# CSS3-текст

**CSS3-текст** включает свойства для работы с текстом, позволяющие решить проблемы с переносом текста и обрезкой текста в пределах содержимого.

## Обрезка строки text-overflow

Свойство позволяет ограничивать длину текста в случае, когда он не умещается в контейнер, визуально обрезая его или отображая многоточием. Свойство работает только при задании следующих условий: должна быть определена ширина контейнера, элемент должен иметь значения overflow: hidden и white-space: nowrap. Не наследуется.

|  |  |
| --- | --- |
| **text-overflow** | |
| **clip** | Значение по умолчанию. Текст обрезается в пределе области содержимого, при этом может отобразиться лишь часть символа. |
| **ellipsis** | Замещает текст, не уместившийся в блок, с помощью многоточия. |
| **initial** | Устанавливает значение свойства в значение по умолчанию. |
| **inherit** | Наследует значение свойства от родительского элемента. |

**Синтаксис:**

**p {**

**width: 300px;**

**overflow: hidden;**

**white-space: nowrap;**

**text-overflow: ellipsis;**

**}**

**text-overflowПример 1. Обрезка текста с помощью свойства text-overflow**

## Перенос внутри слов word-break

В обычной ситуации слова переносятся на другую строку в местах пробелов или дефисов («мягкий перенос»), или же в случае принудительного переноса с использованием элемента <br>. Данное свойство позволяет установить правила переноса внутри слов для того, чтобы длинные строки полностью заполняли пространство внутри контейнера. Не используется для CJK-языков (китайский-японский-корейский). Наследуется.

|  |  |
| --- | --- |
| **word-break** | |
| **normal** | Значение по умолчанию. Текст в контейнере отображается в привычном для нас виде в соответствии с правилами, установленными в языке. |
| **break-all** | Пробелы в словах могут быть оформлены между любыми двумя буквами. |
| **keep-all** | Запрещает разрывы между парами букв слова. |
| **initial** | Устанавливает значение свойства в значение по умолчанию. |
| **inherit** | Наследует значение свойства от родительского элемента. |

**Синтаксис:**

p {word-break: normal;}

p {word-break: break-all;}

p {word-break: keep-all;}

## Перенос слов в строке word-wrap

Применяется в случаях, когда текст не умещается в свой контейнер и выходит за его границу. Позволяет разрывать длинные слова и переносить их на следующую строку. Наследуется. Срабатывает лишь в случае, если значение white-spaceразрешает разрыв строк.

|  |  |
| --- | --- |
| **word-wrap** | |
| **normal** | Значение по умолчанию. Слова разрываются только по правилам, принятым в языке. |
| **break-word** | Разрыв слов и перенос на следующую строку происходит в месте границы контейнера. |
| **initial** | Устанавливает значение свойства в значение по умолчанию. |
| **inherit** | Наследует значение свойства от родительского элемента. |

**Синтаксис:**

p {word-wrap: normal;}

p {word-wrap: break-word;}

# CSS3-градиент

**CSS3-градиент** представляет собой переходы от одного цвета к другому.

Градиенты создаются с помощью функций **linear-gradient()** и radial-gradient().

Градиентный фон можно устанавливать в свойствах background, background-image, border-image и list-style-image.

## linear-gradient-axisЛинейный градиент linear-gradient()

**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 / сторона или угол наклона с помощью ключевого слова (пары ключевых слов), первый цвет, второй цвет и т.д.);}

**Направление** градиента может быть задано двумя способами:

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

div {

height: 200px;

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 {

height: 200px;

background: linear-gradient(to right, #F6EFD2, #CEAD78);

}

Если направление задано парой ключевых слов, например, to top left, то начальная точка градиента будет расположена в противоположном направлении, в данном случае справа внизу.

div {

height: 200px;

background: linear-gradient(to top left, powderblue, pink);

}

Для неравномерного распределения цветов указывается начальная позиция каждого цвета через точки остановки градиента, так называемые **color stops**. **Точки остановки** задаются в %, где 0% — начальная точка, 100% — конечная точка, например:

div {

height: 200px;

background: linear-gradient(to top, #E4AF9D 20%, #E4E4D8 50%, #A19887 80%);

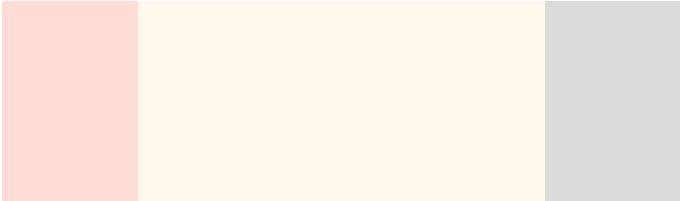
}

Для чёткого распределения цветных полос каждый последующий цвет нужно начинать с точки остановки предыдущего цвета:

div {

height: 200px;

background: linear-gradient(to right, #FFDDD6 20%, #FFF9ED 20%, #FFF9ED 80%, #DBDBDB 80%);

}

## Радиальный градиент radial-gradient()

**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 градиента / размер / позиция центра, первый цвет, второй цвет и т.д.);}

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

div {

height: 200px;

background: radial-gradient(white, #FFA9A1);

}



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

div {

height: 200px;

background: radial-gradient(at top, #FEFFFF, #A7CECC);

}

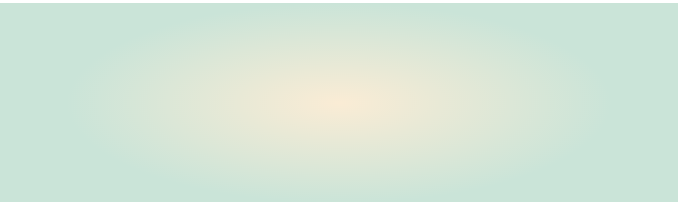
С помощью пары значений, указанных в единицах длины %, em или px, можно управлять размером эллипсообразного градиента. Первое значение задает ширину эллипса, второе — высоту.

div {

height: 200px;

background: radial-gradient(40% 50%, #FAECD5, #CAE4D8);

}

**Размер градиента** задаётся с помощью ключевых слов. Значение по умолчанию farthest-corner (к дальнему углу).

|  |  |
| --- | --- |
| **Значение** | **Описание** |
| **closest-side** | Размер градиента рассчитывается из расстояния до любой ближней стороны блока для circle или до ближних сторон по X и по Удля ellipse. |
| **farthest-side** | Размер рассчитывается из расстояния до дальних сторон. |
| **closest-corner** | Размер рассчитывается из расстояния до ближних углов. |
| **farthest-corner** | Размер рассчитывается из расстояния до дальних углов. |

div {

height: 200px;

background: radial-gradient(circle farthest-corner at 100px 50px, #FBF2EB, #352A3B);

}

С помощью радиального градиента можно создавать реалистичные объёмные фигуры, такие как мячи, кнопки.

## Приложение

## Мяч

div {

width: 200px;

height: 200px;

border-radius: 50%;

margin: 0 auto;

background: radial-gradient(circle at 65% 15%, aqua, darkblue);}

## Кнопка

<div class="wrap"><div class="button"></div></div>

.wrap {

height: 200px;

padding: 50px 0;

background: #cccccc;

}

.button {

width: 100px;

height: 100px;

border-radius: 50%;

margin: 0 auto;

background: radial-gradient(farthest-side ellipse at top left, white, #aaaaaa);

box-shadow: 5px 10px 20px rgba(0,0,0,0.3), -5px -10px 20px rgba(255,255,255,0.5);

}

## Почтовая марка

Используя в градиентах прозрачный цвет, можно создавать вот такие эффекты.

<div class="container">

<div class="wrap">

<img src="https://html5book.ru/wp-content/uploads/2015/10/flowers-by-barbara-florchik-2.jpg">

</div>

</div>

.container {

background: #B7D1D8;

padding: 25px;

}

.wrap {

background: radial-gradient(transparent, transparent 4px, white 4px,white);

padding: 10px;

width: 300px;

height: 220px;

background-size: 20px 20px;

background-position: -10px -10px; /\*обрезаем узор по краю\*/

margin: 0 auto;

}

img {width: 100%;}

## Повтор градиента

В добавление к линейному и радиальному градиентам существует повтор градиента, который задается с помощью функций repeating-linear-gradient() и repeating-radial-gradient() соответственно. Фон из повторяющихся градиентов выглядит неаккуратно, поэтому рекомендуется начинать следующий цвет с точки остановки предыдущего, создавая таким образом полосатые узоры.

div {

height: 200px;

background: repeating-linear-gradient(45deg, #606dbc, #606dbc 10px, #465298 10px, #465298 20px);

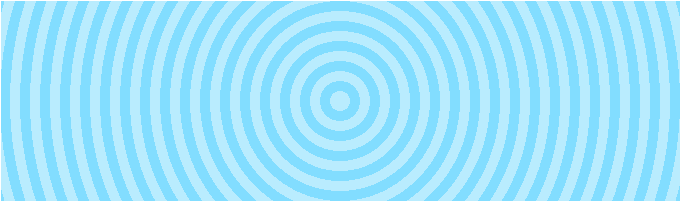
}

div {

height: 200px;

background: repeating-radial-gradient(circle, #B9ECFE, #B9ECFE 10px, #82DDFF 10px, #82DDFF 20px);

}



## Кроссбраузерный градиент

Для корректного отображения градиентов во всех браузерах необходимо добавить кроссбраузерную запись.

**Линейный градиент**

-ms-filter: "progid:DXImageTransform.Microsoft.gradient (GradientType=0, startColorstr=#1471da, endColorstr=#1C85FB)"; /\* IE 8-9 \*/

background: -webkit-linear-gradient(left, red, #f06d06); /\* Safari 5.1, iOS 5.0-6.1, Chrome 10-25, Android 4.0-4.3 \*/

background: -moz-linear-gradient(left, red, #f06d06); /\* Firefox 3.6-15 \*/

background: -o-linear-gradient(left, red, #f06d06); /\* Opera 11.1-12 \*/

background: linear-gradient(to right, red, #f06d06); /\* Opera 15+, Chrome 25+, IE 10+, Firefox 16+, Safari 6.1+, iOS 7+, Android 4.4+ \*/

**Повтор линейного градиента**

background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%); /\* Safari 5.1 - 6.0 \*/

background: -o-repeating-linear-gradient(red, yellow 10%, green 20%); /\* Opera 11.1-12.0 \*/

background: -moz-repeating-linear-gradient(red, yellow 10%, green 20%); /\* Firefox 3.6-15 \*/

background: repeating-linear-gradient(red, yellow 10%, green 20%); /\* Стандартный синтаксис \*/

**Радиальный градиент**

background: -webkit-radial-gradient(red, yellow, green); /\* Safari 5.1-6.0 \*/

background: -o-radial-gradient(red, yellow, green); /\* Opera 11.6-12.0 \*/

background: -moz-radial-gradient(red, yellow, green); /\* Firefox 3.6-15 \*/

background: radial-gradient(red, yellow, green); /\* Стандартный синтаксис \*/

background: -webkit-radial-gradient(60% 55%, farthest-side, red, yellow, black); /\* Safari 5.1-6.0 \*/

background: -o-radial-gradient(60% 55%, farthest-side, red, yellow, black); /\* Opera 11.6-12.0 \*/

background: -moz-radial-gradient(60% 55%, farthest-side, red, yellow, black); /\* Firefox 3.6-15 \*/

background: radial-gradient(farthest-side at 60% 55%, red, yellow, black); /\* Стандартный синтаксис \*/

**Повтор радиального градиента**

background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%); /\* Safari 5.1-6.0 \*/

background: -o-repeating-radial-gradient(red, yellow 10%, green 15%); /\* Opera 11.6-12.0 \*/

background: -moz-repeating-radial-gradient(red, yellow 10%, green 15%); /\* Firefox 3.6-15 \*/

background: repeating-radial-gradient(red, yellow 10%, green 15%); /\* Стандартный синтаксис \*/

## Комбинация градиента и фонового изображения

За счёт комбинации градиента и изображения можно создавать интересные эффекты. Для градиента нужно использовать полупрозрачные цвета, чтобы картинка оставалась видимой.

div {

height: 453px;

background: linear-gradient(45deg, rgba(103, 0, 31, .8), rgba(34, 101, 163, .5)), url(https://html5book.ru/wp-content/uploads/2016/12/photo-8.jpg);

background-size: cover;

}

