**Що таке CSS насправді?**

Як і HTML, CSS не є справжньою мовою програмування. Це лише *мова таблиць стилів*, яка дозволяє задавати стилі обраним елементам у HTML документах. Наприклад, для того, щоб вибрати **усі** параграфи на HTML сторінці та зробити колір їхнього тексту червоним, потрібно написати такий CSS:

p {

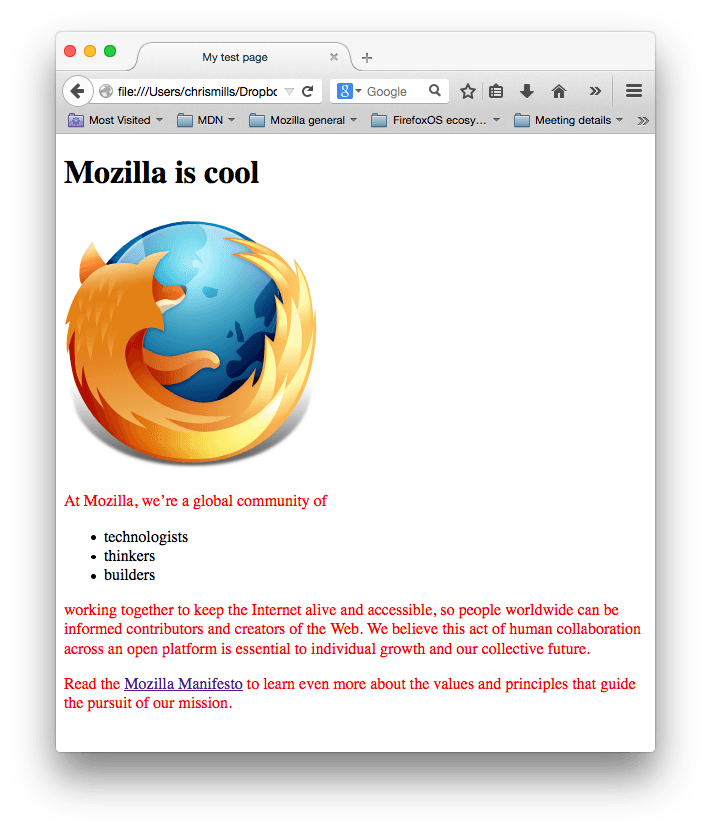
color: red;

}

Давайте спробуємо: вставте ці три строки CSS у новий файл у вашому редакторі коду, та збережіть файл як style.css у вашій теці styles.

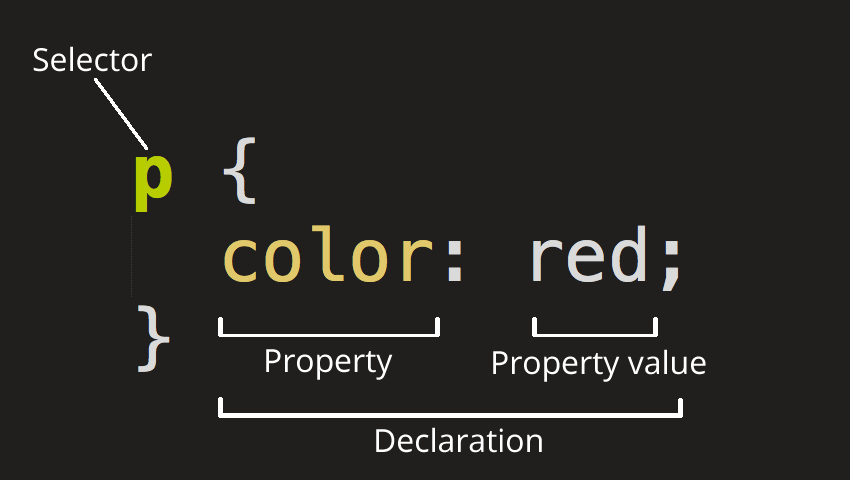
Окрім того, нам необхідно приєднати CSS до вашого HTML документу, інакше стилі CSS не вплинуть на те, як браузер відобразить HTML документ. (Якщо ви щойно приєдналися до нашого проекту, прочитайте [Роботу з файлами](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/Dealing_with_files) та [Основи HTML](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/HTML_basics)щоб дізнатися, що необхідно зробити на початку.) Відкрийте ваш файл index.html та вставте наступну стрічку кудись у head, між тегами <head> і </head>:

1. <link href="styles/style.css" rel="stylesheet" type="text/css">
2. Збережіть index.html та відкрийте його у вашому браузері. Ви побачите щось на зразок наведеного зображення:

Якщо текст ваших параграфів став червоним, вітаємо! Ви щойно написали ваш перший CSS!

Анатомія правила CSS

Подивимось на попередній CSS більш детально:



Вся структура називається "**rule set" — "набір правил"**(частіше, скорочено, "правило"). Його частини також мають свої назви:

**Selector — Селектор**

Назва елементу HTML на початку правила. Селектор вибирає елемент чи елементи, які будуть стилізовані (у нашому випадку, елементи p). Для стилізації інших елементів просто змініть селектор.

**Declaration — Визначення (декларація)**

Одне правило на зразок color: red; вказує, яку з властивостей елементу ви бажаєте стилізувати.

**Properties — Властивості**

Шляхи, якими ви можете стилізувати даний HTML елемент. (У цьому випадку, color — це властивість елементів p). У CSS ві обираєте властивості, які хочете змінити у вашому правилі.

**Property value — Значення властивості**

Праворуч від властивості, після двокрапки, ми вказуємо **значення властивості,**обираючи з багатьох можливих значень для наданої властивості (є багато іншіх значень властивості color окрім red).

Зверніть увагу на інші важливі частини синтаксису:

* Кожен набір правил (окрім селектора) повинен бути оточений фігурними дужками ({}).
* У кожному визначенні використовується двокрапка для відокремлення властивостей від їх значення.
* У кожному наборі правил використовується крапка з комою для відокремлення декларацій від наступних.

Отже, для того, щоб змінити декілька значень властивостей одночасно, їх достатньо записати, розділяючи крапкою з комою, наприклад:

p {

color: red;

width: 500px;

border: 1px solid black;

}

Вибір багатьох елементів одночасно

Ви також можете вибрати декілька типів елементів та застосувати один набір правил для них усіх. Додайте декілька селекторів, розділяючи їх комою, наприклад:

p,li,h1 {

color: red;

}

Різні типи селекторів

Є велика кількість різних типів селекторів. Вище ми розглянули лише **селектори елементів**, які вибирають всі елементи заданого типу у наданих HTML документах. Але ми можемо застосовувати більш специфічні селектори. Подивіться на найбільш розповсюджені типи селекторів:

| **Назва селектору** | **Що він вибирає** | **Приклад** |
| --- | --- | --- |
| Селектор елементу (іноді називають селектором тегу) | Всі HTML елементи даного типу | p Вибирає <p> |
| Селектор по ID | Елемент на сторінці, який має заданий ID (на HTML сторінці не може бути елементів з однаковими ID). | #my-id Вибирає <p id="my-id"> чи <a id="my-id"> |
| Селектор по класу | Елементи на сторінці, які мають задані класи (на сторінці можуть бути елементи з однаковими класами). | .my-class Вибирає <p class="my-class">та <a class="my-class"> |
| Селектор по атрибуту | Елементи на сторінці із вказаним атрибутом | img[src] Вибирає <img src="myimage.png"> але не <img> |
| Селектори псевдокласів | Вказані елементи, але тільки тоді, коли вони знаходяться у вказаному стані, наприклад, коли на них наводять курсор миші. | a:hover Вибирає <a>, але тільки коли курсор миші вказує на посилання. |

Інші селектори ви можете побачити на сторінці [Selectors guide](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started/Selectors).

**Шрифти та текст**

Дослідивши деякі основи CSS, давайте додамо більше правил та інформації в файл style.css, щоб зробити наш приклад більш привабливим. Для початку покращимо вигляд наших шрифтів та тексту.

1. В першу чергу, знайдіть збережений вами [вивід з сайту Google Fonts](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/What_should_your_web_site_be_like" \l "Font), який ви зберегли раніше. Додайте елемент <link ... > у head файлу index.html (будь-куди між тегами <head> та </head>). Елемент буде виглядати приблизно так:

<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>

1. Видаліть правило, яке зараз знаходиться у вашому файлі style.css. Це був гарний приклад, але червоний текст виглядає не дуже файно.
2. Додайте наступні строки до вашого файлу стилей, замінивши placeholder на строку font-family, що ви отримали з сайта Google Fonts. (font-family означає шрифти, які ви застосуєте для вашого тексту). Це правило встановить глобальний базовий шрифт і його розмір для всієї сторінки (так як <html> — це батьківський елемент всієї сторінки, всі елементи у ньому успадкують однаковий font-size та font-family):
3. html {
4. font-size: 10px; /\* px означає 'пікселі': базовий розмір шрифта встановлюється рівним 10 пікселів заввишки \*/
5. font-family: placeholder: замініть цей плейсхолдер на строку із сайту Google fonts

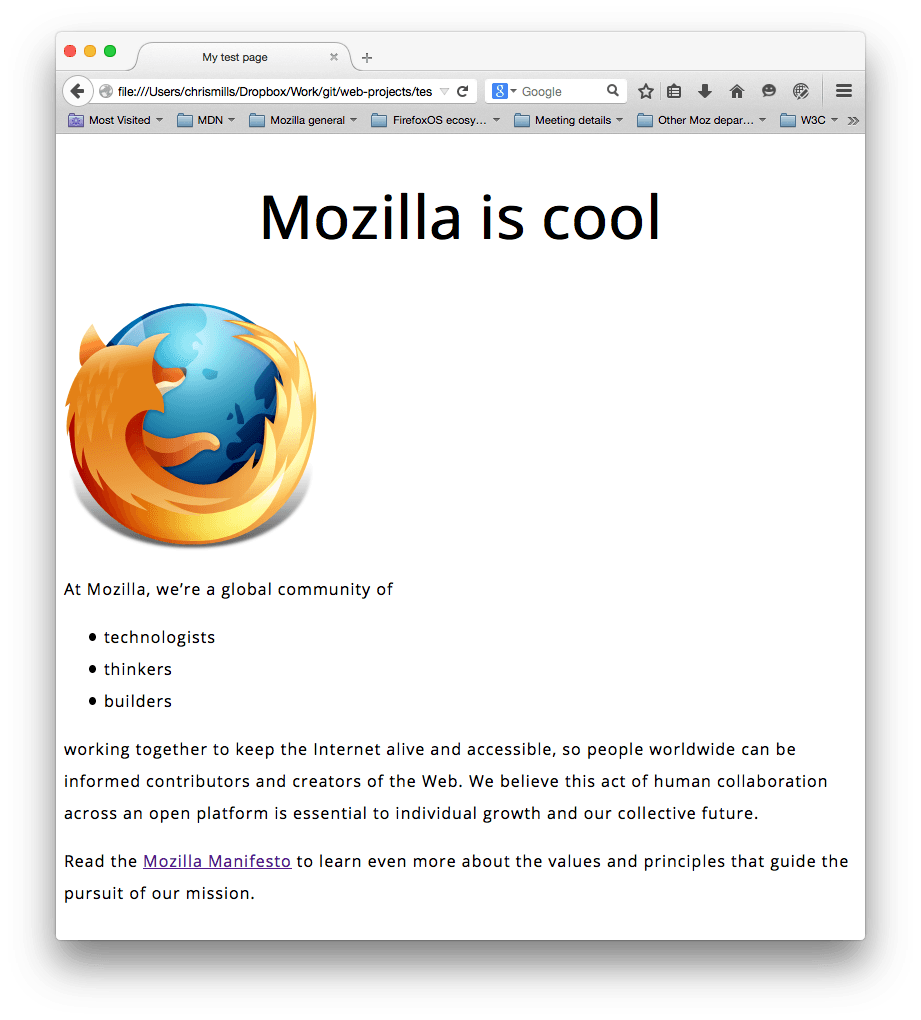
}

**Примітка**: Я додав коментар із поясненням, що означає "px". Все у CSS документі, що знаходиться  між /\* та \*/ — це **коментар CSS**, який ігнорується браузером. Коментарі застосовують для приміток та пояснень, що допоможуть вам в роботі.

1. Встановимо розміри шрифта для елементів, що містять у собі текст всередині HTML body ([<h1>](https://developer.mozilla.org/uk/docs/Web/HTML/Element/h1), [<li>](https://developer.mozilla.org/uk/docs/Web/HTML/Element/li), та [<p>](https://developer.mozilla.org/uk/docs/Web/HTML/Element/p)). Також відцентруємо текст нашого заголовку та встановимо висоту строки і проміжок між літерами нашого контенту, щоб зробити його зручнішим для читання.
2. h1 {
3. font-size: 60px;
4. text-align: center;
5. }
6. p, li {
7. font-size: 16px;
8. line-height: 2;
9. letter-spacing: 1px;

}

Ви можете змінювати числа як завгодно, щоб ваш дизайн виглядав так, як ви хочете. Загалом він буде виглядати так:



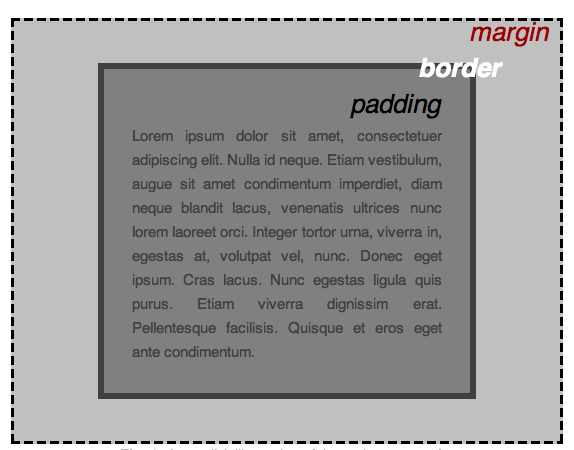
**Блоки, блоки, і ще раз про блоки**

Одна річ, яку ви помітите при написанні CSS, полягає в тому, що багато використовуються блоки - встановлення їх розміру, кольору, позиції тощо. Більшість HTML-елементів на вашій сторінці можуть розглядатися як блоки, що розташовані один на одному.



Не дивно, що макет CSS заснований переважно на блочній моделі *(box model)*. Кожен із блоків, що займає простір на вашій сторінці, має такі властивості:

* padding, простір лише навколо контенту (наприклад, навколо параграфу тексту)
* border, суцільна лінія (межа), яка розташована ззовні від padding
* margin, простір навколо зовнішнього елемента



В цьому розділі ми також використовуємо:

* width (ширина елемента)
* background-color, колір за контентом і padding елемента
* color, колір змісту елементу (зазвичай тексту)
* text-shadow: встановлює тінь на тексті в середині елемента
* display: встановлює режим відображення елемента (не хвилюйтесь про це покищо)

Отже, давайте почнемо та додамо ще кілька CSS на нашу сторінку! Додайте ці нові правила до нижньої частини сторінки, і не бійтеся експериментувати зі зміною значень, щоб побачити, як це працює.

Зміна кольору сторінки

html {

background-color: #00539F;

}

Це правило встановлює фоновий колір для всієї сторінки . Змініть код кольору на будь-який інший, [що ви обрали при плануванні свого сайту](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/What_should_your_web_site_be_like" \l "Theme_color).

Задання стилів для body

body {

width: 600px;

margin: 0 auto;

background-color: #FF9500;

padding: 0 20px 20px 20px;

border: 5px solid black;

}

Тепер стилі для елемента body. Тут багато параметрів, тому давайте розглянемо по-одному:

* width: 600px; — цей параметр робить body шириною завжди 600 пікселів.
* margin: 0 auto; — Коли ви встановлюєте два значення у властивостях marginабо padding, перше значення впливає на верхню  **та** нижню сторони елемента (зроблено 0 в даному випадку), та друге значення лівої **та** правої сторони (тут auto - це спеціальне значення, яке рівномірно розподіляє доступний горизонтальний простір між лівою і правою стороною). Ви можете також використовувати один, три або чотири значення, як задокументовано [тут](https://developer.mozilla.org/en-US/docs/Web/CSS/margin#Values).
* background-color: #FF9500; — як і раніше, це встановлює фоновий колір елемента. Ми використали якийсь червоно-оранжевий для body, а не синій для елемента [<html>](https://developer.mozilla.org/uk/docs/Web/HTML/Element/html), але не бійтеся експериментувати, підставляючи інші кольори.
* padding: 0 20px 20px 20px; — ми маємо чотири значення, встановлені для padding, щоб зробити трохи простору навколо нашого вмісту. Цього разу ми не встановлюємо padding зверху від body (значення 0px), а встановлюємо 20 пікселів ліворуч, знизу та праворуч. Значення встановлено в такому порядку: вгорі, праворуч, внизу та ліворуч.
* border: 5px solid black; — встановлює безперервну чорну рамку border товщиною 5 піксель з усіх боків body.

Позиціонування та стиль заголовка головної сторінки

h1 {

margin: 0;

padding: 20px 0;

color: #00539F;

text-shadow: 3px 3px 1px black;

}

Ви, мабуть, помітили, що у верхній частині body є чималенький розрив. Це відбувається тому, що браузери застосовують деякий **стиль за умовчанням** до елемента [<h1>](https://developer.mozilla.org/uk/docs/Web/HTML/Element/h1) (серед інших), навіть якщо ви не застосували жодного CSS! Це може здаватись поганою ідеєю, але ми хочемо навіть незавершена веб-сторінка мала базову читальну здатність. Щоб позбутися розриву, ми перекриваємо стандартний **стиль за умовчанням**, встановлюючи margin: 0;.

Далі ми встановили верхній та нижній відступ (padding) заголовка до 20 пікселів і зробили заголовок текстом того ж кольору, що і колір фону (background) всієї сторінки html.

Одна досить цікава властивість, яку ми використовували тут, це - text-shadow, яка застосовує текстову тінь до текстового вмісту елемента. Її чотири значення означають наступне:

* Перше значення пікселя встановлює **горизонтальне зміщення** тіні від тексту - як далеко вона рухається: негативне значення має перемістити тінь вліво.
* Друге значення пікселя встановлює **вертикальне зміщення** тіні від тексту - як далеко вона рухається в такому випадку; негативне значення має рухати тінь вгору.
* Третє значення пікселя встановлює **радіус розмивання** тіні - більша величина означатиме більш розмиту тінь.
* Четверте значення встановлює базовий колір тіні.

Знову ж таки, спробуйте експериментувати з різними значеннями, щоб побачити, що ви можете придумати.

Центрування зображення

img {

display: block;

margin: 0 auto;

}

Нарешті, ми будемо центрувати зображення, щоб воно виглядало краще. Ми могли б використати margin: 0 auto як і раніше для body, але ми також потребуємо зробити дещо інше. Елемент body є **блочним рівнем**, тобто він займає місце на сторінці, і до нього можуть бути додані поля та інші значення відступів. Зображення, з іншого боку, є **вбудованими елементами (inline)**, тобто вони не можуть мати відступів. Таким чином, щоб застосувати поля до зображення, ми повинні надати поведінку блочного рівня зображенню за допомогою display: block;.