

Что такое CSS,
преимущества.
Подключение CSS к HTML.
Правила и селекторы CSS.
Псевдоэлементы и
псевдоклассы

Что такое CSS?

CSS – это язык таблицы стилей.

Преимущества CSS

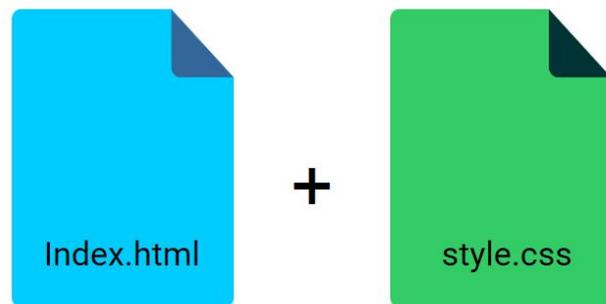
- CSS позволяет значительно сократить размер кода и сделать его читабельным.
- CSS позволяет задавать такие параметры, которые нельзя задать только языком HTML. Например, убрать подчеркивание у ссылок.
- CSS позволяет легко изменять внешний вид страниц. Представьте, вы сделали сайт из 50 страниц, на которых все заголовки синего цвета. Через какое-то время, вы захотели поменять синий цвет на зеленый. Вам придется пройтись по всем 50 страницам и поменять цвет в соответствующем атрибуте. С CSS вам придется сделать это лишь один раз, в таблице стилей.
- С CSS связана так называемая блочная верстка сайта.

Для чего нужен и принцип подключения к HTML

CSS позволяет нам "**оформить, стилизовать**" нашу страницу - например:

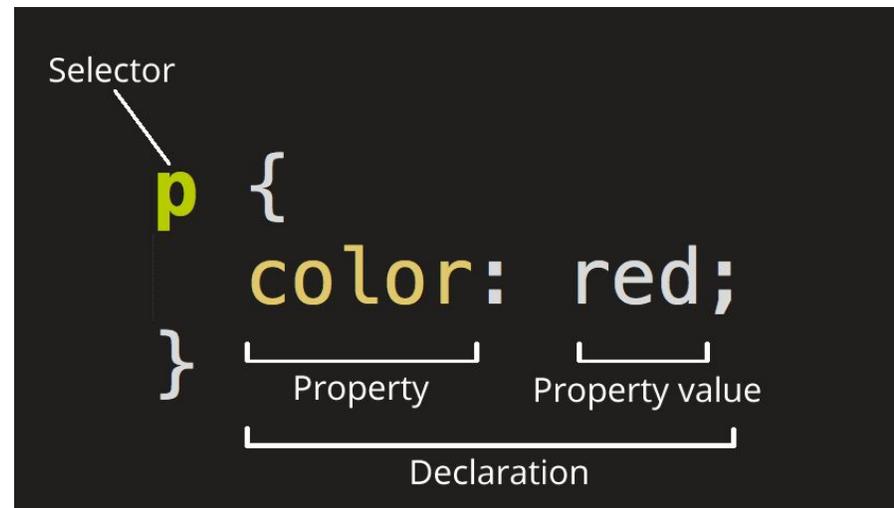
1. задать шрифт для текста (Arial, Helvetica, Times New Roman и т.д.)
2. задать размер шрифта (12px, 14px, 16px и т.д.)
3. задать цвет текста (желтый, голубой, сиреневый, зеленый, темно-зеленый и т.д.)
4. задать цвет фона страницы (желтый, белый, голубой и т.д.)

и многое другое



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

Формат CSS



Селектор (Selector) - Имя HTML-элемента в начале набора правил.

Объявление (Declaration) - указывает, какие из свойств элемента вы хотите стилизовать.

Свойства (Properties) - Способы, которыми вы можете стилизовать определённый HTML-элемент

Значение свойства (Property value) - Справа от свойства, после двоеточия, у нас есть значение свойства, которое выбирает одно из множества возможных признаков для данного свойства

Типы селекторов ([CSS Selector](http://cssselector.html5css.ru) (html5css.ru))

Имя селектора	Что выбирает	Пример
Селектор элемента (иногда называемый селектором тега или типа)	Все HTML-элемент(ы) указанного типа.	p Выбирает <p>
ID селектор	Элемент на странице с указанным ID на данной HTML. Лучше всего использовать один элемент для каждого ID (и конечно один ID для каждого элемента), даже если вам разрешено использовать один и тот же ID для нескольких элементов.	#my-id Выбирает <p id="my-id"> или
Селектор класса	Элемент(ы) на странице с указанным классом (множество экземпляров класса может объявляться на странице).	.my-class Выбирает <p class="my-class"> и
Селектор атрибута	Элемент(ы) на странице с указанным атрибутом.	img[src] Выбирает но не
Селектор псевдокласса	Указанные элемент(ы), но только в случае определённого состояния, например, при наведении курсора.	a:hover Выбирает <a>, но только тогда, когда указатель мыши наведён на ссылку.

Псевдоэлементы и псевдоклассы

(Псевдоклассы и псевдоэлементы

(danfimov.github.io))

Псевдоэлементы — это элементы, которых не существует в HTML-разметке

- Псевдоэлементы позволяют создавать стилизованные "виртуальные" элементы внутри других элементов, такие как `::before` и `::after`.
- Они позволяют добавлять контент или декорацию к элементу, не добавляя дополнительных элементов в HTML.
- Например, `::before` может использоваться для добавления контента перед содержимым элемента, а `::after` - после содержимого.

<pre>[селектор]::псевдоэлемент { свойство: значение; }</pre>	<pre>p::before { content: "Префикс: "; font-weight: bold; }</pre> <pre>button::after { content: "Нажми меня"; background-color: green; color: white; }</pre>	<pre><div class="box">Пример</div> .box::before { content: "До "; font-weight: bold; }</pre>
--	---	---

Таблица псевдоэлементов

Псевдоэлемент	Описание
::before	Добавляет контент перед содержимым элемента
::after	Добавляет контент после содержимого элемента
::first-line	Применяет стили к первой строке блочного элемента
::first-letter	Применяет стили к первой букве блочного элемента
::selection	Применяет стили к выделенному тексту на странице
::marker	Применяет стили к маркеру элемента списка
::placeholder	Применяет стили к плейсхолдеру в инпутах и textarea
::backdrop	Применяет стили к заднему фону элемента, например для модальных окон

Псевдоэлементы и псевдоклассы

(Псевдоклассы и псевдоэлементы

(danfimov.github.io)

Псевдоклассы — особый вид селектора, который уточняет тип или состояние. Это как заранее заданные классы.

```
[селектор]:псевдокласс {  
  свойство: значение;  
}
```

```
a:hover {  
  color: red;  
}
```

```
input[type="checkbox"]:checked + label {  
  color: green;  
}
```

```
input:focus {  
  border: 2px solid blue;  
}
```

```
input:focus {  
  border: 2px solid blue;  
}
```

```
li:nth-child(odd) {  
  background-color: lightgray;  
}
```

```
li:nth-child(odd) {  
  background-color: lightgray;  
}
```

```
:not(.special) {  
  font-style: italic;  
}
```

```
:not(.special) {  
  font-style: italic;  
}
```

Таблица псевдоклассов

:hover	Применяется, когда пользователь наводит указатель мыши на элемент
:active	Применяется, когда элемент активен (например, кнопка нажата)
:focus	Применяется, когда элемент получает фокус ввода (например, при клике на input)
:visited	Применяется к ссылкам, которые были посещены
:nth-child(n)	Применяет стили к элементам, которые являются n-ными потомками своего родителя
:first-child	Применяется к первому потомку родителя
:last-child	Применяется к последнему потомку родителя
:nth-of-type(n)	Применяет стили к элементам, которые являются n-ыми потомками своего типа
:first-of-type	Применяется к первому потомку своего типа
:last-of-type	Применяется к последнему потомку своего типа
:not(selector)	Применяет стили ко всем элементам, кроме тех, которые соответствуют указанному селектору
:nth-last-child(n)	Применяет стили к n-ному потомку с конца своего родителя

Примеры

Пример 1: Стилизация первого параграфа

1.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="styles.css">
  <title>Пример 1</title>
</head>
<body>
  <p>Первый параграф</p>
  <p>Второй параграф</p>
</body>
</html>
```

styles.css

```
p:first-child {
  font-weight: bold;
}
```

Пример 2: Стилизация ссылок

1.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="styles.css">
  <title>Пример 2</title>
</head>
<body>
  <a href="#">Ссылка 1</a>
  <a href="#">Ссылка 2</a>
  <a href="#">Ссылка 3</a>
</body>
</html>
```

CSS:

styles.css

```
a {
  color: blue;
  text-decoration: none;
}
a:hover {
  text-decoration: underline;
}
```

Примеры

Пример 3: Стилизация элементов списка

1.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="styles.css">
  <title>Пример 3</title>
</head>
<body>
  <ul>
    <li>Элемент 1</li>
    <li>Элемент 2</li>
    <li>Элемент 3</li>
  </ul>
</body>
</html>
```

styles.css

Copy code

```
ul li:nth-child(odd) {
  background-color: #f2f2f2;
```

Пример 4: Стилизация формы

1.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="styles.css">
  <title>Пример 4</title>
</head>
<body>
  <form>
    <input type="text" placeholder="Имя">
    <input type="email" placeholder="Email">
    <input type="submit" value="
Отправить">
  </form>
</body>
</html>
```

styles.css

Copy code

```
input[type="text"] {
  border: 1px solid #ccc;
}

input[type="email"] {
  border: 1px solid #ccc;
}

input[type="submit"] {
  background-color: #008CBA;
  color: #fff;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}
```

Самостоятельные задания

Задача 1: Стилизация кнопок. Создайте несколько кнопок с использованием HTML. Используйте псевдоклассы для стилизации кнопок при наведении (:hover) и активации (:active).

Задача 2: Выравнивание элементов. Создайте блок с несколькими вложенными элементами (например, div и p). Используйте псевдоклассы для выравнивания элементов внутри блока (например, :first-child, :last-child).

Задача 3: Заголовки с разными стилями. Создайте несколько заголовков разных уровней (h1, h2, h3). Используйте псевдоэлементы ::before и ::after для добавления декоративных элементов к заголовкам.

Задача 4: Подчеркивание при наведении. Создайте несколько ссылок с использованием HTML. Используйте псевдокласс :hover для добавления подчеркивания к ссылкам при наведении.

Задача 5: Стилизация списков. Создайте нумерованный список (ordered list) с несколькими элементами. Используйте псевдокласс :hover для добавления эффектов при наведении на элементы списка.

Задача 6: Стилизация формы. Создайте форму с несколькими полями ввода (input) и кнопкой (button). Используйте псевдокласс :focus для стилизации элементов формы при фокусировке.

Задача 7: Скрытие элемента. Создайте элемент с содержимым. Используйте псевдокласс :nth-child или :nth-of-type для скрытия определенного элемента.

Задача 8: Стилизация последнего элемента в списке. Создайте нумерованный список (ordered list) с несколькими элементами. Используйте псевдокласс :last-child для стилизации последнего элемента в списке.