**Что такое CSS?**

Добро пожаловать в CSS!

CSS означает Каскадные Таблицы Стилей (Cascading Style Sheets).

- Каскадные относится к способу применения CSS одного стиля к вершине другого.

- Таблицы Стилей управляют отображением веб-документов.

CSS и HTML работают рука об руку:

- HTML сортирует структуру страницы.

- CSS определяет, как будут отображены элементы HTML.

**Для чего используется CSS?**

CSS позволяет вам применять специфические стили к специфическим HTML элементам.

Основным преимуществом CSS является то, что он позволяет разделять стиль от содержимого.

Используя только HTML, все стили и форматирование будут в одном месте, что приведет к намного более сложному управлению страницей при ее росте.

**Встроенный/внутренний (Inline) CSS**

Использование встроенного стиля является одним из способов для вставки таблицы стилей. С помощью встроенного стиля, применяется уникальный стиль к одному элементу.

Чтобы использовать встроенный стиль, добавьте атрибут style к соответствующему тегу.

В нижеуказанном примере отображено, как создавать параграф с серым фоном и белым текстом:

<p style="color:white; background-color:gray;">

This is an example of inline styling.

</p>

Атрибут style может содержать любое свойство CSS.

**Вложенный(Embedded) CSS**

Вложенные стили определяются с помощью элемента <style> , внутри секции head HTML страницы.

Например, следующий код определяет стиль всех параграфов:

<html>

<head>

<style>

p { color:white; background-color:gray; }

</style>

</head>

<body>

<p>This is my first paragraph. </p>

<p>This is my second paragraph. </p>

</body>

</html>

Все параграфы имеют белый цвет текста и серый фон:

Внутренняя таблица стилей может использоваться, если один документ имеет уникальное оформление.

**Внешний(External) CSS**

В этом способе, все правила стилей содержатся в одном текстовом файле, который сохраняется с расширением .css.

Затем на этот файл ссылается HTML с помощью тега <link>. Элемент <link> располагается внутри секции заголовка (head).

Например:

<head>

<link rel="stylesheet" href="example.css">

</head>

<body>

<p>This is my first paragraph.</p>

<p>This is my second paragraph. </p>

<p>This is my third paragraph. </p>

</body>

p { color:white;

background-color:gray;} – в css файле пишем, ново созданном

И относительный и абсолютный пути могут быть использованы для определения href для CSS файла. В нашем примере путь относительный, так как CSS файл находится в одной директории с HTML файлом.

Какой из трех типов стилизации является наиболее полезным в рамках оптимизации веб сайта?

* Internal (внутренний) – непосредственно в артибуте
* Inline (встоенный) – в шапке стиль прописывается, в документе html
* External (внешний) – почему???? – это файл css отдельный

**Синтаксис CSS**

CSS состоит из правил стиля, которые браузер интерпретирует и затем применяет к соответствующим элементам в вашем документе.

Каждое правило имеет три части**: селектор, свойство, и значение.**

h1 { color: orange; }

где:

h1 – это селектор (Селектор указывает на элемент HTML, стиль которого вы хотите изменить.)

color: - это свойство

orange; - значение

**Блок объявлений** содержит одно или несколько объявлений, разделенных точкой с запятой.

Каждое объявление включает в себя имя свойства и значение, разделенные точкой с запятой.

**Селекторы по Типу**

Наиболее общие и легкие для понимания селекторы, это селекторы по типу. Эти селекторы выделяют типы элемента на странице.

Например, для выделения всех параграфов на странице:

p { color: red; font-size:130%;}

Объявления в CSS всегда заканчиваются точкой с запятой, а группы объявлений объединяются в фигурные скобки.

**Селекторы идентификаторов и классов**

Селекторы идентификаторов позволяют вам изменять стиль элемента HTML, который имеет атрибут **id,** несмотря на его позицию в дереве документа. Пример селектора id:

**HTML**

<div id="intro">

<p> Этот абзац находится во вводном разделе css.</p>

</div>

<p> Этого абзаца нет во вводном разделе css </p>

**CSS**

#intro { color: red; background-color: gray;}

Чтобы выбрать элемент со специфическим id, используйте символ решётки(#), а затем добавьте id элемента.

**Селектор потомков**

Эти селекторы используются для выбора элементов, которые являются потомками других элементов. При выборе уровней, вы можете выбрать столько уровней в глубину, сколько вам нужно.

Например, чтобы выбрать только элементы <em> в первом параграфе секции "intro":

**HTML:**

<div id="intro">

<p class="first">This is a <em> paragraph.</em></p>

<p> This is the second paragraph. </p>

</div>

<p class="first"> This is not in the intro section.</p>

<p> The second paragraph is not in the intro section. </p>

**CSS:**

#intro .first em {color: pink; background-color:gray;}

*(Урок 4) 16.10.2021*

**Комментарии**

Комментарии необходимы для того, чтобы объяснить ваш код, и они также могут помочь, когда вы редактируете исходный код спустя время. Комментарии игнорируются браузерами.

CSS комментарий выглядит следующим образом: **/\* Comment goes here \*/**

Комментарий может занимать несколько строк.

**Каскадирование**

Итоговый вид веб страницы является результатом различных правил.

Тремя главными источниками информации о стиле, которые формируют каскад, являются:

- Таблица стилей, созданная **автором страницы**

- **Стандартные стили браузера**

- Стили, определенные **пользователем**

**Наследование**

Наследование относится к способу, которым свойства протекают на странице. Обычно потомок принимает все характеристики родительского элемента, если не определено другое.

<head> <style

body { color: green; - цвет всей страницы зеленый

font-family: Arial;} - семейство шрифтов

</style> </head>

*(Урок 7) 17.10.2021*

**Свойство font-family**

Свойство font-family определяет шрифт элемента.

Существует два типа имен семейств шрифтов:

**- font family:** специфическое семейство шрифтов (как Times New Roman или Arial)

**- generic family:** группа подобных семейств шрифтов (как Serif или Monospace)

Разделяйте каждое значение запятой чтобы указать, что они являются альтернативными.

Если имя семейства шрифта содержит более одного слова, то оно должно быть указано в кавычках: "Times New Roman".

Свойство font-family должно содержать несколько имен шрифтов, в качестве "запасных". При определении веб шрифта в стиле CSS, добавьте более одного имени шрифта, чтобы избежать неопределенного поведения. Если компьютер клиента по каким-либо причинам не имеет шрифт, который вы выбрали, то он попробует другой.

Указание общего семейства шрифтов является хорошей практикой, чтобы позволить браузеру выбрать похожий шрифт в семействе, если другие недоступны.

body { font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;}

Если браузер не поддерживает шрифт Arial, то он пробует следующие шрифты (Helvetica Neue, затем Helvetica). Если браузер не поддерживает ни один из них, то он попытается использовать sans-serif.

Не забывайте использовать кавычки, если имя шрифта состоит более чем из одного слова.

**Свойство font-size**

Свойство font-size устанавливает размер шрифта. Одним из способов установки размера шрифта является использование ключевых слов. Например xx-small, small, medium, large, larger, и т.д.

Ключевые слова полезны, если вы не хотите, чтобы пользователь мог увеличить размер шрифта, потому что это может неблагоприятно сказаться на внешнем виде вашего сайта.

Вы также можете использовать числовые значения в пикселях или em для управления размером шрифта.

Установка размера шрифта в пикселях (px) является хорошим выбором, когда вам необходима точность в пикселях, это дает вам полный контроль над размером текста.

Использование em является другим способом установки размера шрифта (em является относительной единицей измерения). Оно позволяет большинству браузеров изменять размер текста. Если вы не установили самостоятельно размер шрифта на странице, то браузер будет использовать размер по умолчанию, который равен 16px.

Чтобы посчитать размер в em, используйте следующую формулу: em = pixels / 16

Например:

h1 { font-size: 20px;}

h1 { font-size: 1.25em; }

Оба примера выведут одинаковый результат в браузере, потому что 20/16=1.25em.

**Свойство font-style**

Свойство font-style обычно используется для применения курсивного начертания к тексту.

HTML:

<p class="italic">This is a paragraph in italic style.</p>

CSS:

p.italic { font-style: italic;}

Свойство font-style имеет три значения:

* normal
* italic
* oblique

Oblique(наклонный шрифт) очень похож на курсив, но является менее поддерживаемым.

HTML тег **<i>** выведет такой же результат, как и **курсивный стиль шрифта.**

**Свойство font-weight**

Свойство font-weight управляет толщиной текста. Могут быть установлены значения:

* normal (размер по умолчанию)
* bold
* bolder
* lighter

Если необходимо, можно задать толщину шрифта с помощью числа от

* 100 (тонкий) до
* 900 (жирный).
* 400 это то же самое, что normal,
* а 700 то же самое, что bold.

HTML тег **<strong>** также делает шрифт **жирным.**

**Свойство font-variant**

Свойство CSS font-variant позволяет вам конвертировать ваш шрифт в маленькие заглавные буквы. Значениями могут быть установлены:

* normal
* small-caps
* inherit

Не каждый шрифт поддерживает свойство CSS font-variant, так что не забудьте его протестировать.

**Свойство color**

Свойство CSS **color** определяет цвет текста.

Один метод спецификации цвета текста использует **название цвета**: например red(красный), green(зеленый), blue(голубой), и т.д.

Другим способом определения цвета является использование **шестнадцатеричных значений и RGB.**

Шестнадцатеричная форма состоит из знака решётки (#), после которой следуют максимум, **6 значений** (0-F).

RGB определяет индивидуальные значения для **Red**(Красного**), Green**(Зеленого), и **Blue**(Голубого).

В нижеуказанном примере мы используем шестнадцатеричное значение для смены цвета заголовка на голубой, и RGB для смены цвета параграфа на красный.

*(Урок 13) 21.10.2021*

**Свойство text-align**

Свойство text-align определяет горизонтальное расположение текста в элементе. По умолчанию, текст на вашем веб-сайте выравнен по левому краю. Однако, вам может понадобиться иное расположение.

Значениями свойства text-align являются:

* left
* right
* center
* justify

Если свойство text-align включено на **"justify",** то каждая строка растягивается таким образом, что будет иметь равную ширину, а левое и правое поля будут ровными (как в журналах и газетах).

**Свойство vertical-align**

Свойство vertical-align устанавливает вертикальное положение элемента. Используемыми значениями являются

* top
* middle
* bottom

Свойство vertical-align также принимает следующие значения:

* baseline
* sub
* super
* %
* px (или pt, cm).

Вместо значений px, вы можете использовать pt, cm (сантиметры) и % (проценты).

Свойство расположения по вертикали не действует одинаково на все элементы.

Например, дополнительная CSS стилизация необходима для элементов div.

display: inline-table; и display: table-cell; правила применяются для работы свойства vertical-align с divs.

**Свойство text-decoration**

Свойство text-decoration определяет, как будет декорирован текст.

Наиболее часто используемые значения:

**none** - Значение по умолчанию, определяет нормальный текст

**inherit** - Наследует это свойство от его родительского элемента

**overline** - Рисует горизонтальную линию над текстом

**underline**  - Рисует горизонтальную линию под текстом

**line-through** - Рисует горизонтальную линию сквозь текст (заменяет HTML тег <s> )

Вы можете комбинировать значения **underline, overline,** или **line-through** указывая их через пробел, чтобы добавить несколько линий.

Другим значением свойства text-decoration является **blink** которое заставляет текст мигать.

CSS синтаксис выглядит следующим образом:

text-decoration: blink;

Данное значение является доступным, но не рекомендуется к использованию, поэтому большинство браузеров его игнорируют.

Свойство text-indent

Свойство text-indent определяет, сколько места по горизонтали должно быть оставлено до начала первой строки текста. Значениями свойства являются

* length (px, pt, cm, em, и т.д.), %,
* inherit.

**HTML:**

<p> Это пример

<strong> text-indent </strong> свойство.

Первая строка нашего текста смещена вправо в 60 пикселей.

Помимо пикселей вы также можете использовать другие единицы измерения,

например pt, cm, em и т. д. </p>

**CSS:**

p { text-indent: 60 px;}

Также разрешено использование отрицательных значений. Первая строка будет выровнена по левому краю, если значение является отрицательным.

**Свойство text-shadow**

Свойство text-shadow добавляет к тексту тень.

Оно принимает четыре значения: первое значение определяет длину тени по оси **x (по горизонтали),** второе значение определяет длину по оси **y (по вертикали),** третье значение определяет **размытие тени**, а четвертое значение определяет **цвет.**

**HTML:**

<h1>Text-shadow example</h1>

**CSS:**

h1 {

color: blue;

font-size: 30pt;

text-shadow: 5px 2px 4px;

}

В вышеуказанном примере мы создали тень со следующими параметрами:

5px - X-координата

2px - Y-координата

4px - радиус размытия

grey - цвет тени

Чтобы добавить более одной тени тексту, добавьте список теней, разделенных запятыми.

**эффект размытия свойства text-shadow**

При работе с тенями вы можете использовать любые поддерживаемые CSS цветовые форматы.

Для значений x и y могут применяться различные единицы измерений (как **px, cm, mm, in, pc, pt**, и т.д.).

Отрицательные значения также доступны для использования.

В нижеуказанном примере создается синяя тень, которая на два пикселя выше основного текста, на один пиксель левее его и с радиусом размытия 0.5em:

**HTML:**

<h1>Text-shadow with blur effect</h1>

CSS:

h1 {

font-size: 20pt;

text-shadow: rgba(0, 0, 255, 1); -1px -2px 0.5em:;

}

Браузер Internet Explorer версии 9 и старше не поддерживает свойство text-shadow.

**Свойство text-transform**

CSS свойство text-transform определяет регистр текста. Например, с его помощью можно сделать каждое слово в тексте с заглавной буквы.