

# Лекция 4

## CSS. Современные ВОЗМОЖНОСТИ

Дмитрий Зайцев



образование

# План на сегодня

- CSS, боль
- Кто это придумал?
- Возможности
- Препроцессоры, инструменты
- Разработка под мобильные устройства
- Единицы измерения в CSS
- Область просмотра (viewport)
- Нативное взаимодействие
- ДЗ

# Минутка бюрократии

- Внимание
- Отметки о посещении занятий
- Обратная связь о лекциях



# CSS, боль

CSS. 501b

CSS

IS

AWESOME

## CSS. Боль



Кто это  
придумал?

# CSS. W3C — [www.w3.org](http://www.w3.org)

**W3C (World Wide Web Consortium — Консорциум Всемирной паутины)** — организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины. Консорциум возглавляет **Тимоти Джон Бернерс-Ли**

Любой стандарт W3C проходит 5 стадий согласования:

- Черновик спецификации (*Draft*)
- Рабочий проект (*Working Draft*);
- Последний созыв (*Last Call*);
- Возможная рекомендация (*Candidate Recommendation*);
- Предлагаемая рекомендация (*Proposed Recommendation*);

и только после этого официально становится рекомендацией W3C.



# CSS. WHATWG — [www.whatwg.org](http://www.whatwg.org)

**WHATWG (Web Hypertext Application Technology Working Group)** — сообщество людей, заинтересованных в развитии Интернета. Было основано в 2004 году производителями браузеров: Apple, Mozilla Foundation и Opera Software. Основным направлением сообщества является развитие HTML и API, необходимого для веб-приложений.

По сути, является подобием **W3C**. WHATWG была недовольна медленными темпами развития стандартов и уклоном W3C в сторону HTML, основанного на XML-синтаксисе

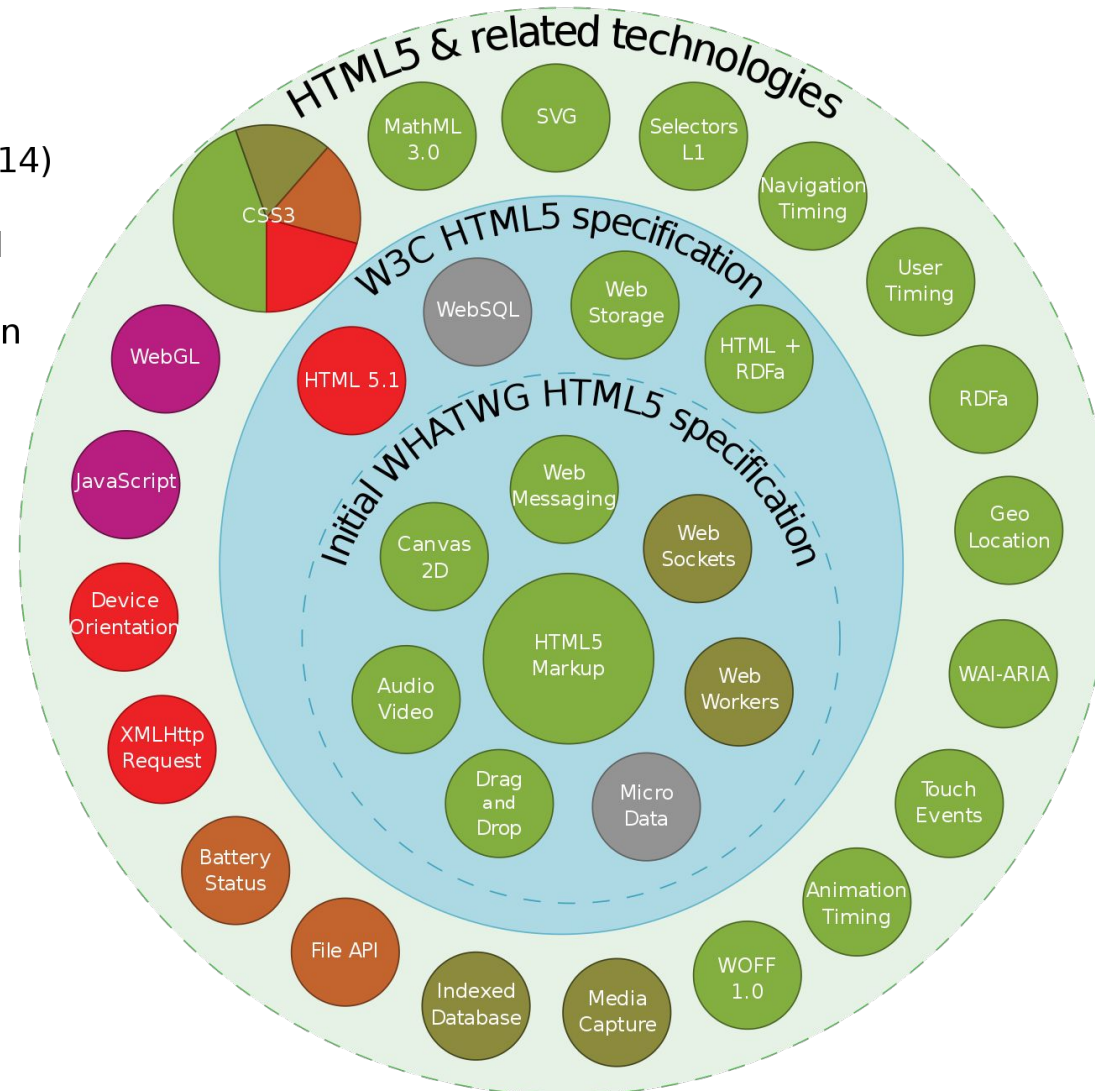
Сейчас WHATWG активно разрабатывает спецификации: HTML, DOM Standard, Fetch Standard, Web workers, Storage Standard, Streams Standard.

# CSS. HTML Status

## HTML5

Taxonomy & Status (October 2014)

- Recommendation/Proposed
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated or inactive



# CSS. Cascading Style Sheets

CSS одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название «стандарты Web»

[wiki.csswg.org](http://wiki.csswg.org) — рабочая группа CSS в рамках консорциума W3C

- CSS level 1 — декабрь 1996
- CSS level 2 — май 1998
- CSS level 2.1 — июнь 2011
- CSS level 3 — с июня 2012 ... (по сути, ещё не принята, постоянно в разработке)
- ~~CSS level 4~~ — никогда не появится, т.к. CSS level 3 разбил все на отдельные модули, которые постоянно развиваются

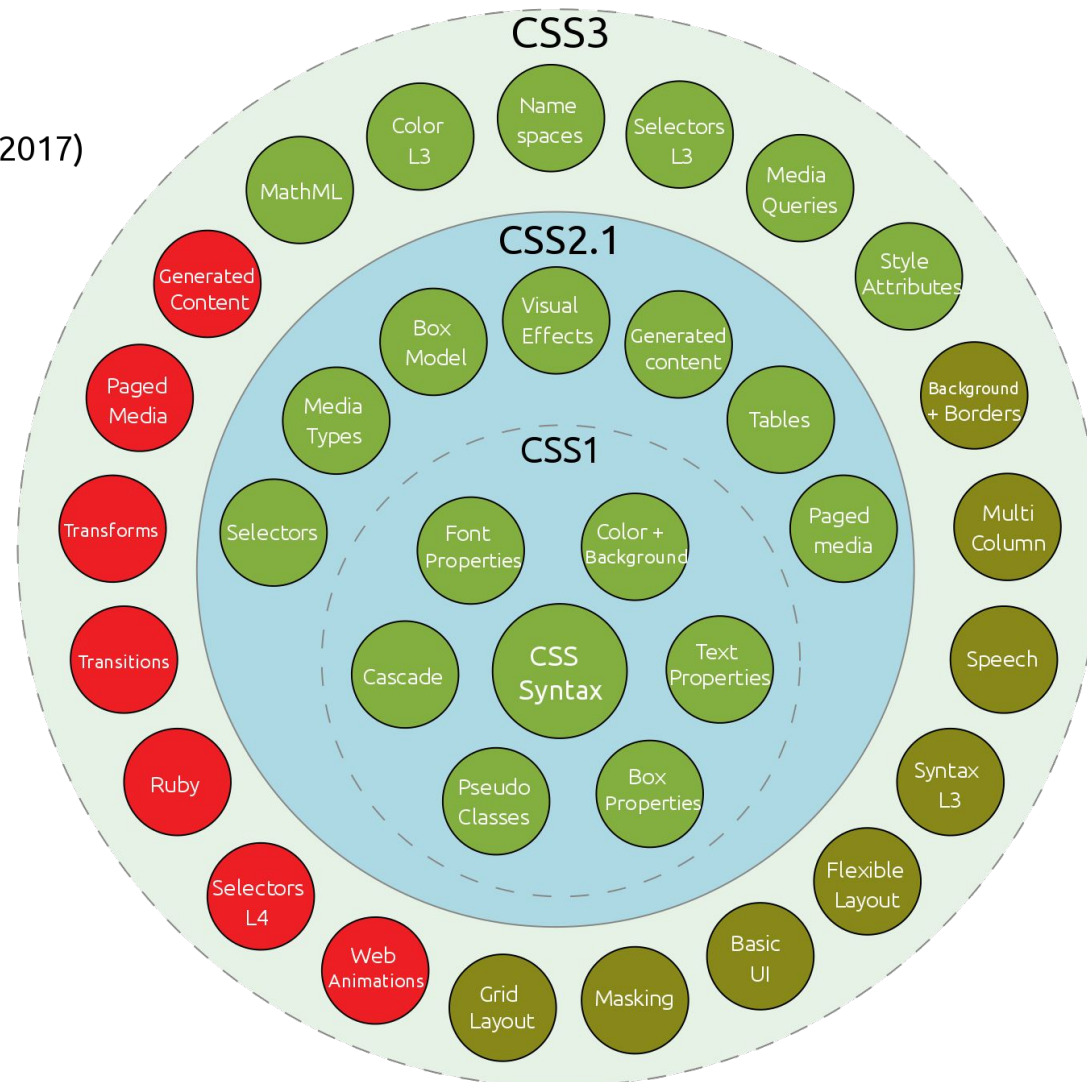
Все CSS-спецификации — [www.w3.org/Style/CSS/specs.en.html](http://www.w3.org/Style/CSS/specs.en.html)

# CSS. CSS Status

## CSS3

Taxonomy & Status (September 2017)

-  W3C Recommendation
-  Candidate Recommendation
-  Last Call
-  Working Draft
-  Obsolete or inactive



# Возможности

# CSS. Управление разметкой. display: block;

Блочные элементы – всегда начинаются с новой строки и занимают всю доступную ширину.

```
<address>  
<article>  
<aside>  
<blockquote>  
<canvas>  
<dd>  
<div>  
<dl>  
<dt>  
<fieldset>  
<figcaption>  
<figure>  
<footer>  
<form>
```

```
<h1>-<h6>  
<header>  
<hr>  
<li>  
<main>  
<nav>  
<noscript>  
<ol>  
<p>  
<pre>  
<section>  
<table>  
<tfoot>  
<ul>  
<video>
```

# CSS. Управление разметкой. display: inline;

Строчные элементы – всегда **НЕ** начинаются с новой строки и занимают столько ширины, сколько требуется.

```
<a>  
<abbr>  
<acronym>  
<b>  
<bdo>  
<big>  
<br>  
<button>  
<cite>  
<code>  
<dfn>  
<em>  
<i>  
<img>  
<input>  
<kbd>
```

```
<label>  
<map>  
<object>  
<output>  
<q>  
<samp>  
<script>  
<select>  
<small>  
<span>  
<strong>  
<sub>  
<sup>  
<textarea>  
<time>  
<tt>  
<var>
```

# CSS. Управление разметкой. Позиционирование элементов

```
.element-left { float: left; }

.container-center-old {
  display: block;
  width: 150px;
  margin: 0 auto;
}

.container-absolute-center-old {
  height: 100px;
  width: 100px;
  position: absolute;
  top: 50%;
  left: 50%;
  margin-left: -50px;
  margin-top: -50px;
}
```

```
.container-flex {
  display: flex;
}

.container-grid {
  display: grid;
}

.container-center-new {
  display: flex;
  justify-content: center;
  align-items: center;
}
```

[A Complete Guide to Flexbox](#)

[A Complete Guide to Grid](#)

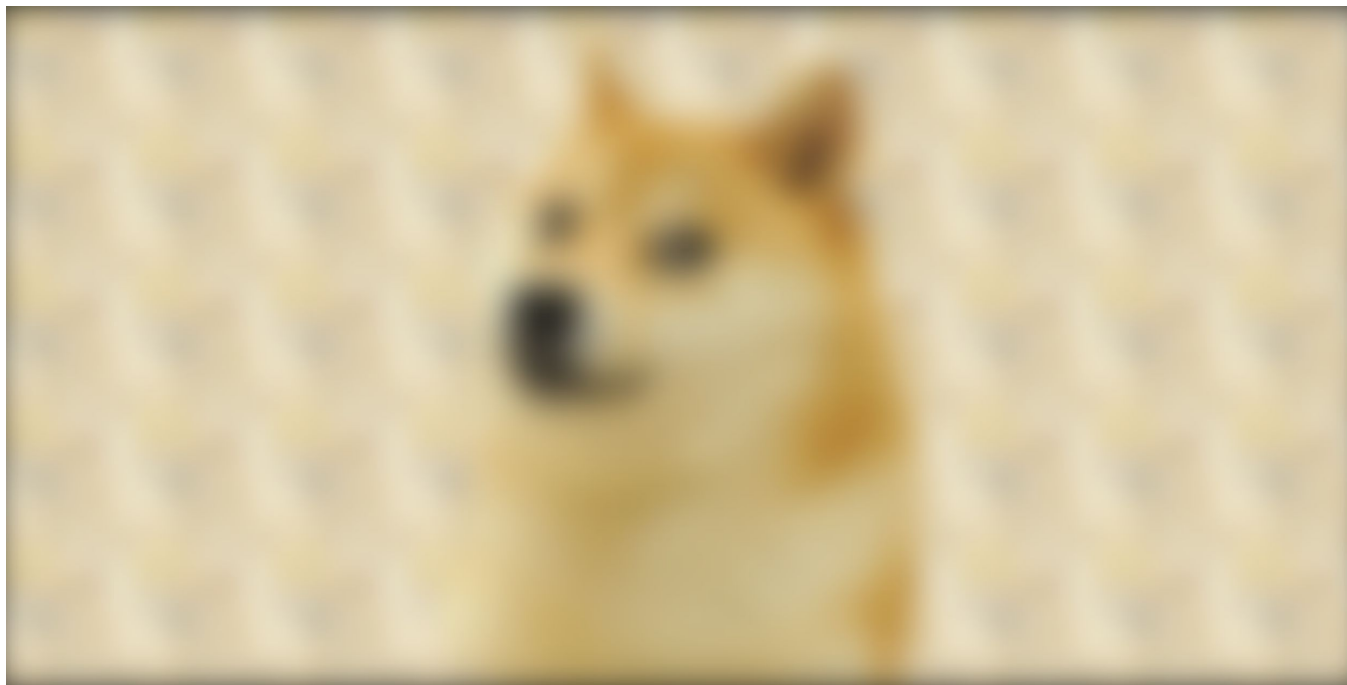


# CSS. Filter

[Пример MDN](#)

[Пример CSS TRICKS](#)

```
img {  
  filter: blur(25px);  
}
```



# CSS. Transition

## Пример MDN

```
/* Apply to 1 property */
/* property name | duration */
transition: margin-right 4s;

/* property name | duration | delay */
transition: margin-right 4s 1s;

/* property name | duration | timing function */
transition: margin-right 4s ease-in-out;

/* property name | duration | timing function | delay */
transition: margin-right 4s ease-in-out 1s;

/* Apply to 2 properties */
transition: margin-right 4s, color 1s;

/* Apply to all changed properties */
transition: all 0.5s ease-out;
```

# CSS. Animation

[Пример MDN](#)

[Пример css-tricks](#)

# CSS. Transform

[Пример 1](#)

[Пример w3cschools](#)

[Свойство will-change](#)

[Статья про рендер CSS на GPU](#)

[Статья про рендер CSS на GPU 2](#)

# CSS. Variables (custom properties)

```
:root {  
  --my-color: #FFF;  
}  
  
.color-picker {  
  box-shadow: 0 0 5em var(--my-color, white);  
}
```

[Пример MDN](#)

Статья [Medium](#)

# CSS. Variables from JS

```
// get value
document.documentElement.style.getPropertyValue('--my-color');

// set value
document.documentElement.style.setProperty('--my-color', 'green');
document.documentElement.style.setProperty('--my-color', 'var(--fancy-color)');
```


# CSS. Variables

Can I use

css variables

? ⚙ Settings

2 results found

# CSS Variables (Custom Properties)  - CR

Usage 

% of all users

 ?

Global 90.54% + 0.06% = 90.6%

Permits the declaration and usage of cascading variables in stylesheets.

Current aligned Usage relative Date relative Apply filters Show all ?

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsung Internet
	12-14		4-47		10-34										
	15	2-30	48	3.1-9	35	3.2-9.2									4
6-10	16-17	31-65	49-72	9.1-11.1	36-57	9.3-11.4		2.1-4.4.4	7	12-12.1			10		5-7
11	18	66	73	12	58	12.1	all	67	10	46	71	64	11	11.8	8.2
		67-68	74-76	12.1-TP		12.2									

Notes Known issues (3) Resources (8) Feedback

1

Enabled through the "Experimental Web Platform features" flag in `chrome://flags`

2

Partial support is due to bugs present (see known issues)

# Препроцессоры, инструменты



# CSS. Preprocessors

SASS/SCSS — Syntactically Awesome Stylesheets [sass-scss.ru](http://sass-scss.ru)

LESS — [lesscss.org](http://lesscss.org)

Stylus — [stylus-lang.com](http://stylus-lang.com)

[Статья](#) про препроцессоры.

# CSS. SCSS

```
$font-stack:    Helvetica, sans-serif;
$primary-color: #333;
.login-form {
  color: $primary-color;
  &__input {
    font: 18px $font-stack;
    &:active {
      background-color: white;
    }
  }
}
```

# CSS. SCSS—МИКСИНЫ

```
@mixin border-radius($radius) {  
    -webkit-border-radius: $radius;  
    -moz-border-radius: $radius;  
    -ms-border-radius: $radius;  
    border-radius: $radius;  
}  
  
.button {  
    @include border-radius(10px);  
}
```

# CSS. PostCSS

[PostCSS](#) is a tool for transforming styles with JS plugins. These plugins can *lint* your CSS, *support variables and mixins*, *transpile future CSS syntax*, *inline images*, and *more*.

# CSS. PostCSS - config

```
// postcss.config.js
module.exports = {
  plugins: [
    require('precss') ({/* ...options */}),
    require('autoprefixer') ({/* ...options */}),
    ...
  ]
};

// main.js
import 'styles.css';
```

# CSS. CSS Modules

## CSS Modules

```
/* button.css */  
  
.button {  
  width: 200px;  
  height: 48px;  
  border-radius: 12px;  
}  
  
.primary {  
  background-color: green;  
  font-weight: 500;  
}
```

# CSS. CSS Modules

```
// button.js
import styles from './button.css';
export default const Button = ({title, primary}) => {
  return (
    <button
      className={`${styles.button} ${primary ? styles.primary : ''}`}
    >
      {title}
    </button>
  );
}
```

# CSS. CSS Modules

```
<!-- результирующий HTML -->  
<button class="button-213ge1hw primary-jh4gd318">  
  Kek!  
</button>  
<button class="button-213ge1hw">  
  Kek is not Schreck  
</button>
```



# CSS. JSS

## JSS

```
// main.js
import jss from 'jss';
import preset from 'jss-preset-default';
import color from 'color';

// One time setup with default plugins and settings
jss.setup(preset());

const styles = {
  button: {
    width: 200,
    background: color('blue').darken(0.3).hex(),
  },
};
```

# CSS. JSS

## JSS

```
// app.js
const { classes } = jss.createStyleSheet(styles).attach();
document.body.innerHTML = `
  <button class="${classes.button}">
    Button
  </button>
`;
```

# CSS. Styled Components

## Styled Components

```
const Button = styled.a`
  display: inline-block;
  border-radius: 3px;
  padding: 0.5rem 0;
  margin: 0.5rem 1rem;
  width: 11rem;
  background: transparent;
  color: white;
  border: 2px solid white;

  ${props => props.primary && css`
    background: white;
    color: palevioletred;
  `}
`;
```

# CSS. Styled Components

## Styled Components

```
render(  
  <div>  
    <Button  
href="https://github.com/styled-components/styled-components"  
  target="_blank"  
  rel="noopener"  
  primary  
  >  
    GitHub  
  </Button>  
  
    <Button as={Link} href="/docs" prefetch>  
      Documentation  
    </Button>  
  </div>  
)
```

# CSS?

Вопросы?





Перерыв! (10 минут)

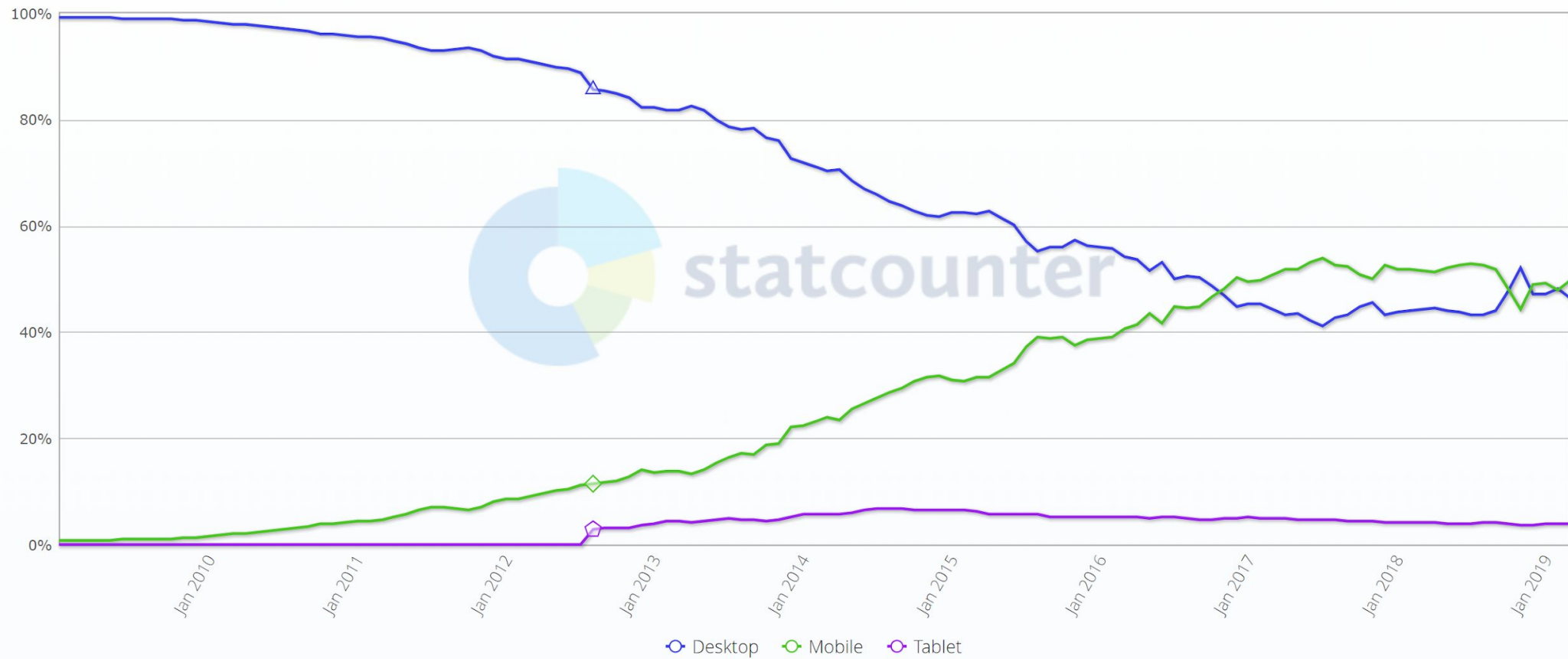
Препо (с)

# Разработка под мобильные устройства

# CSS. Device usage

## Desktop vs Mobile vs Tablet Market Share Worldwide

Jan 2009 - Mar 2019

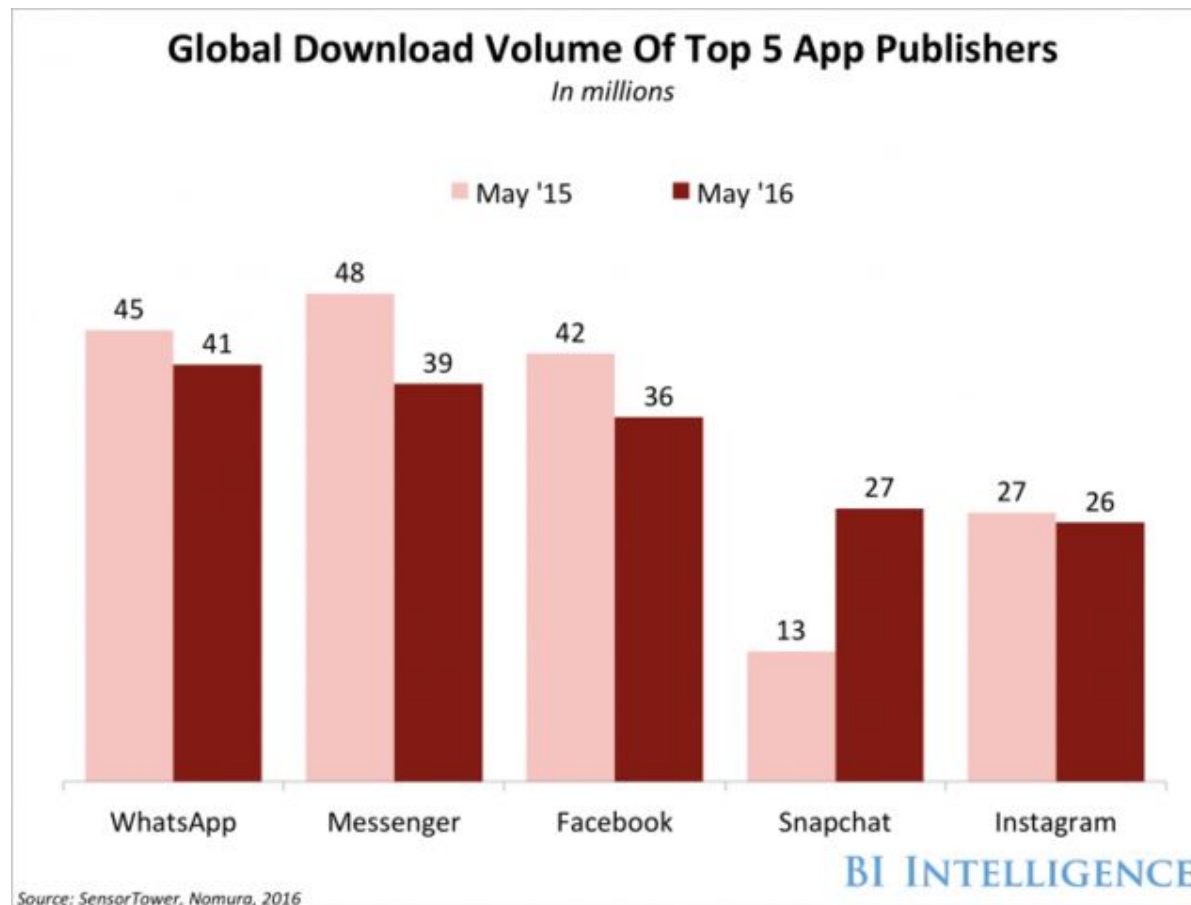




# CSS. Native vs. Web stats

[Статистика](#)

[Статистика 2](#)



# CSS. Подходы к разработке под мобильные устройства

- Отдельная мобильная версия
- Резиновый ( liquid ) дизайн
- Отзывчивый или адаптивный ( responsive ) дизайн
- Mobile-first подход
- **Прогрессивные веб-приложения**
  - graceful degradation
  - progressive enhancement

# CSS. Как определить возможности устройства?

- Какой тип устройства (экран, телевизор, часы, микроволновка, тостер...)?
- Какая плотность пикселей?
- Какое разрешение у дисплея
- Какой размер устройства?
- многое другое

# CSS. Медиа-запросы (Media Queries)

A *media query* is a method of testing certain aspects of the user agent or device that the document is being displayed in

```
@media screen and (max-width: 1900px) {  
  .container {  
    width: 70vw;  
    max-width: 1200px;  
  }  
}
```

# CSS. Медиа-запросы. Использование в CSS

```
@media screen and (color) { /* Для цветных экранов */  
  body { background: whitesmoke; }  
}  
  
/* Для широкоформатных экранов */  
@media screen and (min-device-aspect-ratio: 16/10) {  
  ...  
}
```

# CSS. Медиа-запросы. Использование в CSS

```
/* Для широкоформатных экранов */  
@media screen and (min-device-aspect-ratio: 16/10) { ... }  
  
@media projection { /* проектор */ }  
@media handheld { /* смартфоны и носимые устройства */ }  
@media tv { /* телевизоры */ }  
@media braille { /* устройства для слабовидящих */ }  
  
@media (hover) { /* если на устройстве работает hover */ }  
  
@media (pointer: coarse) { /* тачскрины, управляемые пальцами */ }  
@media (pointer: fine) { /* мышь или стилус */ }  
@media (pointer: none) { /* нет курсора */ }
```

# CSS. Медиа-запросы. Использование в HTML

```
<link rel="stylesheet"  
      media="all and (orientation : portrait)"  
      href="portrait.css">
```

```
<link rel="stylesheet"  
      media="all and (orientation : landscape)"  
      href="landscape.css">
```

# CSS. CSS директива @supports

```
@supports (display: flex) {  
  div { display: flex; }  
}
```

```
@supports not (display: flex) {  
  div { float: left; } /* задан альтернативный стиль */  
}
```



# CSS. CSS директива @supports

```
@supports (display: -webkit-flex) or
          (display: -moz-flex) or
          (display: flex) {
  /* добавляем сюда ваших клёвых стилей */
}
```

```
// использование в JS
const supportsFlex1 = CSS.supports('display', 'flex');
const supportsFlex2 = CSS.supports('(display: flex)');
```

# Единицы измерения в CSS

# CSS. Единицы измерения CSS

## Спецификация

### Относительные

- `em` — font size of the element
- `ex` — x-height of the element's font
- `ch` — width of the "0" (ZERO, U+0030) glyph in the element's font
- `rem` — font size of the root element
- `vw` — 1% of viewport's width
- `vh` — 1% of viewport's height
- `vmin` — 1% of viewport's smaller dimension
- `vmax` — 1% of viewport's larger dimension

# CSS. Единицы измерения CSS

## Спецификация

### Абсолютные

- **cm** —  $1\text{cm} = 96\text{px}/2.54$
- **mm** —  $1\text{mm} = 1/10\text{th of } 1\text{cm}$
- **Q** —  $1\text{q} = 1/40\text{th of } 1\text{cm}$
- **in** —  $1\text{in} = 2.54\text{cm} = 96\text{px}$
- **pc** —  $1\text{pc} = 1/6\text{th of } 1\text{in}$
- **pt** —  $1\text{pt} = 1/72\text{th of } 1\text{in}$
- **px** —  $1\text{px} = 1/96\text{th of } 1\text{in} *$

# CSS. Типы пикселей

[Статья](#)

[Статья 2](#)

- **Аппаратные пиксели** — физический пиксель матрицы дисплея
- **Аппаратно-независимые пиксели** (Device-independent pixels, dip) — пиксели дисплея, приведённые к единому масштабу, чтобы соответствовать примерно одинаковому углу зрения на всех девайсах (с учётом расстояния, на котором мы их держим)
- **Пиксель CSS** — единица измерения вёрстки

# CSS. Retina



# CSS. Retina

[Статья Retina](#)

## RETINA

2880px across

1800px high

HTML  
OBJECT

physically  
**800x600**  
on screen

## Actual CSS

```
div {  
  width: 400px;  
  height: 300px;  
}
```

## NORMAL

1440px across

900px high

HTML  
OBJECT

physically  
**400x300**  
on screen

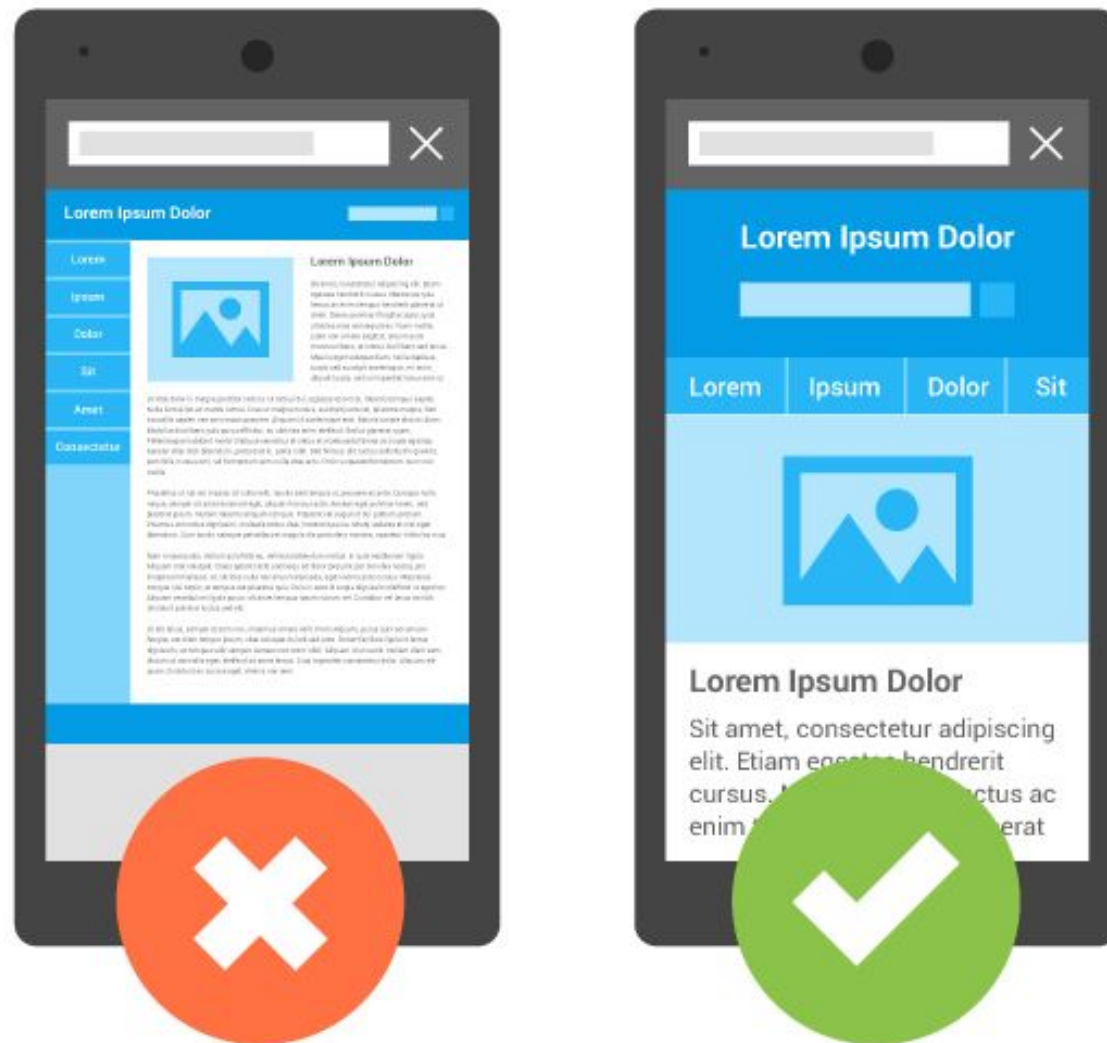
## Actual CSS

```
div {  
  width: 400px;  
  height: 300px;  
}
```

# Область просмотра (viewport)



# CSS. Область просмотра (viewport)



# CSS. Область просмотра (viewport)

## Значения:

- width, height
- initial-scale
- minimum-scale, maximum-scale
- user-scalable

```
<meta name=viewport content="width=device-width, initial-scale=1">
```

# CSS. Область просмотра (viewport)

```
@viewport {  
  width: device-width;  
  height: device-height;  
  zoom: 2;  
  user-zoom: fixed;  
}
```

# Нативное взаимодействие

# CSS. Нативное взаимодействие

Проблемы:

- Зоопарк событий
- 300-ms задержка
- Нет нужных событий

# CSS. Зоопарк событий

События mouse-events:

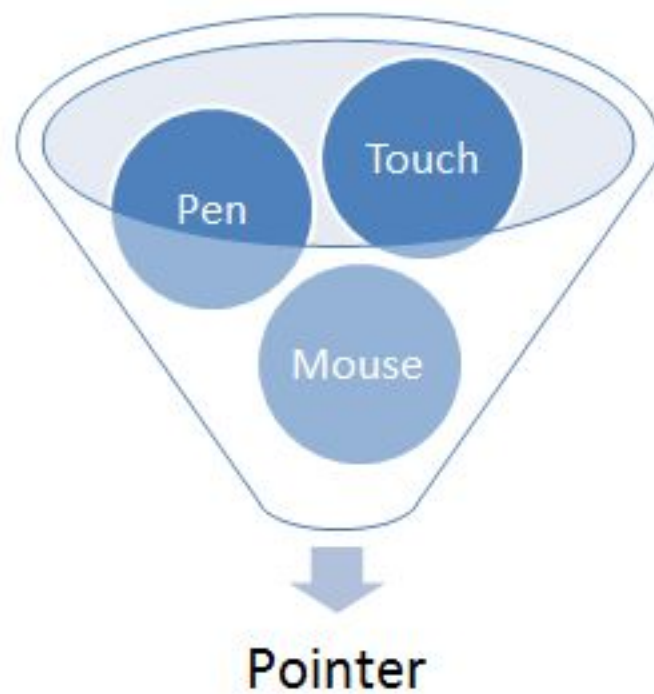
- mousedown
- mouseup
- click
- dblclick
- contextmenu
- mouseover (-out, -move)

# CSS. Зоопарк событий

События touch-events:

- touchstart
- touchmove
- touchcancel
- touchend

# CSS. Зоопарк событий. Решение





# CSS. Зоопарк событий

События pointer-events:

- pointerdown
- pointerup
- pointercancel
- pointermove
- pointerover
- pointerout
- gotpointercapture
- lostpointercapture

# CSS. Зоопарк событий

Can I use

pointer events



Settings

5 results found

#

CSS pointer-events (for HTML) - UNOFF

Usage % of all users ?  
Global 96.55%

This CSS property, when set to "none" allows elements to not receive hover/click events, instead the event will occur on anything behind it.

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baid Brow:
		2-3.5		3.1-3.2	10-12.1										
6-10	12-17	3.6-68	4-76	4-12.1	15-60	3.2-12.3		2.1-4.4.4	12-12.1				4-9.2		
11	18	69	77	13	62	13.1	all	76	46	76	68	12.12	10.1	1.2	7.1
	76	70-71	78-80	TP											

# CSS. 300-ms задержка

Между событием touchend и click проходит 300-350ms

```
html {  
  touch-action: manipulation;  
  touch-action: auto;  
  touch-action: pan-x;  
  touch-action: pinch-zoom;  
}
```

# CSS. Дополнительные события

- pan
- swipe
- rotate
- pinch/zoom
- doubletap

# CSS. Дополнительные события. Решение

```
// HammerJS - http://hammerjs.github.io/  
const hammer = new Hammer(element, options);  
hammer.on('swipe', function(event) {  
    console.log(event);  
});
```

# CSS. PWA, TWA

PWA:

<https://developers.google.com/web/progressive-web-apps/checklist>

<https://habr.com/ru/company/google/blog/414609/>

<https://web.archive.org/web/20151103001838/http://www.luster.io/blog/9-29-14-mobile-web-checklist.html>

TWA:

<https://developers.google.com/web/updates/2019/02/using-twa>

<https://habr.com/ru/post/439238/>

[Lighthouse](#)

# Домашнее задание №4

1. Добавить “украшательства”, анимации, транзишны
2. Добавить viewport
3. Приблизить дизайн своего приложения к макетам
4. ...

Расширенное описание задания, подсказки, а также презентации с лекций всегда есть в репозитории.

Срок сдачи

*27 октября*

# Спасибо за внимание!





# Пока!

Присоединяйтесь к сообществам про образование в VK

- [VK Джуниор](#)
- [VK Образование](#)

