

Лекция №2.1

План:

1. Подключение CSS
2. CSS – синтаксис
3. Переопределение правил CSS

1. Подключение CSS

Существует четыре способа связать стили с вашим HTML-документом. Наиболее часто используемые методы — встроенный CSS и внешний CSS.

1.1 Внедренный CSS — элемент <style>

Вы можете поместить свои правила CSS в документ HTML, используя элемент <style>. Этот тег находится внутри тегов <head> ... </head>. Правила, определенные с использованием этого синтаксиса, будут применяться ко всем элементам, доступным в документе. Вот общий синтаксис

```
<html>
<head>
  <style type = "text/css" media = "all">
    body {
      background-color: linen;
    }
    h1 {
      color: maroon;
      margin-left: 40px;
    }
  </style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Атрибуты, связанные с элементами <style>:

атрибут	Значение	Описание
тип	<i>text/css</i>	Указывает язык таблицы стилей как тип содержимого (MIME-тип). Это обязательный атрибут.
средства массовой информации	<i>screen</i> <i>TTY</i> <i>TV</i> <i>projection</i> <i>portable</i> <i>print</i> <i>braille</i> <i>ear</i> <i>all</i>	Указывает устройство, на котором будет отображаться документ. Значением по умолчанию является <i>все</i> . Это необязательный атрибут.

1.2 Встроенный CSS — Атрибут стиля

Вы можете использовать атрибут `style` любого элемента HTML для определения правил стиля. Эти правила будут применяться только к этому элементу. Вот общий синтаксис

```
<element style = "...style rules....">
```

Атрибуты

атрибут	Значение	Описание
<code>style</code>	<code>style rules</code>	Значение атрибута <code>style</code> является комбинацией объявлений стилей, разделенных точкой с запятой (;).

Ниже приведен пример встроенного CSS на основе приведенного выше синтаксиса:

```
<html>
  <head>
  </head>
  <body>
    <h1 style = "color:#36C;">
      This is inline CSS
    </h1>
  </body>
</html>
```

1.3 Внешний CSS — Элемент `<link>`

Элемент `<link>` можно использовать для включения внешнего файла таблицы стилей в ваш HTML-документ.

Внешняя таблица стилей — это отдельный текстовый файл с расширением `.css`. Вы определяете все правила стиля в этом текстовом файле, а затем можете включить этот файл в любой HTML-документ, используя элемент `<link>`.

Вот общий синтаксис включения внешнего файла CSS:

```
<head>
  <link type = "text/css" href = "... " media = "... " />
</head>
```

Атрибуты, связанные с элементами `<style>`:

атрибут	Значение	Описание
<code>type</code>	<code>css text</code>	Указывает язык таблицы стилей как тип содержимого (MIME-тип). Этот атрибут обязателен.
<code>HREF</code>	<code>URL</code>	Указывает файл таблицы стилей, имеющий правила стиля. Этот атрибут является обязательным.
<code>mass media</code>	<code>screen</code> <code>TTY</code> <code>TV</code> <code>projection</code> <code>portable</code> <code>print</code> <code>braille</code> <code>ear</code>	Указывает устройство, на котором будет отображаться документ. Значением по умолчанию является все. Это необязательный атрибут.

	all	
--	-----	--

Рассмотрим простой файл таблицы стилей с именем `mystyle.css`, имеющий следующие правила:

```
h1, h2, h3 {
  color: #36C;
  font-weight: normal;
  letter-spacing: .4em;
  margin-bottom: 1em;
  text-transform: lowercase;
}
```

Теперь вы можете включить этот файл `mystyle.css` в любой HTML-документ следующим образом:

```
<head>
  <link type = "text/css" href = "mystyle.css" media = "all" />
</head>
```

1.4 Импортированный CSS — правило `@import`

`@import` используется для импорта внешней таблицы стилей аналогично элементу `<link>`. Вот общий синтаксис правила `@import`.

```
<head>
  <@import "URL";
</head>
```

Здесь URL — это URL-адрес файла таблицы стилей с правилами стиля. Вы также можете использовать другой синтаксис —

```
<head>
  <@import url("URL");
</head>
```

Ниже приведен пример, показывающий, как импортировать файл таблицы стилей в документ HTML:

```
<head>
  @import "mystyle.css";
</head>
```

2. CSS – синтаксис

CSS содержит правила стилей, которые интерпретируются браузером и затем применяются к соответствующим элементам в вашем документе. Правило стиля состоит из трех частей:

- **Селектор** . Селектор – это HTML-тег, к которому будет применен стиль. Это может быть любой тег типа `<h1>` или `<table>` и т. Д.
- **Свойство** – свойство является типом атрибута тега HTML. Проще говоря, все атрибуты HTML преобразуются в свойства CSS. Это могут быть *цвета* , *границы* и т. Д.
- **Значение** – значения присваиваются свойствам. Например, свойство *color* может иметь значение *red* или *#F1F1F1* и т. Д.

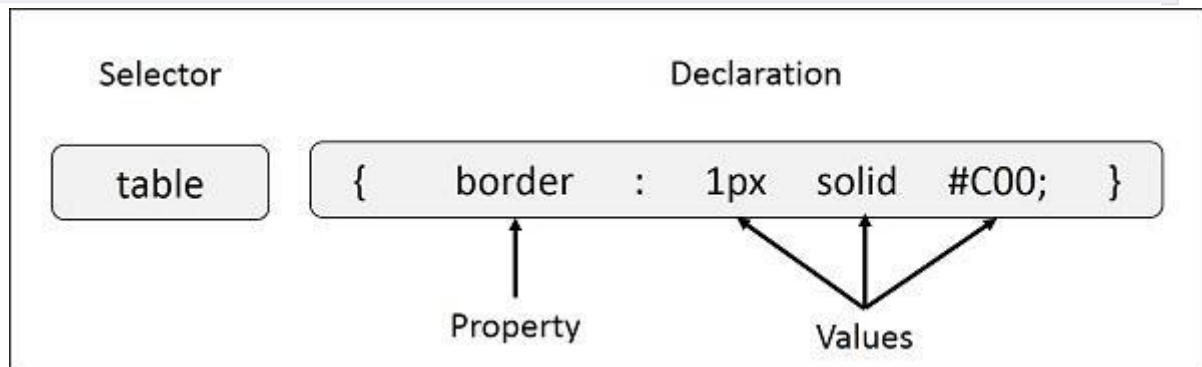
Селектор . Селектор – это HTML-тег, к которому будет применен стиль. Это может быть любой тег типа `<h1>` или `<table>` и т. Д.

Свойство – свойство является типом атрибута тега HTML. Проще говоря, все атрибуты HTML преобразуются в свойства CSS. Это могут быть *цвета*, *границы* и т. Д.

Значение – значения присваиваются свойствам. Например, свойство *color* может иметь значение *red* или *#F1F1F1* и т. Д.

Вы можете поместить синтаксис правила стиля CSS следующим образом:

```
selector { property: value }
```



Пример – Вы можете определить границу таблицы следующим образом –

```
table { border : 1px solid #C00; }
```

Здесь *table* – селектор, а *border* – свойство, а заданное значение *1px solid # C00* – значение этого свойства.

Вы можете определить селекторы различными способами, исходя из вашего удобства. Позвольте мне поставить эти селекторы один за другим.

Селекторы типов

Это тот же селектор, который мы видели выше. Снова, еще один пример, чтобы дать цвет всем заголовкам уровня 1 –

```
h1 {  
  color: #36CFFF;  
}
```

Универсальные селекторы

Вместо того, чтобы выбирать элементы определенного типа, универсальный селектор просто совпадает с именем любого типа элемента –

```
* {  
  color: #000000;  
}
```

Это правило отображает содержимое каждого элемента в нашем документе в черном цвете.

Селекторы потомков

Предположим, что вы хотите применить правило стиля к определенному элементу, только когда оно находится внутри определенного элемента. Как показано в следующем примере, правило стиля будет применяться к элементу `` только тогда, когда он находится внутри тега ``.

```
ul em {  
  color: #000000;  
}
```

Селекторы класса

Вы можете определить правила стиля на основе атрибута класса элементов. Все элементы этого класса будут отформатированы в соответствии с заданным правилом.

```
.black {  
  color: #000000;  
}
```

Это правило отображает содержимое в черном для каждого элемента с атрибутом класса, установленным в нашем документе как *черный*. Вы можете сделать это немного более конкретным. Например –

```
h1.black {  
  color: #000000;  
}
```

Это правило отображает содержимое в черном цвете только для элементов `<h1>` с атрибутом `class`, установленным в *черный цвет*.

Вы можете применить более одного селектора класса к данному элементу. Рассмотрим следующий пример –

```
<p class = "center bold">  
  This para will be styled by the classes center and bold .  
</p>
```

Селекторы ID. Вы можете определить правила стиля на основе атрибута *id* элементов. Все элементы с таким *идентификатором* будут отформатированы в соответствии с заданным правилом.

```
#black {  
  color: #000000;  
}
```

Это правило отображает содержимое в черном цвете для каждого элемента с атрибутом *id*, установленным в нашем документе как *черный*. Вы можете сделать это немного более конкретным. Например –

```
h1#black {  
  color: #000000;  
}
```

Это правило отображает содержимое в черном цвете только для элементов `<h1>` с атрибутом *id*, установленным в *черный цвет*.

Истинная сила селекторов *идентификаторов* заключается в том, что они используются в качестве основы для селекторов-потомков, например:

```
#black h2 {  
  color: #000000;  
}
```

В этом примере все заголовки уровня 2 будут отображаться черным цветом, когда эти заголовки будут находиться в тегах с атрибутом *id*, установленным на *черный*.

Селекторы ребенка. Вы видели потомков селекторов. Существует еще один тип селектора, который очень похож на потомков, но имеет различную функциональность. Рассмотрим следующий пример –

```
body > p {  
  color: #000000;  
}
```

Это правило отображает все абзацы черным, если они являются прямыми дочерними элементами элемента `<body>`. Другие абзацы, помещенные в другие элементы, такие как `<div>` или `<td>`, не будут влиять на это правило.

Селекторы атрибутов

Вы также можете применять стили к элементам HTML с определенными атрибутами. Приведенное ниже правило стиля будет соответствовать всем элементам ввода, имеющим атрибут `type`, со значением *текста* –

```
input[type = "text"] {  
    color: #000000;  
}
```

Преимущество этого метода в том, что на элемент `<input type = "submit" />` это не влияет, а цвет применяется только к нужным текстовым полям.

К селектору атрибутов применяются следующие правила.

- **p [lang]** – выбирает все элементы абзаца с атрибутом *lang* .
 - **p [lang = "fr"]** – выбирает все элементы абзаца, чей атрибут *lang* имеет значение равно "fr".
 - **p [lang ~ = "fr"]** – выбирает все элементы абзаца, атрибут *lang* которых содержит слово "fr".
 - **p [lang | = "en"]** – выбирает все элементы абзаца, атрибут *lang* которых содержит значения, которые в точности равны "en" или начинаются с "en-".
- p [lang]** – выбирает все элементы абзаца с атрибутом *lang* .
- p [lang = "fr"]** – выбирает все элементы абзаца, чей атрибут *lang* имеет значение равно "fr".
- p [lang ~ = "fr"]** – выбирает все элементы абзаца, атрибут *lang* которых содержит слово "fr".
- p [lang | = "en"]** – выбирает все элементы абзаца, атрибут *lang* которых содержит значения, которые в точности равны "en" или начинаются с "en-".

Несколько стилевых правил

Возможно, вам придется определить несколько правил стиля для одного элемента. Вы можете определить эти правила для объединения нескольких свойств и соответствующих значений в один блок, как определено в следующем примере –

```
h1 {  
    color: #36C;  
    font-weight: normal;  
    letter-spacing: .4em;  
    margin-bottom: 1em;  
    text-transform: lowercase;  
}
```

Здесь все пары свойств и значений разделяются **точкой с запятой (;)** . Вы можете хранить их в одну или несколько строк. Для лучшей читаемости мы держим их в отдельных строках.

Какое-то время не беспокойтесь о свойствах, упомянутых в приведенном выше блоке. Эти свойства будут объяснены в следующих главах, и вы можете найти полную информацию о свойствах в CSS References

Группировка селекторов. Вы можете применить стиль ко многим селекторам, если хотите. Просто разделите селекторы запятой, как показано в следующем примере –

```
h1, h2, h3 {  
    color: #36C;
```

```
font-weight: normal;  
letter-spacing: .4em;  
margin-bottom: 1em;  
text-transform: lowercase;  
}
```

Это правило определения стиля будет применимо также к элементам h1, h2 и h3. Порядок в списке не имеет значения. Все элементы в селекторе будут иметь соответствующие объявления, примененные к ним.

3. Переопределение правил CSS

Мы обсудили четыре способа включения правил таблиц стилей в HTML-документ. Вот правило для переопределения любого правила таблицы стилей.

- Любая встроенная таблица стилей имеет наивысший приоритет. Таким образом, он переопределит любое правило, определенное в тегах `<style> ... </ style>` или правилах, определенных в любом внешнем файле таблицы стилей.
- Любое правило, определенное в тегах `<style> ... </ style>`, будет переопределять правила, определенные в любом внешнем файле таблицы стилей.
- Любое правило, определенное во внешнем файле таблицы стилей, имеет самый низкий приоритет, и правила, определенные в этом файле, будут применяться только в том случае, если вышеупомянутые два правила неприменимы.

Любая встроенная таблица стилей имеет наивысший приоритет. Таким образом, он переопределит любое правило, определенное в тегах `<style> ... </ style>` или правилах, определенных в любом внешнем файле таблицы стилей.

Любое правило, определенное в тегах `<style> ... </ style>`, будет переопределять правила, определенные в любом внешнем файле таблицы стилей.

Любое правило, определенное во внешнем файле таблицы стилей, имеет самый низкий приоритет, и правила, определенные в этом файле, будут применяться только в том случае, если вышеупомянутые два правила неприменимы.