

# ОСНОВИ WEB – РОЗРОБКИ

## Практична робота 2.1

### «Елементи мови гіпертекстової розмітки»

HTML (мова гіпертекстової розмітки) — це код, який використовується для структурування і відображення веб-сторінки та її контенту. Наприклад, контент може бути розбитий на параграфи (абзаци), містити список, зображення чи таблицю. Ця стаття має дати базове розуміння мови HTML та її призначення.

## Що таке HTML?

HTML — це не мова програмування, це *мова розмітки*, яка каже вашому браузеру, як відобразити вміст веб-сторінки, яку ви переглядаєте. Вона може бути простою чи складною, залежно від бажання веб-дизайнера, який її створює. HTML складається з серії (**елементів**), які використовуються для вміщення, або "загортання" в них різних частин контенту, щоб вони відображались чи діяли в певний спосіб. Ці елементи за допомогою початкової і кінцевої (**тегів**) можуть зробити слово чи зображення посиланням на будь-яку іншу сторінку, можуть відобразити текст курсивом, збільшити чи зменшити шрифт тощо. Для прикладу, візьмемо такий контент:

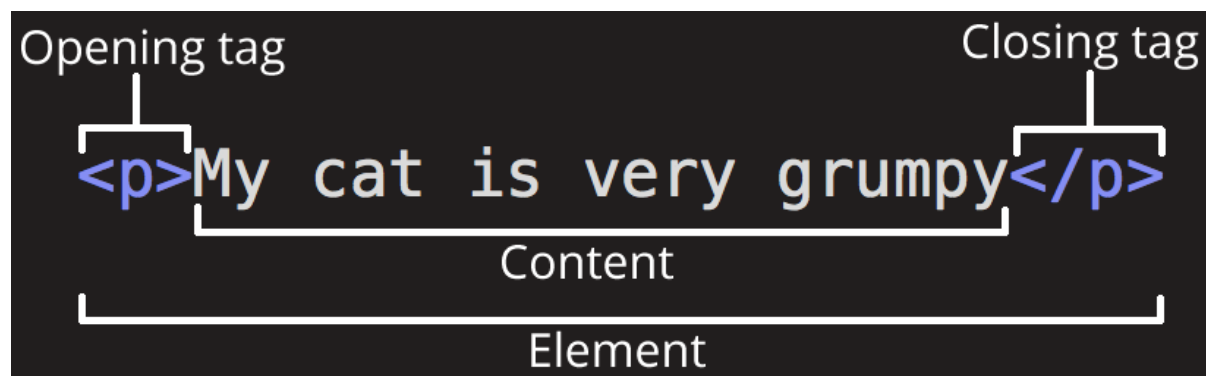
Мій кіт дуже сумний.

Якщо ми хочемо, щоб цей рядок був сам по собі, ми можемо вказати, що це параграф (абзац), додавши до нього відповідні теги:

```
<p>Мій кіт дуже сумний.</p>
```

### Анатомія елемента HTML

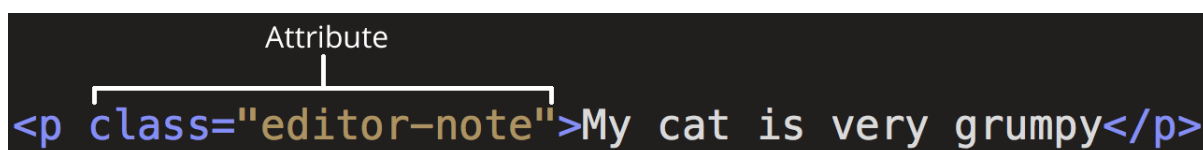
Погляньмо на елемент "параграф" детальніше.



Основні частини елемента такі:

1. **Початковий тег:** містить назву елемента (в даному випадку, p), загорнену в **кутові дужки**. Цей тег позначає початок елемента, або місце, де елемент починає діяти. У даному випадку — це місце, де починається параграф.
2. **Кінцевий тег:** такий самий тег, як і початковий, тільки тепер він містить скісну риску перед назвою елемента. Цей тег позначає місце закінчення елемента — у даному випадку, місце, де закінчується параграф. Одна з найпоширеніших помилок початківців — це забути поставити кінцевий тег, що може призвести до несподіваних результатів.
3. **Вміст:** вміст елемента, у даному випадку — просто текст.
4. **Елемент:** початковий тег плюс кінцевий тег плюс вміст між ними — дорівнює елементу.

Елементи також можуть мати атрибути, які виглядають так:



```
<p class="editor-note">My cat is very grumpy</p>
```

Атрибути містять додаткову інформацію про елемент, яка не відображається в самому контенті. У прикладі вище `class` це *ім'я* атрибута, а `editor-note` — це *значення* атрибута. Атрибут `class` дозволяє дати елементу певний ідентифікатор, який пізніше можна використати для того, щоб доступитися до цього елемента і змінити його стиль.

Атрибут завжди повинен мати:

1. Пробіл між ім'ям атрибута і назвою елемента (або попереднім атрибутом, якщо такий є).
2. Ім'я атрибута, за яким іде знак дорівнює.
3. Значення атрибута, загорнене в прямі лапки.

## Вкладені елементи

Ви можете додати елемент всередину іншого елемента. Це називається **вкладенням**. Якщо ми хочемо наголосити, що наш кіт **дуже** сумний, ми можемо загорнути слово "дуже" в елемент `<strong>`, який означає, що слово особливо сильно наголошене:

```
<p>Мій кіт <strong>дуже</strong> сумний.</p>
```

Важливо пам'ятати правильне вкладення елементів: у прикладі вище ми спершу відкрили елемент `<p>`, а потім елемент `<strong>`, що означає, що першим ми маємо закрити внутрішній елемент `<strong>`, а потім зовнішній `<p>`. Приклад нижче неправильний:

```
<p>Мій кіт <strong>дуже сумний.</p></strong>
```

Елемент має закриватися або всередині, або ззовні іншого елемента. Якщо кінцевий тег зовнішнього елемента іде перед кінцевим тегом внутрішнього елемента, ваш браузер спробує здогадатися, що ви хотіли цим всім сказати, і результати цього можуть бути неочікуваними. Тому не робіть так!

## Порожні елементи

Деякі елементи не містять жодного вмісту. Вони називаються **порожніми елементами**. Для прикладу візьмемо елемент `<img>`:

```

```

Цей елемент має два атрибути, при цьому не має кінцевого тегу і жодного внутрішнього вмісту. Елементу `image` не потрібно загортати контент, його мета — вставити зображення на певне місце на веб-сторінці.

## Анатомія HTML-документу

Все зазначене вище є основами HTML-елементів, але окремо взяті елементи самі по собі мало на що здатні. Тепер ми поглянемо на те, як окремі елементи формують цілу веб-сторінку. Візьмемо для прикладу код з файлу `index.html` (який використовується в статті [Робота з файлами](#)):

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Моя тестова сторінка</title>
  </head>
  <body>
    
  </body>
</html>
```

Тут ми маємо:

- `<!DOCTYPE html>` — тип документу. В сиву давнину, коли HTML був молодим (приблизно в 1991-92 роках), типи документів мали слугувати набором посилань на правила, яким HTML-сторінка мала слідувати, щоб вважатися правильним HTML (наприклад, автоматична перевірка помилок та інше). Однак, у наш час мало хто переймається типами документів, і вони є швидше історичним артефактом, який потрібно додавати, щоб все працювало правильно. Поки що це все, що вам потрібно знати про тип документу.
- `<html></html>` — елемент `<html>`. Цей елемент загортає весь контент на всій сторінці. Також його називають кореневим елементом (root element).
- `<head></head>` — елемент `<head>`. Цей елемент слугує контейнером для всієї інформації, яку ви хочете додати до своєї сторінки, але яка *не є власне контентом* сторінки, який будуть переглядати користувачі. Це, зокрема, [ключові слова](#), опис сторінки, який буде з'являтися в пошукових результатах, CSS для стилів контенту, набір символів (кодування) тощо.
- `<body></body>` — елемент `<body>`. Цей елемент містить *увесь контент*, який ви хочете показати користувачам, коли вони відвідують вашу сторінку: текст, зображення, відео, ігри, аудіо тощо.
- `<meta charset="utf-8">` — цей елемент визначає набір символів, які використовує сторінка. Зазвичай це UTF-8, який містить найбільше символів з більшості людських писемних мов. По суті, ваша сторінка зможе відобразити практично будь-який текстовий контент. Немає жодних причин пропускати цей елемент, він може допомогти уникнути певних проблем пізніше.

- `<title></title>` — елемент `<title>`. Цей елемент містить заголовок вашої сторінки, який буде відображатися у вкладці браузера при завантаженні сторінки. Також текст у цьому елементі буде відображатися в закладках, якщо додати сторінку до закладок.

---

## Зображення

Повернемося до елементу `<img>`:

```

```

Як ми вже з'ясували раніше, цей елемент вставляє зображення на сторінку в тому місці, це воно має з'явитися. Це відбувається за допомогою атрибута `src` (source — джерело), який містить шлях до зображення.

Також ми додали атрибут `alt` (alternative). У цьому атрибуті потрібно вказувати альтернативний текст-опис зображення для тих випадків, коли користувачі не зможуть побачити картинку, наприклад, через те, що:

1. Мають порушення зору. Користувачі з порушенням зору часто використовують скрін-рідери, які читають описи зображень.
2. Щось пішло не так при завантаженні зображення. Спробуйте навмисно змінити шлях до зображення в атрибуті `src`, щоб він став неправильним. Якщо ви тепер перезавантажите сторінку, ви побачите, що на місці зображення буде текст:

**My test image**

Основним при створенні атрибута `alt` має бути описовість тексту. Цей атрибут має дати користувачу достатньо інформації, щоб скласти уявлення про те, що є на зображенні. У нашому випадку, текст "Моє тестове зображення" нікуди не годиться. Набагато кращий варіант для логотипу Firefox був би "Логотип Firefox: палаюча лисиця навколо Земної кулі."

Спробуйте придумати кращий альтернативний текст для свого зображення.

**Примітка:** дізнайтеся більше про доступність на сторінці [MDN's Accessibility](#).

---

## Розмітка тексту

У цьому розділі ми розглянемо базові HTML-елементи, які використовуються для розмітки тексту.

Заголовки

Елементи "Heading" дозволяють вам вказати, що певні частини контенту вашої сторінки є заголовками або підзаголовками. Так само, як у книжці є назва книги, назви розділів і підзаголовки, у HTML-документі це також є. HTML має заголовки шістьох рівнів, <h1>–<h6>, хоча зазвичай використовуються перші 3-4.

```
<h1>Мій головний заголовок</h1>
<h2>Мій важливий заголовок</h2>
<h3>Мій підзаголовок</h3>
<h4>Мій менш важливий підзаголовок</h4>
```

Спробуйте додати підзаголовки до вашої HTML-сторінки перед елементом <img>.

## Параграфи

Як пояснювалося вище, елемент <p> містить частину тексту, яка становить один абзац. Параграфи використовуються дуже часто, особливо при розмітці звичайного текстового контенту:

```
<p>Це один параграф.</p>
```

Додайте зразок тексту (він має бути готовий зі статті [Як виглядатиме ваш сайт?](#)) в один чи два параграфи і розмістіть їх прямо під елементом <img>.

## Списки

Чимало контенту на веб-сторінках є списками, і в HTML є спеціальні елементи для них. Розмітка списків завжди має принаймні два елементи. Найпоширеніші типи списків - це нумерований і ненумерований списки:

1. **Ненумерований список** — це перелік, у якому порядок пунктів не суттєвий, наприклад, список покупок. Такий список загортається в елемент <ul>.
2. **Нумерований список** — це перелік, у якому порядок пунктів має значення, наприклад, рецепт приготування. Такий список загортається в елемент <ol>. Кожен пункт всередині списку загортається в свій окремий елемент <li> (list item).

Наприклад, якщо ми хочемо перетворити такий параграф у список:

```
<p>Mozilla – це глобальна спільнота технологів, мислителів і творців, які працюють разом...</p>
```

Ми можемо так змінити розмітку:

```
<p>Mozilla – це глобальна спільнота</p>
<ul>
  <li>технологів</li>
  <li>мислителів</li>
  <li>творців</li>
</ul>
```

```
<p>які працюють разом... </p>
```

Спробуйте додати нумерований чи нелінійний список до вашої сторінки.

---

## Посилання

Посилання дуже важливі — це те, що робить мережу мережею! Щоб додати посилання, потрібно використовувати елемент `<a>` ("a" скорочено від "anchor" — якор). Щоб перетворити параграф на посилання, робіть так:

1. Виберіть текст. Ми вибрали "Mozilla Manifesto".
2. Загорніть цей текст в елемент `<a>`, отак:

```
<a>Mozilla Manifesto</a>
```

3. Додайте до елемента `<a>` атрибут `href`, отак:

```
<a href="">Mozilla Manifesto</a>
```

4. Додайте значення цьому атрибуту — адресу посилання:

```
<a href="https://www.mozilla.org/en-US/about/manifesto/">Mozilla Manifesto</a>
```

Не пропускайте частину `https://` чи `http://` на початку адреси, так званий *протокол*, інакше результати можуть бути неочікуваними. Після того, як елемент доданий, клікніть на посилання і переконайтеся, що вас відправило туди, куди ви хотіли.

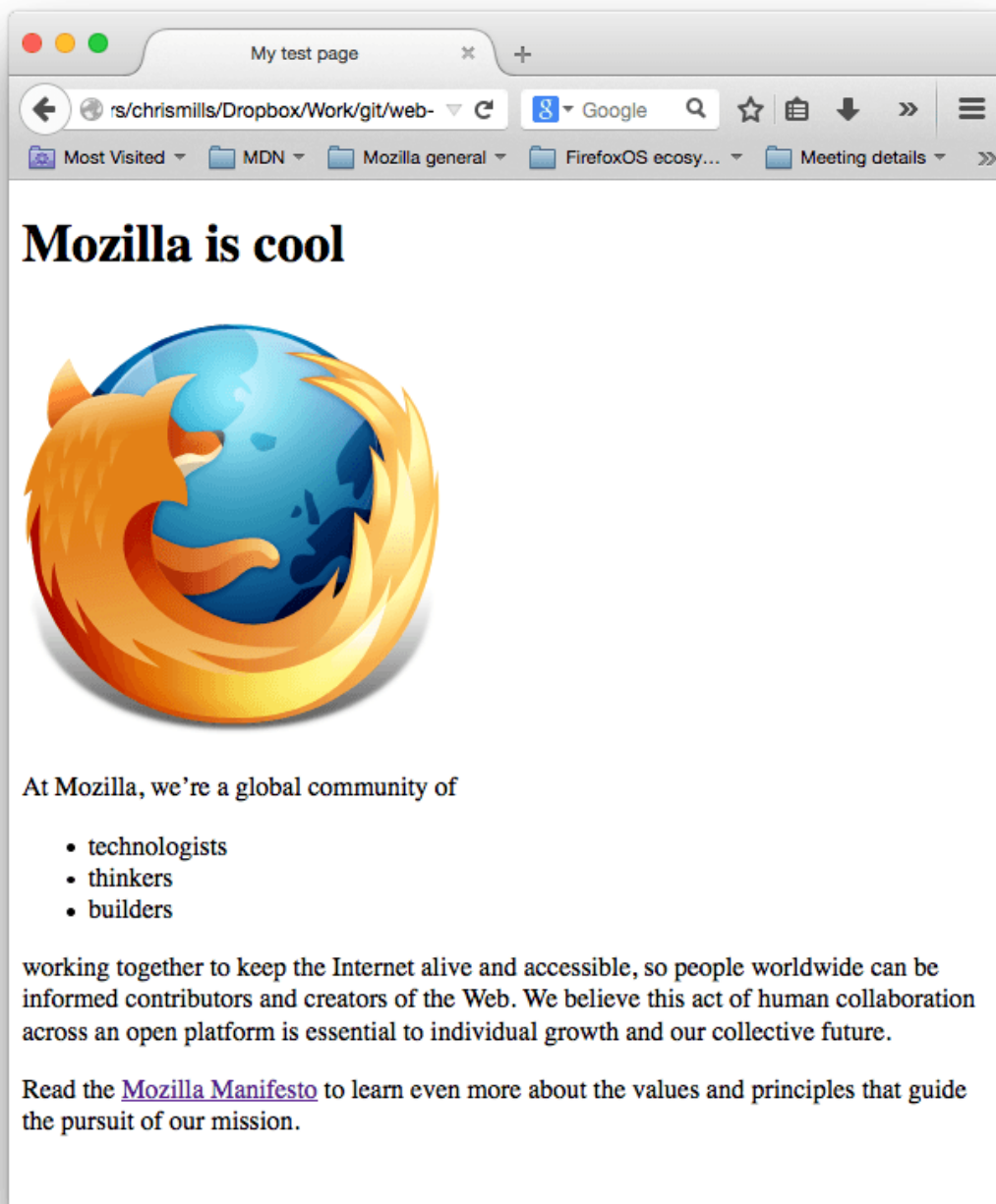
`href` може видаватися дивним ім'ям для атрибута, який складно запам'ятати. Він означає *hypertext reference* — гіпертекстове посилання.

Тепер додайте посилання до своєї сторінки.

---

## Висновок

Якщо ви слідували інструкціям у цій статті, у вас мала вийти сторінка, яка виглядає приблизно так (також можна переглянути [тут](#)):



Якщо ви застрягли, ви завжди можете порівняти свій код із завершеним прикладом коду на [GitHub](#).  
У цій статті ми розглянули тільки базові поняття HTML. Щоб дізнатися більше, зайдіть на сторінку [HTML Learning topic](#).