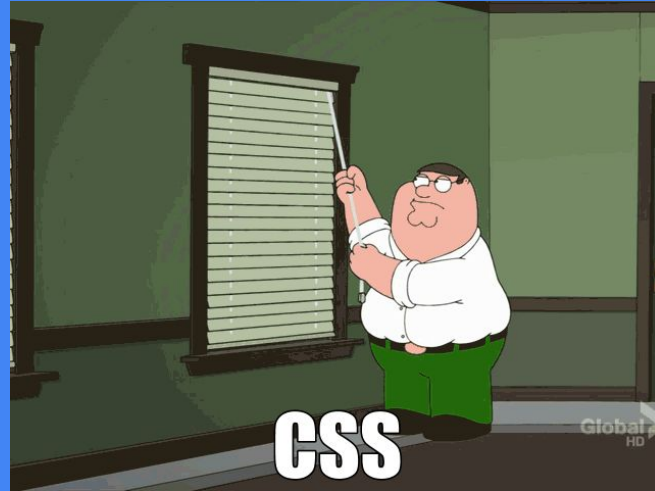


# CSS

Cascading Style Sheets



# Ретроспектива

домашнее задание

- 1) Переключиться на ветку main
- 2) Обновить main → `git pull`
- 3) Создать ветку → `git create branch`
- 4) Сделать изменения
- 5) Сохранить изменения → `git commit`
- 6) Отправить → `git push`
- 7) Создать Pull request
- 8) Ответить на комментарии

In case of fire



1. `git commit`



2. `git push`



3. leave building

# CSS документация

<https://www.w3.org/Style/CSS/specs.en.html>

<https://www.w3.org/TR/?tag=css>

<https://www.w3schools.com/cssref/default.asp>

<https://w3schoolsrus.github.io/cssref>

<https://caniuse.com/>

# CSS синтаксис (css правило)

```
селектор {  
    СВОЙСТВО: значение;  
}
```

# Селекторы

# CSS селекторы

универсальный

`* {}`

по тегу

`h1 {}`

по классу

`.heading {}`

по ид

`#heading {}`

по атрибуту

`a[href="#"]`

# CSS комбинированные селекторы

.class1.class2 {}      <div class="class1 class2"></div>

.class1 .class2 {}      <div class="class1">  
                                 <div class="class2">  
                                 </div>  
                                 </div>

p.class {}      <p class="class"></p>  
p .class {}      <p>  
                         <em class="class"></em>  
                         </p>

.class1,  
.class2 {}      Перечисление селекторов



# CSS комбинированные селекторы

h1	все теги h1
#main	все теги с ID = main
.class1	все теги с классом class1
.class1.class2	все теги с двумя классами class1 и class2
.class1 .class2	все теги с классом class2 с родителем class1
.class1, .class2	все теги с классом class1 или class2

Свойства

# Классификация свойств

состояние - Псевдоклассы (:)

декоративные элементы - Псевдоэлементы (::)

функции

анимации (animation, transition)

трансформации (transform)

фон (background)

работа с текстом (text, font)

размеры и отступы (width, height, padding, margin)

позиционирование (position)

списки (list)

гриды (grid)

флексбоксы (flex)

переполнение

обтекаемость (float)

рамки, обводки тени (border)

взаимодействия

svg

директивы (@)

колонки (column)

CSS специфичность

# CSS специфичность

!important

inline-style

id

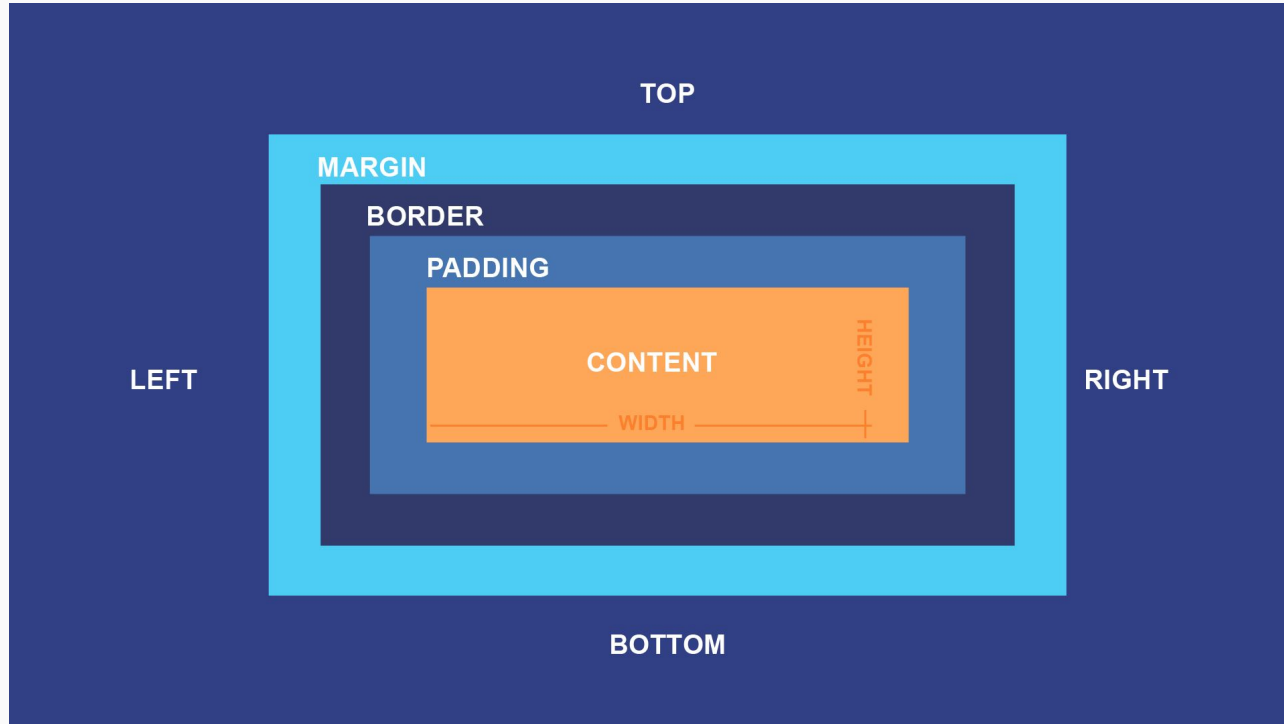
class

tag

0 . 0 . 0 . 0 . 0

# Блочная модель

# Box model



# Единицы измерения



# Абсолютные измерения

px	1px или 1/96 дюйма.
cm	
in	дюйм = 2.54cm
mm	
Q	четверть миллиметра.
pc	пики. 1pc = 1/16in.
pt	1pt = 1/72in.

## Относительные измерения

em	относительно размера шрифта родительского блока
rem	относительно размера html элемента
vw	относительно ширины вьюпорта
vh	относительно высоты вьюпорта
vmin	относительно меньшей стороны вьюпорта
vmax	относительно большей стороны вьюпорта
%	относительно размера родительского блока

БЭМ

<https://ru.bem.info/>

## БЭМ (Блок, Элемент, Модификатор)

`block__element_modifier`

<code>.block {}</code>	- блок
<code>.block__elem1 {}</code>	- элемент
<code>.block__elem2 {}</code>	- элемент
<code>.block__elem3 {}</code>	- элемент
<code>.block_modifier {}</code>	- модификатор блока
<code>.block__elem1_modifier {}</code>	- модификатор элемента

# Flexbox

# FlexBox свойства элементов

<u>order</u>	порядок
<u>flex-grow</u>	соотношение как блок будет расти относительно других
<u>flex-shrink</u>	соотношение как блок будет уменьшаться относительно других
<u>flex-basis</u>	начальная длина
<u>flex</u>	сокращение flex-grow, flex-shrink, flex-basis
<u>align-self</u>	выравнивание по элемента по горизонтали относительно других

GRID

# GRID основные понятия

<u>column</u>	вертикальные столбцы
<u>row</u>	горизонтальные ряды
<u>column gap</u>	зазоры между столбцами
<u>row gap</u>	зазоры между рядами
<u>line</u>	линии между рядами и столбцами



# GRID

grid-template-columns

количество столбцов

grid-template-row

количество рядов

justify-content

выравнивания внутри контейнера

align-content

вертикальное выравнивания внутри контейнера

# GRID элементы

grid-column

занимаемые элементом столбцы

grid-row

занимаемые элементом строки

grid-area

занимаемые элементом столбцы и строки

названия областей

# Адаптивный дизайн

# meta viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

device-width

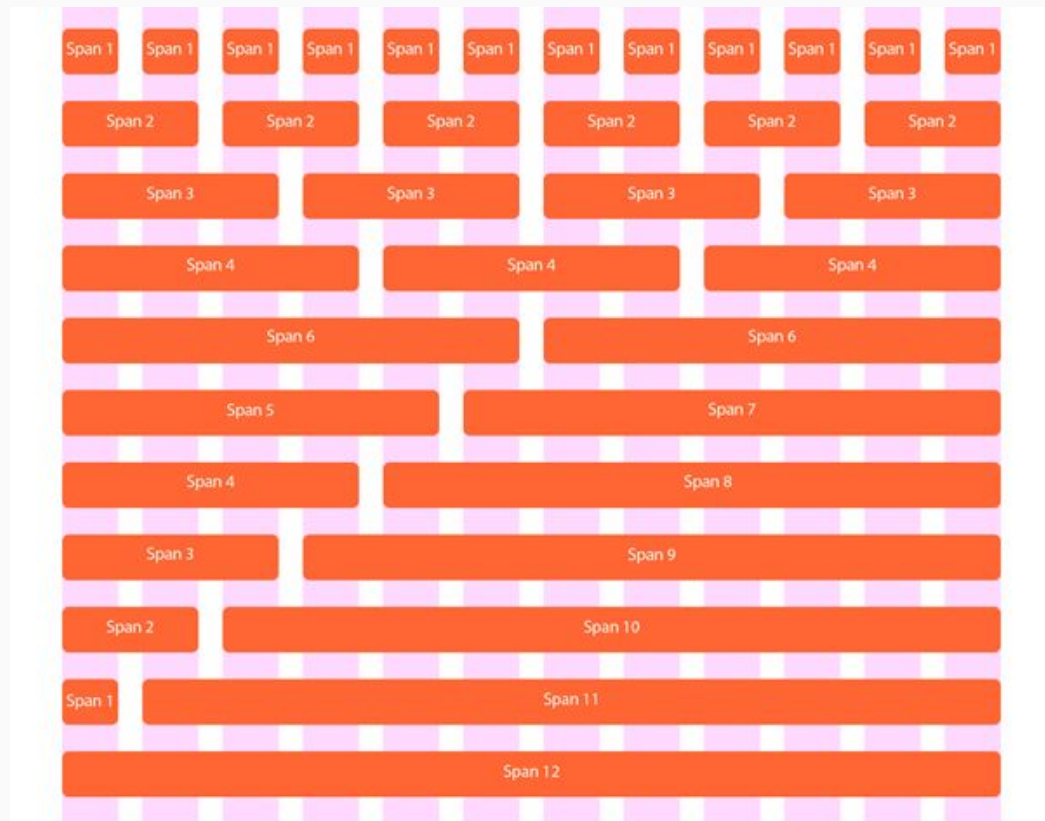
устанавливает размер в соответствии с размером экрана

initial-scale

начальный уровень масштабирования

# Сетки

```
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```



# Media query

```
@media only screen and (max-width: 600px) {  
  
}
```

```
@media only screen and (max-width: 600px) {...}
```

```
@media only screen and (min-width: 600px) {...}
```

```
@media only screen and (min-width: 768px) {...}
```

```
@media only screen and (min-width: 992px) {...}
```

```
@media only screen and (min-width: 1200px) {...}
```

# Адаптивные картинки

Retina displays

```
.selector { background: url(../path_to_png/apple.png) no-repeat; }
```

```
@media (-webkit-min-device-pixel-ratio: 2) {
```

```
  .selector { background-image: url(../path_to_png/apple@2x.png); background-size: cover; }
```

```
}
```

```

```

# Адаптивные картинки

```
<picture>
  <source media="(min-width:650px)" srcset="red.jpg">
  <source media="(min-width:465px)" srcset="black.jpg">
  
</picture>
```

```
@media only screen and (min-width: 400px) {
  body {
    background-image: url('img_small.jpg');
  }
}
```



# Анимации и трансформации

# Примеры

<https://codepen.io/FilipVitas/pen/GdMbOX>

<https://codepen.io/juliangarnier/pen/krNqZO>

<https://pattle.github.io/simpsons-in-css/>



<https://codepen.io/SampathParavasthu/pen/qBWPabx>

# Анимации

animation-name	имя анимации
animation-duration	время анимации (s, ms)
animation-iteration-count	количество повторений (infinite)
animation-direction	проигрывать в обратном направлении
animation-timing-function	развитие анимации - linear   ease   ease-in   ease-out   ease-in-out
cubic-bezier	
animation-delay	задержка анимации
animation-play-state	поставить анимацию на паузу
animation-fill-mode	применять стили ключевых кадров - none   forwards   backwards
animation	объединяет все свойства выше
@keyframes	ключевые кадры (from to   0% - 100%)

# animation-timing-function

linear	равномерно
ease	медленно / очень быстро / медленно
ease-in	медленно / быстро
ease-out	быстро / медленно
ease-in-out	медленно / быстро / медленно
cubic-bezier	<a href="https://cubic-bezier.com">https://cubic-bezier.com</a>

# Transition - переход между состояниями элемента

transition-property	указываем свойство, которое хотим плавно изменить
transition-duration	длительность перехода
transition-timing-function	развитие анимации
transition-delay	задержка перед началом изменения
transition	объединяет все свойства выше

# Transform - не затрагивает соседние элементы

[playground](#)

translate

rotate

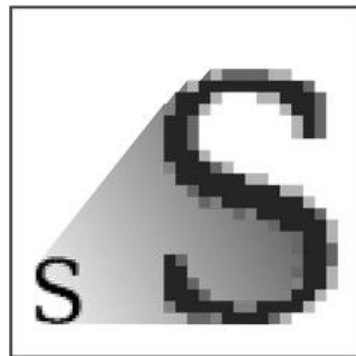
scale

skew

SVG

Scalable Vector Graphics (SVG)

работает при любых размерах



**ПАСТР**  
.jpeg .gif .png



**BEKTOP**  
.svg



# SVG

fill

заливка

stroke

обводка

# Bootstrap

# Bootstrap

<https://getbootstrap.com/>