

Cascading Style Sheets (CSS)

CSS - Cascading Style Sheets
(каскадные таблицы стилей) - это средство, позволяющее задавать различные свойства HTML-тегам.

Используя CSS можно:

- Задавать поля, отступы, размер и тип шрифта, цвета текста и фона для отдельных элементов страницы (абзацев, слов, букв), оформлять красные строки, буквицы.
- Изменять оформление целого сайта, состоящего из сотен файлов не прикасаясь к HTML-коду, редактируя всего лишь один css-файл.
- Уменьшать количество тегов в HTML-документе, по возможности отделяя информационное наполнение HTML-файла от его внешнего представления на экране

**Программист может
использовать в HTML-документе
три типа стилей:**

Встроенный (inline)

Внедренный (embedded)

Связанный (linked)

- ***Встроенный (inline)***. Стиль, описанный внутри тега при помощи атрибута `style`. Контролирует представление отдельного тега.
- Задается атрибутом `style`. Этот атрибут можно использовать практически в каждом теге, задавая его специфические свойства.

Пример

Вид программы:

```
<P style="font-size: 1cm; color: blue">
```

К этому абзацу применено стилевое определение.

Результат отображения:

К этому абзацу применено стилевое определение.

- ***Внедренный (embedded)***. Стиль, описанный в заголовке HTML-файла при помощи тегов `<STYLE>...</STYLE>`. Контролирует представление отдельного HTML-документа.
- Можно задавать стиль для тега или группы тегов так, чтобы определение работало на протяжении всего HTML-документа.

Пример

Вид программы:

```
<HTML> <HEAD>
```

```
<TITLE>Пример задания стиля для заголовков</TITLE>
```

```
<STYLE type="text/css"> <!--
```

```
H1,H2,H3,H4,H5,H6
```

```
{ text-align: left; color: maroon;
```

```
font-family: Arial Cyr, Geneva, Helvetica, sans-serif; } -->
```

```
</STYLE></HEAD> <BODY bgcolor=#DFF0D5 text=black>
```

```
<H2>Пример задания стиля для заголовков</H2><HR>
```

```
<P>Это обычный текст <H3>Это заголовок</H3>
```

```
<P>Это снова обычный текст</BODY>
```

```
</HTML>
```

Результат отображения **Пример задания стиля для заголовков**

Это обычный текст

Это заголовок

Это снова обычный текст

Код стиля

```
<STYLE type="text/css">  
<!--  
  H1,H2,H3,H4,H5,H6  
  {  
    text-align: left;  
    color: maroon;  
    font-family: Arial Cyr, Geneva, Helvetica, sans-serif;  
  }  
-->  
</STYLE>
```

- *Связанный (linked)*. Стилль, описанный в отдельном css-файле. Контролирует представление многих HTML-документов. Для ссылки на стилевой файл в головной части HTML-документа записывают тег **<LINK>**.

Пример

Содержимое файла style.css:

```
BODY { margin-left: 40px; }
```

```
H1,H2,H3,H4,H5,H6
```

```
{
```

```
text-align: left;
```

```
color: maroon;
```

```
font-family: Arial Cyr, Geneva, Helvetica, sans-serif;
```

```
}
```

В самих же HTML-файлах в разделе

<HEAD>...</HEAD> на файл style.css

расположена ссылка:

```
<LINK rel="stylesheet" type="text/css"  
href="style.css">
```

Комбинирование стилей

На протяжении одного документа могут применяться все описанные выше три стилевых механизма.

Для браузера самыми главными являются встроенные стили, затем по убыванию старшинства, внедренный и связанный. Самым младшим в стилевой иерархии является стиль «по умолчанию».

Пример

```
<FONT color="red">
```

```
<P style="color:blue">
```

Это будет показано синим.

```
</P>
```

```
</FONT>
```

Это будет показано синим.

```
<P style="color:blue">
```

Это будет показано синим, а

```
<FONT color="red">
```

это будет показано красным.

```
</FONT>
```

```
</P>
```

Это будет показано синим , а это будет показано красным.

Классы

Стилевые определения можно описывать без указания тега. В этом случае каждому определению присваивается имя, которое можно использовать для сопоставления заданного стиля конкретному тегу. Такие стилевые определения называются *классами*.

Класс записывается следующим образом:

.ИМЯ

{

характеристика: величина;

...

характеристика: величина;

}

***Например, можно определить
стилевой класс с именем "def":***

```
.def  
{  
  color:red;  
  font-family:Geneva, Helvetica, sans-serif;  
}
```


Теги DIV и SPAN

Эти теги позволяют выделять в документе отдельные области, задавая для них специфические свойства.

После блока `<DIV>...</DIV>` браузер переходит на новую строку, а после блока `...` -- нет.

Пример. Использование тега

**

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

```
  BODY { font-family: Arial, sans-serif;} .letter {  
  color: red; font-size: 200%; font-family: serif;  
  position: relative; top: 5px;} </style> </head>
```

```
<body>
```

```
<p><span class="letter">Р</span>азумные  
  люди приспособляются к окружающему  
  миру.</p>
```

```
</body>
```

Результат отображения:

Разумные люди приспосабливаются к окружающему миру.

Пример. Использование тега

<div>

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

```
.block1 { width: 200px; background: #ccc; padding: 5px;
padding-right: 20px; border: solid 1px black; float: left; }
```

```
.block2 { width: 200px; background: #fc0; padding: 5px;
border: solid 1px black; float: left; position: relative; top:
40px; left: -70px; } </style>
```

```
</head>
```

```
<body>
```

```
<div class="block1">Разумные люди
приспосабливаются к окружающему миру.</div>
```

```
<div class="block2">Неразумные люди
приспосабливают мир к себе.</div>
```

```
</body>
```

Результат отображения:

Разумные люди
приспосабливаются к
окружающему миру.

Неразумные люди
приспосабливают мир к
себе..

Абсолютное позиционирование

Используя CSS можно отображать элементы на экране используя реальные координаты, отсчитываемые от левого верхнего угла окна браузера. Такую возможность предоставляет стилевое свойство `position` со значением `absolute`. Сами координаты задаются при помощи свойств `left` (горизонтальная координата) и `top` (вертикальная координата).

Пример

```
<HTML><HEAD>
```

```
  <META http-equiv="Content-Type" content="text/html;  
    charset=windows-1251"><TITLE>Тег DIV</TITLE>
```

```
<STYLE type="text/css"><!--
```

```
.area1 { position: absolute; top:10; left:10;  
color:red;font-weight:bolder; font-size:40pt; background:aqua }
```

```
.area2 { position: absolute; top:20; left:100;  
color:maroon;background:#CFB597; padding: 12pt; }
```

```
.area3 { position: absolute; top:70; left:90;  
color:blue;background:#C0C0C0; padding: 12pt; }
```

```
--> </STYLE>
```

```
</HEAD> <BODY bgcolor=white text=black>
```

```
  <DIV class=area1>Где</DIV>
```

```
  <DIV class=area2>начало того тега,</DIV>
```

```
  <DIV class=area3>которым оканчивается начало?</DIV>
```

```
</BODY></HTML>
```

Результат отображения:

ГДЕ

НАЧАЛО ТОГО ТЕГА,

КОТОРЫМ ОКОНЧИВАЕТСЯ НАЧАЛО?

Z-index

Это стилевое свойство позволяет указывать в каком слое (на каком уровне) находится элемент на экране. Номер основного уровня (на который выводятся обычные элементы без стилизованных указаний) равен нулю. Следовательно, элементы с отрицательным z-index расположены ниже (дальше), с положительным -- выше (ближе) основного экранного слоя. Если элементы имеют одинаковый z-index, то они расположены в одном слое. В противном случае, "ближе" к нам расположен слой, имеющий больший z-index.

Пример

```
<HTML>
<HEAD></HEAD>
<BODY bgcolor=white text=black>
  <DIV style="position:absolute; top:50; left:260;
    height:30; width:100; background:blue;
    font-size:30; color:white; z-index:1">
    Это дальше</DIV>
  <DIV style="position:absolute; top:50; left:110;
    height:130; width:100; background:red;
    font-size:60; color:white; z-index:3">
    Это ближе</DIV>
</BODY>
</HTML>
```

Результат отображения:

