Unicode-символы

Помимо тех символов, которые помещаются у нас на клавиатуре, в стандарте кодирования Unicode представлены знаки почти всех письменных языков, а также разные специальные символы. В разметке HTML возможно написать любой символ Unicode, используя специальные ссылки-мнемоники: в виде слова, например, `&название;` или в числовом обозначении с использованием десятичного или шестнадцатиричного кода, например, `&#224;`.

Примеры Unicode-символов:

`&copy;`	`&#169;`	©
`&alpha;`	`&#945;`	α
`&para;`	`&#182;`	¶

Есть специальный сервис, где можно найти и скопировать нужный символ.

Свойство text-shadow

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

Смещение текстовой тени по горизонтали

Рассмотрим подробнее параметры текстовой тени:

```
text-shadow: 0px 0px 0px #333333;
```

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

Смещение задаётся в абсолютных и относительных единицах измерения: `px`, `em`, `pt` и других.

Смещение текстовой тени по вертикали

```
text-shadow: 0px 0px 0px #333333;
```

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

Размытие текстовой тени

```
text-shadow: 0px 0px 0px #333333;
```

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.

Цвет текстовой тени

```
text-shadow: 0px 0px 0px #333333;
```

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

Если цвет тени не указан, то он совпадает с цветом текста. Цвет задаётся в любом цветовом формате, может быть полупрозрачным.

Эффект вдавленного текста

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;
```

С помощью множественных теней можно добиваться необычных результатов. Например, можно добавить тексту эффект «вдавленности» с помощью двух однотонных теней: более тёмная смещается немного вверх и влево, а более светлая — вниз и вправо.

```
h1 {

  margin: 30px;

  text-transform: uppercase;

  text-shadow:

    -1px -1px 1px #1e181c,

    2px 2px 1px #52424d;
```

```
  font-weight: normal;

  font-size: 100px;

}
```

Декоративная ретро-тень

Ещё один текстовый эффект создаётся из двух резких теней.

Тени смещены в одну сторону. Нижняя тень смещена чуть сильнее и её цвет отличается от цвета фона, а верхняя тень смещена слабее и цвет её совпадает с цветом фона. Получается интересный эффект обводки.

Такой эффект хорошо подходит для заголовков с винтажным стилем.

Логотип: нестандартный шрифт

Помимо теней, необычных текстовых эффектов можно добиться с помощью подключения нестандартных шрифтов. В части про возможности HTML и CSS уже рассматривался пример подключения шрифтов с помощью конструкции `@font-face`.

Логотип: иконочный шрифт

Помимо букв шрифт может содержать и другие графические символы. Существуют инструменты, позволяющие собрать свой собственный шрифт из SVG-объектов.
Их довольно много: icomoon.io/app, fontastic.me, glyphter.com и другие.

Преимуществом шрифтовых иконок перед растровыми картинками является то, что их легко смасштабировать или перекрасить в другой цвет с помощью CSS-свойств, применимых к тексту. Также на экранах с большой плотностью пикселей шрифт не «замыливается», как это происходит с обычными картинками.

Логотип: выравнивание размеров

Давайте сделаем надпись «WARMPHOTO» плотнее, уменьшив межсимвольное расстояние.

Это делается с помощью свойства `letter-spacing`. Браузер по умолчанию устанавливает межсимвольное расстояние автоматически согласно определенному значению у каждого шрифта. Значение по умолчанию соответствует `letter-spacing: 0`.

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

Декоративная стилизация строки, шаг 1

Давайте создадим ещё один интересный элемент дизайна. Стилизуем надпись «WARMPHOTO» с помощью подключаемого шрифта, размеров и отступов так, чтобы буквы сместились на новую строку и образовали квадрат.

Сначала создадим рамку и зададим тексту нужный шрифт и размер.

** Warmphoto **

```
.square {
  margin: 150px auto;
  width: 260px;
  border: 5px solid #333333;
  text-align: center;
  text-transform: uppercase;
  font-size: 72px;
  font-family: "Montserrat Alternates", sans-serif;
}
```

Декоративная стилизация строки, шаг 2

Теперь нам нужно изменить правила переноса, чтобы символы, не поместившиеся по ширине в контейнер, переносились на новую строку.

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.

Это как раз то, что нам нужно: зададим нужный перенос слова, увеличим межсимвольное расстояние и поправим отступ.

Также нужно заметить, что свойство `word-wrap` — изначально проприетарное расширение Microsoft, а в текущем черновике спецификации CSS переименовано в `overflow-wrap`. Название `word-wrap` сейчас обозначено как «альтернативное» именование.

```
.square {
```

```
  margin: 150px auto;
  width: 260px;
  border: 5px solid #333333;
  text-align: center;
  text-transform: uppercase;
  font-size: 72px;
  font-family: "Montserrat Alternates", sans-serif;
  word-wrap:break-word;
  letter-spacing:24px;
  padding-left:24px;
}
```

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.

Значение свойства может принимать положительное или отрицательное значение, задаётся в `px`, `em`, `pt` или других единицах длины CSS. При задании значения в процентах, отступ первой строки вычисляется в зависимости от ширины блока.

Псевдоэлемент ::first-letter

В CSS существуют особые псевдоэлементы, позволяющие стилизовать первую букву в слове или первую строку в блоке текста.

Крупная, отличная от прочих, первая буква блока текста называется буквицей. Исторически буквицей украшали главы и разделы печатных книг и рукописей. Для создания буквицы воспользуемся псевдоэлементом `::first-letter`:

```
p::first-letter { color: red; }
```

Псевдоэлемент ::first-line

Первую строку блока текста можно стилизовать при помощи псевдоэлемента `::first-line`.

```
p::first-line { color: green; }
```

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.

Уже сейчас во всех популярных современных браузерах можно использовать многоколоночную разметку CSS.

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.

Значение свойства положительное и задаётся в `px`, `em`, `pt` или других единицах длины CSS.

Следует отметить, что если одиночная строка может включать от `45` до `75` символов, чтобы быть читабельной, то для колонок текста рекомендуется придерживаться ширины, включающей `40-50` символов.

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).

Однако помимо направления текста свойство `direction` также влияет на позицию полосы прокрутки в блоке.

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`.

Направление текста и таблицы

Также свойство `direction` влияет на порядок колонок в таблице, меняя его на соответствующее значение.

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

Свойство— `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.

Свойство `word-spacing` также можно использовать для задания отступов между блочно-строчными элементами, а также изображений, ведь изображения тоже являются строчными.