

# HTML & CSS

Cascading Style Sheets



# HTML & CSS

Тема

CSS

# CSS

## 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.

# CSS

## 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.

## 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;
}
```

# CSS

## 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>
```

## 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>
```

## Сложные селекторы

Есть несколько основных сложных конструкций:

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



## 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>

## 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(имя тега).класс {}`

**Селектор(имя тега).класс**

**{**

**...**

**}**

**...**

**<селектор class="класс">**

**...**

**</селектор>**

## Конструкция использования нескольких селекторов

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

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

```
.класс1.класс2 {}
```

```
.класс1.класс2
```

```
{
```

```
...
```

```
}
```

```
...
```

```
<селектор class="класс1 класс2">
```

```
...
```

```
</селектор>
```

## 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>
```

Q&A