

# Введение в CSS

# CSS синтаксис

Структура CSS правила:

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

# Подключение CSS

Существует три способа подключить CSS на страницу: тег `<link>`, тег `<style></style>` и атрибут `style`.

```
<link rel="stylesheet" href="style.css">
```

```
<style> немного стилей </style>
```

```
<div style="немного стилей">
```

# Селекторы CSS

`* { }` /\* По всем тегам на странице\*/

`div { }` /\* По тегу \*/

`.container { }` /\* По классу \*/

`div.container { }` /\* По тегу с классом \*/

`#header { }` /\* По id\*/

`.header a { }` /\* По тегу внутри тега с классом \*/

`.header .logo { }` /\* По классу внутри тега с классом \*/

# Селекторы CSS

`.logo + .menu { }` /\* Соседние \*/

`.header .logo + .menu { }` /\* Контекстные и соседние \*/

`ul > li` /\* Дочерние \*/

`input[checked]` /\* По атрибуту \*/

`input[type="text"]` /\* По атрибуту со значением \*/

# Псевдоклассы CSS

Псевдоклассы это дополнения к обычным селекторам. Мы можем использовать их для более точной выборки или по состоянию элемента.

**селектор:псевдокласс { }** /\*Выборка по псевдоклассу\*/

## Первый и последний элемент

**:first-child** - выбирает первый дочерний элемент родителя

**:last-child** - выбирает последний дочерний элемент родителя

**li:last-child { }**

# Псевдоклассы CSS

*/\*По порядковому номеру\*/*

**селектор:** `nth-child(выражение) { }`

В качестве выражения может быть порядковый номер (число) или функция.

`li:nth-child(5) { }` - выберет пятый элемент

`li:nth-child(2n) { }` - выберет четные элементы

`li:nth-child(2n + 1) { }` - выберет нечетные элементы

`li:nth-child(even) { }` - выберет четные элементы

`li:nth-child(odd) { }` - выберет нечетные элементы

# Псевдоклассы CSS

**селектор:link** { } /\* Не посещенные ссылки\*/

**селектор:visited** { } /\* Посещенные ссылки\*/

**селектор:hover** { } /\* При наведении (на любой объект)\*/

**селектор:active** { } /\* Активные ссылки\*/

Если вы задаете одному элементу правила по этим псевдоклассам, то порядок селекторов должен быть таким иначе не все могут сработать.

# Псевдоклассы CSS

**селектор:focus { }** /\*При фокусе\*/

Срабатывает когда на элемент произошел фокус, к примеру если мы используем навигацию на странице с помощью Tab или поставили в поле курсор. По умолчанию фокус на элементе это обводка голубого цвета.

**селектор:empty { }** /\*Выбирает пустые элементы <p></p>\*/

**селектор:not(селектор) { }** /\*Не соответствующие селектору в функции\*/

**div:not(.red)** - выберет все блоки у которых нет класса red

**body :not(p)** - выберет все элементы которые не p

# Псевдоэлементы CSS

Псевдоэлемент в CSS — это ключевое слово, добавляемое к селектору, которое позволяет стилизовать определенную часть выбранного элемента.

`селектор::before { }` /\*Первый потомок\*/

`селектор::after { }` /\*Последний потомок\*/

Создают псевдоэлементы которые являются потомками выбранного элемента.

```
a::after {  
  content: ">>";  
}
```

# Псевдоэлементы CSS

**селектор::first-line { }** /\*Первая строка\*/

Применяет стили к первой строке блочного элемента. К этому псевдоэлементу можно применить свойства связанные со шрифтами, цветом, фоном.

**селектор::first-letter { }** /\*Первая буква\*/

Применяет стили к первой букве первой строки. Только если перед первой строкой нет другого элемента.

# CSS-свойства

## Делятся на категории

- Оформление текста.
- Блочная модель (размеры и отступы).
- Управление потоком элементов.
- Управление позиционированием.
- Фоны.
- Декоративные эффекты.
- Анимация и динамические эффекты.

# Цифровые значения CSS

## Абсолютные

height: 100px;

font-size: 18px;

## Относительные

height: 100%;

font-size: 1em;

font-size: 2rem;

# Цветовые значения CSS

`color: #000;` // короткая запись

`color: #000000;` // полная запись

`color: rgb(255, 255, 255);`

`color: rgba(255, 255, 255, 0.5);`

`color: hsl(0, 100%, 50%);`

`color: hsla(0, 100%, 50%, 0.5);`

# CSS-свойства для текста

**.text {**

**font-family: Arial, sans-serif;** – определяет вид и семейство шрифта.

**font-size: 13px;** – устанавливает размер шрифта в пикселях или процентах.

**font-style: italic;** – преобразует текст в наклонный (еще его называю курсивным).

**font-weight: bold;** – определяет жирный текст.

**letter-spacing: 2px;** – дополнительное расстояние между символами в пикселях.

**text-align: center;** – выравнивание.

**text-decoration: underline;** – выводит подчеркнутый текст.

**text-transform: uppercase;** – преобразует все буквы в заглавные.

**line-height: 1.2;** – междустрочный интервал.

**}**

# CSS-свойства блочных элементов

```
.box {
```

```
  display: block; - отображение на странице.
```

```
  float: right; - обтекание.
```

```
  width: 100px; - ширина.
```

```
  height: 100px; - высота.
```

```
  margin: 10px; - внешние отступы.
```

```
  padding: 10px; - внутренние отступы.
```

```
}
```

# CSS-свойства позиционирования

**.box-position {**

**position: absolute;** - позиционирование.

**top: 100px;** - отступ сверху.

**right: 100px;** - отступ справа.

**bottom: 100px;** - отступ снизу.

**left: 100px;** - внешние слева.

**z-index: 25;** - индекс слоя.

**}**

# CSS-свойства оформления

**.box-style {**

**background-color: red;** - цвет фона.

**border-radius: 50px;** - закругление углов.

**border: 1px solid red;** - рамка.

**color: green;** - цвет текста.

**opacity: 0.5;** - уровень прозрачности.

**}**

# Наследование в CSS

Наследование в CSS — механизм, с помощью которого значения свойств элемента-родителя передаются его элементам-потомкам.

```
<div class="big-text">  
  <p>Самые крутые машины на этом сайте!</p>  
</div>
```

```
.big-text { font-size: 50px; }
```

Не все свойства могут наследоваться дочерними элементами.

# Наследование в CSS

## Наследуемые

К этому типу относятся свойства стилизации текста и некоторые другие.

`font-size`, `font-family`, `font-style`, `font-weight`, `color`, `text-align`, `text-transform`, `text-indent`, `line-height`, `letter-spacing`, `list-style`, `cursor`, `visibility`, `border-collapse`.

## Ненаследуемые

Это параметры позиционирования, размеров, отступов, фона, рамок и т. д.

`background`, `border`, `padding`, `margin`, `width`, `height`, `position`.

# Каскадность в CSS

К одному и тому же элементу может применяться несколько CSS-правил. В этом случае правила комбинируются.

```
<p class="color-red bold">Самые крутые машины на этом сайте!</p>
```

```
.color-red { color: red; }
```

```
.bold { font-weight: bold; }
```

# Переопределение в CSS

Когда в разных CSS-правилах есть разные свойства, то одно переопределяет другое.

```
<div class="container bg-color">  
  <p>Самые крутые машины на этом сайте!<p>  
</div>
```

```
.container { background-color: green; }  
.bg-color { background-color: red; }
```

# Специфичность и приоритеты

Чтобы понять, кто кого переопределит используют понятия специфичности и приоритетов. Список ниже показывает приоритеты от большего к меньшему.

`#container {}`

`.hero .title {}`

`div .title {}`

`.title {}`

`div {}`

`<div id="container" class="hero title"></div>`

# Подключение шрифтов

```
@font-face {
```

```
    font-family: Output Sans;
```

```
    src: url(fonts/output-sans.woff2) format('woff2'),
```

```
         url(fonts/output-sans.woff) format('woff');
```

```
}
```

# Шрифты google

Можно также подключать удаленные шрифты к примеру через [fonts.google.com](https://fonts.google.com)

```
<link href="https://fonts.googleapis.com/css?family=Roboto" rel="stylesheet">
```

```
body {  
  font-family: 'Roboto', sans-serif;  
}
```

Альтернатива тегу link

```
<style> @import url('https://fonts.googleapis.com/css?family=Roboto'); </style>
```