**Модуль 5**. **Каскадные таблицы стилей. Введение**

**Применение таблиц стилей: способы внедрения стилей в документ, каскадность**

(Продолжительность модуля 2 пары)

Каскадные таблицы стилей/CSS это язык стилей, определяющий отображение HTML-документов. Cascading Style Sheets (CSS) это поразительное изобретение для улучшения вида ваших web-сайтов. Оно поможет сэкономить уйму времени и предоставит вам совершенно новые возможности в дизайне web-сайтов. CSS совершенно необходим каждому, работающему с web-дизайном.

CSS это язык стилей, определяющий отображение HTML-документов. Например, CSS работает с шрифтами, цветом, полями, строками, высотой, шириной, фоновыми изображениями, позиционированием элементов и многими другими вещами.

**Базовый синтаксис CSS.**

selector {property: value;}

Value – значение свойства

Property – свойство стиля

Селектор – к какому тегу (тегам) применяется свойство

Пример: body {background: #FF0000;}

Из примера мы видим, что выбирается тег **body** и ему присваивается свойство **background** cо значением цвета **#FF0000**, в итоге мы получаем фон веб-страницы.

Но где размещать CSS-код?

Есть три способа применить правила CSS к HTML-документу. Ниже мы рассмотри эти три метода. Рекомендуется сосредоточиться на третьем - то есть внешней/external таблице стилей.

**Метод 1**: Инлайн/In-line (атрибут style)

Можно применять CSS к HTML с помощью HTML-атрибута style

Пример в файле (inline.html)

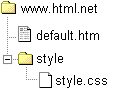
**Метод 2**: Внутренний (тэг style)

Второй способ вставки CSS-кодов - HTML-тэг **<style>**

Пример в файле (style.html)

**Метод 3**: Внешний (ссылка на таблицу стилей)

Рекомендуемый метод - создание ссылки на так называемую внешнюю таблицу стилей. Внешняя таблица стилей - это просто текстовый файл с расширением .css. Вы можете поместить таблицу стилей на ваш web-сервер или на жёсткий диск, как и другие файлы. Например, скажем, ваша таблица стилей называется style.css и находится в папке style. Это можно проиллюстрировать так:

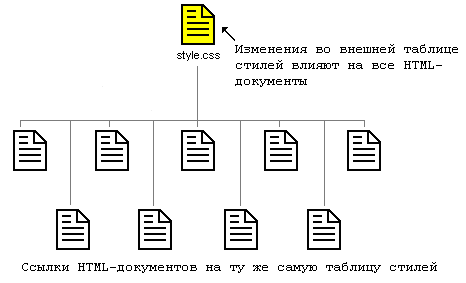


Весь фокус состоит в том, чтобы создать ссылку из HTML-документа (default.htm) на таблицу стилей (style.css). Это можно сделать одной строчкой HTML-кода:

**<link rel="stylesheet" type="text/css" href="style/style.css" />**

Эту строку кода нужно вставлять в разделе header HTML, то есть между тэгами **<head>** и **</head>.**

Эта ссылка указывает браузеру, что он должен использовать правила отображения HTML-файла из CSS-файла. Самое важное здесь то, что несколько HTML-документов могут ссылаться на одну таблицу стилей. Иначе говоря, один CSS-файл можно использовать для управления отображением множества HTML-документов.



Пример в файле (link.html)

Это поможет вам сэкономить уйму времени и сил. Если вы, например, хотите изменить цвет фона web-сайта из 100 страниц, таблица стилей избавит вас от необходимости вручную изменять все сто HTML-документов. Используя CSS, эти изменения можно сделать за несколько секунд, просто изменив один код в центральной таблице стилей.

**Селекторы, типы селекторов: классы, селекторы-потомки, идентификаторы, псевдоклассы**

Cелектор - это формальное описание того элемента или группы элементов, к которым применяется указанное правило стиля.

**1. \***

\* { margin: 0; padding: 0; }

Начнем с самого простого, а потом уже перейдем к более продвинутым вещам.

Этот CSS-селектор выделяет каждый элемент на странице. Многие разработчики используют его для того, чтобы скинуть у всех элементов значения margin и padding. На первый взгляд это удобно, но все-таки в рабочем коде так лучше не делать. Этот CSS-селектор слишком сильно грузит броузер.

\* также можно использовать для выделения дочерних элементов.

#container \* { border: 1px solid black; }

В данном случае выделяться все дочерние элементы #container. Опять же, старайтесь не злоупотреблять им.

**2. #X**

#container { width: 960px; margin: auto; }

Знак решетки перед CSS-селектором Х выделит нам элемент с id = Х. Это очень просто, но будьте аккуратны при использовании id.

id жестко привязывают стиль к элементу и не дает возможности повторного использования. Более предпочтительным будет использование классов, названий тэгов или даже псевдо-классов.

**3 .Х**

.error { color: red; }

Это CSS-селектор класса Х. Разница между id и классом заключается в том, что одному классу может принадлежать несколько элементов на странице. Используйте классы, когда вы хотите применить стиль к нескольким однотипным элементам. При использовании id вам придется писать стиль для каждого отдельного элемента.

**4. Х Y**

li a { text-decoration: none; }

CSS-селектор дочерних элементов встречается чаще всего. Если вам надо выделить элементы определенного типа из множества дочерних элементов, используете этот селектор. Например, вам надо выделить все ссылки, которые находятся в элементе li. В этом случае используйте этот селектор.

**5. X**

a { color: red; }

ul { margin-left: 0; }

Что делать, если вы хотите, чтобы охватить все элементы данного типа на странице? Будьте проще, используйте CSS-селектор типа. Если вы должны выделить все неупорядоченные списки, используйте ul {}.

**6. Х+Y**

ul + p { color: red; }

Выделяет последующий элемент. Он будет выбирать только элемент типа Y, который идет сразу после элемента Х. В примере текст первого абзаца после каждого ul будет красного цвета.

**7. Х>Y**

div#container > ul { border: 1px solid black; }

Разница между стандартными Х Y и X > Y состоит в том, что рассматриваемый CSS-селектор будет выбирать только непосредственные дочерние элементы. Например, рассмотрим следующий код.

<div id="container">

    <ul>

        <li> Элемент списка

            <ul>

                <li> Дочерний элемент</li>

            </ul>

        </li>

        <li> Элемент списка </li>

        <li> Элемент списка </li>

        <li> Элемент списка </li>

    </ul>

</div>

CSS-селектор #container > ul выберет только ul-ы, которые являются непосредственными дочерними элементами div с id =container . Он не выберет, например, ul-ы, являющиеся дочерними элементами первых li .

Поэтому можно получить выигрыш в производительности используя данный CSS-селектор. В самом деле, это особенно рекомендуется при работе с jQuery или другими библиотеками, выбирающими элементы на основе правил CSS -селекторов.

**8. Х ~ Y**

ul ~ p { color: red; }

Этот CSS-селектор очень похож на X + Y, однако, является менее строгим. При использовании ul + p будет выбрать только первый элемент, идущий за Х. В данном случае будут выбраны все элементы p, идущие на ul.

**Псевдоклассы и псевдоэлементы**

Псевдоклассы и псевдоэлементы позволяют задать особые стили. От названия элемента отделяются двоеточием. Рассмотрим псевдоклассы ссылок:

**:hover** – срабатывает при наведение курсора на элемент

**:active –** срабатывает при клики левой кнопкой мыши на элемент

**:visited** – срабатывает при посещенной пользователем страницей

**Свойства и их значения в CSS**

Выбирать элементы мы с Вами научились, давайте же теперь рассмотрим свойства элементов (их довольно много). Начнем конечно же со свойств текста.

Выравнивание текста.

CSS данную задачу берет на себя свойство **text-align**, которое выравнивает текстовое содержание относительно элемента родителя (например, блока div) или же окна браузера.

**text-align** (так же как и htmlловский атрибут align) имеет следующие значения:

* **left** - Выровнять текст по левому краю элемента (по умолчанию).
* **right** - Выровнять текст по правому краю.
* **center** - Выровнять текст по центру.
* **justify** - Выровнять текст по обоим краям.

Вертикальное выравнивание

Вертикальное выравнивание текста в строке устанавливает свойство **vertical-align**

Возможные значения свойства vertical-align:

**baseline** - Выравнивает базовую линию элемента по базовой линии родителя.

**bottom** - Выравнивает элемент по нижней части строки.

**middle** - Выравнивает средину элемента по базовой линии родителя и прибавляет половину высоты родительского элемента.

**sub** - Нижний индекс (размер шрифта не меняется).

**super** - Верхний индекс (размер шрифта не меняется).

**text-bottom** - Нижняя граница элемента выравнивается по нижнему краю строки.

**text-top** - Верхняя граница элемента выравнивается по верхнему краю строки.

**top** - Выравнивает элемент по верхней части строки.

Оформление текста.

Свойство **text-decoration** позволяет декорировать текст, присвоив ему одно или несколько значений из ниже представленных вариантов оформления текста.

Возможные значения:

**blink** - Текст будет мигать.

**line-through** - Делает текст перечеркнутым.

**overline** - Надчёркивание текста.

**underline** - Подчеркивание текста.

**none** - Текст без оформления.

Отступ первой строки.

Свойство **text-indent** - задаёт отступ первой строки в текстовом блоке с левой стороны, проще говоря делает "красную строку".

Расстояние от левого края окна браузера или же элемента родителя (блока в который помещен блок с текстом) может быть заданно в процентах от ширины окна браузера или же единицах измерения принятых в CSS.

Трансформация текста

Своийство **text-transform** трансформирует символы в указанном текстовом блоке, делая их заглавными или прописными по одному из правил в зависимости от присужденного значения данному свойству.

Значения:

**none** - Текст отображается без каких-либо изменений.(по умолчанию)

**capitalize** - Каждое слово в тексте отображается с заглавного символа.

**lowercase** - Все символы преобразуются в нижний регистр.

**uppercase** - Все символы преобразуются в верхний регистр.

Пробелы и перенос строки.

Набранный текст, в каком либо текстовом редакторе браузерами по умолчанию выводится на экран в виде сплошного текста, где переносы строк расставляются автоматически, а так же убираются лишние (более одного) пробелы между символами.

Свойство **white-space** имитирует работу тега <pre>, определяя показывать или нет пробелы между символов, если таковых больше чем один, а так же разрешает или запрещает перенос строки.

Может иметь следующие значения:

**normal** - текст выводится как обычно (лишние пробелы убираются), переносы строк определяются автоматически. (по умолчанию)

**nowrap** - запрещает автоматический перенос строки.

**pre** - показывает текст в том виде в котором он был набран. пробелы и переносы строки не удаляются.

Расстояние между словами.

Свойство **word-spacing** задаёт расстояние между словами (группами символов не разделенными пробелом) в строке.

Значения:

**normal** - Нормальное расстояние. (по умолчанию)

**px** - Расстояние задаётся в пикселях или любых других единицах измерения принятых в CSS.

Межсимвольное расстояние.

А вот свойство **letter-spacing** определяет расстояние между символами в тексте и так же, как и может **word-spacing** быть задано следующими значениями:

**normal** - Нормальное расстояние. (по умолчанию)

**px** - Расстояние задаётся в пикселях или любых других единицах измерения, принятых в CSS.

Расстояние между строками текста.

Расстояние между строками текста можно задать используя свойство **line-height**, сделать это можно следующими способами:

**normal** - Норма (по умолчанию).

**%** - Проценты. за сто процентов берется высота шрифта

**0.5** - Множитель. Может быть использовано любое число больше ноля. Так, например множитель 0.5 будет равняться половинному межстрочному расстоянию, а 2 - двойному.

**px** - Пиксели и любые другие единицы измерения, принятые в CSS.

ВСЕ ПРИМЕРЫ В ФАЙЛЕ **SIMPLES.HTML**

**Свойства шрифта.**

Свойство **font-style**, в зависимости от выбранного значения, определяет стиль шрифта.

Шрифт может иметь следующие стили:

**normal** - обычный (по умолчанию)

**italic** - курсив

**oblique** – наклонный

Начертание шрифта

Весьма интересное свойство шрифта **font-variant** позволяет делать строчные буквы заглавными и уменьшенными.

Значения:

**normal** - нормальный (по умолчанию)

**small-caps** - все буквы заглавные и уменьшенные

Размер шрифта

Свойство CSS **font-size** - определяет размер шрифта.

Размер шрифта может быть задан в процентах или пикселях и любых других допустимых единицах измерения CSS, а также абсолютным или относительным значением.

значения абсолютного размера шрифта:

**xx-small** - очень очень маленький

**x-small** - очень маленький

**small** - маленький

**medium** - средний

**large** - большой

**x-large** - очень большой

**xx-large** - очень очень большой

значения относительного размера шрифта:

**larger** - больше чем размер шрифта родительского элемента

**smaller** - меньше чем размер шрифта родительского элемента

Жирность шрифта

Свойство **font-weight** - определяет жирность шрифта. Насыщенность шрифта может быть задана относительно шрифта элемента родителя с помощью следующих значений:

**normal** - обычный шрифт

**bold** - полужирный шрифт

**bolder** - жирный шрифт

**lighter** - тонкий шрифт

А также выражается в условном числовом значении от 100 до 900 с шагом 100 (100, 200, 300... 900) где значение 100 тонкий шрифт, а 900 - сверх жирный.

Семейство шрифта

Атрибут CSS **font-family** - указывает одно, два или три имени шрифта из библиотеки шрифтов.

Возможность указывать до трёх имен шрифтов через запятую используется разработчиками во избежание возможных проблем связанных с отсутствием, по тем или иным причинам, указанных имен в библиотеке шрифтов на компьютере пользователя.

Так например запись в стилевом описании P {**font-family: Times New Roman, Arial, Verdana;**} - будет указывать браузеру пользователя, что данный параграф следует писать с помощью шрифта Times New Roman, а если его по каким то мифическим причинам не окажется в его библиотеке шрифтов то следует использовать шрифт Arial, и уж если и его нет, тогда писать шрифтом Verdana.

Если же браузер не найдёт в библиотеке шрифтов пользователя ни одного шрифта из указанных то он будет использовать тот шрифт который указан в его настройках "по умолчанию"

Однако также можно указать браузеру не только какой то конкретный шрифт, но и обозначить предпочтительное семейство шрифтов из перечисленных ниже возможных

**serif** - шрифты с засечками

**sans-serif** - рубленые шрифты

**cursive** - курсивные шрифты

**fantasy** - декоративные шрифты

**monospace** - моношириные шрифты

Например, если в файле CSS написать P {**font-family: Times New Roman, sans-serif;**} то это будет значить что если вдруг не окажится шрифта с именем Times New Roman, то следует использовать любой (или определённый в настройках браузера) доступный шрифт из семейства sans-serif - рубленых шрифтов.