

Что такое CSS?

Способы подключения стилей на страницу.

Селекторы тега, класса и идентификатора

Цели занятия

- повторить работу со списками и параграфами, разобрать таблицы;
- разобрать устаревшие теги и зачем нужен Iframe;
- добавить уникальности проекту с помощью Favicon;

Полезные материалы

- [Справочник CSS](#)
- [Правила оформления CSS-кода](#)
- [Валидатор CSS](#)

Что такое CSS?

CSS (Cascading Style Sheets, каскадные таблицы стилей) — язык описания внешнего вида HTML-документа. Это одна из базовых технологий в современном интернете. Практически ни один сайт не обходится без CSS, поэтому HTML и CSS действуют в единой связке.

Основы CSS помогут вам понять, что вам нужно для начала работы. Мы ответим на такие вопросы как: Как сделать мой текст черным или красным? Как сделать так, чтобы контент появлялся в определённом месте страницы? Как украсить мою веб-страницу с помощью фоновых изображений и цветов?

Наверняка многие из вас слышали про Sass, Less, PostCSS или Bootstrap — это инструменты, которые позволяют упростить некоторые моменты в CSS. Они все основаны на CSS и браузер понимает только CSS, поэтому знание “чистого” CSS очень важно для фронтенд-разработчика, а изучение инструментов вторично и зависит от компании. Например, в divan.ru используется PostCSS и не используются все остальные и их знание не требуется.

Способы подключения стилей на страницу

Внутренние стили

Внутренний или встроенный стиль является по существу расширением для одиночного тега используемого на текущей веб-странице. Для определения стиля используется атрибут `style`, а его значением выступает набор стилевых правил. Используется только в исключительных ситуациях и правилом хорошего кода считается избегать такой стилизации.

Пример:

```
<body>
  <p style="font-size: 120%; color: #cd66cc">Пример текста</p>
</body>
```

Глобальные стили

При использовании глобальных стилей свойства CSS описываются в самом документе и располагаются в заголовке веб-страницы. Также не рекомендуется к использованию, если ваш проект имеет больше одной страницы или несколько сложных блоков на странице.

Пример:

```
<head>
  <title>Глобальные стили</title>
  <style>
    h1 {
      font-size: 120%;
      font-family: Verdana, Arial, Helvetica, sans-serif;
```

```
color: #333366;
}
</style>
</head>
```

Связанные стили

При использовании связанных стилей описание селекторов и их значений располагается в отдельном файле, как правило, с расширением `css`, а для связывания документа с этим файлом применяется тег `<link>`. Данный тег помещается в контейнер `<head>`. Именно этот способ является основным и будет использоваться в дальнейшем.

Пример:

```
<head>
  <meta charset="utf-8">
  <title>Стили</title>
  <link rel="stylesheet" href="styles.css">
</head>
```

Файл со стилями `styles.css`

```
h1 {
  color: #000080;
  font-size: 200%;
  font-family: Arial, Verdana, sans-serif;
  text-align: center;
}

p {
  padding-left: 20px;
}
```

Структура CSS

Как и HTML, CSS на самом деле не является языком программирования. Это не язык разметки — это язык таблицы стилей. Это означает, что он позволяет применять стили выборочно к элементам в документах HTML. Например, чтобы выбрать все элементы абзаца на HTML странице и изменить текст внутри них с чёрного на красный, вы должны написать этот CSS:

```
p {  
  color: red;  
}
```

Весь CSS-файл состоит из наборов правил, а каждое правило из отдельных частей:

Селектор — имя HTML-элемента в начале набора правил. Он выбирает элементы для применения стиля (в данном случае, элементы `<p>`). Для стилизации другого элемента, просто измените селектор.

Объявление — единственное правило, например `color: red;` указывает, какие из свойств элемента вы хотите стилизовать.

Свойства — способы, которыми вы можете стилизовать определённый HTML-элемент (в данном случае, `color` является свойством для элементов `<p>`). В CSS вы выбираете, какие свойства вы хотите затронуть в вашем правиле.

Значение свойства — справа от свойства, после двоеточия, у нас есть значение свойства, которое выбирает одно из множества возможных признаков для данного свойства (существует множество значений `color`, помимо `red`).

Обратите внимание на важные части синтаксиса:

- Каждый набор правил (кроме селектора) должен быть обёрнут в фигурные скобки (`{}`).
- В каждом объявлении необходимо использовать двоеточие (`:`), чтобы отделить свойство от его значений.
- В каждом наборе правил вы должны использовать точку с запятой (`;`), чтобы отделить каждое объявление от следующего.

Селекторы тега, класса и идентификатора

В примере выше был показан селектор, который применяется на все теги `<p>` на данной странице. Но это слишком грубая выборка и чаще всего нам необходимо стилизовать лишь часть тегов, а не все сразу.

И селекторы позволяют это делать, но для этого необходима выборка по классу. Класс это атрибут, который мы назначаем необходимому тегу или группе тегов, имеющий смысловое название блока. Например, мы хотим вставить на страницу три параграфа, два из которых будут с красным текстом, а третий с синим:

```
<body>
  <p class="red-paragraph">Я буду красным</p>
  <p class="blue-paragraph">А я синим</p>
  <p class="red-paragraph">А я тоже красным</p>
</body>
```

Тогда стилизовать их можно следующим способом (обратите внимание на точку перед селектором, она означает что мы ищем элементы с указанным классом)

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

Классы являются основным способом выбирать нужные теги на странице, но далеко не единственным. Например, иногда мы хотим выбрать все красные параграфы, но один из них сделать ещё и с другим размером текста. Тогда мы могли бы использовать идентификатор для данного параграфа:

```
<p class="red-paragraph">Я буду красным</p>
<p class="blue-paragraph">А я синим</p>
<p class="red-paragraph" id="big">Тоже красный, но с другим
размером</p>
```

А стилизовать его можно с помощью селектора по идентификатору (обратите внимание на символ решётки):

```
.red-paragraph {  
  color: red;  
}  
.blue-paragraph {  
  color: blue;  
}  
#big {  
  font-size: 40px;  
}
```

Стоит помнить, что в рамках всей страницы идентификатор должен быть уникален и не может повторяться. А также то, что это лишь один из способов решить задачу, но её можно было решить и с помощью селекторов по классу, вообще не используя идентификатор и в большинстве случаев нужно стремиться именно к этому варианту.