

# Каскадные таблицы стилей

## [3.0 Введение](#)

## [3.1 Селекторы и блоки объявлений](#)

## [3.2 Свойства шрифта](#)

## [3.3 Свойства цвета](#)

## [3.4 Списки](#)

## [3.5 Выравнивание текста](#)

## [3.6 Фоновые изображения](#)

## [3.7 Блочная верстка](#)

### [3.7.1 Рамки](#)

### [3.7.2 Внешние и внутренние отступы](#)

## 3.0 Введение

### ЧТО ЭТО ТАКОЕ

Каскадные таблицы стилей (или сокращенно CSS – от английского Cascading Style Sheets) используются для оформления веб-страниц. Иными словами, они определяют внешний вид документа, созданного при помощи языка HTML.

Средства языка HTML также позволяют задавать позиционирование элементов (с помощью таблиц) и их стили (с помощью тегов шрифта и т. д.). Однако эти задачи гораздо удобнее решать с помощью каскадных таблиц стилей, поскольку такой способ позволяет разделять код содержания (HTML) и код оформления (CSS).

### ПРИМЕНЕНИЕ

Существует три вида CSS.

- **Встроенные стили**
  - Такие таблицы стилей (style sheets) применяются к контенту одного элемента и определяются внутри тега этого элемента.
  - Для этого используется атрибут style:  

```
<p style="color: red;">Это CSS!</p>
```

### Пример

```
<!DOCTYPE html>
<html>

<head>
<title>Моя страница</title>
</head>

<body>
<p style="color:red">текст</p>
</body>

</html>
```

- Встроенные стили применять не рекомендуется, поскольку при этом смешивается код CSS и HTML, а это затрудняет работу веб-мастера и увеличивает объем загружаемых страниц.

- **Внутренние стили**

- Такие таблицы стилей (style sheets) применяются ко всему документу.
- Они указываются с помощью элемента style между тегами <head> документа:

```
<style>
p {
color: red;
}
</style>
```

### Пример

```
<!DOCTYPE html>
<html>

<head>
<title>Моя страница</title>

<style type="text/css">
p {
color: red;
}
</style>
```

```
</head>

<body>
<p>текст</p>
</body>

</html>
```

- Преимущество такого подхода заключается в том, что стили и правила CSS отделены от HTML-кода. Недостаток же этого способа в том, что стили описываются внутри файла HTML, что делает их недоступными для других документов.

- **Внешние стили**

- Такие таблицы стилей (style sheets) можно применять к любому количеству файлов HTML.
- Внешние таблицы стилей подключаются с помощью тега `<link>`, который добавляется в тег `<head>`:

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

- К документу можно подключить несколько таблиц стилей (style sheets):

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

**Пример**

```
<!DOCTYPE html>
<html>

<head>
<title>Моя страница</title>
<link rel="stylesheet" href="my_styles.css">
```

```
</head>

<body>
<p>текст</p>
</body>

</html>
```

- Использование внешнего файла для стилей позволяет разделить код CSS и HTML. Это упрощает понимание и редактирование кода CSS и позволяет подключать его к разным документам. При этом браузеру не нужно несколько раз загружать код CSS в разных файлах: он будет подключаться из файла с таблицами стилей, сохраненного в кеше.

СОВЕТ. Каждый браузер имеет собственную таблицу стилей (style sheet), используемых по умолчанию, например для размера шрифта в заголовках или для отступов до и после абзаца. Полный список таких таблиц можно найти [здесь](#).

## ГЛОССАРИЙ

Каскадные таблицы стилей (Cascading Style Sheets) удобно использовать для определения приоритета в случае конфликта правил для одного или нескольких элементов.

## 3.1 Селекторы и блоки объявлений

### ПРИМЕНЕНИЕ

Каждое правило CSS состоит из трех частей:

1. селектор;
2. свойства;
3. значения.

Селектор показывает, к каким элементам применяется правило.

Свойство (color – цвет) и значение (red – красный) образуют блок объявления стиля.

Например, можно установить красный текст для всех элементов p с помощью такого правила:

```
p { color: red; }
```

В этом примере p – селектор,

a color – свойство. После свойства всегда ставится двоеточие (:).

red – это значение. После значения всегда ставится точка с запятой (;).

Правило может содержать несколько блоков объявлений:

```
p {color: red; font-size: 12px;}
```

Для удобства чтения блоки объявлений рекомендуется размещать на разных строках:

```
p {  
    color: red;  
    font-size: 12px;  
}
```

**СОВЕТ.** Ставьте точку с запятой после каждого блока объявлений, даже если он один. Это поможет избежать ошибок при добавлении новых блоков.

## ПРИМЕНЕНИЕ

Существует несколько видов селекторов:

### ПОПРОБУЙТЕ

1. Название элемента.

Значение свойства применяется ко всем именованным элементам.

```
h1 {color: red;}
```

Одно и то же значение можно присвоить нескольким элементам:

```
h2, h3 {color: green;}
```

2. Селектор классов.

Выбор элементов по атрибуту class выполняется так:

HTML:

```
<p class="highlight">Ой-ой-ой!</p>
```

CSS:

```
p.highlight {  
    color: red;  
    background-color: yellow;  
}
```

Чтобы применить значения к нескольким элементам одного класса, опустите название элемента:

#### HTML:

```
<p class="highlight">Ой-ой-ой!</p>
...
<blockquote class="highlight">Какая-нибудь
цитата.</blockquote>
```

#### CSS:

```
.highlight {
    color: red;
    background-color: yellow;
}
```

### 3. Селектор идентификаторов

Применение стиля к одному элементу.

```
#intro {
    font-size: 18pt;
}

<p id="intro">Размер шрифта - 18 пунктов.</p>
```

### 4. Универсальный селектор

Универсальный селектор применяется ко всем элементам в документе и обозначается звездочкой (\*):

```
* {
    color: yellow;
}
```

**ВНИМАНИЕ!** Универсальные селекторы использовать не рекомендуется. Это может снизить эффективность и затруднить отладку кода CSS, поскольку правила применяются ко всем элементам.

## ГЛОССАРИЙ

**Селекторы** используются для поиска элементов HTML, к которым нужно применить стиль. Это может быть название элемента или класса либо идентификатор: `h1`, `#intro`, `.warning` и т. д.

**Блоки объявления стиля** состоят из двух частей – свойства и его значения, например

```
color: red.
```

**СОВЕТ.** Чтобы код CSS было удобно читать, используйте комментарии. Они добавляются в таком формате:

```
/* Это комментарий. Его формат не имеет значения. */
```

## 3.2 Свойства шрифта

### ЧТО ЭТО ТАКОЕ

В веб-документах используются различные размеры, семейства (Arial, Helvetica и т. д.), стили (полужирный, курсив, подчеркнутый) и другие свойства шрифтов.

### ПОПРОБУЙТЕ

1. Шрифты указываются в порядке предпочтения:

```
font-family: Arial, Helvetica, Futura, sans-serif
```

2. Размер шрифта указывается так:

```
font-size: 14pt
```

3. Выделение текста полужирным курсивом:

```
font-style: italic
```

4. Выделение текста полужирным шрифтом:

```
font-weight: bold
```

Собираательные свойства

С помощью свойства `font` можно указать сразу несколько значений:

```
font: bold 12pt Arial
```

### ГЛОССАРИЙ

Свойство `font-family` определяет семейство или гарнитуру шрифта (Helvetica, Arial и т. д.).

Свойство `font-size` задает размер текста.

Свойство `font-style` определяет стиль текста (курсив, полужирный или обычный).

## 3.3 Свойства цвета

### ЧТО ЭТО ТАКОЕ

В CSS доступны 16 стандартных цветов: `aqua` (голубой), `black` (черный), `blue` (синий), `fuchsia`

(розовый), gray (серый), green (зеленый), lime (салатный), maroon (коричневый), navy (темно-синий), olive (оливковый), purple (пурпурный), red (красный), silver (серебристый), teal (морской волны), white (белый) и yellow (желтый). Вместо них можно использовать числовые значения, которые показывают соотношение красной, зеленой и синей составляющих цвета. Ниже приведены три варианта определения красного цвета. В первом используются числа в десятичной системе счисления, а во втором – в шестнадцатеричной (FF в шестнадцатеричной соответствует 255 в десятичной системе счисления).

```
h1 {color: rgb(255, 0, 0);}
h1 {color: #FF0000;}
h1 {color: red;}
```

Вы также можете задать цвет фона элемента. В примере ниже показано, как задать красный цвет текста на желтом фоне для элемента h1.

```
h1 {
  background-color: yellow;
  color: red;
}
```

## ПОПРОБУЙТЕ

```
p.bright {color: red; background-color: yellow}
<p> Красный шрифт на желтом фоне. </p>

h1 {
    color: red;
}
<h1>Заголовок красного цвета.</h1>
```

## ГЛОССАРИЙ

Свойство color определяет цвет элемента. Если элемент содержит текст, то это свойство задает цвет текста на переднем плане. В качестве значений можно использовать названия цветов (black, red и др.), а также их шестнадцатеричные или десятичные коды, например #ff0000 или rgb(255, 0, 0).

Свойство background-color определяет цвет фона элемента.

## 3.4 Выравнивание текста и позиционирование

### Выравнивание текста



## ЧТО ЭТО ТАКОЕ

Текст можно выравнивать по левому краю (по умолчанию), по правому, по центру или по обоим краям. Также можно задать отступ для первой строки абзаца.

В CSS можно использовать различные единицы измерения для позиционирования элементов текста на экране или бумаге: Это могут быть, например, пиксели (px), сантиметры (cm), дюймы (in) и др.

## КАК

Свойство text-align определяет выравнивание текста, а text-indent – отступ первой строки.

## ПОПРОБУЙТЕ

```
<style>

                .body {
text-align: justified;
text-indent: 10px; }
</style>
```

```
<p class="body">
```

Отступ от края в первой строке этого абзаца составляет 10 пикселей. Все следующие строки абзаца не имеют отступов и выровнены по обоим краям.

```
</p>
```

## Позиция

### ЧТО ЭТО ТАКОЕ

Свойство position определяет один из четырех методов позиционирования элемента.

#### 1. Статическое позиционирование

Этот метод используется по умолчанию. Такие элементы размещаются на странице так, как это указано в коде HTML.

#### 2. Фиксированное позиционирование

При прокрутке страницы такой элемент остается на месте.

#### 3. Относительное позиционирование

Размещение элемента рассчитывается относительно его нормального положения.

#### 4. Абсолютное позиционирование

Положение отсчитывается от ближайшего родительского элемента, к которому применен любой метод позиционирования, кроме статического. Если такого родительского элемента нет, то точкой отсчета служит элемент <html>. Такой элемент при прокрутке перемещается вместе с документом.

## ПОПРОБУЙТЕ

```
<style>

                .box1 {
position: relative;
```

```
        left: 10px; }
.box2 {
        position: fixed;
        bottom: 20px; }
</style>

<p class="box1">
    Отступ от края в первой строке этого абзаца
составляет 10 пикселей.
</p>
<p class="box2">
    Этот абзац остается на месте при прокрутке
страницы, а отступ от края в его первой строке составляет
20 пикселей.
</p>
```

## ГЛОССАРИЙ

Свойство `text-indent` определяет величину отступа от левого края для первой строки абзаца.

Свойство `text-align` определяет выравнивание текста в элементе.

## 3.5 Фоновые изображения

### ЧТО ЭТО ТАКОЕ

Свойство `background-image` задает фоновое изображение элемента.

### ПРИМЕНЕНИЕ

В свойстве `background-image` нужно указать путь к файлу:

```
h1 {background-image: url(/images/gradient.png);}
```

**СОВЕТ.** Пути к файлам фоновых изображений указываются относительно документа, содержащего правило. Поэтому размещайте графические файлы в той же папке, что и файл CSS. Например, если вы задали изображение `background-image: url(gradient.png)` в файле `global.css`, то браузер будет искать файл `gradient.png` в том каталоге, где находится `global.css`.

### ПОПРОБУЙТЕ

Примените фоновое изображение к элементу `body` веб-документа.

## 3.6 Каскадирование (Cascade)

### ЧТО ЭТО ТАКОЕ

Иногда для одного элемента CSS создается несколько конфликтующих правил. Например, для всех элементов в теге `<p>` одно правило может задавать розовый цвет, а другое – синий. В таких случаях приходится определять, какое из них имеет более высокий приоритет. Для этого используется каскадирование (cascade).

### ПРИМЕНЕНИЕ

Для каждого элемента может использоваться несколько блоков объявлений, а каскадирование (cascading) позволяет определить, какой из них будет применен в конечном итоге. В случае конфликта объявлений используется последнее объявление с самой узкой областью применения.

### ПОПРОБУЙТЕ

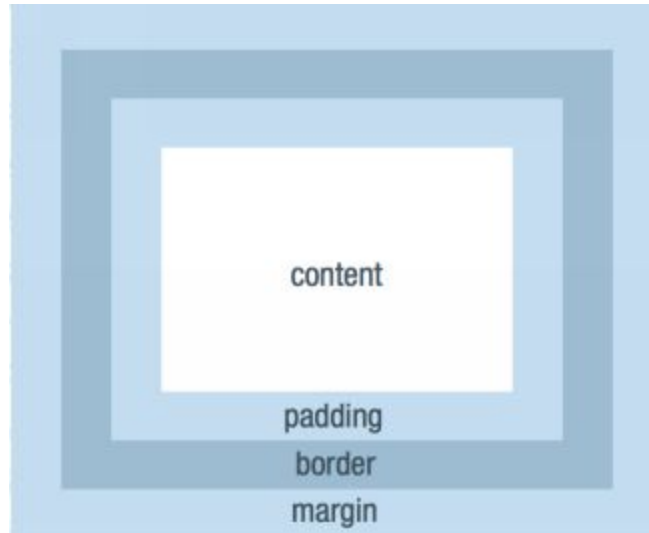
```
<style>
    div {
        color: blue;
        font-style: italic; }
    p {
        color: pink; }
</style>

<div>
<p> Этот розовый шрифт. </p>
</div>
```

## 3.7 Блочная верстка

### ЧТО ЭТО ТАКОЕ

Представьте себе документ, в котором каждый элемент представляет собой прямоугольник. Такое структурирование называется блочной версткой. В этом случае каждый элемент HTML как бы заключен в прямоугольный контейнер, состоящий из четырех частей: внешнего отступа (margin), рамки (border), внутреннего отступа (padding) и содержания (content). Схема такого блока показана этом рисунке:



Содержание (content) – это фактический контент элемента HTML, например текст или изображение.

Внутренний отступ (padding) – это область, прилегающая к содержанию.

Внутренний отступ окружен рамкой (border). Для нее можно использовать свойства `border-style`, `border-width` и `border-color`.

Внешний отступ (margin) – это пространство (справа, слева, сверху и снизу) между двумя соседними элементами.

### 3.7.1 Рамки

#### ЧТО ЭТО ТАКОЕ

Свойства `border` определяют наличие или отсутствие рамки у элемента, а также ее стиль. К параметрам рамки относятся стиль, цвет, ширина и другие.

#### ПРИМЕНЕНИЕ

Для каждой стороны рамки (правой, левой, верхней и нижней) можно указать собственный стиль, цвет и ширину. Ниже перечислены три основных свойства рамок.

- **border-style**
  - Это свойство определяет стиль рамки: *dotted* (рамка из точек), *dashed* (пунктирная) или *double* (двойная).
  - По умолчанию для свойства `border-style` используется значение `none` (рамка отсутствует).
  - Стиль каждой из четырех сторон элемента определяется свойствами `border-top-style`, `border-bottom-style`, `border-left-style` и `border-right-style` (для верхней, нижней, левой и правой соответственно).
- **border-width**
  - Это свойство задает толщину рамки. Возможные значения: `thin` (тонкая), `medium` (средняя), `thick` (толстая) или числовое (в пикселях).

- По умолчанию для элемента border-width используется значение medium.
- Толщина каждой из четырех сторон элемента определяется свойствами border-top-width, border-bottom-width, border-left-width и border-right-width (для верхней, нижней, левой и правой соответственно).
- **border-color**
  - Это свойство задает цвет рамки. В качестве значений можно использовать названия стандартных цветов и шестнадцатеричные коды.
  - Значение по умолчанию – black.
  - Стиль каждой из четырех сторон элемента определяется свойствами border-top-color, border-bottom-color, border-left-color and border-right-color (для верхней, нижней, левой и правой соответственно).

## ПОПРОБУЙТЕ

```
<head>
<link rel="stylesheet" type="text/css" href="boxmodel.css">
</head>
```

*boxmodel.css:*

```
table {
border-top-width: thin;
border-bottom-width: thick;
border-top-color: red;
border-bottom-color: yellow;
border-top-style: dotted;
}

p {
border-width: thick;
border-color: aqua;
border-style: dashed;
}
```

Теперь добавьте таблицу и элемент p (параграф), сохраните файл и откройте его в браузере, чтобы просмотреть заданные свойства рамки.

## ГЛОССАРИЙ

Рамка – область между внешним и внутренним отступами.

Свойство border-style определяет линии рамки (например, рамка в виде пунктирной или двойной линии).

Свойство border-width задает толщину рамки.

Свойство border-color задает цвет рамки.

## 3.7.2 Внешние и внутренние отступы

### ЧТО ЭТО ТАКОЕ

Пространство между содержанием элемента и его рамкой называется внутренним отступом.

Пространство между рамкой элемента и соседним элементом (справа, слева, сверху или снизу) называется внешним отступом.

### ПРИМЕНЕНИЕ

Свойства внешнего отступа содержат слово `margin`, свойства внутреннего – слово `padding`. Эти свойства можно применить к четырем сторонам изображения: верхней, нижней, левой и правой.

Таким образом, для внешнего отступа элемента используются свойства `margin-top`, `margin-bottom`, `margin-left` и `margin-right`, а для внутреннего отступа – `padding-top`, `padding-bottom`, `padding-left` и `padding-right`.

Величину отступов можно задавать в сантиметрах, пикселях и других единицах измерения.

### ПОПРОБУЙТЕ

```
<head>
<link rel="stylesheet" type="text/css" href="margins.css">
</head>
```

*margins.css:*

```
    p.one {
        margin: 0.1in;
        padding: 0.1in;
        background-color: red;
    }
    p.two {
        margin: 0.2in;
        padding: 0.2in;
        background-color: blue;
    }
    p.three {
        margin: 20px;
        padding: 10px;
        background-color: yellow;
    }
    p.four {
        margin: 40px;
```

```
background-color: red;  
}
```

Теперь добавьте 4 элемента p (параграф) с именами класса one, two, three и four и откройте созданный файл в браузере, чтобы посмотреть, как выглядят внутренний и внешний отступы. Не забудьте сохранить проект, прежде чем перейти к следующему разделу.

## **ГЛОССАРИЙ**

Внешний отступ (margin) – это пространство между содержанием элемента и соседним элементом.

Внутренний отступ (padding) – это пространство между элементом и его рамкой.