

CSS Intro

<<Cascading Style Sheets>>

CSS

Каскадные таблицы стилей (**CSS**) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки

Преимущественно используется как средство описания, оформления внешнего вида **веб-страниц**, написанных с помощью языков разметки **HTML** и **XHTML**, но может также применяться к любым **XML-документам**

Основной целью разработки **CSS** являлось **разделение** описания логической структуры веб-страницы (которое производится с помощью **HTML** или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка **CSS**)

Такое разделение может **увеличить** доступность документа, предоставить **большую гибкость** и возможность управления его представлением, а также **уменьшить сложность** и **повторяемость** в структурном содержимом

Подключение

– описание стилей находится в отдельном файле

```
<head>  
  <link rel="stylesheet" href="style.css">  
</head>
```

– стили описаны внутри документа

```
<head>  
  <style>  
    body{  
      color:grey;  
    }  
  </style>  
</head>
```

– стили описаны в атрибутах отдельного элемента

```
<body>  
<p style="font-size: 20px; color: red">.....</p>  
</body>
```

Правила построения CSS

каждое правило **CSS** из файла имеет две основные части — **селектор** и **блок объявлений**.

Селектор, расположенный в левой части правила до знака «{» определяет, на какие части документа (возможно, специально обозначенные) распространяется правило.

Блок объявлений располагается в правой части правила. Он помещается в фигурные скобки, и, в свою очередь, состоит из одного или более объявлений, разделенных знаком «;».

Каждое объявление представляет собой сочетание **свойства CSS** и **значения**, разделенных знаком «:».

Селекторы могут группироваться в одной строке через запятую. В таком случае свойство применяется к каждому из них.

```
селектор, селектор {  
    свойство: значение;  
    свойство: значение;  
    свойство: значение;  
}
```

Виды селекторов

- универсальный селектор → `* {color: red}`
- селектор тегов → `p, h2 {color: green}`
- селектор классов → `.className {color: blue}`
- селектор идентификаторов → `#idName {color: yellow}`
- селектор атрибутов → `a[href="https://tel-ran.com"] {color: magenta}`
- селектор потомков (контекстный селектор) → `div#divId p.myClass {color: grey}`
- селектор дочерних элементов → `p.myClass > b {color: red}`
- селектор элементов одного уровня → `h1 + p {font-size: 40px}`
- селектор псевдоклассов → `a:active {text-decoration: line-through}`
- селектор псевдоэлементов → `p::first-letter {font-size: 16px}`

Наследование и Каскадирование

Применение **CSS** к документам **HTML** основано на принципах **наследования** и **каскадирования**.

Принцип **наследования** заключается в том, что свойства **CSS** объявленные для элементов-предков, почти всегда, наследуются элементами-потомками.

Принцип **каскадирования** применяется в случае, когда какому-то элементу **HTML** одновременно поставлено в соответствие более одного правила **CSS**, то есть, когда происходит конфликт значений этих правил. Чтобы разрешить такие конфликты, вводятся правила приоритета.

- стиль браузера
- стиль, заданный пользователем браузера в его настройках
- стиль, заданный непосредственно автором страницы:
- стили, наследуемые в документе элементом от своих предков;
- стили, заданные во внешних таблицах стилей, подключенных к документу;
- стили, заданные непосредственно в контейнерах **style** данного документа;
- стили определён как **style="..."** в теге;
- самым высоким приоритетом обладают стили, объявленные автором страницы или пользователем, с помощью сопроводительного правила **!important**.

Приоритеты селекторов

Селектор тега:	1
Селектор класса:	10
Селектор ID:	100
Inline-стиль:	1000

Когда селектор состоит из нескольких других селекторов, необходимо посчитать их общий вес.

Селектор	ID	Класс	Тег	Общий вес
p	0	0	1	1
.your_class	0	1	0	10
p.your_class	0	1	1	11
#your_id	1	0	0	100
#your_id p	1	0	1	101
#your_id .your_class	1	1	0	110
p a	0	0	2	2
#your_id #my_id .your_class p a	2	1	2	212

Если случилось так, что два селектора имеют одинаковый вес, то приоритет отдается тому стилю, который находится ниже в коде.

Свойства шрифта

font-family: [Arial, Roboto, sans-serif]	семейство (тип) шрифта
font-size: 20 [px, pt, em, cm, in, mm]	размер
font-weight: [normal, bold, 100, 200 ... 600]	жирность
font-style: [normal, italic]	стиль начертания
font: style weight size family	сокращенная запись

```
p{  
  font-family: Roboto, Arial, sans-serif;  
  font-size: 20px;  
  font-weight: 600;  
  font-style: italic;  
}  
  
p{ font: italic 600 20px Roboto,Arial,sans-serif; }
```


Свойства текста

text-align: [center, left, right, justify]	выравнивание
line-height: 1.7	межстрочный интервал
text-decoration: [underline, overline, line-through]	декорация линияей
color: [#FF0000, red, rgb(255,0,0), rgba(100,255,0,0.8)]	цвет текста

```
p{  
  text-align: center;  
  line-height: 1.7;  
  text-decoration: line-through;  
  color: #333333;  
}
```

Фон элемента

background-color: [#FFF, red, rgba(100,255,0,1.0)]	цвет
background-image: url("../img/bg.jpg")	изображение
background-repeat: [no-repeat, repeat-x, repeat-y]	правила повторения изображения
background-position: [left, right,center,100px][top,center,bottom,100px]	позиция изображения
background: color url() h-pos v-pos repeat-rules	сокращенная запись

```
body{  
    background-color: grey;  
    background-image: url("../img/bg.jpg");  
    background-repeat: no-repeat;  
    background-position: 300px center;  
}  
  
body{ background: red url("../img/bg.jpg") 200px top no-repeat; }
```

Размеры для блоков

width: [100px, 100%]	ширина блока
max-width: 100px	максимальная ширина
min-width: 100px	минимальная ширина
height: 200px	высота блока
max-height: 200px	максимальная высота
min-height	минимальная высота

```
div.wrapper{  
  min-width: 100px;  
  max-width: 300px;  
  min-height: 200px;  
  max-height: 400px;  
}
```

Рамка

border-color: [#333, grey]	цвет
border-width: 2px	толщина
border-style: [solid, dotted, dashed]	вид
border-radius: 5px 5px 5px 5px	радиус скругления углов
border-[top,left,bottom,right]: width style color	сокращенная запись для одной стороны
border: width style color	сокращенная запись для всей рамки

```
div.container{
  width: 200px;
  height: 200px;
  border-color: red;
  border-width: 5px;
  border-style: dotted;
  border-radius: 30px 0 30px 0;
}

div.container{
  border: green solid 5px;
  border-radius: 30px 30px 0 0 ;
}
```

Отступы

margin-[top,bottom,left,right]: [30px,auto]	Внешний отступ с одной стороны
margin: 30px 0 30px 0	Внешний отступ сокращенная запись
padding-[top, bottom, left, right]: 20px	Внутренний отступ с одной стороны
padding: 20px 25px 0 0;	Внутренний отступ сокращенная запись

```
div.content{  
  margin: 20px 35px;  
  padding: 20px 40px;  
}
```

Схлопывание отступов

У блочных элементов расположенных рядом друг с другом по вертикали наблюдается эффект схлопывания, когда отступы не суммируются, а объединяются между собой. Само схлопывание действует на два и более блока (один может быть вложен внутрь другого) с отступами сверху или снизу, при этом примыкающие отступы комбинируются в один. Для отступов слева и справа схлопывание никогда не применяется.

Схлопывание не срабатывает:

- для элементов, у которых на стороне схлопывания задано свойство **padding**.
- для элементов, у которых на стороне схлопывания задана граница;
- на элементах с абсолютным позиционированием, т.е. таких, у которых **position** установлено как **absolute**;
- на плавающих элементах (для них свойство **float** задано как **left** или **right**);
- для строчных элементов;
- для тега **<html>**.