие, и сразу после него без пробела указывается название псевдокласса.

Псевдоклассы, определяющие состояние элементов

a:link – ссылается на непосещенную ссылку.

a:visited – ссылается на уже посещенную ссылку.

a:hover – ссылается на любой элемент, по которому проводят курсором мыши.

a:focus – ссылается на любой элемент, над которым находится курсор мыши.

a:active – ссылается на элемент, который был активизирован пользователем.

:valid – выберет поля формы, содержимое которых прошло проверку в браузере на соответствие указанному типу.

:invalid – выберет поля формы, содержимое которых не соответствует указанному типу.

:enabled – выберет все доступные (активные) поля форм.

:disabled – выберет заблокированные поля форм, т.е. находящиеся в неактивном состоянии.

:in-range – выберет поля формы, значения которых находятся в заданном диапазоне.

:out-of-range – выберет поля формы, значения которых не входят в установленный диапазон.

:lang() – выбирает абзацы на указанном языке.

:not(селектор) – выберет элементы, которые не содержат указанный селектор, например класс, идентификатор или селектор элемента.

:not([type="submit"]). :target – выбирает элемент с символом #, на который ссылаются в документе.

:checked – выбирает выделенные (выбранные пользователем) элементы.

Структурные псевдоклассы

:nth-child(odd) – выбирает нечетные дочерние элементы.

:nth-child(even) – выбирает четные дочерние элементы.

:nth-child(3n) – выбирает каждый третий элемент среди дочерних.

:nth-child(3n+2) – выбирает каждый третий элемент, начиная со второго дочернего элемента (+2).

:nth-child(n+2) – выбирает все элементы, начиная со второго.

:nth-child(3) – выбирает третий дочерний элемент.

:nth-last-child() – в списке дочерних элементов выбирает элемент с указанным местоположением, аналогично с :nth-child(), но начиная с последнего, в обратную сторону.

:first-child – позволяет оформить только самый первый дочерний элемент тега.

:last-child – позволяет форматировать последний дочерний элемент тега.

:only-child – выбирает элемент, являющийся единственным дочерним элементом.

:empty – выбирает элементы, у которых нет дочерних элементов.

:root – выбирает элемент, являющийся корневым в документе (элемент html).

Псевдоклассы по типу дочернего элемента

:nth-of-type() – выбирает элементы по аналогии с :nth-child(), при этом берет во внимание только тип элемента.

:first-of-type – позволяет выбрать первый дочерний элемент.

:last-of-type – выбирает последний тег конкретного типа.

:nth-last-of-type() – выбирает элемент заданного типа в списке элементов в соответствии с указанным местоположением, начиная с конца.

:only-of-type – выбирает единственный элемент указанного типа среди дочерних элементов родительского элемента.

Пример использования псевдоклассов

HTML	CSS

   - Первый элемент
   - Второй элемент
   - Третий элемент
 	
	li:first-child{
	  font-size: 24px;
	  color: #F23401;
	}

Чтобы понять, как работает данный псевдокласс, рассмотрим простой пример. Задаем элементу списка <li> псевдокласс first-child, и у него прописываем определенный стиль. Как видно, маркированный список состоит из трех элементов. Заданный стиль будет применен ТОЛЬКО к первому элементу данного списка. Это происходит потому, что первый элемент списка <li> будет именно первым дочерним у тега <ul>.

Комбинирование псевдоклассов

Псевдоклассы можно комбинировать в одном селекторе, просто перечисляя их через двоеточие, как показано в следующем примере.

```
/* При наведении на не посещенную ссылку цвет текста будет зеленым */
a:link:hover {
    color: #0F0;
}

/* При наведении на посещенную ссылку цвет текста будет красным*/
a:visited:hover {
    color: #F00;
}
```

Псевдоэлементы

Псевдоэлементы – практически то же самое, что и псевдоклассы, только они позволяют ввести несуществующие элементы в веб-документ и придать им определенные стили. Псевдоэлементы появились еще в CSS1, но пошли в релиз только в CSS2.1. В самом начале в синтаксисе использовалось одно двоеточие, но в CSS3 используется двойное двоеточие для отличия от псевдоклассов. Современные браузеры умеют понимать оба типа синтаксиса псевдоэлементов, кроме Internet Explorer 8, который воспринимает только одно двоеточие. Поэтому надежнее использовать одно. С помощью свойства content можно изменить внешний вид части элемента.

В CSS существует 4 псевдоэлемента:

:first-letter – выбирает первую букву каждого абзаца, применяется только к блочным элементам.

:first-line – выбирает первую строку текста элемента, применяется только к блочным элементам.

:before – вставляет генерируемое содержимое перед элементом.

:after – добавляет генерируемое содержимое после элемента.

Пример использования псевдоэлементов

HTML	CSS

   - Первый элемент
   - Второй элемент
   - Третий элемент
 	
	li:after{
	  content: "new";
	  color: #F00;
	}

После элемента списка li появится текст new красного цвета.

Для чего нужны таблицы в HTML

Основное назначение таблиц в HTML – представление табличных данных, таких как: просмотр информации о пользователях, просмотр заказанных товаров в интернет-магазине, просмотр отчетов о продажах и др. Второе назначение таблиц в HTML таково, что при помощи их можно верстать веб-страницы. А верстать – это значит разбивать страницу на элементы, т.е. формировать различные блоки сайта, такие как шапка, меню, контент, подвал и другие. Табличный способ верстки в настоящее время считается неправильным и устаревшим, потому что таблицы используются не по назначению, но многие сайты либо уже сверстаны при помощи таблиц, либо продолжают верстаться этим способом. Поэтому разработчику сайтов важно уметь пользоваться как табличным способом верстки, так и верстской слоями. Оба способа верстки мы рассмотрим в данном курсе.

Структура таблицы в HTML

```
<table>
      <tr>
           <td>Столбец 1</td>
           <td>Столбец 2</td>
           <td>Столбец 3</td>
       </tr>
</table>
```

Любая таблица в HTML помещается в контейнер <table>, после чего в контейнер строки <tr> помещаются уже столбцы таблицы, которые помещаются в тег <td>. В данном примере таблица будет состоять из трех столбцов и одной строки, т.е. в таблице будет 3 ячейки. Чтобы добавить еще одну строку в данную таблицу, нужно вложить в контейнер <table> еще один контейнер строки <tr>, и в него поместить то же количество столбцов <td>, т.е. 3 столбца.

```
<table>
      <tr>
           <td>Столбец 1</td>
           <td>Столбец 2</td>
           <td>Столбец 3</td>
      </tr>
      <tr>
           <td>Столбец 1</td>
           <td>Столбец 2</td>
           <td>Столбец 3</td>
      </tr>
</table>
```

Создание ячеек таблицы

Элемент <td> создает ячейки таблицы с данными, добавляя их в строку таблицы. Парные теги <td>…</td>, расположенные между тегами соответствующей строки, определяют количество ячеек в пределах одной строки. Количество пар ячеек таблицы должно быть равным количеству пар ячеек заголовка.

Элемент <th> создает заголовок – специальную ячейку, текст в которой выделяется полужирным шрифтом. Количество ячеек заголовка определяется количеством пар тегов <th>…</th>.

Группировка строк и столбцов таблицы

Элемент <colgroup> создает структурную группу столбцов, выделяя логически однородные ячейки. Группирует один или более столбцов для форматирования, позволяя применить стили к столбцам, вместо того чтобы повторять стили для каждой ячейки и для каждой строки. Добавляется непосредственно после тегов <table> и <caption>. Элемент <col> формирует неструктурные группы столбцов, которые делят таблицу на разделы, не относящиеся к общей структуре, т.е. не содержащие информацию одного типа. Позволяет задавать свойства столбцов для каждого столбца в пределах элемента <colgroup>. С помощью атрибута <style> можно изменить основной цвет фона ячеек. Для элемента <col> доступен атрибут span, задающий количество столбцов для объединения.

Объединение ячеек таблицы

Чтобы в полной мере начать использовать таблицы в HTML, необходимо научиться объединять ячейки. Для этого у тега <td> или <th> существуют атрибуты colspan и rowspan.

- colspan объединяет ячейки по горизонтали.
- rowspan объединяет ячейки по вертикали.

<table>
<tr><td rowspan="2">**rowspan** – объединение по вертикали (строк)</td><td colspan="2">**colspan** – объединение по горизонтали (столбцов)</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

В данном примере при помощи атрибута rowspan в первом столбце объединяются 2 строки. При помощи атрибута colspan в первой строке объединяются уже два столбца.

Посмотрим, как это выглядит в HTML.

```
<table>
    <tr>
        <td rowspan="2">Столбец 1</td>
        <td colspan="2">Столбец 2</td>
    </tr>
    <tr>
        <td>Столбец 5</td>
        <td>Столбец 6</td>
    </tr>
    <tr>
        <td>Столбец 7</td>
```

```
        <td>Столбец 8</td>
        <td>Столбец 9</td>
    </tr>
</table>
```

В первой строке данной таблицы будет всего два столбца, в первом столбце первая строка объединяется со второй строкой, поэтому у первой ячейки указываем атрибут rowspan со значением 2, что означает, что нужно объединить две ячейки. В этой же строке во втором и третьем столбце объединяем две ячейки по горизонтали при помощи атрибута colspan, также со значением 2. Далее во второй строке нужно учесть, что не надо указывать для нее первый столбец, т.к. он уже объединен с первой строкой. Поэтому во второй строке указываются два столбца, начиная со второго. В третьей строке присутствуют все три столбца по порядку.

Стилевое оформление таблиц

По умолчанию таблица и ячейки таблицы не имеют видимых границ и фона, при этом ячейки внутри таблицы не прилегают вплотную друг к другу. Ширина ячеек таблицы определяется шириной их содержимого, поэтому может быть разной. Высота всех ячеек сетки строки по умолчанию одинаковая.

Границу таблице можно задавать аналогично любому другому элементу, это мы рассматривали на прошлом занятии. Единственное, если задать свойство border только тегу <table>, то граница будет задана только внешним границам таблицы. Это происходит потому, что свойство border не наследуется. Чтобы задать границы для ячеек, нужно задать их тегу <td>. Здесь уже можно задавать границы ячейкам любого стиля.

```
/*Внешняя граница таблицы*/
table {
      border: 1px solid #000;
}
/*Границы для ячеек таблицы*/
td {
      border: 1px solid #000;
}
```

Ширину и высоту таблицы задаем при помощи свойств CSS width и height.

```
table {
      width: 400px;
      height: 200px;
}
```

По умолчанию ширина и высота таблицы определяются содержимым ее ячеек. Если не задать ширину, то она будет равна ширине самого широкого ряда.

Ширину таблицы и ее столбцов можно задать с помощью свойства width. Если для таблицы задана ширина table {width: 100%;}, то она будет равна ширине блока-контейнера, а ширина столбцов

установится в соответствии с шириной содержимого ячеек. Чаще всего ширину таблицы и столбцов задают в px или %.

Есть ещё одно полезное свойство для оформления таблиц. Если для всех ячеек задать атрибут cellspacing=”0” или CSS свойство border-spacing=”0”, то границы соседних ячеек будут двойными. Для того, чтобы убрать двойные границы, в CSS существует свойство border-collapse.

```
table {
       border-collapse: collapse;
}
```

Свойство border-collapse может принимать два значения. По умолчанию установлено separate, при котором рамка двойная, а при значении collapse рамка становится одинарной.

Основные теги для верстки

При верстке сайта с помощью слоев самым часто используемым HTML-тегом является , который как раз и формирует слой на веб-странице. Он является блочным тегом. Второй тег, который используется при верстке, – это строчный тег . Сами по себе эти теги ничего на экране не отображают, и оформляются они стилями CSS.

```
<div>Блочный элемент</div>
<span>Строчный элемент</span>
```

Блочный элемент создает разрыв строки перед тегом и после него. Он образуют прямоугольную область, по ширине занимающую всю ширину веб-страницы или блока-родителя, если для него не задано значение width.

Блочные элементы могут содержать внутри себя элементы любого типа. Нельзя размещать блочные элементы внутри строчных, за исключением элемента <img>. Для блочных элементов можно задавать margin и padding. Свойства width и height устанавливают ширину и высоту области содержимого элемента. Фактическая ширина элемента складывается из ширины полей (внутренних отступов), границ и внешних отступов.

Строчные элементы не создают блоки, они отображаются на одной строке с содержимым рядом стоящих тегов. Строчные элементы являются потомками блочных элементов. Они игнорируют верхние и нижние margin и padding, но, если для элемента задан фон, он будет распространяться на верхний и нижний padding, заходя на соседние строки текста.

Ширина и высота строчного элемента зависит только от его содержания, задать его размеры с помощью CSS нельзя. Можно увеличить расстояние между соседними элементами по горизонтали с помощью горизонтальных полей и отступов.

Давайте вспомним, какие основные элементы HTML относятся к блочным, а какие к строчным.

Блочные элементы

- <h1>...</h1>
- <h2>...</h2>
- <article>…</article>
- <aside>…</aside>
- <output>…</output>
- …

- <h3>...</h3>
- <h4>...</h4>
- <h5>...</h5>
- <h6>...</h6>
- …
- <legend>…</legend>
- <nav>…</nav>
- …
- <body>…</body>
- <blockquote>…</blockquote>
- …
- <fieldset>…</fieldset>
- <footer>…</footer>
- <form>...</form>
- <header>…</header>
- ...
- <pre>…</pre>
- <section>…</section>
- …

Строчные элементы

- ...
- ...
- ...
- ...
- ...
- ...
- <img>
- <input>
- ...

В этом списке тег картинки <img> является замещаемым строчным элементом, т.е. при помощи замещаемых элементов указывается, что в данном месте должен быть какой-то сторонний объект, в данном случае картинка. Замещаемому элементу можно задать ширину и высоту, но он все равно будет являться строчным.

Домашнее задание

Добавление контента на нашей странице, создание первых блоков и эффектов на ссылки сайта.

1. В подвале сайта, под разделительной чертой hr необходимо добавить список
   a. Произвольный текст 6 слов
   b. Номер телефона
   c. Имейл
2. Данное домашнее задание посвящено разбиению нашего сайта на блоки
3. Вам представлено изображение сайта сайта(не макет, просто картинка), на нем разными цветами выделены блоки, на которые необходимо разбить нашу страницу
4. Никаких новых элементов из макета добавлять не нужно, вы с ним начнете работать начиная с 6 урока.
5. Всем блокам необходимо задать логичное название класса, запрещается использовать перечисление (block1, block2, block3... block378)
6. Для страницы контакты, также представлено изображение страницы, только без разбиения на блоки, тут вам необходимо самостоятельно разделить ваш сайт на блоки.

Дополнительные материалы

Популярно о псевдоэлементах :Before и :After.
Псевдоклассы и псевдоэлементы.
Дополнительный материал о таблицах.
div & span.

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. http://htmlbook.ru/samcss/psevdoelementy.
2. http://htmlbook.ru/samcss/psevdoklassy.
3. http://html5book.ru/osnovy-css/.
4. http://html5book.ru/psevdoklassy/.
5. http://html5book.ru/html-table/.

6. http://htmlbook.ru/css/border-collapse.
7. http://technologyweb.org/div-%D0%B8-span/.