



HyperText Markup Language

HTML&CSS

Lesson 5

CSS

HyperText Markup Language

История CSS

Таблицы стилей существовали и раньше, начиная с 1970-х годов.

Но принципиально новую технологию в **1994** году предложил Хоакин Виум Ли, на данный момент генеральный технический директор Opera Software.

Первая версия изначально была названа CHSS(Cascade HTML Style Sheets).

Позже название упростилось до CSS(Cascade Style Sheets).

В **1997** году вышла версия CSS 2.

В **2006** на ряду с HTML5 вышла спецификация CSS 3. На данный момент еще недореализована в браузерах.

HyperText Markup Language

Способы интеграции CSS в страницы

Есть три основных способа интегрировать CSS таблицы на ваши html страницы:

1. Через атрибут **style**.

```
<p style="text-align:right">...</p>
```

2. Через тег **<style>**, размещенный в теге **<head>**.

```
<head>...  
<style type="text/css">...</style>  
...</head>
```

3. Через отдельный файл с расширением **.css**, подключенный к странице.


HyperText Markup Language

Семантика CSS

Семантика CSS достаточно проста:

Селектор

```
{  
Правило: значение;  
Правило: значение;  
Правило: значение;  
...  
}
```



Стиль

В качестве селектора может выступать как имя тега, так и отдельный класс либо идентификатор.

HyperText Markup Language

Селектор – имя тега

Если в качестве селектора указано имя тега, то в этом случае все элементы данного тега будут соответствовать данному стилю.

При этом, если внутри этих тегов будут находиться текстовые элементы, они будут наследовать определенные правила.

```
p
{
  font-family: Segoe UI;
  font-size: 18pt;
  color: Green;
}
```

HyperText Markup Language

Селектор - класс

Класс - особенный вид селектора, которым можно пометить как один, так и несколько элементов.

Чтобы указать селектор класс, в CSS необходимо использовать конструкцию `.имя класса`

Чтобы привязать к этому классу элемент, в открывающем теге необходимо указать атрибут `class = "имя класса"`

```
...  
.class1  
  {  
    font-family: Segoe UI;  
    font-size: 18pt;  
    color: Green;  
  }  
...  
<p class="class1">Параграф</p>
```

HyperText Markup Language

Селектор - идентификатор

Идентификатор - особенный вид селектора, которым можно пометить один элемент. По умолчанию предполагается, что элемент с таким селектором на странице один.

Чтобы указать селектор идентификатор, в CSS необходимо использовать конструкцию #имя идентификатора

Чтобы привязать к этому идентификатору элемент, в открывающем теге необходимо указать атрибут id = "имя идентификатора"

```
...  
#Id1  
  {  
    font-family: Segoe UI;  
    font-size: 18pt;  
    color: Green;  
  }
```

...

```
<p id="Id1">Параграф</p>
```

HyperText Markup Language

Сложные селекторы

Есть несколько основных сложных конструкций:

1. Конструкция наследования
2. Конструкция использования нескольких селекторов
3. Конструкция многозначного селектора

HyperText Markup Language

Конструкция наследования

Если необходимо определить стиль для элемента, вложенного в другой, который найти проще, то лучшим решением является конструкция наследования.

Сперва указывается селектор элементов, в которых должны будут размещены элементы, соответствующие второму селектору.

(родительский_элемент) (дочерний_элемент)

{

...

}

<родительский элемент>

<дочерний элемент>...</дочерний элемент>

</родительский элемент>

HyperText Markup Language

Конструкция использования нескольких селекторов

Если необходимо создать стиль для элемента, который привязан именно к нескольким селекторам, то необходимо использовать данную конструкцию.

Все селекторы, которым обязан соответствовать элемент, перечисляются без пробелов по очереди.

селектор.селектор {}

Селектор.класс

{

...

}

...

<селектор class="класс">

...

</селектор>

HyperText Markup Language

Конструкция использования многозначного селектора

Данная конструкция позволяет указать и объединить при описании стиля несколько селекторов в один, перечислив их через запятую.

Данный стиль будет применен к каждому члену селектора.

```
div, .class1, p
{
...
}

...
<p> ... </p>
<span class="class1"> ... </span>
<div> ... </div>
```