# Свойство text-shadow

Сначала познакомимся с текстовыми тенями.

Тени для текста создаются с помощью CSS-свойства **text-shadow**. Оно очень похоже на **box-shadow**, рассмотренное в одной из предыдущих глав, но имеет свои особенности:

свойство **text-shadow** применяется к тексту;

1. форма тени повторяет форму текстовых символов;
2. можно управлять смещением тени, её цветом, а также размытием;
3. нельзя управлять растяжением текстовой тени;
4. можно создавать множественные тени.

Первый параметр свойства **text-shadow** обязателен и задаёт смещение тени по горизонтали относительно текста. Положительное значение этого параметра сдвигает тень вправо, отрицательное — влево.

Второй параметр свойства **text-shadow** тоже обязателен и задаёт смещение тени по вертикали. Положительное значение сдвигает тень вниз, отрицательное — вверх.

Третий параметр свойства **text-shadow** необязателен и задаёт радиус размытия тени. Если этот параметр не задан, то по умолчанию устанавливается равным **0**. Чем больше значение параметра, тем сильнее тень размывается и становится светлее.

Последний параметр свойства **text-shadow** необязателен и задаёт цвет тени.

У текста может быть задано сразу несколько теней. Для этого определения теней нужно перечислить через запятую. При этом тени распределяются по правилу: первая тень в списке — самая верхняя, последняя в списке — самая нижняя.

**text-shadow:**

**1px 1px 1px #111111,**

**2px 2px 2px #222222;**

# Выравнивание размеров

Это делается c помощью свойства **letter-spacing**. Браузер по умолчанию устанавливает межсимвольное расстояние автоматически согласно определенному значению у каждого шрифта

Увеличение или уменьшение значения **letter-spacing** изменит расстояние между символами на заданную величину. Значение задаётся в **px**, **em**, **pt** или других единицах длины CSS.

# Перенос символов

Для этого в CSS существует свойство **word-wrap**. Оно принимает значения: **normal** и **break-word**. В случае **normal** слова переносятся на новую строку по обычным правилам (то есть по пробелам). А при значении **break-word** перенос производится браузером в тех местах, где слова перестают помещаться в контейнер.

# Свойство text-indent

Рассмотрим свойства, с помощью которых можно улучшить читабельность или украсить большие тексты.

И первым свойством будет **text-indent**. Оно устанавливает отступ для первой строки блока текста. С помощью этого свойства удобно делать красную строку в абзацах.

# Колоночная разметка: свойство column-count

Многоколоночная разметка CSS — замечательная возможность, позволяющая разбить блок с текстом на несколько колонок. Обычно очень длинные строки сложно читать: если приходится слишком долго перемещать взгляд с конца одной строки на начало другой, можно легко потерять нужную строку. Чтобы сэкономить место на экране, но при этом сделать текст читабельным, можно разбить один сплошной блок текста на несколько колонок, как это делается в газетах.

Первое из них — **column-count**. Это свойство принимает в качестве параметра целое число и делит блок текста на заданное число колонок, равных друг другу по ширине.

# Колоночная разметка: свойство column-width

Свойство **column-width** задаёт минимальную желаемую ширину колонки. Если свойство **column-count** ещё не было задано, браузер автоматически поделит текст на такое количество колонок, чтобы они уместились во всю доступную ширину.

# Колоночная разметка: свойство column-gap

Между колонками есть промежуток. Рекомендованная ширина промежутка по умолчанию равна **1em**.

Но она может быть изменена при помощи свойства **column-gap**. Единицы измерения тоже **px**, **em**, **pt** и другие.

# Направление текста

Ещё одной интересной, но малоиспользуемой возможностью работы с текстом в CSS является управление направлением текста. Обычно эта возможность применяется для корректного отображения арабского языка и иврита, в которых чтение происходит справа налево. В CSS за направление текста отвечает свойство **direction**, принимающее значения **ltr** (направление слева направо) и **rtl** (направление справа налево).

Если попробовать задать для кириллицы или латиницы обратное направление текста (справа налево) **direction: rtl;**, мы увидим, что текст в блоке выровнялся наоборот, и полоса прокрутки блока сменила своё положение на противоположное. Но при этом сам текст не стал менять своё направление. Это произошло потому, что браузер автоматически задал тексту корректное направление, проанализировав используемые символы Unicode. Чтобы повлиять на это решение браузера существует свойство **unicode-bidi**, принимающее значения:

**normal** — браузер самостоятельно определяет, как ему следует отображать текст на основе используемых символов Unicode;

**embed** — переопределяет направление текста, располагая его согласно свойству **direction** (применяется, когда в блоке текст на двух разнонаправленных языках);

**bidi-override** — переопределяет порядок символов в тексте согласно значению **direction**.

Переполнение текста

Ещё одно интересное текстовое свойство — **text-overflow**. Оно позволяет определить, как будет выглядеть текст, если не поместится в контейнер. Оно принимает разные значения, но универсальными и работающими во всех современных браузерах являются **clip** и **ellipsis**.

Значение **clip** задано по умолчанию, и при нём текст просто обрезается по размеру контейнера, а при **ellipsis** — обрезается и к концу строки добавляется многоточие.

К сожалению, у этого свойства есть ограничения, которые делают его менее полезным: оно применимо только к однострочным текстам, а также к блокам, значение свойства **overflow** которых установлено в **auto**, **scroll** или **hidden**.

# Интервал между словами

И, наконец, последнее свойство, которое мы рассмотрим в этой части, — **word-spacing**. Оно задаёт расстояние между отдельными словами и строчными элементами. Значение может быть положительное или отрицательное и задаётся в **px**, **em**, **pt** или других единицах измерения CSS.