**Словарь основных селекторов (параметры одного селектора пишутся через пробел)**

**width: px** или % - широта

**height** – высота

**color** –цвет (#00FFFF)

**font-size**: 10px (пиксели) или в % от страницы – размер шрифта

**text-align** –выравнивание текста.

left – полевому краю

right – по правому

center – по центру

justify – по ширине

**font-weight**: 1-900 –жирность шрифта

bold – жирный

normal – отменить жирность

**font-style**: italic – курсив (normal – отмена)

font-family – стиль шрифта (Times New Roman (стоит по умолчанию), Arial, Georgia, Verdana, Trebuchet MS, Impact, Arial Black, «Comic Sans MS», «Courier New».)

**line-height**:..px, – межстрочный интервал (Можно также просто написать число или дробь. В этом случае настоящее значение **line-height** можно будет найти, умножив его на значение **font-size**)

font: bold italic 16px/50px Arial (*курсив жирность размер\_шрифта / межстрочный\_интервал семейство)*

**text-indent**: ..px – красная строка

**text-decoration** – эффекты для текста

none – отмена эффектов

underline –подчеркивание

overline – линия на текстом

line-through – перечеркивание

плевдоклассы: :hover – действие при наведении

:link – не посещенная ссылка

:visited – отлавливает посещенную ссылку

:active – отлавливает состояние когда нажали на элемент мышкой но еще не отпустили

Из-за наследования для корректной работы данные псевдоклассы следует размещать именно в таком порядке, как в примере: *:link, :visited, :hover, :active* (ненужные можно не писать). Этот порядок подчиняется следующему мнемоническому правилу: **LoVe HAte**.

**background-color:** цвет фона

**border**: 1px (размер) solid (форма сплошная) dashed (черточная) цвет – задает границу

**border-width** задает толщину границы,

**border-color** - цвет,

**border-style** задает тип границы

solid – сплошная линия

dotted – линия в виде точек

dashed – линия в виде тире

ridge – выпуклая линия

double – двойная линия

**border-left** (левая граница), **border-right** (правая граница), **border-top** (верхняя граница), **border-bottom** (нижняя граница).

**border-radius:** 0-100px ,%– закругления границы (может принимать 4 значения для каждого угла)

**background-image**: url(image/12345.jpg) – фоновая картинка. По умолчанию картинка замостит своими копиями весь блок, однако, это поведение можно отменить с помощью свойства [background-repeat](http://code.mu/ru/markup/manual/css/property/background-repeat/).

Свойство background-repeat задает каким образом повторять фоновую картинку элемента. По умолчанию картинка повторяется и по оси X, и по оси Y, таким образом покрывая собой всю доступную область.

**background-repeat**: no-repeat | repeat | repeat-x | repeat-y | space | round;

no-repeat Картинка не будет повторяться вообще.

repeat-x Картинка будет повторяться по оси X.

repeat-y Картинка будет повторяться по оси Y.

repeat Картинка будет повторяться по оси X и по оси Y.

space Картинка повторится столько раз, чтобы полностью заполнить область, если это не удается, между картинками добавляется пустое пространство.

round Картинка повторится так, чтобы в области поместилось целое число рисунков, если это не удается сделать, то фоновые рисунки масштабируются.

**list-style-type** –маркеры для списков ul

none – убрать маркеры

square – маркеры в виде квадратиков

circle – кружки

disc – закрашенные кружки

**list-style-type** –маркеры для списков ol

lower-roman – римские цифры

upper-roman – заглавные римские цифры

lower-alpha – заглавные латинские буквы

upper-alpha – нумерация большими латинскими цифрами

lower-greek – маленькие греческие буквы

**margin**: внешние отступы (auto – центрирование. margin-left: auto; /\* задаем автоматический левый отступ \*)

**padding:** - внутренние отступы

**box-sizing:** **border-box** – убирает расширение блока

селектор**:hover** – действия при наведении мыши

**opacity**: (0.1-1) – прозрачность

**box-shadow**: 1px 1px 1px red – тень

**display: block** –видимость элемента. Block –виден, none – не виден flex - блок

**text-align**: - выравнивание текста (center, left, right)

**text-decoration: none** – убрать украшение текста (подчеркивание для ссылок например),

underline – добавить подчеркивание текста

**transform** – изменение формы элемента (rotate (45deg) –поворот, scale(0.5) – изменение размера

**Flex элементы**

**justify-content** - выравнивает элементы горизонтально и принимает следующие значения:

flex-start: Элементы выравниваются по левой стороне

контейнера.flex-end: Элементы выравниваются по правой стороне контейнера.

center: Элементы выравниваются по центру контейнера.

space-between: Элементы отображаются с одинаковыми отступами между ними.

space-around: Элементы отображаются с одинаковыми отступами вокруг них.

……………………

**align-items** - выравнивает элементы вертикально и принимает следующие значения:

flex-start: Элементы выравниваются по верхнему краю контейнера.

flex-end: Элементы выравниваются по нижнему краю контейнера.

center: Элементы выравниваются вертикально по центру контейнера.

baseline: Элементы отображаются на базовой линии контейнера.

stretch: Элементы растягиваются, чтоб заполнить контейнер.

……………………

**flex-direction** - задает направление, в котором будут расположены элементы в контейнере и принимает следующие значения:

row: Элементы размещаются по направлению текста.

row-reverse: Элементы отображаются в обратном порядке к направлению текста.

column: Элементы распологаются сверху вниз.

column-reverse: Элементы распологаются снизу вверх.

(Заметь, что когда ты устанавливаешь направление в обратном порядке для ряда или колонки, начало (start) и конец (end) тоже меняются местами.)

Иногда изменения порядка отображения элементов в контейнере недостаточно. В таких случаях мы можем применить свойство **order** для конкретных элементов. По умолчанию, значение этого свойства у элементов равно 0, но мы можем задать положительное или отрицательное целое число этому свойству.

…………………

**flex-wrap** которое принимает следующие значения:

nowrap: Размеры элементов устанавливаются автоматически, чтоб они поместились в один ряд.

wrap: Элементы автоматически переносятся на новую строку.

wrap-reverse: Элементы автоматически переносятся на новую строку, но строки расположены в обратном порядке.

…..

Два свойства flex-direction и flex-wrap используются так часто вместе, что было создано свойство **flex-flow** для их комбинирования. Это свойство принимает значения двух этих свойств, разделеные пробелом.

Например, ты можешь использовать flex-flow: row wrap, чтоб элементы располагались в ряд и автоматически переносились на новую строку.

…….

**align-content** чтобы указать, как несколько рядов должны отделяться друг от друга. Данное свойство принимает следующие значения:

flex-start: Ряды группируются в верхней части контейнера.

flex-end: Ряды группируются в нижней части контейнера.

center: Ряды группируются вертикально по центру контейнера.

space-between: Ряды отображаются с одинаковыми расстояниями между ними.

space-around: Ряды отображаются с одинаковыми расстояниями вокруг них.

stretch: Ряды растягиваются, чтоб заполнить контейнер равномерно.