

# Що таке WEB ?

WEB — система програмування, створена Дональдом Кнудом як перша реалізація його концепції «грамотного програмування»: ідеї, що програмне забезпечення можна створювати як літературний твір. Такий твір складається з фрагментів початкового коду, вбудований у текст, що описує програму природною мовою людини. Це пряма протилежність більшості мов програмування, де текст для читання людиною (документація) вбудовується у тіло програми. WEB складається з двох програм: TANGLE, що генерує текст програми на Паскалі, і WEAVE, що продукує відформатовану TeX-документацію.

Існує версія системи WEB для мови C, CWEB (наприклад, програма Web2C виконує ту ж роль, що і TANGLE для генерації Паскаль-тексту). З інших програм можна відзначити noweb, що не залежить від кінцевої мови програмування.

Найбільш відомі і важливі програми, написані на WEB — це TeX і Metafont.

## І все? А з чого складається?

**Вебсторінка** Інформаційний ресурс, доступний в мережі World Wide Web, який можна переглянути у веб браузері.

**Браузер** Це програма, що дозволяє відображати сторінки мережі Інтернет на екрані комп'ютера, мобільного телефона, планшета, розумного годинника, передавати і отримувати дані з «всесвітньої павутини».

**WEB-розробка** Це процес створення WEB-сайтів або WEB-додатків.

Основними етапами процесу є:

- вебдизайн,
- верстка сторінок,
- програмування для веб на стороні клієнта і сервера,
- конфігурування вебсервера.

## Front end та back end

Front end — це інтерфейс для взаємодії між користувачем і back end. Front end та back end можуть бути розподілені між однією або кількома системами.

В програмній архітектурі може бути багато рівнів між апаратним забезпеченням та кінцевим користувачем. Кожен з цих рівнів може мати як front end, так і back end. Front — це абстракція, спрощення базового компоненту через надання користувачу зручного (user-friendly) інтерфейсу.

### Front - End

Це зовнішній вигляд сайту, його структура

### Back - End

Це вся механіка сайту, його «невидимий двигун»

#### Front - End (Фронтенд)

Все, що браузер може читати, виводити на екран і / або запускати.

Щоб відобразити контент сайту найкращим чином та забезпечити оптимальну взаємодію користувача з сайтом, фронтенд-розробники використовують багато корисних інструментів, найважливіші з яких це

**HTML, CSS та JavaScript.**





**HyperText Markup Language** — мова розмітки гіпертексту це код, який використовується для структурування і відображення веб-сторінки та її контенту. **HTML** це не мова програмування, це мова розмітки, яка каже вашому браузеру, як відображати вміст веб-сторінки, яку ви переглядаєте.

**HTML - документ має свою структуру**

```
<!DOCTYPEhtml>
<html>
<head>
<title>WEB programming</title>
<script></script>
</head>
<body>
<!--Початок коду-->
  <div>
    <!--Контент 1-->
  </div>
  <div>
    <!--Контент 2-->
  </div>
<!--Кінець коду-->
</body>
</html>
```

**Тег** - елемент розмітки гіпертексту. Мітка, яку використовує розробник для вказівки браузеру, як він повинен показувати WEB-сайт. Тег також має свою структуру

**Всі HTML-елементи діляться на п'ять типів:**

- **порожні елементи** - `<area>`, `<base>`, `<br>`, `<col>`, `<embed>`, `<hr>`, `<img>`, `<input>`, `<link>`, `<menuitem>`, `<meta>`, `<param>`, `<source>`, `<track>`, `<wbr>`;
- **елементи з неформатований текстом** - `<script>`, `<style>`;
- **елементи, які виведуть звичайний текст** - `<textarea>`, `<title>`;
- **елементи з іншого простору імен** - MathML і SVG;
- **звичайні елементи** - всі інші елементи.

## Як написати і запустити HTML на комп'ютері?

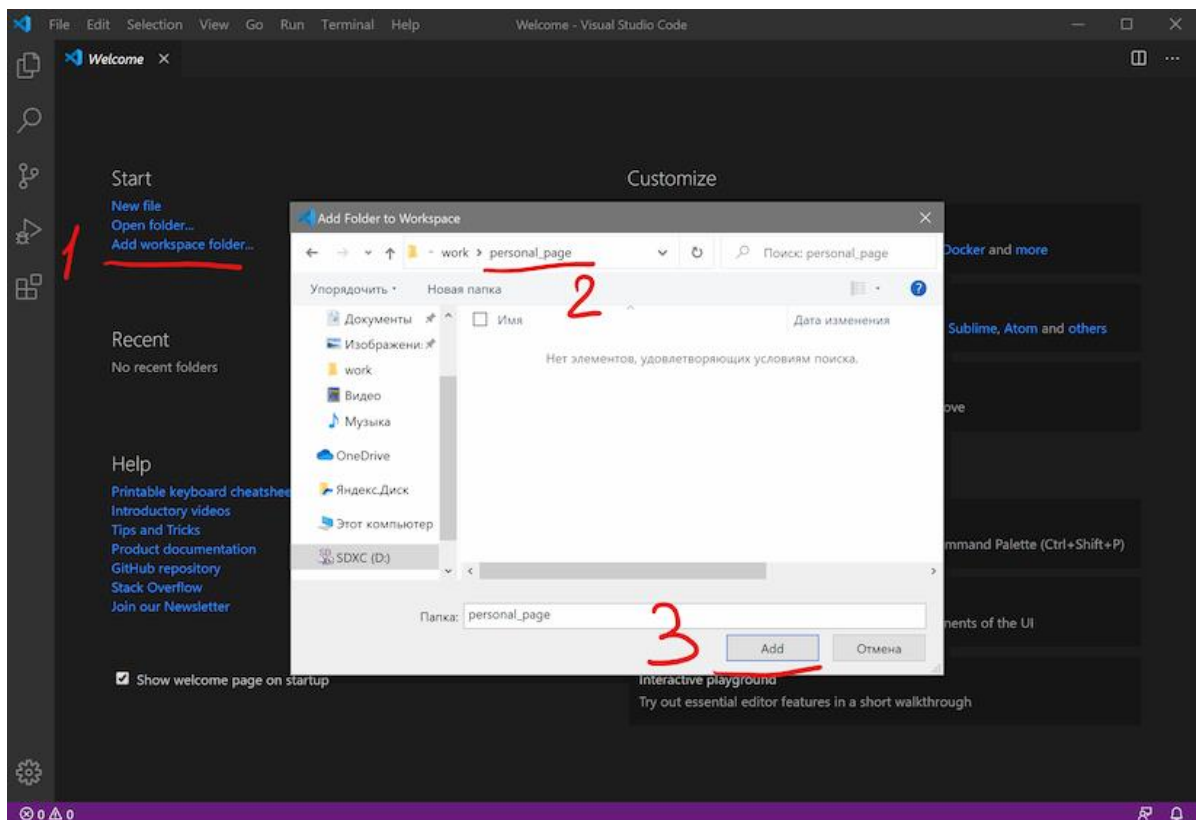
### Крок 1. Качаємо текстовий редактор

Для того, щоб написати код, згодиться взагалі будь-який текстовий редактор. Підійде навіть «Блокнот» на вашому комп'ютері (але в ньому дуже незручно все робити). Ми завантажили та встановили хороший редактор, заточений під веб-розробку. Покажемо все на прикладі Visual Studio Code.

Зайдіть на <https://code.visualstudio.com/> і скачайте редактор. Якщо у вас Windows, то натисніть на будь-яку з синіх кнопок. Якщо OS X або Linux - натисніть Other platforms. Установка пройде як завжди - потрібно запустити файл VSCodeUserSetup, багато раз натиснути «Далі» і поставити пару галочок.

### Крок 2. Запускаємо редактор і оглядаємося

Свіжовстановленому VS Code зустрічає нас екраном з великою кількістю посилань. З ними можна познайомитися пізніше, а зараз потрібно налаштувати все для роботи. Добре б, щоб під час роботи всі потрібні файли лежали в одній папці (поки проект маленький, так можна робити). Для цього додамо робочу папку, щоб VS Code показував нам тільки її вміст.



По кроках на скріншоті:

1. Add workspace folder - відкриває меню вибору папки.
2. Створимо нову папку `personal_page` в будь-якому зручному місці і зайдемо в неї.
3. Натиснемо Add.

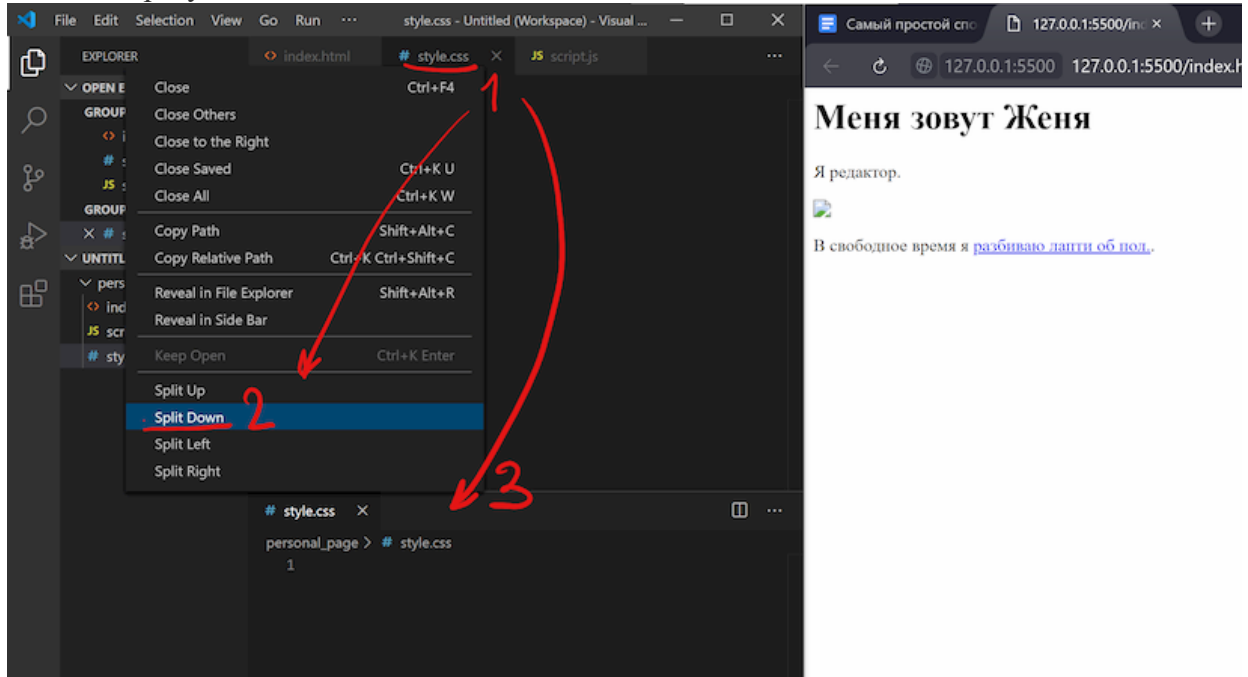
Після цього зліва з'явиться панель Explorer з порожнім робочим простором Untitled (Workspace). Ми створили папку, давайте її наповнимо.

### Крок 3. Додаємо файли

Після створення папка порожня. Клацнемо правою кнопкою по заголовку `personal_page` і додамо три файли, які знадобляться в роботі - `index.html`, `style.css` і `script.js`. Для початку цього вистачить.

#### Крок 4. Робимо роботу зручніше

Зараз все три файли відкриті у вкладках, і між ними не завжди зручно перемикатися. Щоб було зручніше, код зі стилями можна перенести в іншу частину вікна, наприклад, вниз. Для цього натисніть правою кнопкою по вкладці з `style.css` і виберіть `split down`, щоб побачити результат.



#### Крок 5. Додаємо код

Поки відредагуємо тільки `index.html` (файл з розміткою) і `style.css` (файл зі стилями), а `script.js` залишимо на майбутнє. Якщо у вас вже є який-небудь код, напишіть його, або використовуйте готовий - ми, наприклад, візьмемо код з інтерактивних курсів.

*index.html*

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <title>Сайт начинающего верстальщика</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <header>
      <nav>
        На главную
      </nav>
    </header>
    <main>
      <article>
        День первый. Как я забыл покормить кота
```

Кто бы мог подумать, что семантика это так важно, мне срочно нужно было об этом поговорить.

Взгляд упал на кота. Кот издал настойчивое «Мяу». И я понял — пришло время для первой записи в блог. И покормить кота.

</article>

<aside>

Здесь могла быть ваша реклама.

</aside>

</main>

<footer>

Подвал сайта

</footer>

</body>

</html>

style.css

Скопіюємо код зі стилями з файлу <https://htmlacademy.ru/assets/courses/299/outlines.css> - відкрийте його в браузері, скопіюйте всі рядки і вставте в файл style.css в редакторі.

### **Крок 6. Запускаємо код і дивимося на результат**

Найпростіший спосіб - відкрити папку з файлами через провідник і запустити файл index.html. Ви побачите результат верстки в браузері, але це не дуже зручно - при будь-яких змінах доведеться переходити в браузер і оновлювати сторінку.

## **Базові теги HTML**

У загальному випадку тег записується наступним чином:

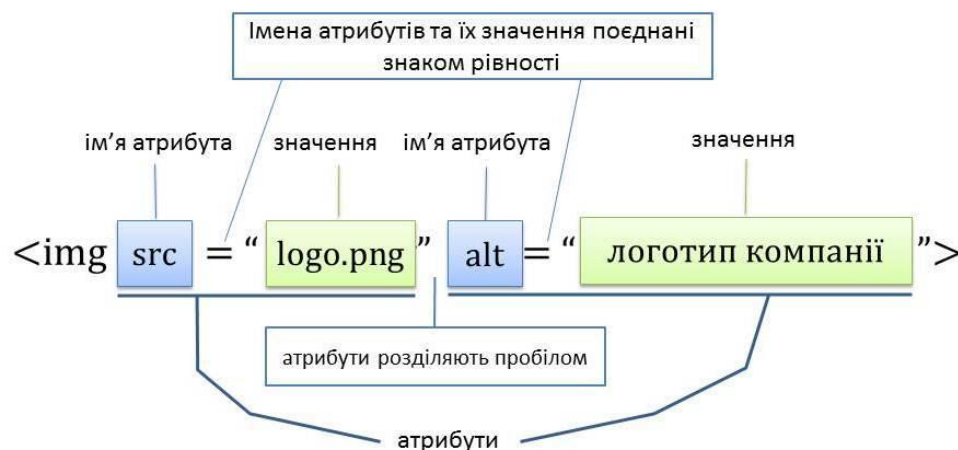
<ім'я тега атрибут1="значення1" атрибут2="значення2" ...>

Ось декілька прикладів тегів з атрибутами:

<p class="important"> ... </p>

<a class="external" href="http://cpto.dp.ua"> ... </a>

<img class = "logo" src = "logo.png">



**Оформлення смислових частин сторінки абзацами:**

- `<p>...</p>` – розбиває текст на параграфи (абзаци);
- `<br>` – обриває рядок і починає новий;
- `<nobr>...</nobr>` – забороняє перенос тексту на інший рядок;
- `<wbr>...</wbr>` – дозволяє розривати довгі слова (наприклад, при виведенні довгих медичних термінів);
- `<!-- ... -->` – тег коментарів; розміщений в ньому текст не відображається на web-сторінці.

### Основні теги форматування тексту та елементів дизайну

- `<h1>...</h1>` – керує розміром символів в заголовках ( $1 \leq n \leq 6$ ); зі збільшенням  $n$  розмір шрифту зменшується.
- `<big>...</big>` – збільшує розмір шрифту на 10% у порівнянні з базовим.
- `<small>...</small>` – зменшує розмір шрифту на 10% у порівнянні з базовим.
- `<pre>...</pre>` – позначає фрагмент тексту, форматування якого здійснено наперед; зручно використовувати при виведенні на екран текстів програм на мовах програмування, або при виведенні тексту, який містить спеціальні символи; в деяких випадках замінює тег `<nobr>...</nobr>`.
- `<code>...</code>` – позначає фрагмент тексту усередині рядка, який відповідає елементам коду.
- `<hr>` – непарний тег горизонтальної лінії; може бути засобом організації тексту і дизайну.

#### Атрибути тега `<hr>`:

- `align=left (right, center)` – вирівнювання лінії на сторінці;
  - `size=n` – товщина лінії в пікселях ( $1 \leq n \leq 175$ ); за замовчуванням 2 пікселі;
  - `color="колір"` – колір лінії;
  - `width=n%` – ширина лінії у відсотках до ширини екрану;
  - `noshade` – атрибут без параметра; показує об'ємні лінії, встановлюється за замовчуванням;
  - `shade` – атрибут без параметра; показує контур лінії; якщо заданий колір, то атрибут не працює.
  - `<left> ... </left>` – вирівнювання тексту по лівому краю сторінки.
  - `<right> ... </right>` – вирівнювання тексту по правому краю сторінки.
  - `<center> ... </center>` – вирівнювання тексту по середині сторінки.
- Наведені вище теги вирівнювання дещо застаріли. Для цього краще використовувати відповідні властивості CSS.
- `<b> ... </b>` – виділення напівжирним шрифтом.
  - `<i> ... </i>` – виділення курсивом.

**HTML-списки** використовуються для групування пов'язаних між собою фрагментів інформації. Існує три види списків:

- **маркований список** - `<ul>` - кожен елемент списку `<li>` зазначається маркером,



- **нумерований список** - `<ol>` - кожен елемент списку `<li>` зазначається цифрою,
- **список** Кожен елемент списку має додатковий блок, розташований збоку, який не бере участі в компонуванні. **визначень** - `<dl>` - складається з пар термін `<dt>` - `<dd>` визначення.

Кожен список являє собою контейнер, усередині якого розташовуються елементи списку або пари термін-визначення.

Елементи списку поводяться як блокові елементи, розташовуючись один під одним і займаючи всю ширину блоку-контейнера.

## 1. Маркований список

**Маркований список** являє собою неупорядкований список (*від англ. Unordered List*). Створюється за допомогою елемента `<ul>`. В якості маркера елемента списку виступає позначка, наприклад, зафарбований кружок.

Браузери за замовчуванням додають наступне форматування блоку списку:

```
ul {
  padding-left: 40px;
  margin-top: 1em;
  margin-bottom: 1em;
}
```

CSS

Кожен елемент списку створюється за допомогою елемента `<li>` (*від англ. List Item*).

Для елемента `<ul>` доступні глобальні атрибути.

```
<ul>
  <li>Microsoft</li>
  <li>Google</li>
  <li>Apple</li>
  <li>IBM</li>
</ul>
```

HTML

- Microsoft
- Google
- Apple
- IBM

ФІГУРА. 1. МАРКОВАНІЙ СПИСОК

## 2. Нумерований список

**Нумерований список** створюється за допомогою елемента `<ol>`. Кожен пункт списку також створюється за допомогою елемента `<li>`. Браузер нумерує елементи по порядку автоматично і якщо видалити один або кілька елементів такого списку, то інші номери будуть автоматично перераховані.

Блок списку також має стилі браузера за замовчуванням:

```
ol {
  padding-left: 40px;
  margin-top: 1em;
  margin-bottom: 1em;
}
```

CSS

Для елемента `<li>` доступний атрибут `value`, який дозволяє змінити номер за умовчанням для вибраного елемента списку. Наприклад, якщо для першого пункту списку задати `<li value="10">`, то інша нумерація буде перерахована щодо нового значення.

Для елемента `<ol>` доступні наступні атрибути:

ТАБЛИЦЯ 1. АТРИБУТИ ЕЛЕМЕНТА `<OL>`

Атрибут	Опис, прийняте значення
reversed	Атрибут reversed задає відображення списку в зворотному порядку (наприклад, 9, 8, 7 ...).
start	Атрибут start задає початкове значення, від якого піде відлік нумерації, наприклад, конструкція <code>&lt;ol start="10"&gt;</code> першого пункту присвоїть порядковий номер «10». Також можна одночасно ставити тип нумерації, наприклад, <code>&lt;ol type="I" start="10"&gt;</code> .
type	Атрибут type задає вид маркера для використання в списку (у вигляді букв або цифр). Прийняті значення: 1- значення за замовчуванням, десяткова нумерація. A- нумерація списку в алфавітному порядку, заголовні букви (A, B, C, D). a- нумерація списку в алфавітному порядку, малі літери (a, b, c, d). I- нумерація римськими великими цифрами (I, II, III, IV). i- нумерація римськими малими цифрами (i, ii, iii, iv).

```
<ol>
<li>Microsoft</li>
<li>Google</li>
<li>Apple</li>
<li>IBM</li>
</ol>
```

HTML

1. Microsoft
2. Google
3. Apple
4. IBM

ФІГУРА. 2. НУМЕРОВАННИЙ СПИСОК

### 3. Список визначень

**Списки визначень** створюються за допомогою елемента `<dl>`. Для додавання терміна застосовується елемент `<dt>`, а для вставки визначення - елемент `<dd>`.

Блок списку визначень має наступні стилі браузера за замовчуванням:

```
dl {
margin-top: 1em;
margin-bottom: 1em;
}
```

CSS

Для елементів `<dl>`, `<dt>` і `<dd>` доступні [глобальні атрибути](#).

```
<dl>
<dt>Режиссер:</dt>
<dd>Петр Точилин</dd>
<dt>В ролях:</dt>
<dd>Андрей Гайдулян</dd>
<dd>Алексей Гаврилов</dd>
<dd>Виталий Гогунский</dd>
<dd>Мария Кожевникова</dd>
</dl>
```

HTML

режисер:  
Петро Точилін



У ролях:

Андрій Гайдулян

Олексій Гаврилов

Віталій Гогунський

Марія Кожевнікова

ФІГУРА. 3. СПИСОК ВИЗНАЧЕНЬ

## 4. Як створити вкладений список

Найчастіше можливостей простих списків не вистачає, наприклад, при створенні змісту ніяк не обійтися без **вкладених пунктів**. Розмітка для вкладеного списку буде наступною:

```
<ul>
```

```
<li>Пункт 1.</li>
```

```
<li>Пункт 2.
```

```
<ul>
```

```
<li>Подпункт 2.1.</li>
```

```
<li>Подпункт 2.2.
```

```
<ul>
```

```
<li>Подпункт 2.2.1.</li>
```

```
<li>Подпункт 2.2.2.</li>
```

```
</ul>
```

```
</li>
```

```
<li>Подпункт 2.3.</li>
```

```
</ul>
```

```
</li>
```

```
<li>Пункт 3.</li>
```

```
</ul>
```

HTML

- Пункт 1.
- Пункт 2.
  - Підпункт 2.1.
  - Підпункт 2.2.
    - Підпункт 2.2.1.
    - Підпункт 2.2.2.
  - Підпункт 2.3.
- Пункт 3.

ФІГУРА. 4. ВКЛАДЕНИЙ СПИСОК

## 5. Багаторівневий нумерований список

Багаторівневий список використовується для відображення елементів списку на різних рівнях з різними відступами. Розмітка для багаторівневого нумерованого списку буде наступною:

```
<ol>
```

```
<li>пункт</li>
```

```
<li>пункт
```

```
<ol>
```

```
<li>пункт</li>
```

```
<li>пункт</li>
```

```
<li>пункт
```

```
<ol>
```

```
<li>пункт</li>
```

```
<li>пункт</li>
```

```
<li>пункт</li>
```

```
</ol>
```

```

        </li>
        <li>пункт</li>
    </ol>
</li>
<li>пункт</li>
<li>пункт</li>
</ol>

```

## HTML

Така розмітка за замовчуванням створить для кожного вкладеного списку нову нумерацію, яка починається з одиниці. Щоб зробити вкладену нумерацію, потрібно використовувати такі властивості:

- counter-reset скидає один або кілька лічильників, задаючи значення для скидання;
- counter-increment задає значення приросту лічильника, тобто з яким кроком нумеруватиметься кожний наступний пункт;
- content - генерується вміст, в даному випадку відповідає за виведення номера перед кожним пунктом списку.

```

ol {
    /* убираем стандартную нумерацию */
    list-style: none;
    /* Идентифицируем счетчик и даем ему имя li. Значение счетчика не указано - по
    умолчанию оно равно 0 */
    counter-reset: li;
}
li:before {
    /* Определяем элемент, который будет нумероваться — li. Псевдоэлемент before
    указывает, что содержимое, вставляемое при помощи свойства content, будет располагаться
    перед пунктами списка. Здесь же устанавливается значение приращения счетчика (по
    умолчанию равно 1). */
    counter-increment: li;
    /* С помощью свойства content выводится номер пункта списка. counters() означает, что
    генерируемый текст представляет собой значения всех счетчиков с таким именем. Точка в
    кавычках добавляет разделяющую точку между цифрами, а точка с пробелом добавляется
    перед содержимым каждого пункта списка */
    content: counters(li, ".") ". ";
}

```

## CSS

1. пункт
2. пункт
  1. пункт
  2. пункт
  3. пункт
3. пункт
4. пункт
  1. пункт
  2. пункт
  3. пункт
4. пункт

# CSS

## Структура WEB проекту

- **Project (Root) Folder:** основна папка, містить всі інші папки та файли
- **CSS Folder:** усі файли, що відносяться до стилів (.css файли)
- **Images Folder:** усі зображення (.jpg, .png, .gif, etc)
- **JavaScript Folder:** усі JavaScript файли (.js)

## *Термінал на Linux*

**ctrl + alt + T - launch terminal**

### Основні команди

`clear` - Очистити консоль

### Навігація

`pwd` - Показати поточний каталог

`ls` - Показати файли в цій папці, крім прихованих

`ls -f` - Показати файли в цій папці, включаючи і приховані

`cd c:/` - Перейти в конкретний каталог

`cd --` - Повернутися назад

`cd ..` - Вийти на 1 рівень вгору

`cd ../..` - Вийти на 2 рівня вгору

### створення каталогів

`mkdir` - Створити папку

`cd !$` - Перейти в тільки що створену папку

`mkdir -p {app1,app2}` - Створити відразу кілька папок

`mkdir -p app/{css,js}` - Створити відразу кілька вкладених папок

### створення файлів

`touch index.html` - Створити файл index.html

`touch app/{css/main.css,js/main.js,index.html}` - Створити відразу кілька файлів, ніяких зайвих прогалин бути не повинно

### видалення файлів

`touch` - дозволяє створювати файли

`rm test` - Видалити порожню папку test

`rm -r test` - Видалити папку test з файлами всередині неї

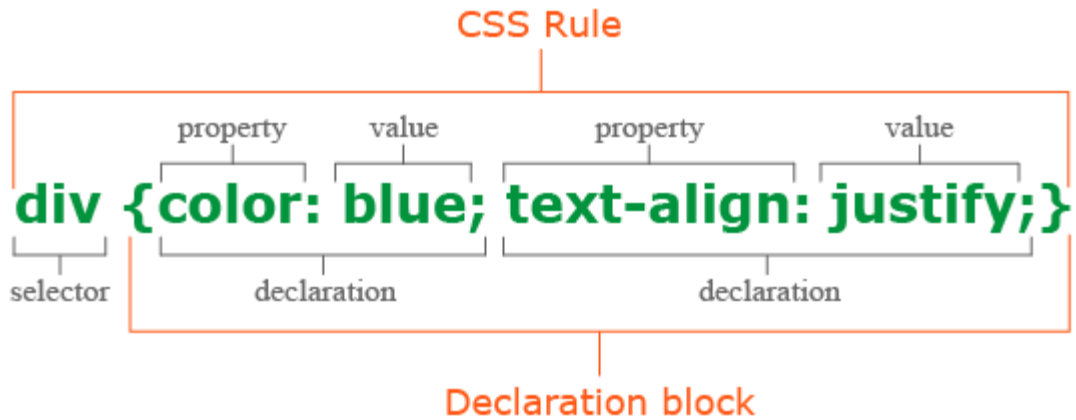
### переміщення файлів

`mv app1/*.* app2` - Перемістити всі файли з папки app1 в папку app2

## *Cascading Stylesheets*

**(CSS, каскадні таблиці стилів)**

це код, що використовується для стилізації сайту. Це мова таблиць стилів, яка дозволяє задавати стилі обраним елементам у HTML документах



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

- **Selector** — **Селектор** Назва елемента 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 властивість елемента ви бажаєте стилізувати.
- **Properties** — **Властивості** Шляхи, якими ви можете стилізувати даний HTML елемент. (У цьому випадку, color — це властивість елементів p). У CSS ви 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 інших значень властивості color окрім red).



Необхідно приєднати **CSS** до вашого **HTML** документу, інакше стилі **CSS** не вплинуть на те, як браузер буде відображати **HTML** документ.

```
<link rel="stylesheet" href="style.css">
```

### *Деякі CSS правила*

color - колір тексту  
background-color - фон  
font-size - розмір шрифту

width - ширина

height – висота

## Tables

**Елемент `<img>`** повинен містити файли у графічному форматі GIF, JPEG, SVG або PNG. Адреса файлу з картинкою задається через атрибут src.

**GIF** (Graphics Interchange Format — «формат обміну зображеннями»). Одними із головних особливостей формату є підтримка анімації та прозорості.

**PNG** (Portable Network Graphics) — збереження графічної інформації, що використовує стиснення без втрат. (Має прозорий фон).

**JPEG** (Joint Photographic Experts Group) – формат для зберігання фотозображень і подібних до них зображень.

**SVG** (Scalable Vector Graphics масштабована векторна графіка) — специфікація мови розмітки, що базується на XML, та формат файлів для двовимірної векторної графіки, як статичної, так і анімованої та інтерактивної.

## Опис

Елемент `<table>` служить контейнером для елементів, що визначають вміст таблиці. Будь-яка таблиця складається з рядків і осередків, які задаються за допомогою тегів `<tr>` і `<td>`. У середині `<table>` допустимо використовувати наступні елементи: `<caption>`, `<col>`, `<colgroup>`, `<tbody>`, `<td>`, `<tfoot>`, `<th>`, `<thead>` і `<tr>`.

Таблиці з невидимою кордоном довгий час використовувалися для верстки веб-сторінок, дозволяючи розділяти документ на модульні блоки. Подібний спосіб застосування таблиць знайшов втілення на багатьох сайтах, поки йому на зміну не прийшов більш сучасний спосіб верстки за допомогою шарів.

## Синтаксис

```
<table>
  <tr>
    <td>...</td>
  </tr>
</table>
```

### атрибути

<a href="#">align</a>	Визначає вирівнювання таблиці.
<a href="#">background</a>	Задає фоновий малюнок у таблиці.
<a href="#">bgcolor</a>	Колір фону таблиці.
<a href="#">border</a>	Товщина рамки в пікселях.
<a href="#">bordercolor</a>	Колір рамки.
<a href="#">cellpadding</a>	Відступ від рамки до вмісту осередку.
<a href="#">cellspacing</a>	Відстань між осередками.
<a href="#">cols</a>	Число колонок в таблиці.
<a href="#">frame</a>	Повідомляє браузеру, як відображати межі навколо таблиці.
<a href="#">height</a>	Висота таблиці.
<a href="#">rules</a>	Повідомляє браузеру, де відображати кордону між осередками.
<a href="#">summary</a>	Короткий опис таблиці.
<a href="#">width</a>	Ширина таблиці.

Також для цього тега доступні [універсальні атрибути](#) і [події](#) .

### закриває тег

Обов'язковий.

## Таблиця має свою структуру

```
<!DOCTYPE html>
<html>
<body>
<h2>Basic HTML Table</h2>

<table style="width:30%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
</body>
</html>
```

### Basic HTML Table

	Firstname	Lastname	Age
Jill	Smith		50
Eve	Jackson		94

**border:** 2px solid green;

**2px** - border width

**solid** - border type

**green** - border color

**Solid**

**Dotted**

**Dashed**

**Inset**



