**C – Таблица (некий набор правил)**

**S – Каскадных (последовательность/наследование)**

**S – Стилей (правила по которым мы задаем вид нашей странице в документе html)**

Стили применяются по каскадному принципу.

**1й способ применения стилей:**

**STYLE=” ”** атрибут который можем применить к любым тегам, кроме парочки исключений

Когда мы применяем стиль к тегу <body> мы применяем стиль ко всей странице

Также стиль можно применить к тегу <html> - вся страница получит этот стиль (пример красный), но если указать стиль заголовку черный, то заголовок будет черным, это называется **наследование и приоритет!**

**COLOR** – свойство, отвечает за цвет текста, чаще пишут в HEX формате (#FF0025), можно англ. слово, но с этим бывают проблемы

Свойство может быть не одно. С помощью**;** можно перечислять еще кучу свойств

Как пример: <html **style="color:red;">**

**2й способ применения стилей:**

С помощью тега <style> <style/>, он является закрывающимся, потому что внутри него нужно писать таблицы правил. Он может находится и в <head> и в <body>…где угодно. От его расположения выше или ниже будет зависеть приоритет. Как примет с таблицей:

<head>

**<style> table {color: blue;} </style>**

</head>

**3й способ применения стилей:**

Сегодня стили не пишут в атрибутах или прямо в документе html, для этого создают отдельно файл и подключают к html документу. Это является неким правилом или стандартом. Очень редко бывают исключения, но лучше этого не делать.

**<link>** с помощью этого тега, подключают стиль для оформления документа. Его можно ассоциировать с ссылкой или связью, что бы браузер знал какой тип файла мы будем связывать мы должны его передать. Делается это с помощью атрибута rel и внутри него пишем, что это таблица стилей или лист со стилями stylesheet

<link **rel= “stylesheet**”

Теперь нужно указать ссылку на подключаемый внешний стиль (абсолютный и относительный путь)

<link rel="stylesheet" **href**="style.css"> (href это гиперссылка, гипер референтс)

Создаем файл style.css

1. Название на английском языке (до точки) или английскими буквами (STYLE, APP, HOME, ADMIN…)
2. .CSS является константой

<link rel="stylesheet" href**="style.css">** это путь, в данном случае относительный, не полный

Во всех браузерах, уже заложен свой стиль для html элементов на странице. Наша задача как веб-разработчика, установить свои правила.

**Правила по приоритету:**

**Структура**

**table** {color: green;}

**Table –** селектор. Это может быть: название тега, класс или сочетание селекторов

{ } – в фигурных скобках указывается свойства. Они записываются в виде имени свойства и его значения {color: green;}

**Селектор** - **это** часть CSS-правила, которая сообщает браузеру, к какому элементу (или элементам) веб-страницы будет применён стиль.

**селекторы**

**Table – селектор,** первым делом мы указываем для кого мы будем менять стиль и внешний вид. Он может быть простым как этот или более точным и развернутым

Можем например использовать как селектор любой атрибут, но он должен быть в квадратных скобках

[href="#ideveloper"]

Если есть несколько одинаковых атрибутов, но в разных тегах, то перед квадратными кавычками ставим название тега без пробелов и этот стиль работает только для этого тега

**b**[height="50px"] {color: orange;}

**class атрибут – также является селектором**

**class атрибут –** также является селектором, но это **специфический** атрибут, он по сути, создан для таблицы каскадных стилей (css).

[class="privet"] = **.privet** (css).

.privet {color: purple;} вместо [class="privet"] {color: purple;}

Этот атрибут особенный, он создан для работы со стилями.

Class**=”name-class”** когда внутри добавляем какое-нибудь слово, создается класс, некая совокупность.

**На примере 4х таблиц:**

1. **Мы хотим сделать 1 и 4 таблицы – красные, а 2 и 3 синие**

Создаем два класса: (в css)

* .Table-red {color:red;}
* .Table-blue { color:blue;}

Эти классы нужно указать в атрибуте таблицы (в html)

<table class=” Table-red” border=”1”> и так ко всем 4м таблицам, смотря какую в какой класс мы определили.

1. **Мы хотим у 2 и 4 таблицы сделать текст жирным**

Создаем еще один класс (css)

* Table-bolt {font-weight: bolt;}

Добавляем этот класс куда нам нужно, так как html элемент может иметь сколько угодно классов. **Перечисление классов** идет через **ПРОБЕЛ.**

Теперь где бы мы не использовали этот класс (Table-bolt), он будет делать текст жирным

**Название класса**

* Латинские буквы (abcdef)
* Цыфры (name1)
* Дефис, нижнее подчеркивание (имя-класса, имя\_класса)
* Рекомендация писать название из двух слов, через дефис
* В нижнем регистре (nameclass)

Не должно быть:

* Без Пробелов
* Не начинаться с цыфры
* (NameClass-не правильно)

Атрибут Class можно применять к любому html тегу на странице.

Силектор Class в таблице каскадных стилей всегда используется как:

.class-name (точка и его название)

Не нужно его использовать как обычный атрибут [class=”class-name”]

**Какой еще есть селектор? – Уникальный! Селектор по идентификатору.**

Он не должен повторятся:

<h1 id=”header”> заголовок </h1>

<p id=”~~header~~”> заголовок </p>

Header – это имя, уникальное, должно использоваться один раз

**Использование в качестве селектора в CSS**

Что бы присвоить тегу с таким уникальным атрибутом id особенный стиль, его так же не используют как запись вида. Его запись похожа на запись класса, только вместо точки пишется #

~~[id=”header”]~~ #header

**Какие есть еще особенности при написании селекторов?**

**html tr {color: yellow;}** тут мы задали всем строкам таблицы желтый цвет (обратившись к королям)

Вместо названия тега, можно использовать класс или идентификатор, атрибут и так далее:

**.class-name**

**#idname**

**[attribute=”vaiue”]**

Если существует список а мы хотим обратиться только к элементам, которые вложены только на первом уровне

* Первый элемент списка
* Второй этемент списка
* [YouTube](https://www.youtube.com/) ссылка на главную страницу
  + второй уровень
  + второй уровень

Первым элементам создаем класс: class=”list” (в html), затем пишем в (CSS) класс в качестве селектора, затем знак > и название тега li

.list > li {color: blue;}, если убрать знак >, то стиль будет применен ко всем уровням синий.

Еще селекторы можно комбинировать, записывая последовательно без пробела. К чему это можно применить? Допустим у нас есть две ссылки имеющие одинаковый размер шрифта, но нужны разные цвета для этих ссылок

1. нужен общий класс для обеих ссылок в CSS, в котором будет указан размер шрифта в пикселях.

.font-size-md {font-size: 18px;}

class="font-size-md – добавляем к обеим ссылкам в html

1. создаем еже два класса:

.font-size-md.color-red {color: red;} – цвет красный

.font-size-md.color-gray {color: gray;} – цвет серый

и также добавляем в документ html . Вот что выходит:

<a class="font-size-md color-red " href="#"> первая ссылка </a>

<br>

<a class="font-size-md color-gray " href="#"> вторая ссылка </a>

[первая ссылка](file:///C:\Users\KiNiSkA\Desktop\web\YouYube\indexCSS.html)  
[вторая ссылка](file:///C:\Users\KiNiSkA\Desktop\web\YouYube\indexCSS.html)

Если в третей ссылке, не будет хоть одного класса, то данное правило работать не будет.

Базовые селекторы: (в интернете полно инфо)

* селекторы по элементу
* селекторы по классу
* селекторы по идентификатору
* универсальный селектор
* селекторы по атрибуту

свойства

Если селекторов много, то свойств вообще орда! Их можно найти в справочнике CSS