

CSS





CSS-списки — набор свойств, отвечающих за оформление списков. Использование HTML-списков очень распространено при создании панелей навигации по сайту. Элементы списка представляют набор блочных элементов.

С помощью стандартных CSS-свойств можно **изменить внешний вид маркера списка, добавить изображение для маркера, а также изменить местоположение маркера**. Высоту блока маркера можно задать свойством `line-height`.

Тип маркера списка `list-style-type`

Свойство изменяет типа маркера или удаляет маркер для маркированного и нумерованного списков. Наследуется.

- `disc` - Значение по умолчанию. В качестве маркера элементов списка выступает закрашенный кружок.
- `circle` - В качестве маркера выступает незакрашенный кружок.
- `decimal` - нумерованный список 1, 2, 3, 4, 5, ...
- `decimal-leading-zero` - нумерованный список с 0 перед цифрой 01, 02, 03, 04, 05, ...
- `lower-alpha` - a, b, c, d, e, ... маленькие буквы
- `upper-alpha` - A, B, C, D, E, ...

Синтаксис:

- `ul {list-style-type: none;} ul {list-style-type: square;}`
- `ol {list-style-type: none;} ol {list-style-type: lower-alpha;}`

CSS-списки: Изображения для списка `list-style-image`

В качестве маркера элементов списка можно использовать изображения и градиентные заливки. Наследуется.

Можно задать через `url` изображение для списка

Синтаксис

```
ul { list-style-image: url("images/romb.png"); }
```

```
ul { list-style-image: linear-gradient(#FF7A2F 0, #FF7A2F 50%, #FFB214 50%); }
```

CSS-списки: Краткая форма стилей списка

Можно объединить все три свойства форматирования списка в одно с помощью `list-style`. Значения свойств могут быть расположены в произвольном порядке, а часть значений может быть опущена. Если присутствует одно значение, то другие свойства примут значения браузера по умолчанию.

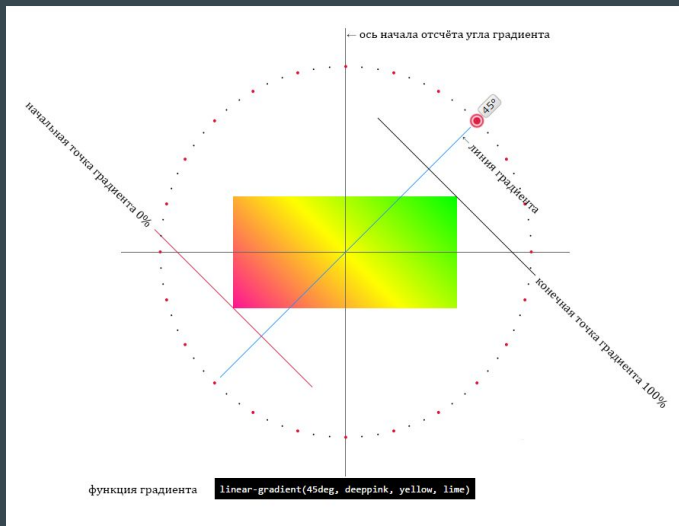
```
ul { list-style: url("images/romb.png") inside; }
```

CSS-градиент

Линейный градиент создается с помощью двух и более цветов, для которых задано направление, или **линия градиента**.

Если направление не указано, используется значение по умолчанию — **сверху-вниз**.

Цвета градиента по умолчанию распределяются равномерно в направлении, перпендикулярном линии градиента.



```
{ background: linear-gradient(угол / пары ключевых слов, первый цвет, второй цвет и т.д.); }
```

Радиальный градиент

Радиальный градиент отличается от линейного тем, что цвета выходят из одной точки (центра градиента) и равномерно распределяются наружу, рисуя форму круга или эллипса.

`background: radial-gradient` (форма / позиция центра, первый цвет, второй цвет);

Форма градиента определяется ключевыми словами circle или ellipse. Если форма не задана, по умолчанию радиальный градиент принимает форму эллипса.

Радиальный градиент

Позиция центра задается с помощью ключевых слов, используемых в свойстве `background-position`, с добавлением приставки `at`. Если позиция центра не задана, используется значение по умолчанию `at center`.

```
background: radial-gradient(at top, #ffffff, #a7cecc);
```

С помощью пары значений, указанных в единицах длины `%`, `em` или `px`, можно управлять размером эллипсообразного градиента. Первое значение задает ширину эллипса, второе — высоту.

```
background: radial-gradient(at top, #ffffff, #a7cecc);
```


Линейный градиент

Направление градиента может быть задано двумя способами:

с помощью угла наклона в градусах `deg`, значение которого определяет угол наклона линии внутри элемента.

```
div { background: linear-gradient (45deg, #EECFBA, #C5DDE8); }
```

с помощью ключевых слов `to top`, `to right`, `to bottom`, `to left`, которые соответствуют углу градиента, равному `0deg`, `90deg`, `180deg` и `270deg` соответственно.

```
div { background: linear-gradient (to right, #EECFBA, #C5DDE8); }
```

CSS content

```
h1:before, h1:after {  
    content: "";  
}
```

Генерируемое содержимое наследует значения свойств от элемента, к которому оно прикрепляется. При этом наследуются только наследуемые свойства.

`content:` `none` | `counter()` | `attr()` | `" "` | `url()` | `open/close-quote` | `no-open/close-quote`

`none` - не добавляет содержимое. Используется в случае, когда нужно удалить генерируемое содержимое для одного элемента из группы элементов (например, элементы списка), для которых уже задано это свойство.

CSS content

counter() - дает возможность создавать счётчики, задавая для них точку отсчёта и приращение на некоторую величину.

attr() - добавляет до или после элемента значение атрибута, заключённого в скобки.

" " - текст, который добавляется на веб-страницу, должен быть заключен в двойные или одинарные кавычки. Пустые кавычки можно использовать для добавления блочного содержимого.

url() - добавляет медиа-содержимое, например, изображение, звук, видео. В качестве значения атрибута в скобках указывается адрес внешнего ресурса, который вставляется в выбранное место документа.

open-quote - добавляет к содержимому открывающую кавычку.

close-quote - добавляет к содержимому закрывающую кавычку.

no-open-quote - удаляет открывающую кавычку.

no-close-quote - удаляет закрывающую кавычку.

CSS content : спец-символы

Добавление специальных символов.

В качестве значения используется символ Юникода. Перед юникодом нужно сделать экранирование с помощью бэкслеша “\”.

```
h1:before, h1:after { content: "\2746"; }
```

Добавление текста.

В качестве генерируемого содержимого между кавычками можно поместить любой текст, и он появится в указанном месте, при этом текст в кавычках выводится как есть.

```
h1:after, h1:after { content: "BOOOOM!"; }
```

Добавление изображения.

```
h1:before { content: url(smiley.gif); }
```

Добавление значения атрибута.

Функция `attr()` позволяет добавить любое значение атрибута, например, url-адрес ссылки, который будет выводиться при печати текста.

```
a:after { content: attr(href); }
```

content: counter()

Добавление счетчика элементов.

Значение функции `counter()` позволяет пронумеровать любые блочные элементы на веб-странице. Счетчики создаются при помощи CSS-свойств `counter-reset` и `counter-increment`.

Свойство `counter-reset` задает имя одного или нескольких счетчиков, позволяя устанавливать или сбрасывать значение счетчика до любого значения. Свойство используется только в сочетании со вторым свойством — `counter-increment`. Работает для любых HTML-элементов.

content: counter()

имя счетчика/число - имя счетчика выбирает один или несколько элементов, для которых требуется изменить значение счетчика. Число задает начальное значение счетчика. Не допускается использование ключевых слов в качестве имени счетчика. Если нужно установить нумерацию с единицы, необходимо задать **counter-reset: 0;**. Если нужно, чтобы нумерация начиналась с нуля, то **counter-reset: -1;**.

content: counter()

Свойство **counter-increment** принимает одно или несколько значений, которые определяют имена счетчиков, которые будут приращены.

имя счетчика/число - имя счетчика определяет, какой счетчик будет приращен. Значение по умолчанию **1**. Если задано отрицательное число, то нумерация элементов будет отрицательная. Число задает начальное значение, а также шаг приращения, т.е. если установить **counter-increment: id 2;**, то первый элемент будет нумероваться с **2**, второй — с **4** и т.д. Если идентификатор ссылается на счетчик, который не был инициализирован с помощью **counter-reset**, то начальное значение будет равно **0**.

Значение свойства **content: counter(имя счетчика)** выводит значение счетчика в качестве содержимого псевдоэлемента.