



# Web-технологии

ОСНОВЫ CSS, LESS, SASS

- Оформление HTML с использованием CSS
- Селекторы, свойства и их значения, псевдоклассы
- Оформление границ, отступов, размеров, ссылок, списков, таблиц, текста, настройка шрифта, использование иконок
- Позиционирование элементов
- Адаптивная верстка, использование grid, flexbox
- Использование шаблонов web-страниц, переменных в CSS
- Работа с CSS с использованием JS
- Использование CSS-фильтров
- Библиотека w3.css
- Знакомство с фреймворками LESS и SASS

# Версии CSS (Cascading Style Sheets)

## Каскадные таблицы стилей

- CSS1 – 1996 год
  - Параметры шрифтов
  - Цвета
  - Атрибуты текста
  - Выравнивание для текста, изображений, таблиц и других элементов
  - Свойства блоков, такие как высота, ширина, внутренние (padding) и внешние (margin) отступы и рамки
  - Ограниченные средства по позиционированию элементов
- CSS2 – 1998 год
  - Блочная верстка
  - Типы носителей
  - Звуковые таблицы стилей
  - Страничные носители
  - Расширенный механизм селекторов
  - Указатели
  - Генерируемое содержание
- CSS3 – 2011 год (?)
  - Возможность создавать анимированные элементы без использования JS
- CSS4
  - Модули CSS4 построены на основе CSS3 и дополняют их новыми свойствами и значениями

# Объявления CSS

- Определяет, как элементы HTML будут отображаться браузером
- Внешние CSS сохраняются в отдельных файлах
- Синтаксис
  - Селектор {свойство:значение; свойство:значение; }
- Объявление
  - свойство: значение;
- Пример
- Тот же пример

```
p {color:green; font-family:arial;}
```

```
p {  
    color: green;  
    font-family: arial;  
}
```

Комментарий заключается в /\* ... \*/  
Пример:  
/\* Комментарий может быть  
многострочным \*/

- **Простые селекторы**

- выбор элементов на основе имени, идентификатора, класса

```
p { color: red; }  
#f { font-size: small; }  
.cl { background: red; }
```

- **Комбинируемые селекторы**

- выбор элементов на основе определенной взаимосвязи между ними

- **Селекторы псевдоклассов**

- выбор элементов на основе определенного состояния

- **Селекторы псевдоэлементов**

- выбор и стилизация части элемента

- **Селекторы атрибутов**

- выбор элементов на основе атрибута или значения атрибута

## Простые

- **A или A, B**
  - p {...}
  - div {...}
  - p, div {...}
- **#id**
  - #tagId {...}
- **.class**
  - p.myP {...}
  - .myP {...}
- **class.class**
  - .myP1.myP2 {...}
  - <p class="myP1 myP2">...
- **\***
  - \* {...}

## Комбинируемые

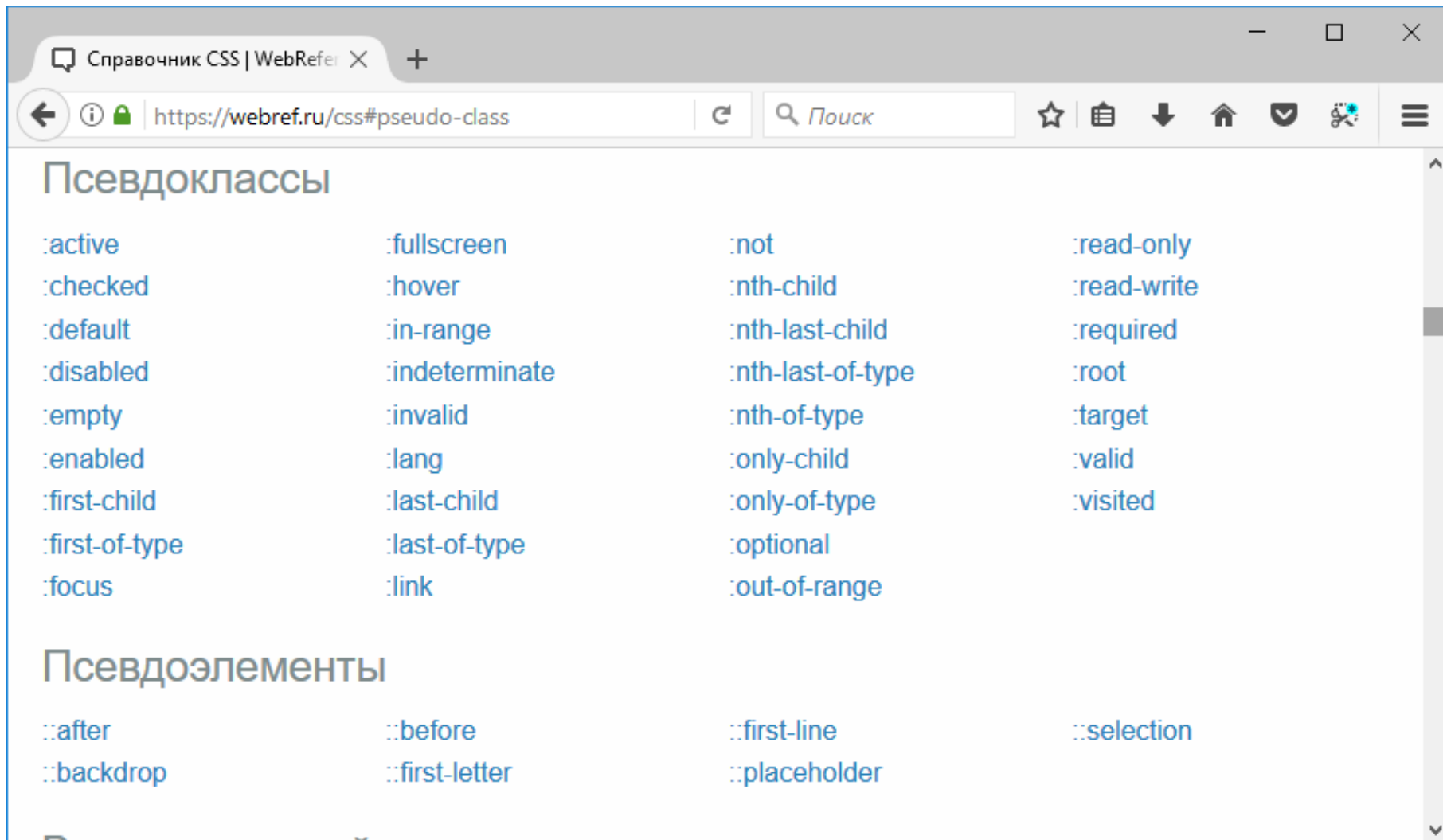
- **A B // одновременно**
  - div p {...}
- **A > B // потомок**
  - div > p {...}
- **A + B // соседний**
  - i b {...}
  - <i>...</i><b> ... </b>

## По атрибутам

- **. A[attr]**
  - a[title] {...}
- **A[attr="value"]**
  - option[name="selected"] {...}

# Псевдоклассы в CSS

7



<https://webref.ru/css#pseudo-class>

# Способы подключения CSS

## Внешний

```
<head>
  <link rel="stylesheet" href="mystyle.css">
</head>
```

### mystyle.css

```
p { color: red; }
#f { font-size: small; }
.cl { background: red; }
```

## Внутренний

```
<head>
  <style>
    p { color: red; }
    #f { font-size: small; }
    .cl { background: red; }
  </style>
</head>
```

```
<head>
  <style> /* последовательная загрузка */
    @import url("/styles/mystyle.css");
  </style>
</head>
```

## Встроенный

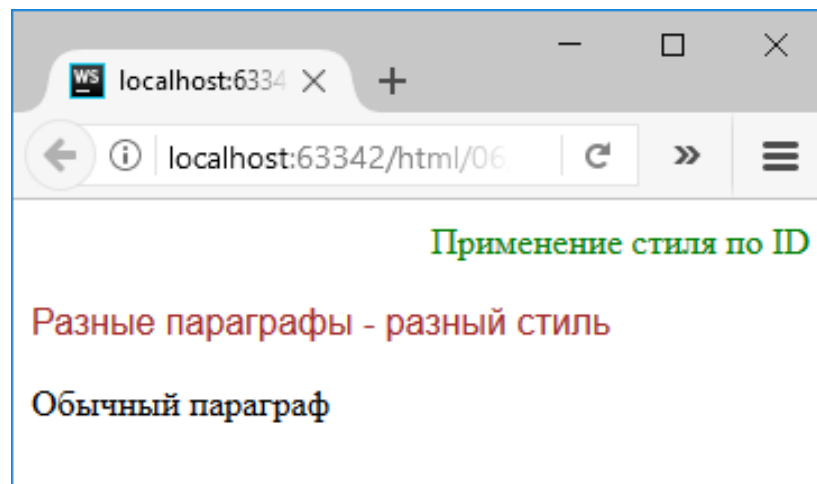
```
<body>
<p style="color: red;font-size: small;">Текст</p>
</body>
```



# CSS (1), id, class

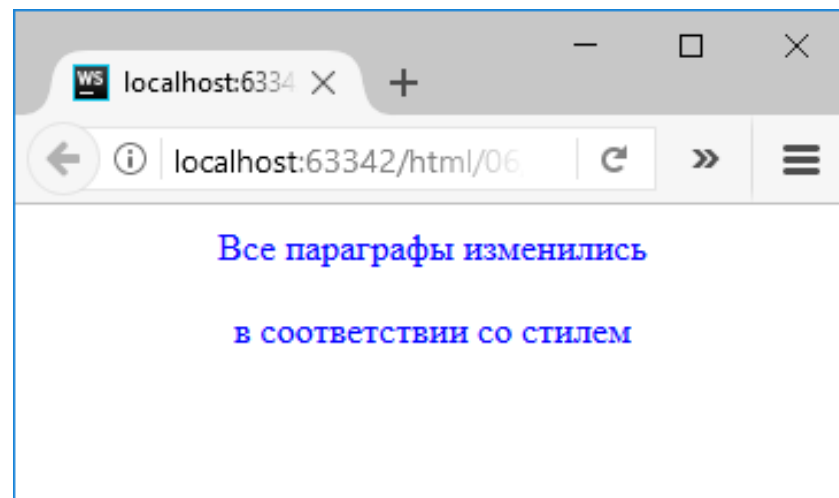
9

```
<html>
<head>
  <meta charset="utf-8">
  <style>
    #test1 {
      text-align: right;
      color: green;
    }
    p.test2{
      font-family: arial;
      color: brown;
    }
  </style>
</head>
<body>
<p id="test1">Применение стиля по ID</p>
<p class='test2'>Разные параграфы - разный стиль</p>
<p>Обычный параграф</p>
</body>
</html>
```



# CSS (2), element

```
<html>
<head>
  <meta charset="utf-8">
  <style type="text/css">
    p {
      text-align: center;
      color: blue;
    }
  </style>
</head>
<body>
<p>Все параграфы изменились</p>
<p>в соответствии со стилем</p>
</body>
</html>
```



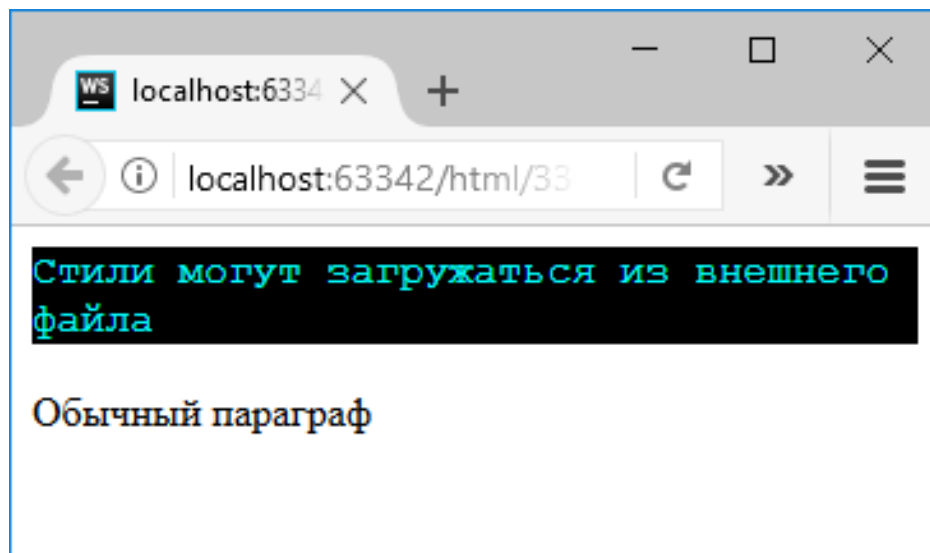
# CSS (3), внешний файл, класс

11

```
<html>
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="simple.css">
</head>
<body>
<p class="test1">Стили могут загружаться из внешнего файла</p>
<p>Обычный параграф</p>
</body>
</html>
```

## simple.css

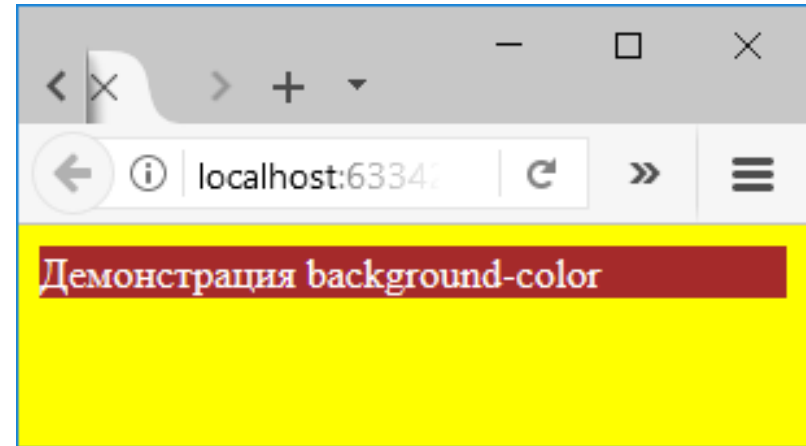
```
p.test1 {
  font-family: courier;
  color: cyan;
  background-color: black;
}
```



# CSS (4), background-color, color

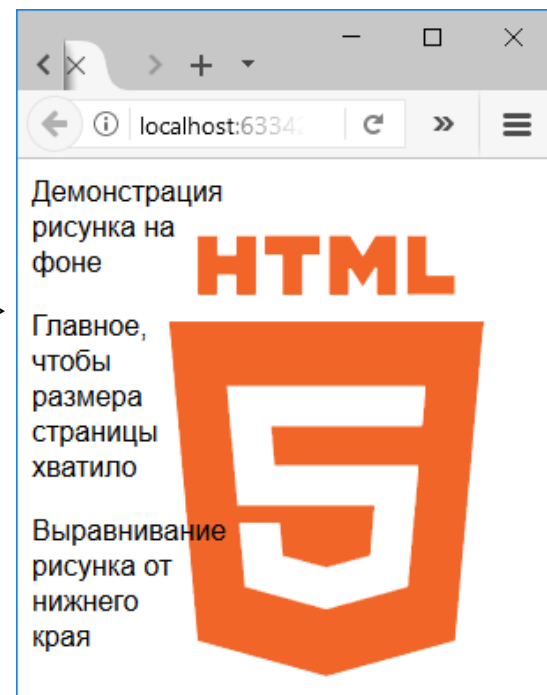
12

```
<html>
<head>
  <meta charset="utf-8">
  <style type="text/css">
    body {
      background-color: yellow;
    }
    p {
      background-color: brown;
      color: white;
    }
  </style>
</head>
<body>
<p>Демонстрация background-color</p>
</body>
</html>
```



# CSS (5), изменение нескольких свойств

```
<html>
<head>
  <meta charset="utf-8">
  <style type="text/css">
    body {
      background: url('../html5.png') no-repeat bottom right;
      padding-right: 200px;
      font-family: arial;
    }
  </style>
</head>
<body>
<p>Демонстрация рисунка на фоне</p>
<p>Главное, чтобы размера страницы хватило</p>
<p>Выравнивание рисунка от нижнего края</p>
</body>
</html>
```



# Приоритеты стилей

- Селектор тега
- Селектор класса
- Селектор ID
- Inline-стиль
- Применение «!important» отменяет все предыдущие стили

```
p {  
  background-color: red !important;  
}
```

# Настройка цвета в CSS

- Ключевые слова: **red, orange, yellow...**
  - **color: silver;**
- RGB в процентах
  - **color: rgb(20%, 80%, 70%)**
- RGB в значениях (до 256)
  - **color: rgb(123, 17, 250)**
- Шестнадцатеричный RGB
  - **color: #3F12EE**
- HSL (hue, saturation и lightness)
  - **color:hsl(140, 51%, 48%)**
- **rgba(red, green, blue, alpha)**
- **hsla(hue, saturation, lightness, alpha)**

**<h1 style="background-color:DodgerBlue;">Фон</h1>**

**<h1 style="color:Tomato;">Текст</h1>**

**<h1 style="border:2px solid Tomato;">Бордюр</h1>**

# ФОН В CSS

16

```
/* Цвет фона */
#i1 { background-color: lightblue; }
/* Цвет фона с прозрачностью в [0, 1] */
#i11 {
    background-color: green;
    opacity: 0.3;
}
/* Рисунок на фоне */
#i2 { background-image: url("image.gif"); }
/* Повторить рисунок */
#i3 { background-image: url("image.png"); }
/* Повторить рисунок по горизонтали */
#i31 {
    background-image: url("image.png");
    background-repeat: repeat-x;
}
/* Фиксированный/прокр. при скролле фон. рис. */
#i4 {
    background-image: url("image.png");
    background-repeat: no-repeat;
    background-position: right top;
    background-attachment: fixed; /*scroll*/
}
/* Краткая запись */
#i41 {
    background: white url("image.png") no-repeat right top;
}
```



# Стиль border (border-style)

- **dotted** – Определяет пунктирную границу
- **dashed** – Определяет пунктирную границу
- **solid** – Определяет сплошную границу
- **double** – Определяет двойную границу
- **groove** – Определяет 3D- groove границу. Эффект зависит от значения цвета границы
- **ridge** – Определяет 3D-ребристую границу. Эффект зависит от значения цвета границы
- **inset** – Определяет границу 3D- inset. Эффект зависит от значения цвета границы
- **outset** – Определяет начальную границу 3D-outset. Эффект зависит от значения цвета границы
- **none** – Нет границы
- **hidden** – Определяет скрытую границу

# Примеры border-style

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

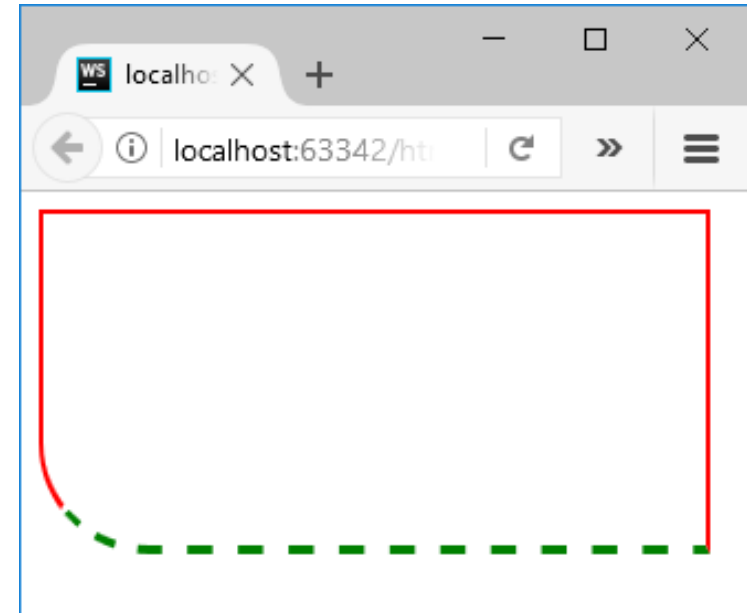
A hidden border.

# Оформление границ

```
p { border-style: solid; }  
/* pixels */  
p.i1 { border-width: 4px; }  
p.i2 { border-width: medium; }  
p.i3 { border-width: thick; }  
/* 15px top & bottom, 20px on the sides */  
p.i4 { border-width: 15px 20px; }  
/* 3px top, 15px right, 42px bottom, 7px left */  
p.i5 { border-width: 3px 15px 42px 7px; }  
p.i6 { border-color: red; }  
/* green top, red right, blue bottom & yellow left */  
p.i7 { border-color: green red blue yellow; }  
p.i8 {  
    border-top-style: solid;  
    border-right-style: dashed;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}  
p.i9 { border: 7px dashed red; }  
p.i10 {  
    border-left: 6px solid red;  
    border-bottom: 7px dashed green;  
    border-radius: 3px; /* rounded */  
}
```

# Пример оформления border

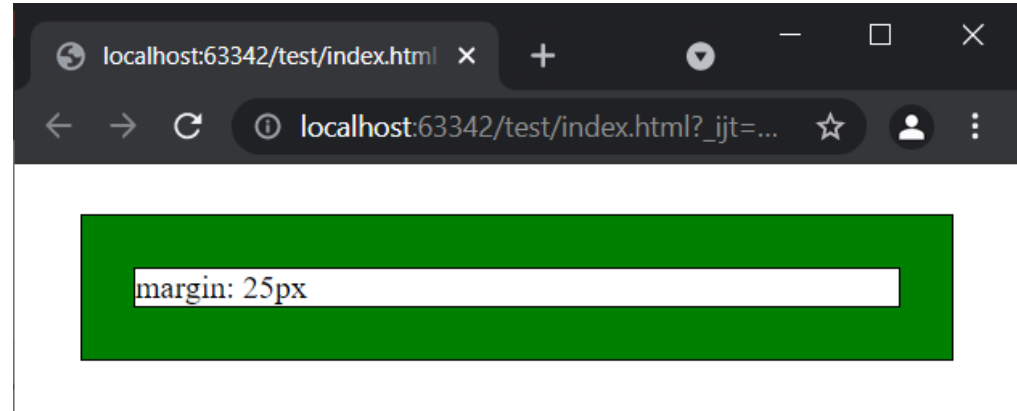
```
<html>
<head>
  <style type="text/css">
    div {
      border: 2px;
      border-style: solid;
      border-color: red;
      border-bottom-width: 4px;
      border-bottom-style: dashed;
      border-bottom-color: green;
      border-bottom-left-radius: 50px;
    }
  </style>
</head>
<body>
  <div style="width: 300px; height: 150px; "></div>
</body>
</html>
```



# Отступы вокруг элемента – margin

21

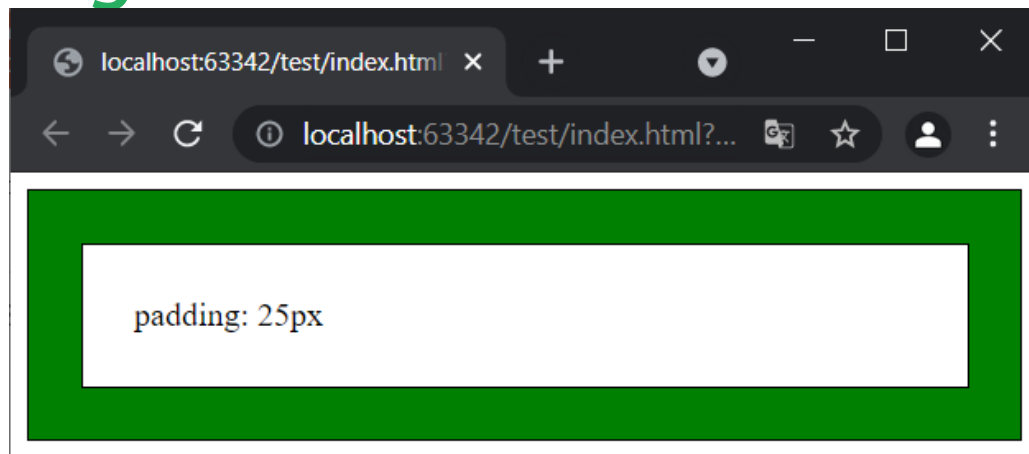
```
p {  
  margin-top: 10px;  
  margin-bottom: 10px;  
  margin-right: 20px;  
  margin-left: 10px;  
}  
  
div { /* top right bottom left */  
  margin: 10px 20px 30px 40px;  
}  
  
div.i1 { /* top right&left bottom */  
  margin: 10px 20px 30px;  
}  
  
div.i2 { /* top&bottom right&left */  
  margin: 10px 20px;  
}  
  
div.i3 { /* top&right&bottom&left */  
  margin: 10px;  
}
```



- auto – автоматическое распределение
- inherit – от родительского элемента

# Место вокруг содержимого элемента – padding

```
p {  
  padding-top: 10px;  
  padding-bottom: 10px;  
  padding-right: 20px;  
  padding-left: 10px;  
}  
  
div { /* top right bottom left */  
  padding: 10px 20px 30px 40px;  
}  
  
div.i1 { /* top right&left bottom */  
  padding: 10px 20px 30px;  
}  
  
div.i2 { /* top&bottom right&left */  
  padding: 10px 20px;  
}  
  
div.i3 { /* top&right&bottom&left */  
  padding: 10px;  
}
```



# Высота (height) и ширина (width)

- **auto** – по умолчанию
- **length** – длина в абсолютных единицах (px...)
- **%** – в процентах от блока
- **initial** – задание начального значения
- **inherit** – наследование от родительского элемента

**max-width, min-width, max-height, min-height**

```
p {  
  height: 100px;  
  width: 30%;  
}
```

# Единицы измерения

- **px** – пиксель на экране
  - mm – 3.8px
  - cm – 38px
  - pt – 4/3 px
  - pc – 16px
- **em** – относительно размера шрифта (пропорция относительно размера шрифта родителя)
- **ex** – размер символа «x»
- **eh** – размер символа «0»
- **%** – относительный размер (в основном относительно такого же свойства родителя)
- **rem** – некоторый «разумный» (reasonable) размер по умолчанию
- **vw** – 1% ширины окна
- **vh** – 1% высоты окна
- **vmin** – наименьшее из (**vw**, **vh**)
- **vmax** – наибольшее из (**vw**, **vh**)

Устарели

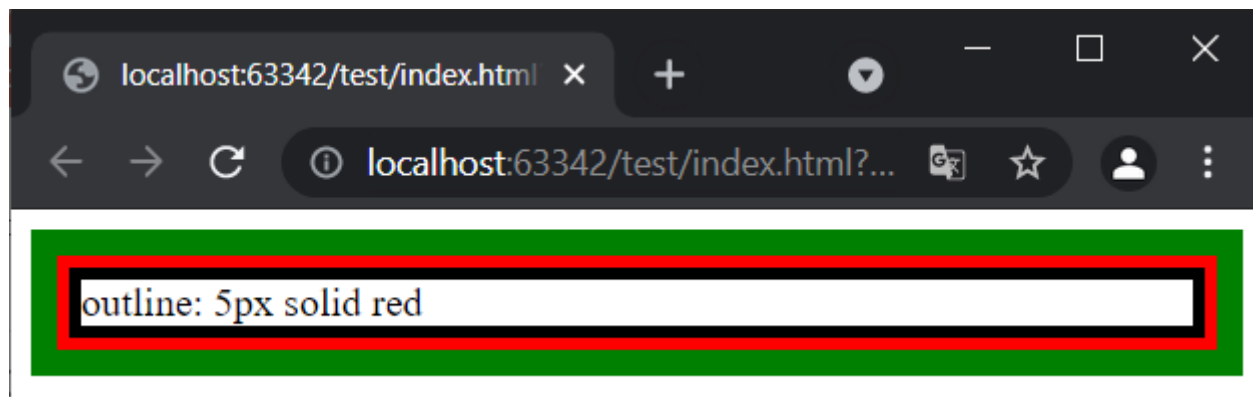
Часто используются  
для мобильных  
устройств



# Граница элемента за пределами

## border – outline

- outline-style
  - dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden
- outline-color
- outline-width
- outline-offset
- outline



```
div.c {  
    border: 5px solid black;  
    outline: 5px solid red;  
    background-color: white;  
}
```

```
<div style="background-color: green; padding: 15px;">  
  <div class="c">outline: 5px solid red</div>  
</div>
```

# Выравнивание текста

*/\* По центру \*/*

**p** { **text-align**: **center**; }

*/\* Слева \*/*

**p** { **text-align**: **left**; }

*/\* Справа \*/*

**p** { **text-align**: **right**; }

*/\* Выравнивание по ширине \*/*

**p** { **text-align**: **justify**; }

*/\* Справа налево \*/*

**p** {  
    **direction**: **rtl**;  
    **unicode-bidi**: **bidi-override**;  
}

*/\* Выравнивание рисунка в тексте \*/*

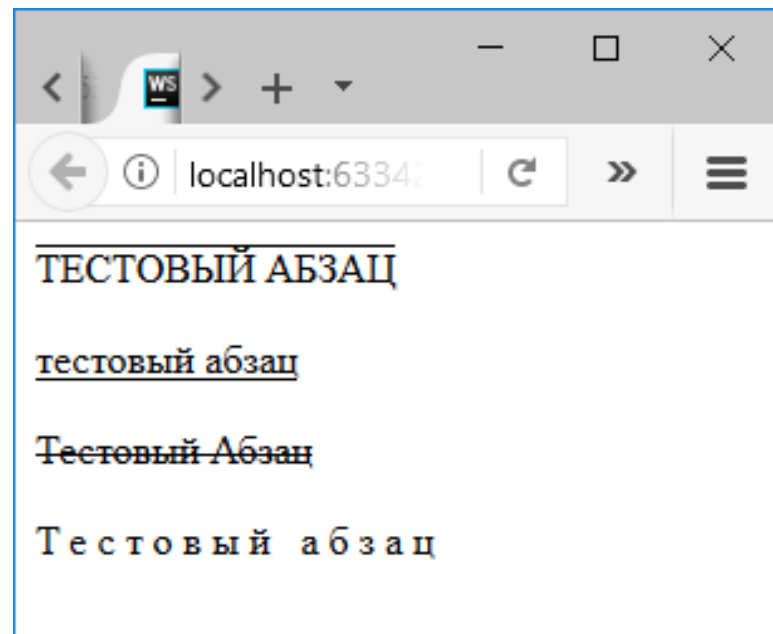
**img** { **vertical-align**: **baseline**; */\* по центру \*/* }

*/\* baseline, text-top, text-bottom, sub, super \*/*

<p>Рисунок <**img src="icon.gif" width="8" height="8"**> в тексте</p>

# Оформление и трансформация текста

```
<html>
<head>
  <meta charset="utf-8">
  <style type="text/css">
    p.upper {
      text-transform: uppercase;
      text-decoration: overline;
    }
    p.lower {
      text-transform: lowercase;
      text-decoration: underline;
    }
    p.del {
      text-transform: capitalize;
      text-decoration: line-through;
    }
    p.space {
      letter-spacing: 4px;
    }
  </style>
</head>
```



```
<p class="upper">Тестовый абзац</p>
<p class="lower">тестовый абзац</p>
<p class="del">Тестовый абзац</p>
<p class="space">Тестовый абзац</p>
</body>
</html>
```

# Отступы в тексте

*/\* Отступ первой строки \*/*

**p { text-indent: 20px; }**

*/\* Расстояние между символами \*/*

**p { letter-spacing: 2px; }**

*/\* Расстояние между строками абзаца \*/*

**p { line-height: 0.9; }**

*/\* Расстояние между словами \*/*

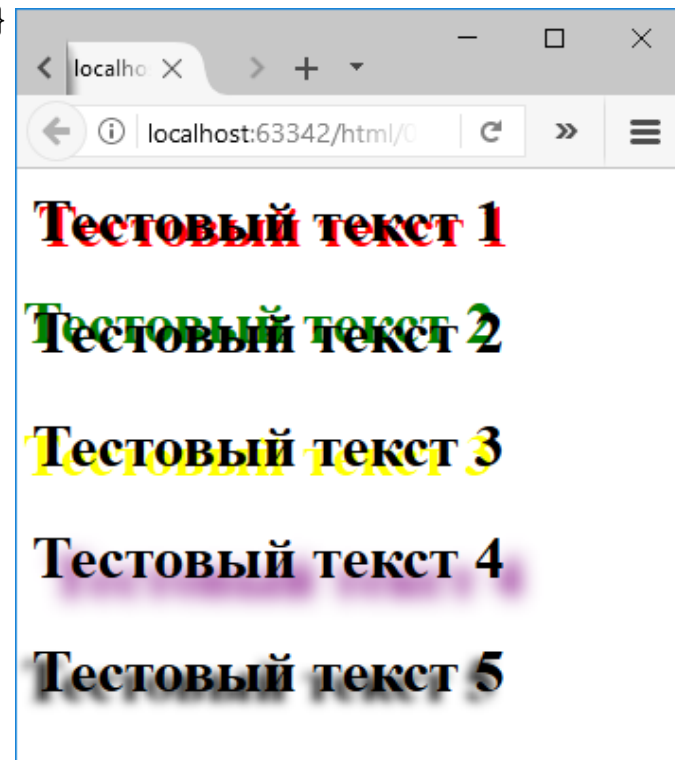
**p { word-spacing: 8px; }**

*/\* Отключение перехода на следующую строку  
в конце страницы \*/*

**p { white-space: nowrap; }**

# Тени в тексте

```
<html>
<head>
  <meta charset='utf-8'>
  <style type='text/css'>
    /* x y размытость цвет */
    .sh1 { text-shadow: 3px 3px red; }
    .sh2 { text-shadow: -5px -5px green; }
    .sh3 { text-shadow: -5px 5px yellow; }
    .sh4 { text-shadow: 10px 10px 10px purple; }
    .sh5 { text-shadow: -5px 5px 5px black; }
  </style>
</head>
<body>
  <h1 class='sh1'>Тестовый текст 1</h1>
  <h1 class='sh2'>Тестовый текст 2</h1>
  <h1 class='sh3'>Тестовый текст 3</h1>
  <h1 class='sh4'>Тестовый текст 4</h1>
  <h1 class='sh5'>Тестовый текст 5</h1>
</body>
</html>
```



# Стиль шрифта

*/\* НАКЛОН \*/*

**font-style: normal;** */\* нормальный \*/*

**font-style: italic;** */\* наклонный \*/*

*/\* ЖИРНОСТЬ \*/*

**font-weight: normal;** */\* нормальный \*/*

**font-weight: bold;** */\* жирный \*/*

*/\* ЖИРНОСТЬ ОТ РОДИТЕЛЯ \*/*

**font-weight: lighter;** */\* легче \*/*

**font-weight: bolder;** */\* жирнее \*/*

*/\* Размер букв \*/*

**font-variant: normal;** */\* обычные \*/*

**font-variant: small-caps;** */\* БОЛЬШИЕ \*/*

# Размер шрифта

*/\* Предопределённые значения \*/*

**font-size: xx-small;**

**font-size: x-small;**

**font-size: small;**

**font-size: medium;**

**font-size: large;**

**font-size: x-large;**

**font-size: xx-large;**

**font-size: xxx-large;**

*/\* Относительные значения \*/*

**font-size: smaller;**

**font-size: larger;**

*/\* Абсолютные значения \*/*

**font-size: 10px;**

**font-size: 0.9em;**

*/\* В процентах \*/*

**font-size: 70%;**

*/\* Глобальные переменные \*/*

**font-size: inherit;**

# Шрифты

*/\* Имя и общее имя \*/*

**font-family:** "Arial", sans-serif;

**font-family:** "Times New Roman", serif;

*/\* Только общее имя \*/*

**font-family:** serif;

**font-family:** sans-serif;

**font-family:** monospace;

**font-family:** cursive;

**font-family:** fantasy;

**font-family:** system-ui;

**font-family:** ui-serif;

**font-family:** ui-sans-serif;

**font-family:** ui-monospace;

**font-family:** ui-rounded;

**font-family:** emoji;

**font-family:** math;

**font-family:** fangsong;

*/\* Глобальные значения \*/*

**font-family:** inherit;

**font-family:** initial;

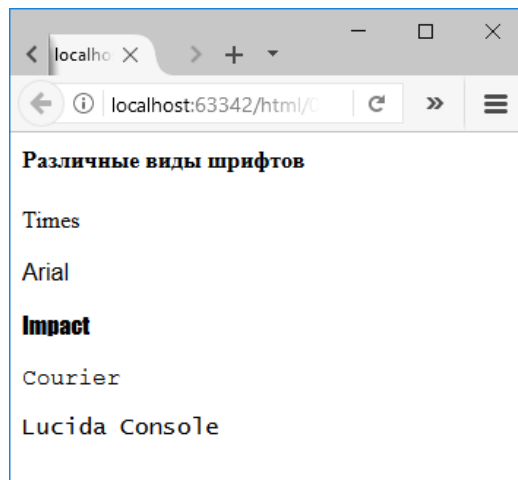
**font-family:** revert;

**font-family:** unset;

## «Безопасные» web-шрифты:

- Arial (sans-serif)
- Verdana (sans-serif)
- Helvetica (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

Есть резервные варианты шрифтов для отображения в любых браузерах





# Загрузка специфического шрифта

## С использованием директив

```
@font-face {  
  font-family: Pompadur; /* Имя шрифта */  
  src: url("fonts/pompadur.ttf"); /* Путь к файлу со шрифтом */  
}  
  
p {  
  font-family: Pompadur;  
}
```

## С использованием link

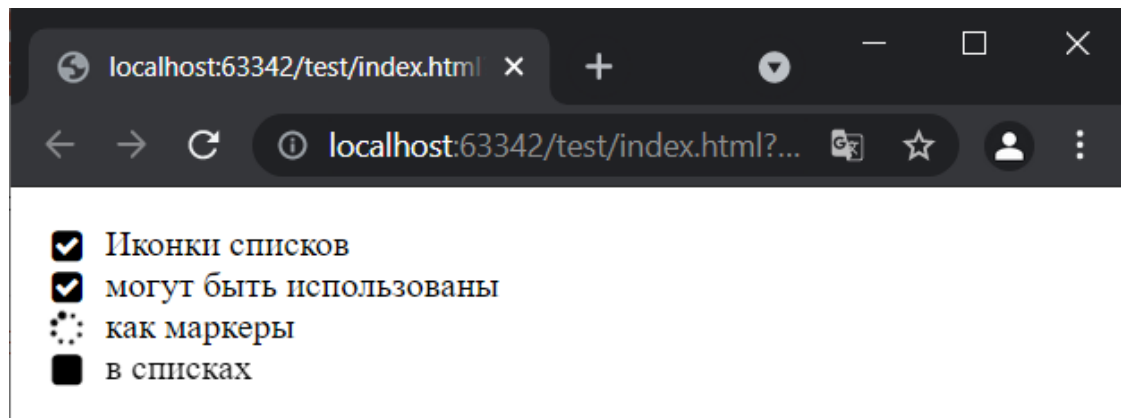
```
<head>  
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">  
  <style>  
    p {  
      font-family: "Sofia", sans-serif;  
    }  
  </style>  
</head>
```

# Иконки (1), fontawesome

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="font-awesome/css/font-awesome.min.css">
</head>
<body>
<ul class="fa-ul">
  <li><i class="fa-li fa fa-check-square"></i>Иконки списков</li>
  <li><i class="fa-li fa fa-check-square"></i>могут быть использованы</li>
  <li><i class="fa-li fa fa-spinner fa-spin"></i>как маркеры</li>
  <li><i class="fa-li fa fa-square"></i>в списках</li>
</ul>
</body>
</html>
```

<https://fontawesome.ru/>

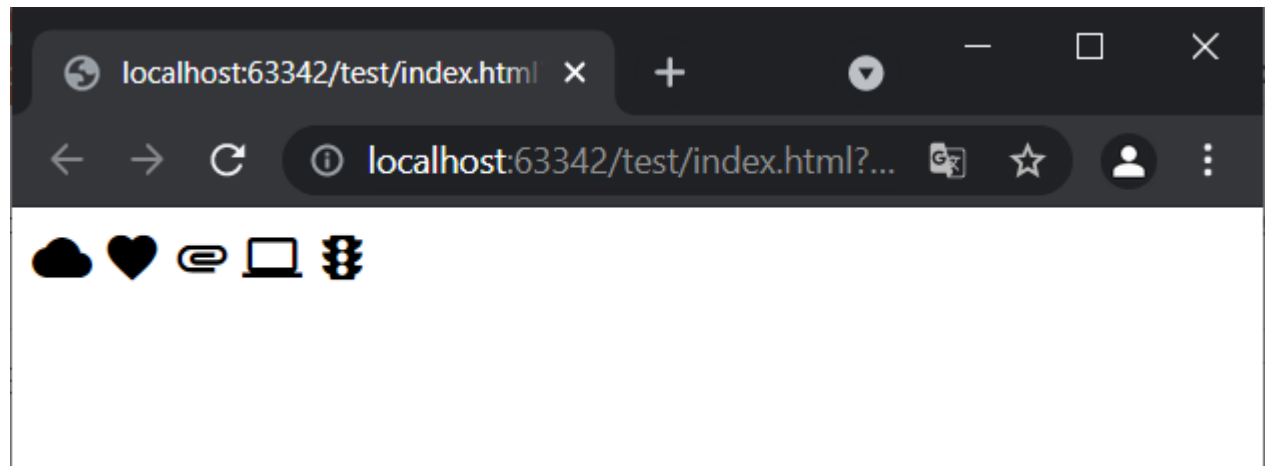
- скачать иконки и шрифты в проект
- подключить link
- использовать классы



# Иконки (2), google

35

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
</head>
<body>
<i class="material-icons">cloud</i>
<i class="material-icons">favorite</i>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">traffic</i>
</body>
</html>
```



<b>a:link</b>	{ <b>text-decoration</b> : none; }	<i>/* ссылка */</i>
<b>a:visited</b>	{ <b>text-decoration</b> : none; }	<i>/* посещённая */</i>
<b>a:hover</b>	{ <b>text-decoration</b> : underline; }	<i>/* при наведении */</i>
<b>a:active</b>	{ <b>text-decoration</b> : underline; }	<i>/* активная */</i>

- **:any-link** – любая ссылка
- **:link** – не посещённая ссылка
- **:visited** – посещённая ссылка
- **:local-link** – на том же домене
- **:target** – на ту же страницу (на якорь)
- **:scope** – для селекторов
  
- **:hover** – мышь поверх ссылки
- **:active** – активирована пользователем
- **:focus** – элемент в фокусе
  
- **:enabled** – доступен для редактирования
- **:valid** – корректный контент
- **:required** – обязательный для заполнения

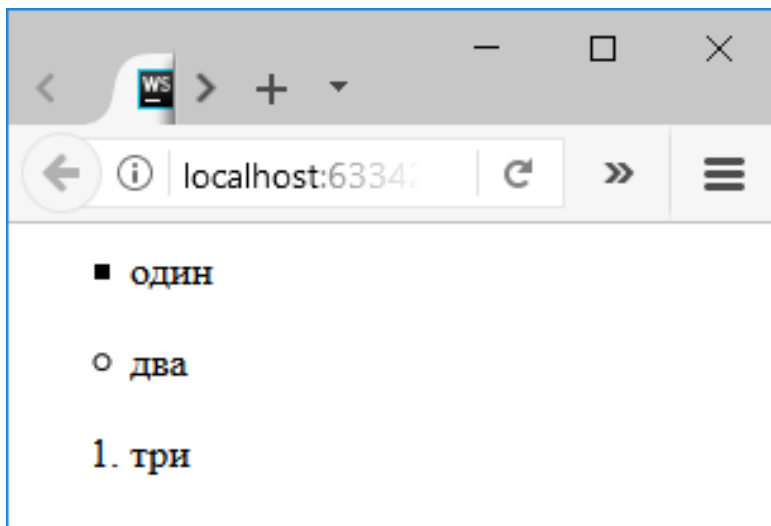
# Внешний вид курсора

```
<html>
<head>
  <meta charset='utf-8'>
  <style type='text/css'>
    div.c1 { width: 100%; height: 50px; cursor: crosshair; }
    div.c2 { width: 100%; height: 50px; cursor: wait; }
    div.c3 { width: 100%; height: 50px;
              cursor: url('../smile.png'), auto; }
  </style>
</head>
<body>
<div class='c1'><p>Курсор в виде креста</p></div>
<div class='c2'><p>Курсор - ожидание</p></div>
<div class='c3'><p>Курсор - рисунок</p></div>
</body>
</html>
```

## Доступные виды курсора:

auto | default | none | context-menu | help | pointer | progress | wait | cell |  
 crosshair | text | vertical-text | alias | copy | move | no-drop | not-allowed |  
 grab | grabbing | e-resize | n-resize | ne-resize | nw-resize | s-resize | se-resize |  
 sw-resize | w-resize | ew-resize | ns-resize | nesw-resize | nwse-resize |  
 col-resize | row-resize | all-scroll | zoom-in | zoom-out

```
<html>
<head>
  <meta charset='utf-8'>
  <style type='text/css'>
    .sq { list-style-type: square; }
    .cir { list-style-type: circle; }
    .dec { list-style-type: decimal; }
  </style>
</head>
<body>
  <ol class='sq'><li>один</li></ol>
  <ol class='cir'><li>два</li></ol>
  <ol class='dec'><li>три</li></ol>
</body>
</html>
```



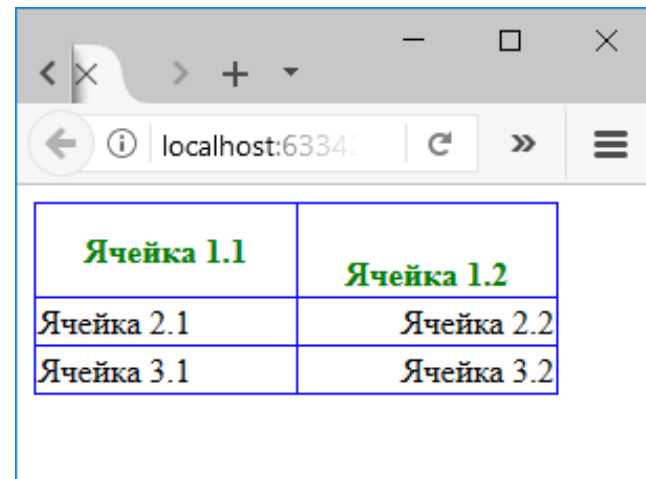
- **list-style-type**
  - **circle**
  - **square**
  - **upper-roman** – римские
  - **decimal** – арабские
  - **lower-alpha** – латиница
  - **none** – удаляет буллиты
- **list-style-position**
  - **outside** – буллит вне текста
  - **inside** – буллит в тексте
- **list-style-image**
  - **url('icon.gif')** – замена буллита
- **list-style** – краткая запись

# Оформление таблиц (1)

```

/* Границы таблицы и ячеек */
table, th, td { border: 2px solid black; }
/* Границы таблицы (БЕЗ ячеек) */
table { border: 2px solid black; }
/* Граница в одну линию */
table { border-collapse: collapse; }
/* Ширина таблицы */
table { width: 80%; }
/* Высота заголовков */
th { height: 50px; }
/* Выравнивание текста в ячейке */
td { text-align: right; }
/* Выравнивание текста в ячейке */
td { vertical-align: bottom; /* top, center */ }
/* Горизонтальные линии под строками */
th, td { border-bottom: 1px solid black; }
/* Подсвечивание строки под курсором */
tr:hover { background-color: gray; }
/* Подсвечивание чётных ячеек (отсчёт с 1) */
tr:nth-child(even) { background-color: gray; }

```



Ячейка 1.1	Ячейка 1.2
Ячейка 2.1	Ячейка 2.2
Ячейка 3.1	Ячейка 3.2

```

table, th, td {
    border: 1px solid blue;
    border-collapse: collapse;
    width: 250px; /* ширина */
}
th {
    height: 45px; /* высота */
    color: green;
}
/* снизу */
th.valign { vertical-align: bottom; }
/* правый край */
td.align { text-align: right; }

```

# Оформление таблиц (2)

```
<div style="overflow-x:auto;">  
  <table>  
    ... содержимое таблицы ...  
  </table>  
</div>
```

Имитация **responsive** – добавит горизонтальную прокрутку, если ширины экрана будет не достаточно

- **border** – задает все свойства границ в одном объявлении
- **border-collapse** – указывает, следует ли сворачивать границы таблицы
  - **separate, collapse**
- **border-spacing** – задает расстояние между границами соседних ячеек (в px...)
- **caption-side** – указывает размещение заголовка таблицы
  - **top, bottom, block-start, block-end, inline-start, inline-end**
- **empty-cells** – указывает, следует ли отображать границы и фон в пустых ячейках таблицы
  - **show, hide**
- **table-layout** – задает алгоритм компоновки, который будет использоваться для таблицы
  - **auto, fixed**



# Отображение элементов

**display: none;** */\* скрыть блок \*/*  
**display: block;** */\* "перевод каретки" \*/*  
**display: inline;** */\* в строке \*/*  
**display: inline-block;** */\* расширение строки \*/*  
**display: flex;** */\* включение модели **flex** \*/*  
**display: grid;** */\* включение модели **grid** \*/*  
**visibility: hidden;** */\* скрыт \*/*  
**visibility: visible;** */\* видим \*/*

# СВОЙСТВО POSITION

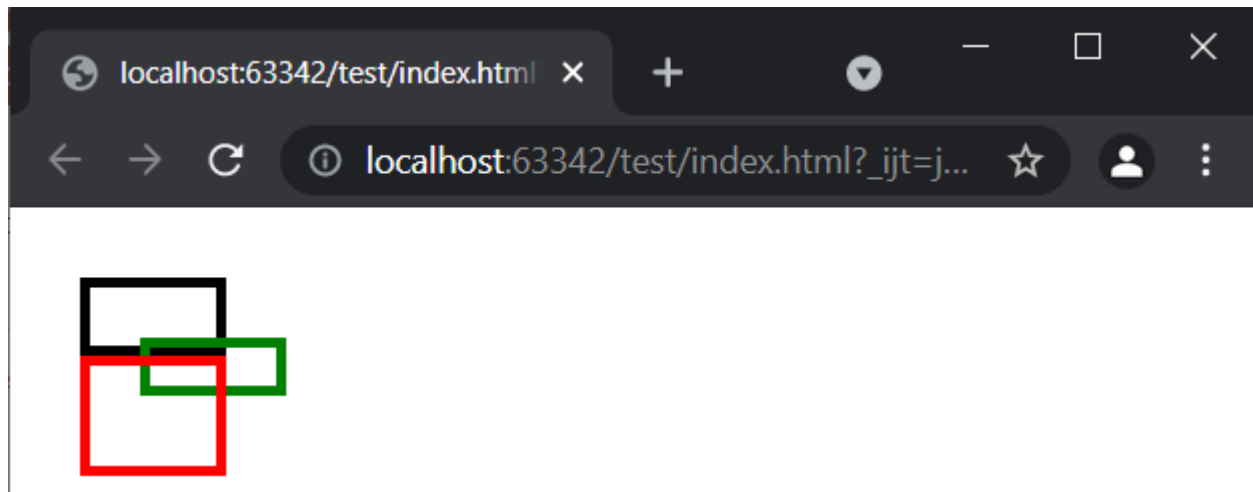
42

position	Описание
<b>absolute</b>	Абсолютное позиционирование. Другие элементы отображаются на веб-странице так, словно этого абсолютного позиционирования на странице нет. Если у элемента стоит <code>position: absolute</code> , то мы также можем управлять его местоположение при помощи каких свойств как: <code>left</code> , <code>top</code> , <code>right</code> и <code>bottom</code> .
<b>fixed</b>	Фиксированное позиционирование. В данном случае мы можем также управлять местоположением при помощи <code>left</code> , <code>top</code> , <code>right</code> и <code>bottom</code> , но при этом он свое положение на странице не будет менять при скролле страницы.
<b>relative</b>	Относительное позиционирование. В данном случае положение элементов устанавливается относительно его исходного места, при этом если мы добавим такие свойства как: <code>left</code> , <code>top</code> , <code>right</code> и <code>bottom</code> , то мы изменим позицию элемента и сдвинем его в ту или иную сторону в зависимости от значений.
<b>static</b>	Статичное позиционирование. В этом случае элементы отображаются, как обычно. Использование таких свойств как: <code>left</code> , <code>top</code> , <code>right</code> и <code>bottom</code> не приводит ни каким результатам.
<b>sticky</b>	Сочетание относительного и фиксированного позиционирования.

# Пример для изучения position

43

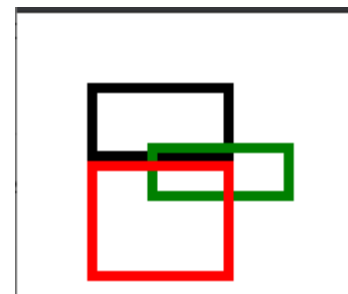
```
<html>
<head>
  <style>
    .pos {
      position: relative;
      top: 20px;
      left: 20px;
      width: 50px;
      border: 4px solid black;
    }
  </style>
</head>
<body>
<div class="pos">
  <div class="pos" style="border-color:green; height: 15px;"></div>
</div>
<div class="pos" style="height: 40px; border-color: red;"></div>
</body>
```



# Варианты position

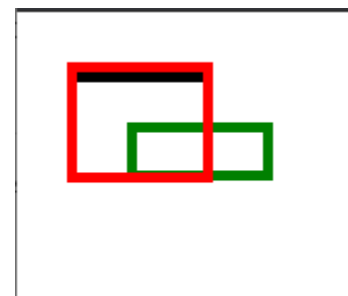
## relative

- родитель (Ч) изменил размер
- (З) сместился на указанное расстояние
- (З) полностью «поместился» в (Ч) родителя



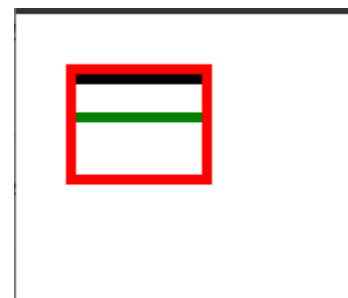
## absolute

- (Ч) родитель не изменил размер
- (З) сместился относительно родителя
- другие (Ч), (К) сместились относительно угла body



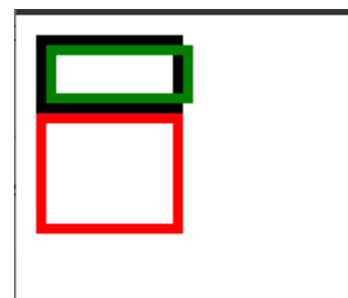
## fixed

- (Ч) родитель не изменил размер
- (З) сместился относительно окна, а не родителя (все три блока друг поверх друга)



## static

- (Ч) родитель изменил размер по высоте
- смещение отсутствует



**Легенда:** Ч – черный, З – зеленый, К – красный

# Переполнение (overflow)

45

*/\* объект отображается за пределами блока \*/*

**overflow: visible;**

*/\* объект за пределами блока невидим \*/*

**overflow: hidden;**

*/\* добавляется прокрутка \*/*

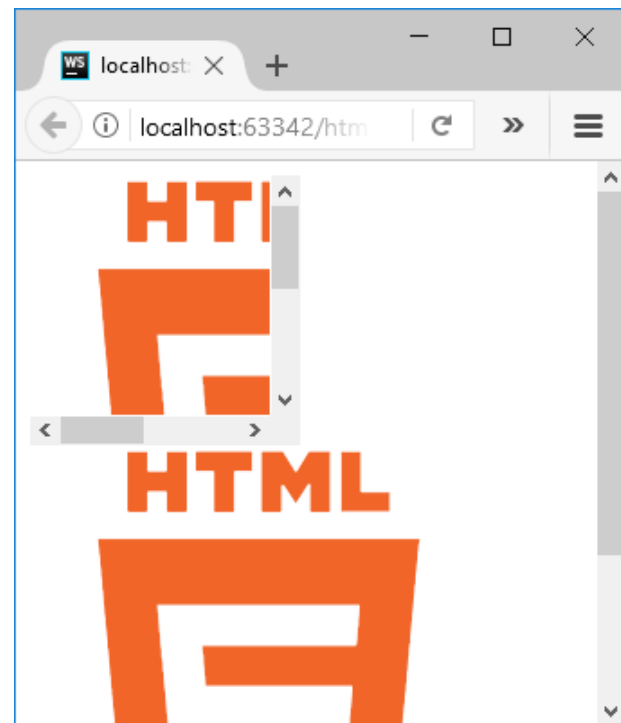
**overflow: scroll;**

*/\* добавляет прокрутку по необходимости \*/*

**overflow: auto;**

---

```
<html>
<head>
  <style type='text/css'>
    div { width: 150px; height: 150px; }
    div.s { overflow: scroll; }
    div.v { overflow: visible; }
  </style>
</head>
<body>
<div class='s'><img src='../html5.png'></div>
<div class='v'><img src='../html5.png'></div>
</body>
</html>
```

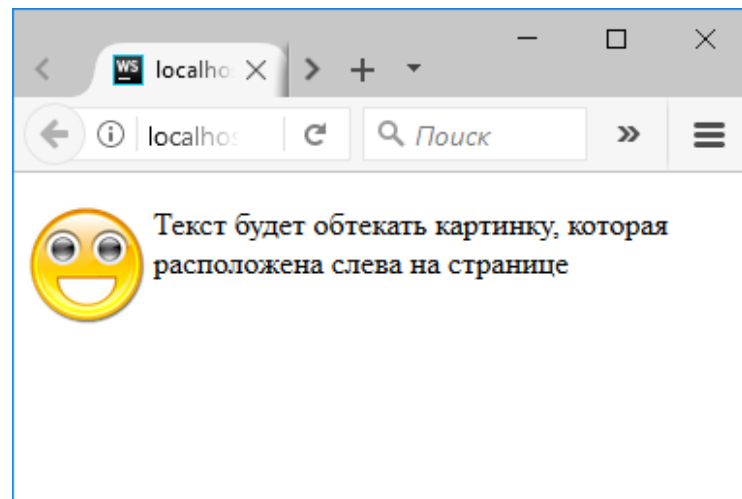
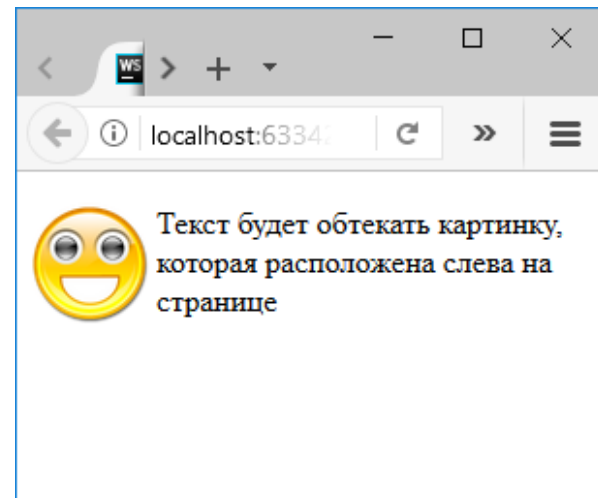


# Обтекание (float)

46

```
<html>
<head>
  <meta charset='utf-8'>
  <style type='text/css'>
    img {
      float: left; /* ВОЗМОЖНЫ right, none */
    }
  </style>
</head>
<body>

<p>Текст будет обтекать картинку, которая
  расположена слева на странице</p>
</body>
</html>
```



Свойство «clear: left;» отключает обтекание «left»

# Псевдоэлементы

*/\* Первый символ \*/*

**p::first-letter** {

*/\* Первая строка \*/*

**p::first-line** {

*/\* До элемента\*/*

**h3::before** { **content**: url(image.gif); }

*/\* После элемента\*/*

**h3::after** { **content**: url(image.gif); }

*/\* Изменяет стиль для списков \*/*

**::marker** { **color**: green; }

*/\* Изменение выделенного пользователем текста \*/*

**::selection** {

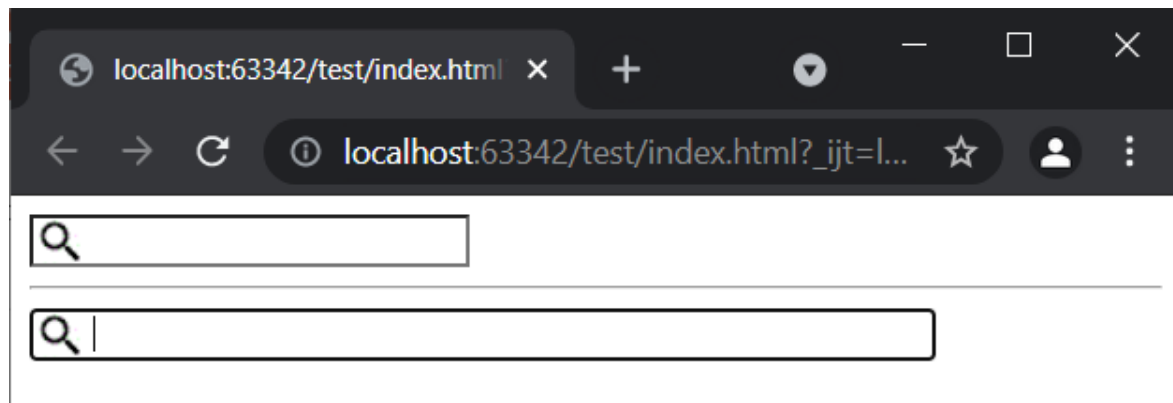
**color**: white;

**background**: black;

}

# Оформление input

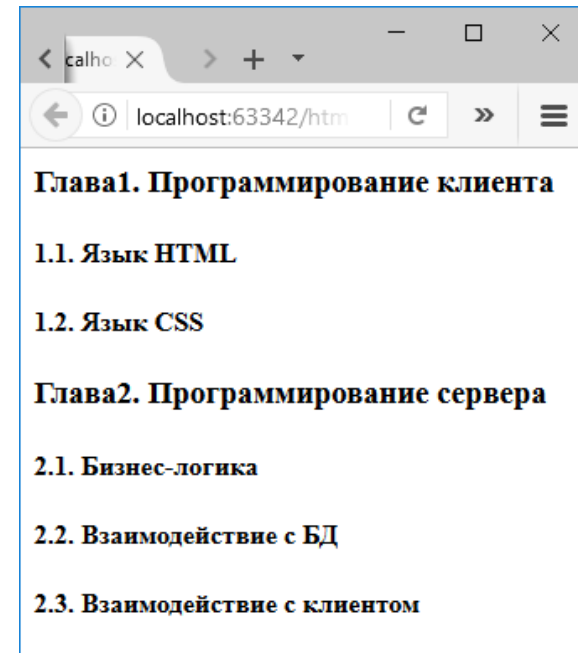
```
<html>
<head>
  <style>
    input[type=text] {
      background: white 1px 0px url('search.gif') no-repeat;
      padding-left: 25px;
    }
    input.animate {
      transition: width 1s ease-in-out;
    }
    input[type=text].animate:focus {
      width: 80%;
    }
  </style>
</head>
<body>
  <input type="text">
  <hr>
  <input type="text" class="animate">
</body>
</html>
```





```
<html>
<head>
  <meta charset='utf-8'>
  <style type='text/css'>
    body { counter-reset: sect; } /* sect = 0 */
    h3 { counter-reset: subsect; } /* сбросить subsect */
    h3:before {
      counter-increment: sect; /* увеличить sect */
      content: "Глава" counter(sect) ". "; /* дописать */
    }
    h4:before {
      counter-increment: subsect; /* увеличить subsect */
      content: counter(sect) "." counter(subsect) ". ";
    }
  </style>
</head>
<body> <!-- нумерация будет добавлена автоматически -->
<h3>Программирование клиента</h3>
<h4>Язык HTML</h4>
<h4>Язык CSS</h4>
<h3>Программирование сервера</h3>
<h4>Бизнес-логика</h4>
<h4>Взаимодействие с БД</h4>
<h4>Взаимодействие с клиентом</h4>
</body>
</html>
```

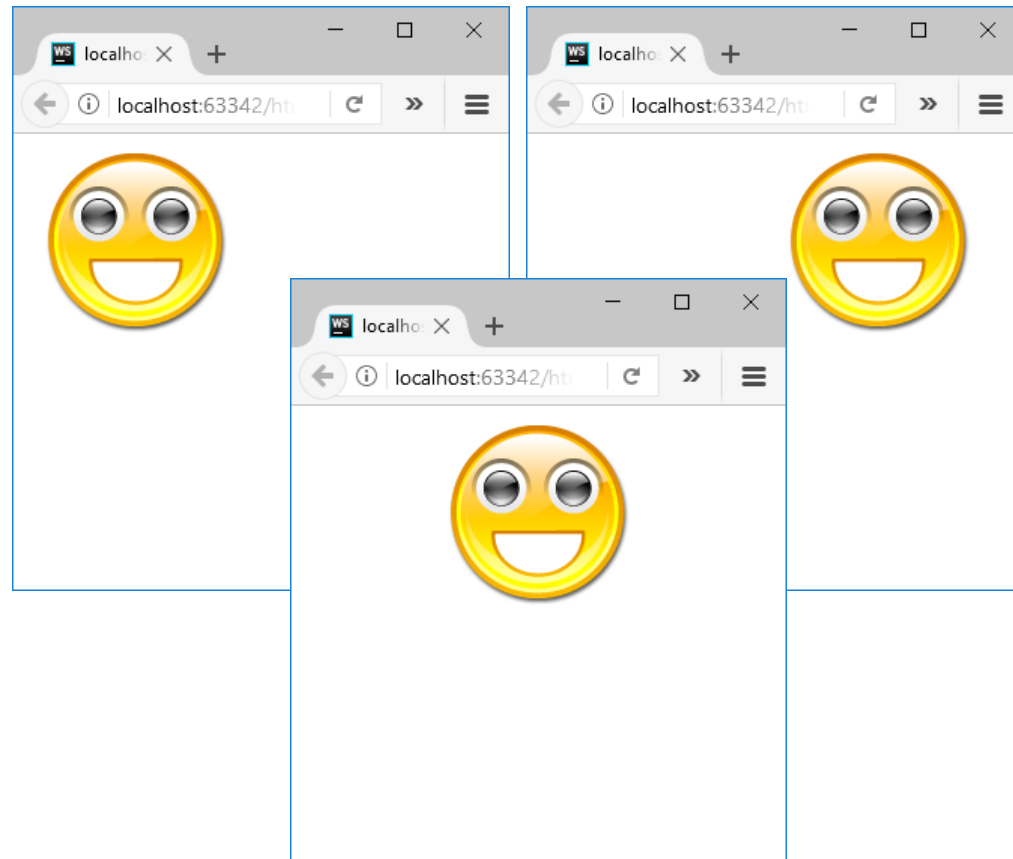
- **counter-reset** – сбрасывает счётчик
- **counter-increment** – увеличивает счётчик
- **content** – добавляет содержимое



# Простейшая анимация

50

```
<html>
<head>
  <style type='text/css'>
    img {
      position: relative;
      animation:mymove 5s infinite;
    }
    @keyframes mymove {
      from { left: 0px; }
      to { left: 200px; }
    }
  </style>
</head>
<body>
  
</body>
</html>
```

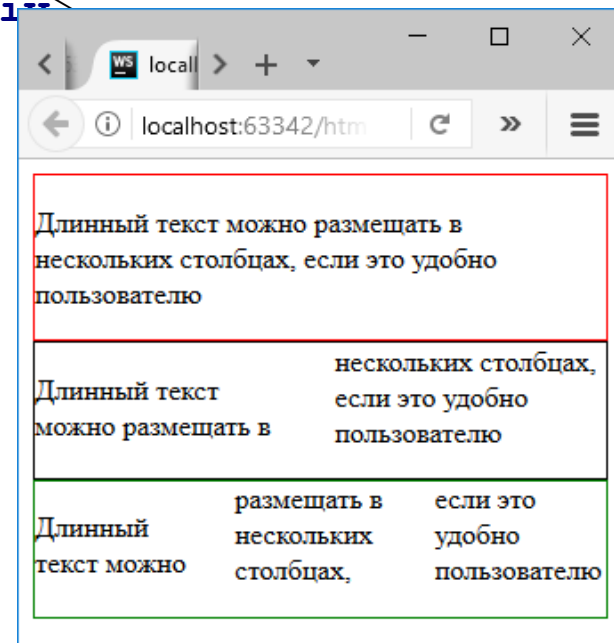


# Столбцы

51

```
<html>
<head>
  <meta charset='utf-8'>
  <style type='text/css'>
    div.c1 { border: 1px solid red; column-count: 1; }
    div.c2 { border: 1px solid black; column-count: 2; }
    div.c3 { border: 1px solid green; column-count: 3; }
  </style>
</head>
<body>
<div class='c1'><p>Длинный текст можно размещать в нескольких
столбцах, если это удобно пользователю</p></div>
<div class='c2'><p>Длинный текст можно размещать
если это удобно пользователю</p></div>
<div class='c3'><p>Длинный текст можно размещать
если это удобно пользователю</p></div>
</body>
</html>
```

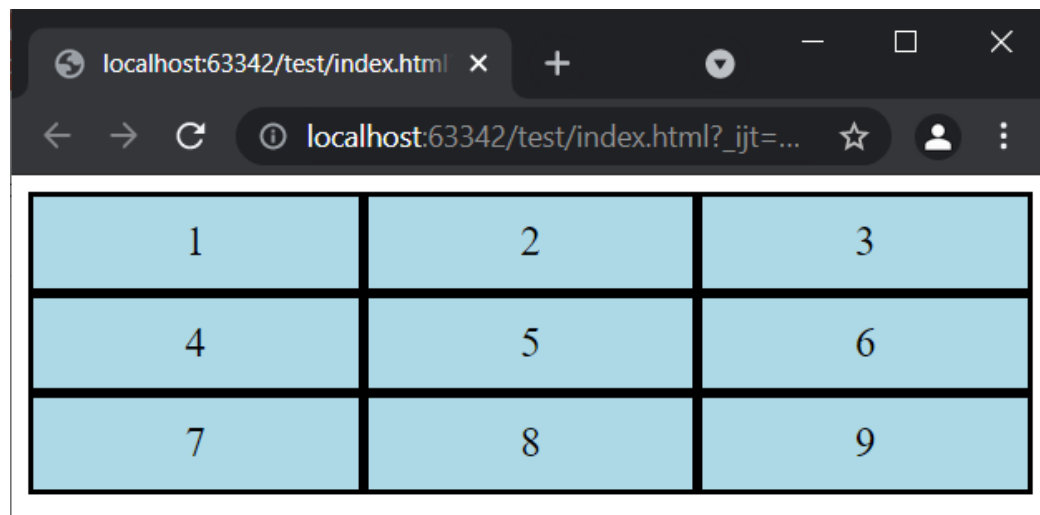
- **column-gap** – зазор между колонками
- **column-rule-style** – вид разделителя
- **column-rule-width** – толщина разделителя
- **column-rule-color** – цвет разделителя
- **column-rule** – разделитель



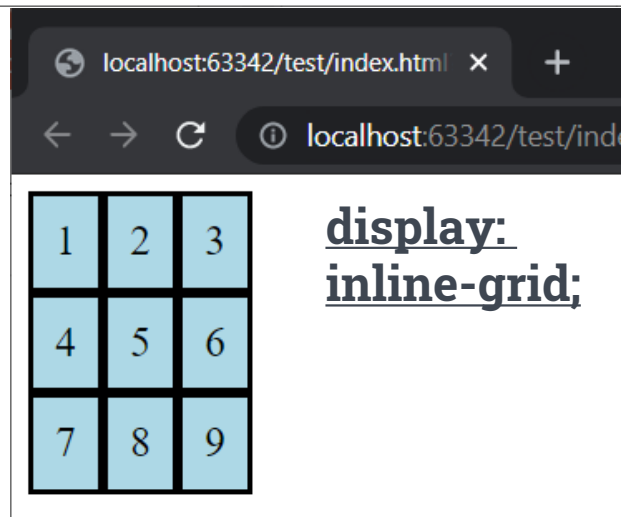
# Решётка (grid)

```
<html>
<head>
  <style>
    .container {
      display: grid;
      grid-template-columns: auto auto auto;
      background-color: lightblue;
    }
    .item {
      border: 3px solid black;
      padding: 10px;
      font-size: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">1</div><div class="item">2</div>
    <div class="item">3</div><div class="item">4</div>
    <div class="item">5</div><div class="item">6</div>
    <div class="item">7</div><div class="item">8</div>
    <div class="item">9</div>
  </div>
</body>
</html>
```

display: grid;



display:  
inline-grid;



# Свойства решёток (1) <sup>53</sup>

```
div { /* разные варианты свойств */
  /* расстояние между столбцами */
  grid-column-gap: 50px;
  /* расстояние между строками */
  grid-row-gap: 50px;
  /* расстояния строки-столбцы */
  grid-gap: 50px 100px;
  /* расстояния строки-столбцы */
  grid-gap: 50px;
}
div { /* растягивание одного элемента в grid */
  grid-column-start: 1;
  grid-column-end: 3;
}
div { /* растягивание одного элемента в grid */
  grid-row-start: 1;
  grid-row-end: 3;
}
div.container { /* пример grid */
  display: grid;
  grid-template-rows: 6em 250px 250px; /* 3 строки */
  grid-template-columns: 300px 6em 50%; /* 3 столбца */
}
.grid-container { /* пример grid */
  display: grid;
  /* эквивалентно grid-template-columns: 50% 25% 25%; */
  grid-template-columns: 2fr 1fr 1fr;
}
```

Новая единица  
длины **fr**  
представляет  
собой долю  
(**fraction**)  
доступного  
пространства в  
решётке (grid-  
контейнере)

# Свойства решёток (2)

```
div { /* разные варианты свойств */
  /* размеры дорожек */
  grid-template-rows: 300px minmax(150px, 1fr);
  /* вычисляемая ширина столбца-строки */
  grid-template-columns: auto 1fr;
  /* задание ограничения на макс. длину контента */
  grid-template-columns: fit-content(40%) fit-content(200px) 1fr;
  /* создание нескольких одинаковых строк-столбцов */
  grid-template-rows: repeat(4, 250px);
}

.grid-container { /* именованные области */
  display: grid;
  grid-template-areas: "header header"
    "sidebar content"
    "sidebar content";
  grid-template-columns: 250px 1fr;
  grid-template-rows: 50px 1fr 1fr;
}

myHeader { /* для моего header */
  grid-area: header; /* +другие свойства */
}

myAside { /* для моего sidebar */
  grid-area: sidebar; /* +другие свойства */
}

myContent { /* для моего content */
  grid-area: content; /* +другие свойства */
}
```

## Полный список свойств:

grid, column-gap, gap, grid-area,  
grid-auto-columns,  
grid-auto-flow, grid-auto-rows,  
grid-column, grid-column-end,  
grid-column-gap,  
grid-column-start, grid-gap,  
grid-row, grid-row-end,  
grid-row-gap, grid-row-start,  
grid-template,  
grid-template-areas,  
grid-template-columns,  
grid-template-rows, row-gap

# Flex-контейнер (flexbox)

55

```
<html>
<head>
  <style>
    .container {
      display: flex;
      background-color: blue;
    }
    .container > div {
      background-color: lightgray;
      margin: 10px;
      padding: 20px;
      font-size: 30px;
    }
  </style>
</head>
<body>
<div class="container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</div>
</body>
</html>
```

Flex-контейнер **не является блочным контейнером**, для его дочерних элементов не работают **float, clear, vertical-align**, не оказывают влияние свойства **column-\***, создающие колонки в тексте и псевдоэлементы **::first-line** и **::first-letter**.



«**display: inline-flex;**» позволит размещать элементы справа

# Свойства flex (1)

```
.container { /* выстроить в столбец */
  display: flex;
  flex-direction: column; /* row */
}

.container { /* в столбец и в обратном порядке */
  display: flex;
  flex-direction: column-reverse; /* row-reverse */
}

.container { /* переход на другую строку */
  display: flex;
  flex-wrap: wrap; /* nowrap, wrap-reverse */
}

.container { /* сокращенная запись */
  display: flex;
  flex-flow: row wrap;
}

.container { /* по центру контейнера (по горизонтали) */
  display: flex;
  justify-content: center;
  /* flex-start, flex-end, space-around, space-between */
}

.container { /* выравнивание по вертикали */
  display: flex;
  height: 500px;
  align-items: center; /* flex-start, flex-end, stretch, baseline */
}
```

## align-content:

- space-between
- space-around
- stretch
- center
- flex-start
- flex-end

Устанавливает распределение пространства между и вокруг элементами контента вдоль поперечной оси



# Свойства flex (2), order

57

```
<html>
<head>
  <style>
    .container {
      display: flex;
      background-color: blue;
    }
    .container > div {
      background-color: lightgray;
      margin: 10px;
      padding: 20px;
      font-size: 30px;
    }
  </style>
</head>
<body>
<div class="container">
  <div style="order: 4">1</div>
  <div style="order: 2">2</div>
  <div style="order: 3">3</div>
  <div style="order: 1">4</div>
</div>
</body>
</html>
```

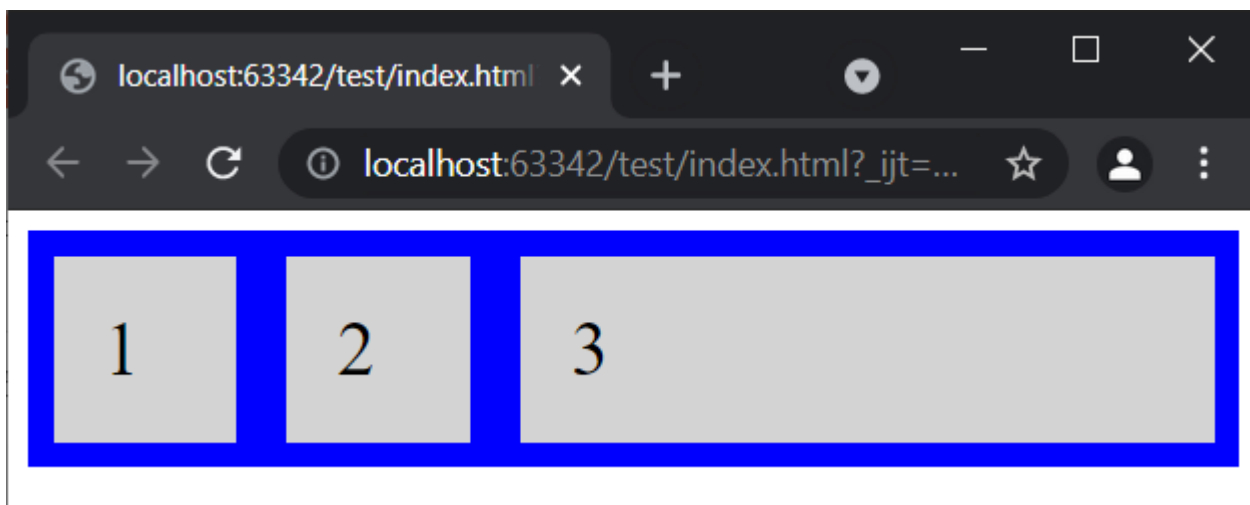
Задаем порядок



# Свойства flex (3), flex-grow

```
<div class="container">  
  <div style="flex-grow: 1">1</div>  
  <div style="flex-grow: 1">2</div>  
  <div style="flex-grow: 12">3</div>  
</div>
```

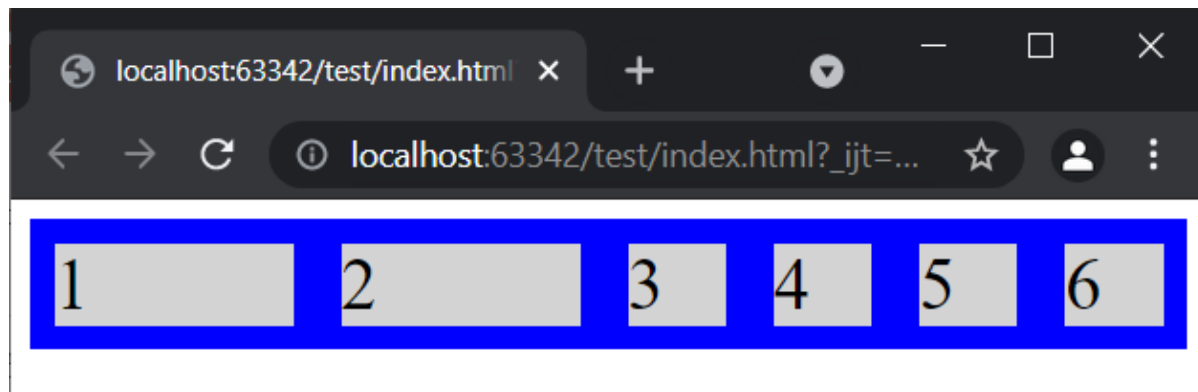
Определяем пропорции



# Свойства flex (4), flex-shrink

```
<html>
<head>
  <style>
    .container {
      display: flex;
      align-items: stretch;
      background-color: blue;
    }
    .container > div {
      background-color: lightgray;
      margin: 10px;
      width: 100px;
      font-size: 30px;
    }
  </style>
</head>
<body>
<div class="container">
  <div style="flex-shrink: 0">1</div>
  <div style="flex-shrink: 0">2</div>
  <div>3</div><div>4</div>
  <div>5</div><div>6</div>
</div>
</body>
</html>
```

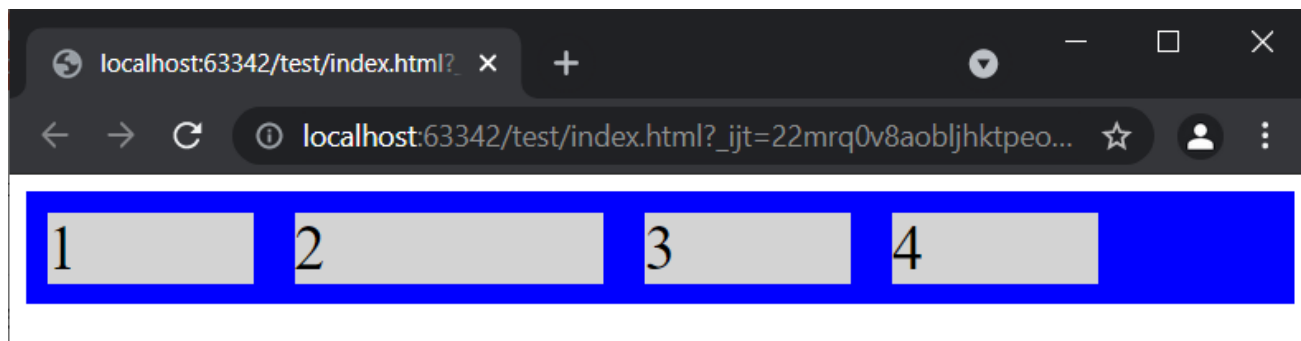
Запрещаем сильно сжиматься



# Свойства flex (5), flex-basis

```
<html>
<head>
  <style>
    .container {
      display: flex;
      align-items: stretch;
      background-color: blue;
    }
    .container > div {
      background-color: lightgray;
      margin: 10px;
      width: 100px;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <div class="container">
    <div>1</div>
    <div style="flex-basis: 150px">2</div>
    <div>3</div><div>4</div>
  </div>
</body>
</html>
```

Задаем начальный размер



# Свойства flex (6), align-self

```
<html>
<head>
  <style>
    .container {
      display: flex;
      height: 100px;
      background-color: blue;
    }
    .container > div {
      background-color: lightgray;
      margin: 10px;
      width: 100px;
      font-size: 30px;
    }
  </style>
</head>
<body>
<div class="container">
  <div>1</div>
  <div style="align-self: center;">2</div>
  <div>3</div><div>4</div>
</div>
</body>
</html>
```

Своя высота



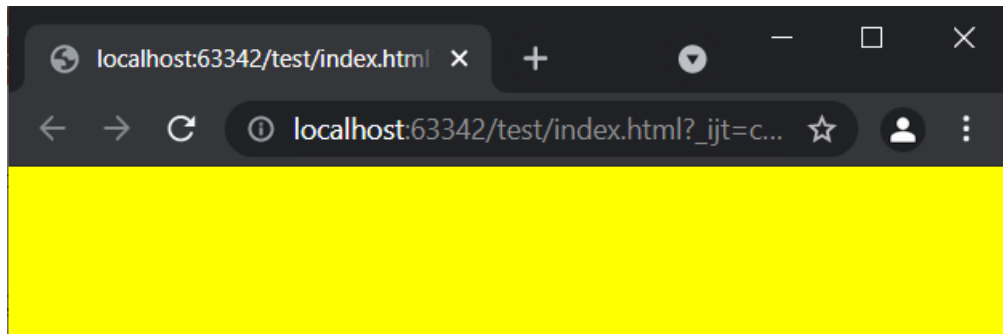
align-self:

- center
- flex-start
- flex-end

# Переменные в CSS

62

```
<html>
<head>
  <style>
    :root { /* переменная в css */
      --main-bg-color: brown;
    }
    body { /* использование переменной */
      background-color: var(--main-bg-color);
    }
  </style>
</head>
<body>
<script>
  document.querySelector(':root').style.setProperty('--main-bg-color', 'yellow');
</script>
</body>
</html>
```

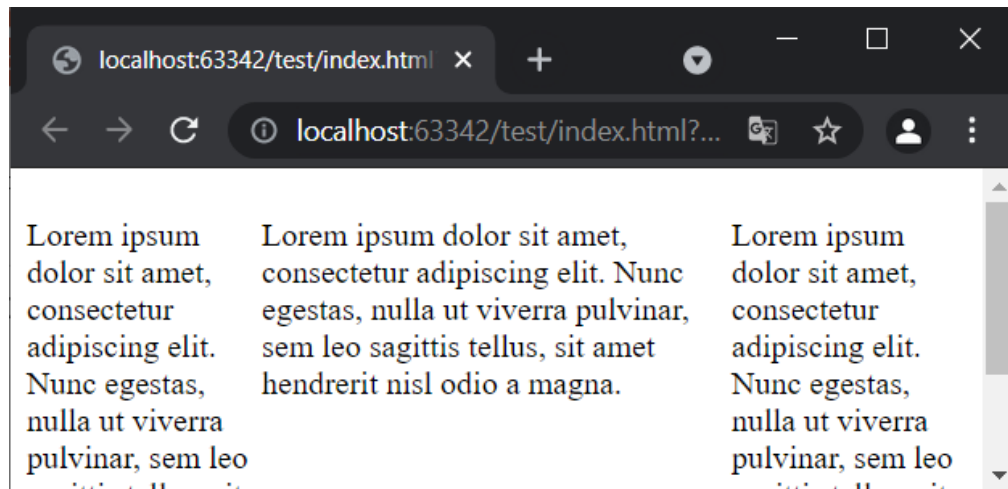


Можно задать переменные для любого селектора

# Responsive элементы

63

```
<html>
<head>
  <style>
    .column { float: left; }
    .column.side { width: 25%; }
    .column.middle { width: 50%; }
    /* Responsive layout */
    @media screen and (max-width: 400px) {
      .column.side, .column.middle {
        width: 100%;
      }
    }
  </style>
</head>
<body>
  <p class="column side">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc egestas,
  <p class="column middle">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc egestas,
  <p class="column side">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc egestas,
</body>
</html>
```



Могут контролировать

- ширину и высоту окна
- ширину и высоту устройства
- ориентацию (планшета/телефона)
- разрешение

## Типы

- all
- print
- screen
- speech

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

```
}  
@media only screen and (orientation: landscape) {  
  div {
```

```
}  
@media screen and (max-width: 800px) and (min-width: 400px), (min-width: 1000px) {  
  p {  
    div {  
  
}
```



# media (повтор)

65

any-hover	any-pointer	aspect-ratio (min, max)	color
color-gamut	color-index	display-mode	forced-colors
grid	height (min, max)	hover	inverted- colors
monochrome	orientation	overflow- block	overflow- inline
pointer	prefers-color- scheme	prefers- reduced- motion	resolution (min, max)
scripting	update	width (min, max)	

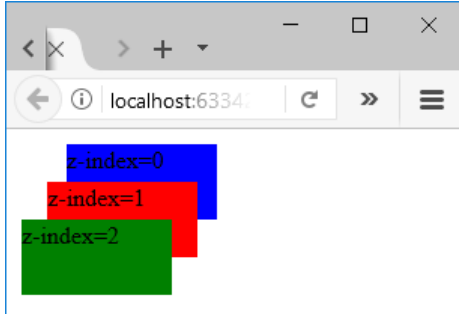
## Используются характеристики:

1. браузера (user agent)
2. устройства вывода (output device)
3. среды (environment)

## Поддерживаются операторы:

- and
- not
- only
- , (запятая)

# z-index



```
<html>
<head>
  <style type="text/css">
    div {
      width: 100px;
      height: 50px;
      position: absolute;
    }
    .first {
      background-color: red;
      left: 27px;
      top: 35px;
      z-index: 1; /* порядок позиционирования */
    }
    .second {
      background-color: blue;
      left: 40px;
      top: 10px;
      z-index: 0; /* порядок позиционирования */
    }

    .third {
      background-color: green;
      left: 10px;
      top: 60px;
      z-index: 2; /* порядок позиционирования */
    }
  </style>
</head>
<body>
  <div class="first">z-index=1</div>
  <div class="second">z-index=0</div>
  <div class="third">z-index=2</div>
</body>
</html>
```

- Примеры фильтров

```
div { filter: grayscale(100%); }
```

```
div { filter: grayscale(50%); }
```

```
div { filter: grayscale(100%) sepia(100%); }
```



# Доступные фильтры в CSS

- **grayscale** (значение 0% — 100%)
- **sepia** (значение 0% — 100%)
- **saturate** (значение 0% — 100%) — насыщенность цвета
- **hue-rotate** (угол 0deg — 360deg)
- **invert** (значение 0% — 100%)
- **opacity** (значение 0% — 100%)
- **brightness** (значение 0% — 100%)
- **contrast** (значение 0% — 100%)
- **blur** (радиус в px) — размытость
- **drop-shadow** (x, y, радиус, цвет) — тень

## CSS

```
p {  
  position: absolute;  
  font-family: sans-serif;  
  background-color: black;  
}
```

## JS

```
e.style.position = "absolute"  
e.style.fontFamily = "sans-serif"  
e.style.backgroundColor = "black"
```

Так называемый  
camelCase

## Особенности применения JS

```
e.style.left = 400 // Неправильно: это число  
e.style.left = "400" // Неправильно: нет единиц измерения  
e.style.left = "400px" // Правильно
```

## Обработка style как текста в JS

```
// Запись в атрибут style (два способа)  
e.setAttribute("style", myStyle);  
e.style.cssText = myStyle;  
// Чтение атрибута style (два способа)  
myStyle = e.getAttribute("style");  
myStyle = e.style.cssText;
```

```
let style = window.getComputedStyle(element)
```

- Доступен только для чтения
- Имеет абсолютное значение
  - размеры в формате “#px”
  - цвет в формате “rgb(##,##,##)” или “rgba(##,##,##,##)”
- Свойства, являющиеся краткой формой записи, не вычисляются
  - вместо **margin** проверяйте **marginLeft**, **marginRight**...
- Свойство **cssText** вычисленного стиля не определено

<https://developer.mozilla.org/ru/docs/Web/API/Window/getComputedStyle>

# Работа со стилями

71

## Таблица стилей document.styleSheets

*// Получение правил (из 1-й таблицы 1-е правило)*

**let firstRule = document.styleSheets[0].cssRules[0]**

*// Запись правил (в 1-ю таблицу в 1-ю позицию)*

**document.styleSheets[0].insertRule("p { color: red; }", 0)**

## Работа с классами из JavaScript

*// добавить класс*

**element.classList.add("className")**

*// удалить класс*

**element.classList.remove("className")**

*// добавить класс, если его нет, иначе удалить*

**element.classList.toggle("className")**

*// проверка наличия класса, возвращает true/false*

**element.classList.contains("className")**

# W3.CSS -

<https://www.w3schools.com/w3css/>

The screenshot shows a web browser window with the address bar displaying 'www.w3schools.com'. The page title is 'W3.CSS Home'. The navigation bar includes links for 'Tutorials', 'References', 'Exercises', 'Spaces' (highlighted in pink), 'Paid Courses', and a 'Log in' button. Below the navigation bar, there is a dark blue menu with links for 'HTML', 'CSS', 'JAVASCRIPT', 'SQL', 'PYTHON', 'PHP', and 'BOOTSTRAP'. A Grammarly advertisement for a 'Free Writing Assistant' is visible. The main heading is 'W3.CSS Tutorial', with '< Home' and 'Next >' buttons. The subheading is 'A Quality Alternative to Bootstrap', followed by the text 'W3.CSS is Smaller, Faster and Easier to Use.' Below this, a green box contains the 'W3 CSS' logo, and a light gray box lists four features of W3.CSS: it is a modern, responsive, mobile first CSS framework; it provides equality for all browsers (Chrome, Firefox, Edge, IE, Safari, Opera); it provides equality for all devices (Desktop, Laptop, Tablet, Mobile); and it is standard CSS only (No jQuery or JavaScript library).

W3.CSS Home

www.w3schools.com

Tutorials References Exercises Spaces Paid Courses Log in

HTML CSS JAVASCRIPT SQL PYTHON PHP BOOTSTRAP

Grammarly  
Free Writing Assistant  
LEARN MORE

## W3.CSS Tutorial

< Home Next >

### A Quality Alternative to Bootstrap

W3.CSS is Smaller, Faster and Easier to Use.

**W3 CSS**

- W3.CSS is a modern, responsive, mobile first CSS framework.
- W3.CSS provides equality for all browsers: Chrome. Firefox. Edge. IE. Safari. Opera.
- W3.CSS provides equality for all devices: Desktop. Laptop. Tablet. Mobile.
- W3.CSS is standard CSS only (No jQuery or JavaScript library).

<https://www.w3schools.com/w3css/4/w3.css>

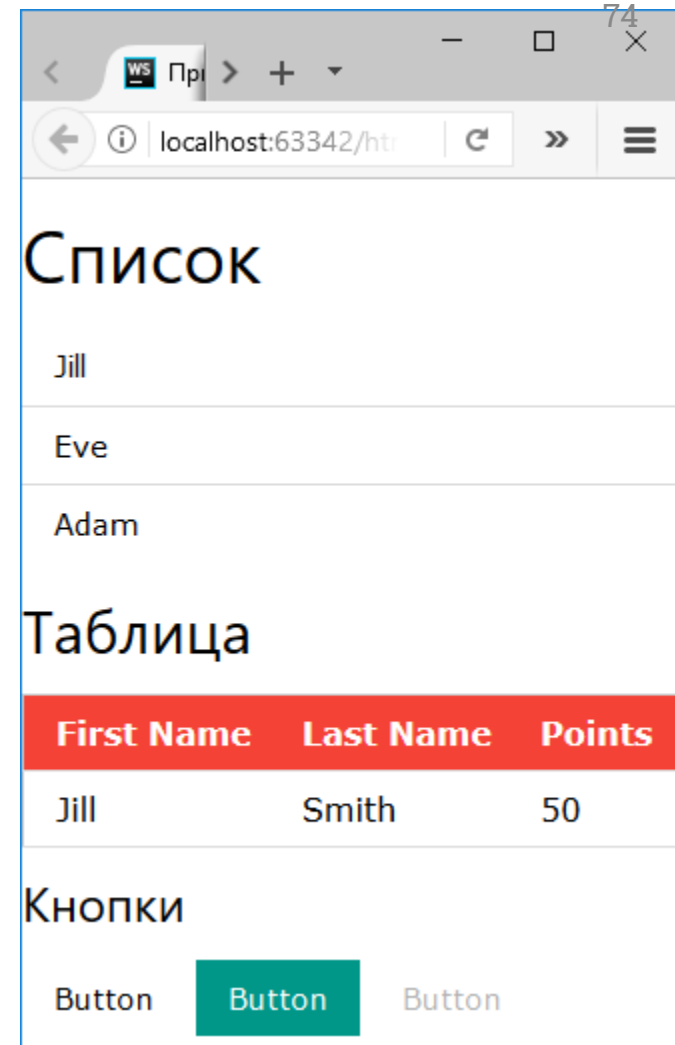


# Особенности W3.CSS

- Меньше и быстрее, чем остальные фреймворки CSS
- Проще для изучения
- Использует только стандартный CSS (без jQuery или библиотек)
- Ускоряет и упрощает веб-разработку
- Поддерживает современный дизайн, учитывающий размер окна
  - ориентирован в т.ч. на мобильного пользователя
- Обеспечивает совместимость со всеми браузерами
  - Chrome, Firefox, Edge, Safari и другие
- Обеспечивает одинаковое отображение для всех устройств
  - компьютеры, ноутбуки, планшеты, телефоны

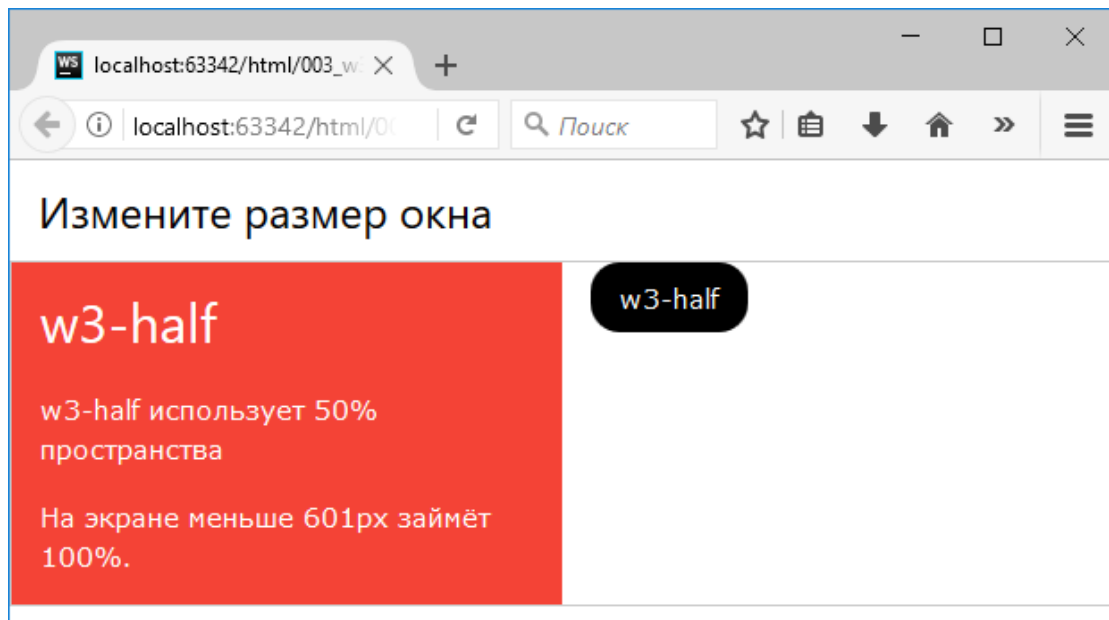
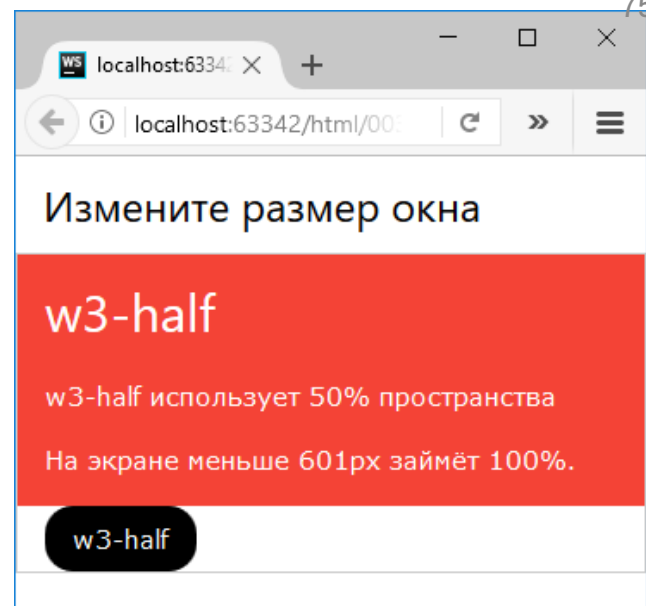
# Пример w3.css (1)

```
<h1>Список</h1>
<ul class="w3-ul">
  <li class="w3-hover-red">Jill</li>
  <li class="w3-hover-blue">Eve</li>
  <li class="w3-hover-green">Adam</li>
</ul>
<h2>Таблица</h2>
<table class="w3-table-all">
  <thead>
    <tr class="w3-red">
      <th>First Name</th>
      <th>Last Name</th>
      <th>Points</th>
    </tr>
  </thead>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
</table>
<h3>Кнопки</h3>
<div class="w3-show-inline-block">
  <div class="w3-bar">
    <button class="w3-btn">Button</button>
    <button class="w3-btn w3-teal">Button</button>
    <button class="w3-btn w3-disabled">Button</button>
  </div>
</div>
```



# Пример w3.css (2)

```
<html>
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="w3.css">
<body>
<div class="w3-container">
  <h3>Измените размер окна</h3>
</div>
<div class="w3-row w3-border">
  <div class="w3-container w3-half w3-red">
    <h2>w3-half</h2>
    <p>w3-half использует 50% пространства</p>
    <p>На экране меньше 601px займёт 100%.</p>
  </div>
  <div class="w3-container w3-half">
    <button class="w3-button w3-round-xlarge w3-black w3-hover-cyan">
      w3-half
    </button>
  </div>
</div>
</body>
</html>
```



# Расширение CSS с помощью LESS

76

Переменные

Вложенные  
блоки

Примеси  
(миксины)

Операторы

Maps

Scope

Import

...

# LESS

<https://lesscss.org/>  
<https://lesscss.org/less-preview/>

77

Альтернатива для  
работы с LESS, SASS  
и др.

<http://koala-app.com/>

```
1 #lib() {  
2   .colors() {  
3     @primary: blue;  
4     @secondary: green;  
5   }  
6   .rules(@size) {  
7     border: @size solid white;  
8   }  
9 }  
10  
11 .box when (#lib.colors[@primary] = blue) {  
12   width: 100px;  
13   height: ($width / 2);  
14 }  
15  
16 .bar:extend(.box) {  
17   @media (min-width: 600px) {  
18     width: 200px;  
19     #lib.rules(1px);  
20   }  
21 }
```

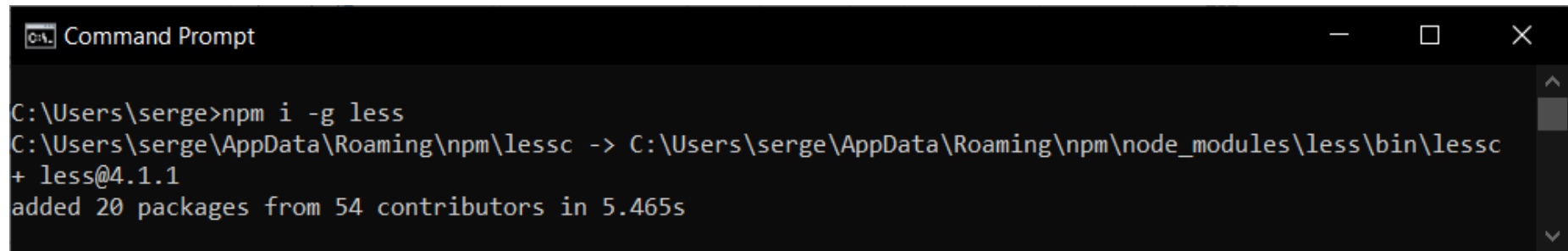
```
1 .box,  
2 .bar {  
3   width: 100px;  
4   height: 50px;  
5 }  
6 @media (min-width: 600px) {  
7   .bar {  
8     width: 200px;  
9     border: 1px solid white;  
10  }  
11 }  
12
```



It's CSS, with just a little more.

# npm i -g less

78



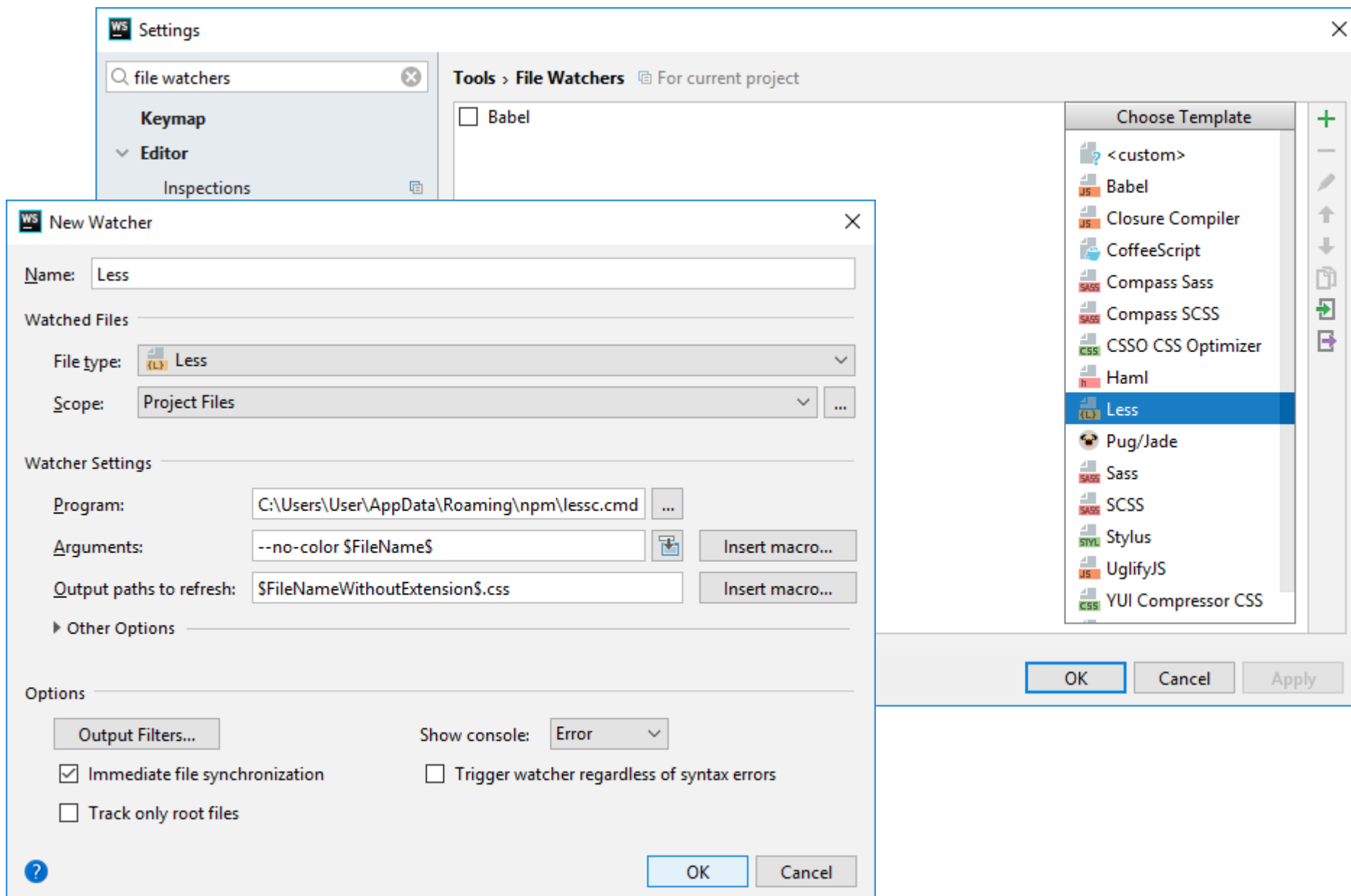
```
Command Prompt
C:\Users\serge>npm i -g less
C:\Users\serge\AppData\Roaming\npm\lessc -> C:\Users\serge\AppData\Roaming\npm\node_modules\less\bin\lessc
+ less@4.1.1
added 20 packages from 54 contributors in 5.465s
```

Запомните путь установки, он понадобится для настройки среды выполнения:

**C:\Users\serge\AppData\Roaming\npm\lessc.cmd**

# Настройка WebStorm

79



# LESS. Переменные

## LESS

```
@width: 10px;  
@height: @width + 10px;  
#header {  
  width: @width;  
  height: @height;  
}
```

## CSS

```
#header {  
  width: 10px;  
  height: 20px;  
}
```



# LESS. Примеси (миксины)

## LESS

```
.a (@x: 20px) {  
  color: red;  
  width: @x;  
}  
  
.mixin-class {  
  .a();  
}  
  
.mixin-id {  
  .a(100px);  
}
```

## CSS

```
.mixin-class {  
  color: red;  
  width: 20px;  
}  
  
.mixin-id {  
  color: red;  
  width: 100px;  
}
```

Миксины — это простой способ присоединения свойств из одного набора правил в другой

- Без параметров: `.mixin { стили }`
- С параметрами: `.mixin(@param) {свойство: @param}`

# LESS. Вложенные правила

82

## LESS

```
#header {  
  background: lightblue;  
  a {  
    color: blue;  
    &:hover {  
      color: green;  
    }  
  }  
}
```

## CSS

```
#header {  
  background: lightblue;  
}  
#header a {  
  color: blue;  
}  
#header a:hover {  
  color: green;  
}
```

# LESS. Использование операций (арифметика, логика)

## LESS

```
.average(@x, @y) {  
  @Average: ((@x + @y) / 2);  
}
```

```
div {  
  .average(12px, 10px); // вызов миксина  
  padding: @Average;    // использование результата  
}
```

## CSS

```
div {  
  padding: 11px;  
}
```

# LESS. Использование функций

## LESS

**@some: black;**

```
div {  
  margin: if((2 > 1), 0px, 3px);  
  color: if((iscolor(@some)), @some, black);  
}
```

## CSS

```
div {  
  margin: 0px;  
  color: black;  
}
```

## Функции:

- if
- escape
- replace
- length
- extract
- range
- each
- ceil
- isnumber
- iscolor
- isurl
- color
- ...

# Расширение CSS с помощью SASS

85

Переменные

Вложенности

Фрагментирование

Импорт

Миксины  
(примеси)

Наследование

Математические  
операторы

# SASS

<https://sass-scss.ru/>  
<https://sass.js.org/>

86

Sass.js Playground - em: X

sass.js.org Sass.js Playground - emscripted libsass in the browser

## Sass.js

an online Playground for Sass.js

Filesystem Sourcemap


### Sass.compile()

SASS	CSS
1 @import "_variables";	1 .imported {
2 @import "_demo";	2 content: "yay, file support!";
3	3 }
4 .selector {	4
5 margin: \$size;	5 .selector {
6 background-color: \$brandColor;	6 margin: 1em;
7	7 background-color: #f60;
8 .nested {	8 }
9 margin: \$size / 2;	9
10 }	10 .selector .nested {

Convert

Sass: Документация на X

sass-scss.ru Sass: Документация на русском



Установка 0

## CSS с суперсилой

Sass является наиболее развитым и стабильным расширением CSS профессионального уровня.



# npm install -g sass

87

```
Командная строка
P:\>npm install -g sass
[.....] / rollbackFailedOptional: verb npm-session 9cad31a3d2d350
C:\Users\sbelyaev\AppData\Roaming\npm\sass -> C:\Users\sbelyaev\AppData\Roamin
g\npm\node_modules\sass\sass.js
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2 (node_modules\
sass\node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevent
s@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x
64"})

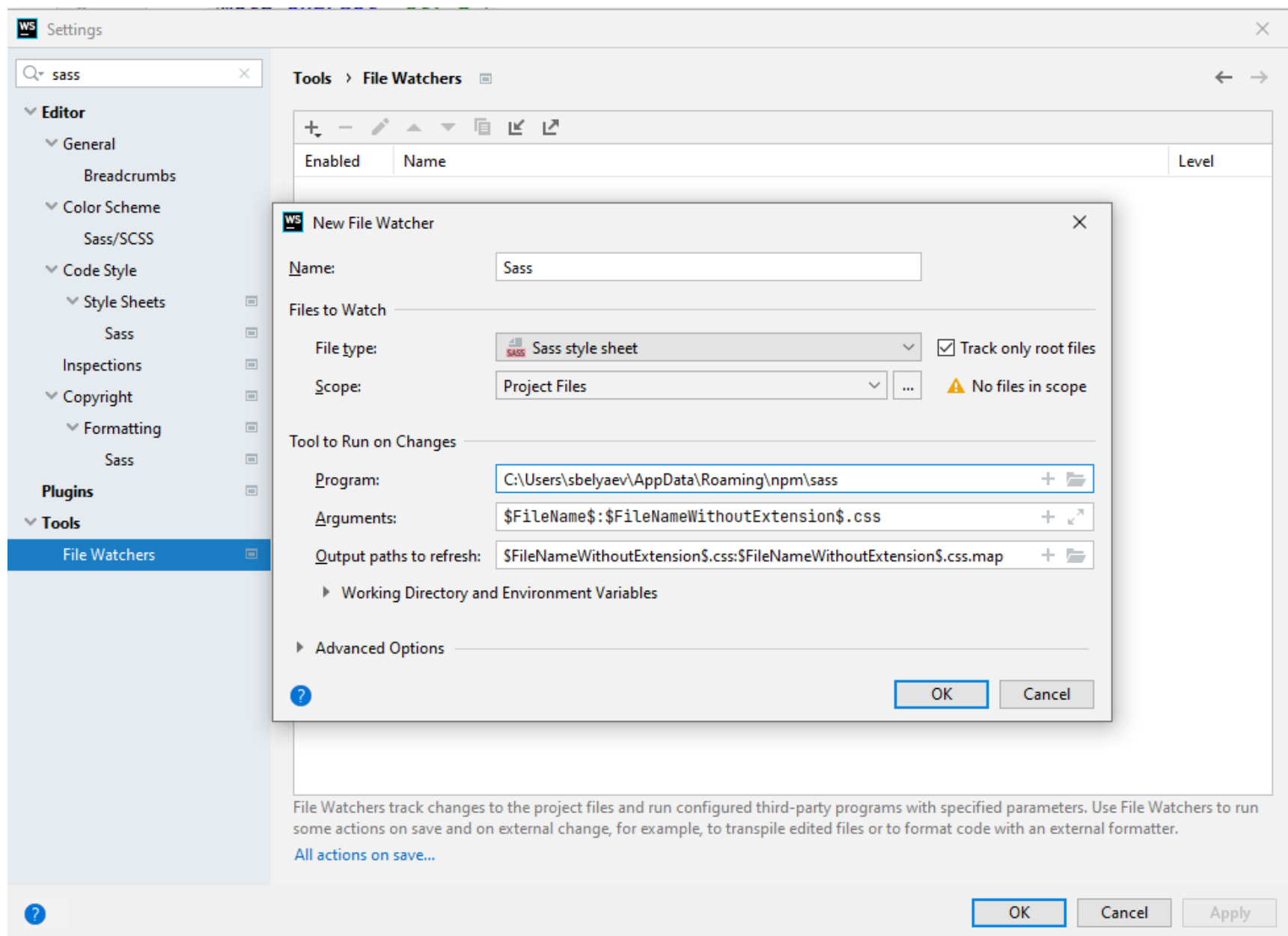
+ sass@1.39.0
added 15 packages in 3.177s
```

## Средства работы с SASS

- **Compass.app** (Условно-бесплатно) Mac Windows Linux
  - <http://compass.kkbox.com/>
- **Koala** (Бесплатно) Mac Windows Linux
  - <http://koala-app.com/>
- **LiveReload** (Условно-бесплатно) Mac Windows
  - <http://livereload.com/>
- **Scout-App** (Бесплатно) Windows Linux Mac
  - <http://scout-app.io/>

# Настройка WebStorm

88





# SASS. Переменные

89

## SASS

### SASS SYNTAX

```
$font-stack: Helvetica, sans-serif  
$primary-color: #333
```

```
body  
  font: 100% $font-stack  
  color: $primary-color
```

## SCSS

```
$font-stack: Helvetica, sans-serif;  
$primary-color: #333;
```

```
body {  
  font: 100% $font-stack;  
  color: $primary-color;  
}
```

## CSS

```
body {  
  font: 100% Helvetica, sans-serif;  
  color: #333;  
}
```

Два формата синтаксиса: SASS и SCSS

# SASS. Вложенности

## SASS

```
nav
  ul
    margin: 0
    padding: 0
    list-style: none

  li
    display: inline-block

  a
    display: block
    padding: 6px 12px
    text-decoration: none
```

## CSS

```
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}

nav li {
  display: inline-block;
}

nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```

# SASS. МИКСИНЫ

91

## SASS

```
=transform($property)
  -webkit-transform: $property
  -ms-transform: $property
  transform: $property
.box
  +transform(rotate(30deg))
```

## CSS

```
.box {
  -webkit-transform: rotate(30deg);
  -ms-transform: rotate(30deg);
  transform: rotate(30deg);
}
```

# SASS. Наследование

92

## SASS

```
/* This CSS will print because %message-  
shared is extended. */
```

```
%message-shared
```

```
border: 1px solid #ccc
```

```
padding: 10px
```

```
color: #333
```

```
// This CSS won't print because %equal-  
heights is never extended.
```

```
%equal-heights
```

```
display: flex
```

```
flex-wrap: wrap
```

```
.message
```

```
@extend %message-shared
```

```
.success
```

```
@extend %message-shared
```

```
border-color: green
```

```
.error
```

```
@extend %message-shared
```

```
border-color: red
```

```
.warning
```

```
@extend %message-shared
```

```
border-color: yellow
```

## CSS

```
/* This CSS will print because %message-  
shared is extended. */
```

```
.warning, .error, .success, .message {
```

```
border: 1px solid #ccc;
```

```
padding: 10px;
```

```
color: #333;
```

```
}
```

```
.success {
```

```
border-color: green;
```

```
}
```

```
.error {
```

```
border-color: red;
```

```
}
```

```
.warning {
```

```
border-color: yellow;
```

```
}
```

# Вопросы для самопроверки

- Какие бывают селекторы? Как ими пользоваться?
- Что такое псевдоклассы и псевдоэлементы?
- Какие способы настройки стилей Вы знаете?
- Какие приоритеты у стилей? Как ими можно управлять?
- Что такое краткая запись CSS? В чем отличие от "полной"?
- Как можно задавать высоту и ширину элемента? Какие используются единицы измерения?
- Какие псевдоклассы используют для ссылок?
- Можно ли изменить внешний вид курсора? Как?
- Как можно сделать responsive-таблицу?
- Какие варианты отображения элементов доступны?
- Какие варианты position есть? Что они означают?
- Что такое счётчики в CSS? Как ими пользоваться?
- Что такое @media? Для чего нужен? Как им пользоваться?
- Что такое grid? Что позволяет? Как настраивается?
- Что такое flex-контейнер? Что позволяет? Как настраивается?
- Как настраиваются переменные в CSS? Как их можно изменить в JS?
- Как в JS можно управлять стилями?
- Что такое w3.css?
- Для чего нужны, что позволяют и как используются LESS и SASS?