

Каскадные таблицы стилей(CSS)

Что же такое стиль?

```
Селектор {  
Правило 1;  
Правило 2;  
}
```

```
p {  
color:red;  
font-size:60px;  
}
```

Виды стилей

1) Внедренные в страницу HTML5

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title></title>
```

```
<meta content="">
```

```
<style></style>
```

```
</head>
```

```
<body></body>
```

```
</html>
```

2) Встроенные в страницу HTML5

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title></title>
```

```
<meta content="">
```

```
</head>
```

```
<body></body>
```

```
</html>
```

3) Внешние

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title></title>
```

```
<link rel='stylesheet'  
href='css/contact.css'/>
```

```
</head>
```

```
<body></body>
```

```
</html>
```

Типы селекторов

1) универсальный

2) типа

3) класса

4) ID-селектор

5) потомков

6) селектор псевдоклассов

Универсальный селектор

* - применяется ко всем

```
*{
```

```
color:orange;
```

```
}
```

Селектор типа(тега)

```
p {  
  text-align: center;  
}
```

Селектор класса

```
.blue {  
    color:blue;  
}
```

ID селектор

```
#abzac {  
    font-weight:bold;  
}
```

Селектор потомков

```
div > p {  
    color:green;  
}
```

Селектор псевдоклассов

```
а { /*изначальный цвет ссылки*/  
    color:red;  
}  
а:visited { /*цвет посещенной ссылки*/  
    color:black;  
}  
а:hover { /*ссылка будет менять цвет при наведении на нее*/  
    color:green  
}
```

[Пример](#)

Цвета в CSS

1) `color:rgb(0.0.255);`

2) `color:blue;`

3) `color:#fff; color:#ffffff;`

[Цветовая
палитра](#)

Создание блока

border:ширина стиль рамки цвет;

```
div {  
    width:200px;  
    height:200px;  
    border:2px solid gold;  
}
```

Также есть такие правила для
блоков

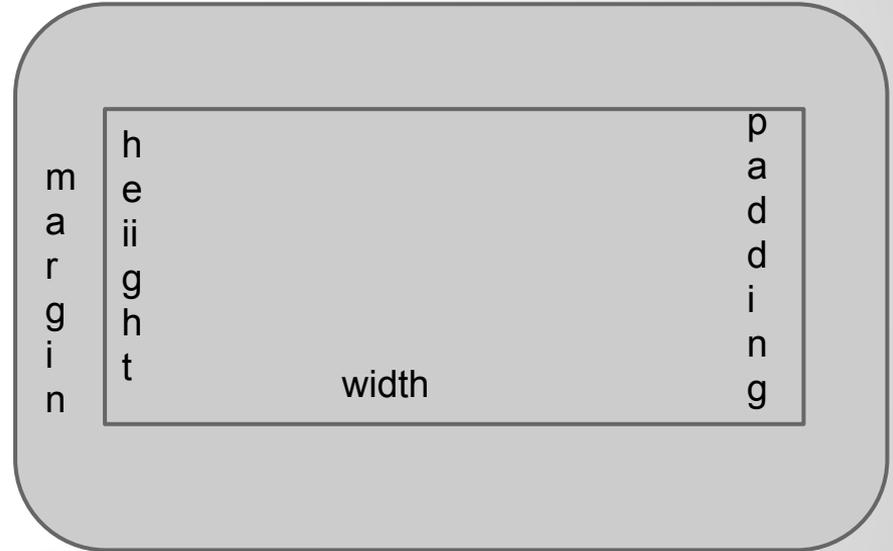
border-width;/*ширина рамки*/
border-style;/*стиль рамки*/
border-color/*цвет рамки*/
border-(top,left,bottom,right)
border-radius/*скругленные края*/

[Пример](#)

margin, padding

```
.first {  
  width:200px;  
  height:200px;  
  border:2px solid green;  
  margin:200px 100px;  
}  
  .two {  
    width:200px;  
    height:200px;  
    border:2px solid red;  
    padding:200px 100px;  
  }
```

[Пример](#)



Стили для текста

font-family:verdana,Tahoma;-шрифт

font-size:12pt-размер шрифта

font-weight:bold;-жирность

font-style:(oblique-наклонный;italic-курсив);

text-indent:40px;красная строка

text-decoration:(none,overline;underline,line-through) подчеркивание

text-align:(center;right;center;left) выравнивание

[Пример](#)

Добавление фона на сайт

```
body {  
  background: #00ff00 url("smiley.gif") no-repeat fixed center;  
}  
h1 {  
  background-color: #00ff00;  
}  
div {  
  background-image: url("paper.gif");  
  background-color: #cccccc;  
}
```

[Пример](#)

position

Синтаксис

position: absolute | fixed | relative | static | inherit

Значения

absolute

Указывает, что элемент абсолютно позиционирован, при этом другие элементы отображаются на веб-странице словно абсолютно позиционированного элемента и нет. Положение элемента задается свойствами `left`, `top`, `right` и `bottom`, также на положение влияет значение свойства `position` родительского элемента. Так, если у родителя значение `position` установлено как `static` или родителя нет, то отсчет координат ведется от края окна браузера. Если у родителя значение `position` задано как `fixed`, `relative` или `absolute`, то отсчет координат ведется от края родительского элемента.

position

fixed

По своему действию это значение близко к `absolute`, но в отличие от него привязывается к указанной свойствами `left`, `top`, `right` и `bottom` точке на экране и не меняет своего положения при прокрутке веб-страницы. Браузер Firefox вообще не отображает полосы прокрутки, если положение элемента задано фиксированным, и оно не помещается целиком в окно браузера. В браузере Opera хотя и показываются полосы прокрутки, но они никак не влияют на позицию элемента.

position

relative

Положение элемента устанавливается относительно его исходного места. Добавление свойств `left`, `top`, `right` и `bottom` изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения.

position

static

Элементы отображаются как обычно. Использование свойств left, top, right и bottom не приводит к каким-либо результатам.

inherit

Наследует значение родителя ■

Пример

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>position</title>
    <style>
      .layer1 {
        position: relative; /* Относительное позиционирование */
        background: #f0f0f0; /* Цвет фона */
        height: 200px; /* Высота блока */
      }
      .layer2 {
        position: absolute; /* Абсолютное позиционирование */
        bottom: 15px; /* Положение от нижнего края */
        right: 15px; /* Положение от правого края */
        line-height: 1px;
      }
    </style>
  </head>
  <body>
    <div class="layer1">
      <div class="layer2">
        
      </div>
    </div>
  </body>
</html>
```

