**Конспект по теме "Что такое CSS. Селекторы и свойства"**

**Что такое css и зачем он используется?**

На предыдущих занятиях вы уже познакомились с разметкой html, которая является базой/структурой веб-страницы. Но дело в том, что html изначально предназначался для описания структуры контента страниц, а не для форматирования этого контента.

<http://info.cern.ch/hypertext/WWW/TheProject.html> – это первая в истории интернета страничка. И если бы не было css, именно так выглядели бы все страницы и сегодня. К счастью, это не так. :)

Что же можно отнести к оформлению, благодаря которому мы имеем современные веб-страницы? Самые очевидные параметры – это начертание шрифта, его размер и цвет. Давайте представим, как мог бы выглядеть наш html, если бы мы взялись программировать все внешние характеристики отображения шрифта с помощью тегов: например, чтобы задать начертание (семейство) шрифта, можно было бы придумать тег <sans-serif>, который нужно было бы размещать вокруг фрагмента, которому мы бы хотели задать эти свойства. Для задания красного цвета, к примеру, придумали бы тег <red>, для наклонного начертания – <italic>, и так нужно было бы по одному тегу на каждое возможное значение! Очевидно, что в этом случае наш html превратился бы в «монстра» с огромным количеством разных тегов, что значительно бы затруднило работу с ним. Только представьте: если в начале работы мы неверно выбрали цвет или начертание шрифта, пришлось бы вручную обойти весь проект и заменить одни теги на другие. Хорошо, что в настоящее время html для оформления не используется. Хотя предпосылки к такому варианту развития были, однако разработчики вовремя остановились. Сейчас у нас есть CSS, который отвечает за все оформление, которое появляется на наших страницах.

**CSS** – *Cascading Style Sheets* – каскадные таблицы стилей – это язык, содержащий набор информации, отвечающий за то, как именно будет отображаться страница сайта визуально. Используя css, мы можем запрограммировать внешний вид страницы: оформление и расположение блоков и элементов. СSS поддерживается абсолютно всеми браузерами, и благодаря свой простоте, де-факто HTML+CSS стали стандартом для построения веб-интерфейсов.

Где написать css?

Для подключения css используется два специальных тега meta и style. Но так как мы будем работать с codepen, то эти теги использовать не будем. Стили пишем в окошко «css».

CSS правило:

123**strong** {

color: red;

}

В терминах css это – правило. Оно всегда состоит из:

* селектора (strong) – указание, к каким тегам применяются указанные в правиле стили;
* пары «свойство» – «значение», разделенных двоеточием и обязательно заканчивающихся точкой с запятой. Объявление правила заключается в фигурные скобки, которые показывают, где начинается и где заканчивается объявление правила.

Что же может быть селектором?

В нашем примере это тег strong, и наше правило применяется ко всем тегам strong на странице. Чтобы обратиться к какому-то тегу, нужно просто указать его название в качестве селектора:

123456**div** {

color: red;

}

**span** {

color: red;

}

Какие свойства можно записать внутри объявления правила?

Допустим, мы хотим увеличить шрифт для заголовка h2. Для этого нам надо написать правило и в нем селектор по имени тега: h2 { }

Для того чтобы увеличить размер шрифта, используется свойство font-size. Самый простой способ – задать размер шрифта в % от того размера, который заголовок имеет сейчас, например:

123**h2** {

font-size: 120%;

}

Также мы можем указать размер шрифта другим способом – задать его в пикселях (px). Один пиксель равен одной точке на экране.

123**h2** {

font-size: 40px;

}

font-weight – задает жирность шрифта (bold, 200) значение по умолчанию – 400 или normal.

123**h2** {

font-weight: 700;

}

font-style — задает начертание шрифта — наклонный italic

123**h2** {

font-style: italic;

}

color — задает цвет шрифта (red, black, yellow)

123**h2** {

color: red;

}

Обратим внимание, что все эти свойства наследуются: если внутри нашего заголовка будет другой тег (например, ссылка a), то заданные нами свойства автоматически будут работать и для внутреннего тега тоже. Почему так происходит? Это определяется стандартами, и на самом деле это очень удобно – если бы не было этого механизма, нам бы пришлось следить за всеми тегами, которые вложены, и повторять все стили.

**Комбинации селекторов (каскад/селекторы, основанные на зависимостях)**

В качестве селектора не обязательно может выступать одиночное название тега — чаще всего это комбинации тегов (каскад):

* div p — выберет все теги p, которые расположены внутри div (на любом уровне вложенности — “потомки всех уровней” — дети, внуки, правнуки, и т.д.)
* div>p — выберет все теги p, которые расположены в первом уровне вложенности внутри div (дочерние селекторы, «потомки первого уровня» – только дети).

Также каскад часто используется для стилизации многоуровневых списков: но нужно быть осторожными – селектор ul li сработает на всех элементах li, даже на элементах вложеных списков, в то время как ul>li выберет также все элементы li, т.к. li всегда вложен в какой-то ul. Немного более сложные комбинации: ul ul li – выберет все элементы списков li, кроме самого верхнего уровня div p>span – выберет все span-ы, являющиеся детьми p, у которых, в свою очередь, есть div-родитель.