# 4.1. Основи HTML

HTML - не є мовою програмування, це мова розмітки, яка використовується для розміщення елементів (тексту, картинок, кнопок, таблиць тощо) на сторінці, тобто, для складання структури. Сторінка може мати складну або просту структуру, все залежить від досвіду веб-розробника.

HTML складається з ряду елементів, які використовують для того, щоб різні частини сторінки мали певний вид або спрацьовували певним способом. За призначення, вигляд та розташування елементу відповідають спеціальні оператори (теги), які представляють собою команду, що поміщена в кутові дужки. Теги, зазвичай, є скороченою назвою або абревіатурою того завдання, що має виконати браузер наприклад:

<p> - тег, що позначає абзац (paragraph).

<tr> - тег, що позначає рядок таблиці (table row).

<h1> - тег, що позначає заголовок 1 рівня (header 1 level).

Для збільшення інформативності тегу використовуються атрибути, що вказують текстові чи числові значення.

<table width=”50%” align=”center”>…</table>

В наведеному прикладі показано тег таблиці, яка має займати 50% від ширини вікна браузера (або блоку, в якому знаходиться) і розміщена по центру.

В мові HTML застосовують теги: одинарні, парні та коментарі.

### Одинарні теги HTML

Одинарні теги складаються з одного оператора. Наприклад, тег <br> - перенесення на новий рядок, <hr> - розділова лінія, <img> - уставляння зображення. Властивості для одиночного тегу вказуються всередині через атрибути, наприклад:

<img src=”image/picture.jpg” width=”200” border=”1”>

В наведеному прикладі в одиночному тезі <img> прописано атрибути, які вказують на місце розташування і назву графічного файлу, ширину зображення в 200пікселів і наявність рамки навколо зображення шириною в 1 піксел.

Певний час вимагалося наприкінці одиночного тегу вказувати слеш (<br />, <img />), натепер такий стиль написання вже не є обов’язковим.

### Парні теги HTML

Парних тегів є значно більше і вони вказують браузеру початок і закінчення певного структурного елементу: рядка, блоку, таблиці, заголовку

Наприклад, для того, щоб певний текст відобразився в браузері як рядок, слід застосувати теги, що позначають елемент-параграф (рис. 1):



Рис. 2.1. Склад HTML-елементу

Основними частинами елемента є:

* **Початковий тег:** Він складається з назви елемента (<p>) і є ознакою початку елемента, з цього моменту тег починає впливати на той вміст, що слідує за ним.
* **Кінцевий тег:** виглядає як і початковий, але містить слеш перед назвою тега (</p>). Він вказує на закінчення елементу. Закінчення в парних тегах є обов’язковими, інакше браузер може невірно відобразити вміст.
* **Вміст:** в цьому випадку вмістом є простий текст.
* **Елемент:** початковий тег + вміст + кінцевий тег = елемент.

Всередині елементів можна вкладати інші елементи - це називається вкладеністю. Наприклад, якщо потрібно зробити акцент на певному слові і виділити його грубішим шрифтом.

<p>Цей фрагмент є <strong>окремим рядком</strong> тексту</p>

Якщо застосовується розміщення елементів всередині інших, слід обов’язково зберігати порядок вкладеності. В наступному прикладі наведено невірний запис:

<p>Цей фрагмент є <strong>окремим рядком тексту</p></strong>

Елементи повинні відкриватися і закриватися правильно таким чином, щоб явно перебувати всередині або зовні один одного. Якщо вони перекриваються так, як в прикладі вище, то браузер сам приймає рішення як відображати, що може привести до непередбачуваного результату.

### Теги для коментарів

Коментування в HTML необхідно для покращення читабельності коду. В коментарях, зазвичай, вказується пояснення ділянки коду, що спрощує процес редагування HTML сторінки в подальшому. Розробнику легше орієнтуватися і вносити зміни, оскільки коли коду стає дуже багато, то в ньому легко заплутатися, можна поставити зайвий тег або, навпаки, не закрити його.

Тег для коментаря починається з кутової дужки, знак оклику, два дефіси, текст коментаря, два дефіси, закінчується кутовою дужкою.

<!-- текст коментаря -->

Коментарі в HTML не відображаються на сторінці в браузері користувачеві, їх можна побачити лише в коді веб-сторінки. Для зручності слід коментувати не лише початок певного блоку коду, а й його завершення, наприклад:

<!-- NAVIGATION -->

<nav> ... </nav>

<!-- END NAVIGATION -->

## Структура HTML-документа

Для сторінок будь якої складності існує стандартна структура, яка має обов’язкові елементи.

<!DOCTYPE html>

<html>

 <head>

 <meta charset = "utf-8">

 <title>Тестова сторінка</title>

 </head>

 <body>

 <p>Моя перша сторінка</p>

 </body>

</html>

* **<!DOCTYPE html>**: посилання на набір правил doctypes, яким має слідувати HTML-сторінка, також вказівка на автоматичну перевірку помилок і інші корисні речі.
* **<html>** є кореневим елементом документа. Всі інші елементи містяться всередині тегів <html> ... </html>. Все, що знаходиться за межами цих тегів, не сприймається браузером як код HTML і не обробляється..
* **<head> елемент** містить службову, довідкову та додаткову інформацію, яка призначена для браузера, пошукових систем, сервера і не відображається для відвідувача сторінки. В цьому елементі міститься технічна інформація про сторінку: ключові слова, короткий опис сторінки, під’єднані шрифти, стилі, скрипти, зазначається тип кодування шрифтів і багато іншого.
* **<body> елемент** містить весь контент, який бачать відвідувачі - текст, зображення, відео, анімація тощо.

Елементи, що знаходяться всередині елементу <html>, утворюють дерево документа, так звану об'єктну модель документа, **DOM (Document Object Model)**. Елемент <html> є кореневим елементом, body і head – структурними елементами.



Рис. 2. Найпростіша структура веб-сторінки

У взаємодії елементів веб-сторінки задіяні "родинні стосунки" між елементами. Відносини між множинними вкладеними елементами підрозділяються на батьківські, дочірні та сестринські.

* **Батьківський елемент** - елемент, що пов'язаний з іншими елементами нижчого рівня, і що знаходиться на дереві вище за них. На рис.2 <html> є батьківським лише для <head> і <body>. Елемент <body> є батьківським для всіх елементів, що містяться в ньому: <h1>, <p>, <span>, <nav> і т.д. Тег <p> є батьківським тільки для <span>.
* **Дочірній елемент** - елемент, що розташований всередині іншого елементу. На рис.2 елементи <h1>, <h2>, <p> і <nav> є дочірніми по відношенню до <body>.
* **Сестринський елемент** - елемент, що має загальний батьківський елемент з даним, так звані елементи одного рівня. На рис.2 <head> і <body> - елементи одного рівня, так само як і елементи <h1>, <h2> і <p> є між собою сестринськими.

### Елемент <head>

Інформація всередині тегу не відображається у вікні браузера, однак, ці дані вказують браузеру, серверу чи пошуковій системі, як слід обробляти сторінку.

* **<title> елемент** є обов'язковим для розділу <head>, оскільки текст, що розміщений всередині нього, відображається у заголовку браузера. Текст заголовку повинен містити максимально повний опис вмісту сторінки.
* **<meta> елемент.** Елемент <head> може містити кілька елементів <meta>, які відповідно до використаних атрибутів несуть різну інформацію. За допомогою <meta> можна зазначити короткий опис сторінки і ключові слова для пошукових машин, автора html-документу та інші відомості.
* **<meta charset="тип кодування">** повідомляє браузеру кодування сторінки. Кодування HTML-сторінки потрібно вказувати для того, щоб веб-браузер міг правильно відображати текст на сторінці. Якщо браузер неправильно «вгадає» кодування, то замість тексту будуть відображатися ієрогліфи. Стандартом на сьогодні є кодування - utf-8, що містить символи зі всіх відомих природних мов, тому цей тег має вигляд <meta charset=" utf-8">.
* **Елемент <style>.** Всередині цього елемента задаються стилі, які використовуються на сторінці.
* **Елемент <link>.** Даний елемент визначає відношення між поточною сторінкою та іншими документами. Таких елементів на сторінці може бути кілька. Наприклад, вказати зв’язок сторінки з файлом зі стилями (окремий файл з розширенням .css), тоді посилання на файл зі стилями буде виглядати так <link href="style.css">. Обов’язковим є вказування атрибуту rel, де вказується як цей документ буде розпізнано браузером (<link rel="stylesheet" href=”style.css”> - документ є таблицею стилів, <link rel="icon" href="favicon.png"> - документ є фавіконкою). Додатково вказують тип документу атрибутом type (<link rel="stylesheet" type="text/css" href="style.css">, <link rel=" icon" type="image/x-icon" href="favicon.png">)
* **Елемент <script>.** Дозволяє приєднувати до документа різні сценарії. Закриваючий тег є обов'язковим, при цьому текст сценарію може розташовуватися або всередині цього елемента, або в зовнішньому файлі. Якщо текст сценарію розташований в зовнішньому файлі, то він під’єднується за допомогою атрибутів елементу, наприклад: <script src=" scripts.js"></script>

### Елемент <body>

Всередині елементу <body> міститься інформація, яка відображається в браузері. В таблиці <https://html5book.ru/html-tags/> наведено повний перелік елементів, що підтримуються HTML5.

За призначенням теги можна умовно розподілити:

* **Контейнерні:** div, article, aside, header, footer, span, p, h1-h6, ul, ol
* **Функціональні:** a, img, table, form, input, audio, video, canvas
* **Фрейми:** iframe

У тегів можуть бути атрибути, які повідомляють браузеру, яким чином має відображатися той чи інший елемент сторінки та особливості відображення даного елементу. Значення атрибута завжди заключають в лапки " ". Назви та значення атрибутів не є чутливими до регістру, але, рекомендується набирати їх у нижньому регістрі.

<тег атрибут1="значення1" атрибут2=”значення1 значення2”>контент</тег>

Атрибути можна поділити на **універсальні (глобальні)** які можна використані для будь-якого HTML-елемента, та **власні**, які притаманні лише цьому тегу.

#### Універсальними атрибутами є

* **title** Вказує додаткову текстову інформацію про елемент, що відображається у спливаючій підказці над елементом, наприклад: **<div title=”Червоний блок”>**.
* **style** Вказує код CSS, що застосовується для оформлення даного елементу, наприклад: **<div style=”width: 100%; background:red; color:#ffffff;”>**.
* **class** Вказує назву класу для елемента, сам клас описано в таблиці стилів, наприклад: **<div class=”red\_box”>**. Атрибут *class* застосовний до всіх HTML-елементів.
Кожному конкретному елементу можна привласнити **кілька значень class**. Множинні значення *class* записуються через пробіл, наприклад, **<div class = "nav top">**. Назви класу повинні складатися з букв, цифр, дефісів і нижніх підкреслень і починатися тільки з літери.
* **id** Вказує унікальний ідентифікатор елемента, наприклад: **<div id=”first\_item”>**. Атрибут *id* застосовний до всіх HTML-елементів. Кожному конкретному елементу можна привласнити **лише одне значення id**. Назви id повинні складатися тільки з букв, цифр, дефісів і нижніх підкреслень і повинні починатися тільки з літер.

#### Умовно елементи можна поділити на групи

* **Блокові елементи.** Характеризуються тим, що починаються з нового рядка, займають всю доступну ширину, висота елемента визначається його вмістом.
* **Рядкові елементи.** Є безпосередньою частиною іншого елемента, наприклад, текстового абзацу. В основному використовуються для зміни вигляду тексту, його логічного виділення чи акцентування.

### Блокові елементи

#### Загальні характеристики блокових елементів

* На початку і в кінці блокових елементів, автоматично ставиться перенесення рядка.
* За замовченням блокові елементи займають всю ширину батьківського елементу.
* В блокових елементах можна керувати шириною і висотою: width, height.
* В блокових елементах можна керувати зовнішніми і внутрішніми відступами.
* Об'єктами, що знаходяться всередині блокового елементу, можна керувати за допомогою горизонтального та вертикального вирівнювання.
* Всередині блокових елементів можна розміщувати інші блокові елементи, а також рядкові елементи.

#### Типові блокові елементи

* **<div>** Універсальний блоковий елемент.
* **<article>, <aside>, <header>, <footer>** Семантичні блоки для позначення структурних ділянок сторінки.
* **<p>, <h1>-<h6>, <blockquote>** Елементи для HTML тексту.
* **<hr>** Горизонтальна лінія.
* **<ol>, <ul>** Встановлює нумерований (маркірований) список.
* **<table>** Створює таблицю.
* **<form>** Встановлює форму на веб-сторінці.

### Рядкові елементи

#### Загальні характеристики рядкових елементів

* На відміну від блокових, рядкові елементи слідують один за одним і НЕ переносяться на новий рядок.
* В рядкових елементах не можна керувати шириною і висотою: width, height.
* В рядкових елементах можна встановлювати лише внутрішні відступи (padding) та зовнішні бічні відступи (margin).
* Рядкові елементи НЕ займають всю ширину батьківського елементу.
* Оскільки рядкові елементи НЕ займають всю ширину батьківського елементу, то елементами, що знаходяться всередині рядкових елементів, не можна керувати за допомогою горизонтального чи вертикального вирівнювання.
* Всередині рядкових елементів можна розміщувати інші рядкові елементи, якщо розмістити там блоковий елемент, то він розірве логічний потік і розміститься з нового рядка.

#### Типові рядкові елементи

**<a>** Є одним з важливих елементів HTML і призначений для створення посилань.

**<span>** Універсальний рядковий елемент.

**<br>** Розірвання рядка.

**<img>** Уставляння зображення.

## HTML текст

Елементи для форматування тексту несуть змістовне навантаження і, зазвичай, задають для тексту структурне та стильове оформлення, наприклад, виділяють текст грубішим шрифтом або відображають його іншим шрифтом.

Грамотно відформатований текст надає для пошукових систем відомості, які фрагменти тексту мають важливий зміст, за якими з них слід ранжирувати веб-сторінку в пошуковій видачі.

#### Елемент абзацу <p>

Розділяє текст на окремі абзаци, відокремлюючи їх збільшеною відстанню. Браузер автоматично додає верхній і нижній відступ.

#### Елементи заголовків <h1>-<h6>

Заголовки є важливим елементом веб-сторінки, вони допомагають впорядкувати текст, сформувати його структуру. У специфікації HTML існує шість рівнів заголовків, завдяки яким можна легко виділяти теми і підтеми.

Заголовки є блоковими елементами, завжди починаються з нового рядка. Браузер автоматично додає перед заголовком і після нього збільшені відступи.

Текст в заголовках впливає на індексацію сайту пошуковими системами, оскільки багато роботів звертають увагу саме на вміст заголовків сайту, тому, краще завжди використовувати ці елементи, міняючи зовнішній вигляд і розмір за допомогою стилів.

При використанні заголовків необхідно враховувати їх ієрархію, тобто за <h1> повинен слідувати <h2> і т.д. Також, не рекомендується вкладення інших елементів в <h1> ... <h6>.

#### Елементи для форматування тексту

* **<strong>** задає грубіше накреслення шрифту, вказуючи браузеру на важливість тексту. Є сучасною альтернативою для попереднього елементу <b>.
* **<em>** відображає текст нахиленим шрифтом (курсивом). Є сучасною альтернативою для елементу <i>, який на сьогоднішній день охоче застосовують для розміщення іконок. Для нахилу тексту можна використати елемент <cite>.
* **<q>** додавання зовнішніх лапок до фрагменту.
* **<code>** зміна шрифту на більш технічний шрифт.
* **<del>** перекреслений текст.
* **<ins>** підкреслений текст.
* **<sub>** використовується для створення нижніх індексів. Зсуває текст нижче рівня рядка, зменшуючи його розмір.
* **<sup>** використовується для створення ступенів. Зсуває текст вище рівня рядка, зменшуючи його розмір.
* **<pre>** дозволяє вивести текст на екран, зберігши початкове форматування. Пробіли і переноси рядків при цьому не видаляються.
* **<blockquote>** Використовується для оформлення фрагменту особливим блоком, наприклад, цитати, виділяючи його відступами і переносами рядків.
* **<br>** Переносить текст на наступний рядок, створюючи розрив рядка.
* **<hr>** Використовується для поділу контенту на веб-сторінці. Відображається у вигляді горизонтальної лінії.

## HTML посилання

HTML посилання складають основу Веб, вони пов'язують між собою різні веб-сторінки.

Посилання, або гіперпосилання, створюються за допомогою тегу <a>. Посилання складається з двох частин: покажчика посилання та адресної частини посилання, наприклад:

<p>Замовити книжки у <a href=”http://shop.ua”>магазині</a></p>

Покажчик посилання представляє фрагмент тексту або зображення, який візуально виділяється в документі (за замовчуванням, синім кольором і підкресленням).

Адресну частину посилання для користувача не видно, вона представляє URL-адресу об'єкта, до якого необхідно перейти.

### Уставляння гіпертекстових посилань в HTML-документ

#### Uniform Resource Locator, URL

Єдиний покажчик ресурсів (URL) визначає місцезнаходження ресурсу. В загальному вигляді має наступний формат:

метод://ІР адреса сервера|доменна адреса сайту/шлях#якір.

Адреси описують, де знаходиться ресурс, наприклад, www.site.ua. Якщо адреси не вказано, то посилання вважається локальним, тобто, воно відноситься до тієї ж машини, на якій знаходиться html-документ, що містить посилання.

Далі вказується шлях (частковий або повний), за яким здійснюється перехід.

Під якорем розуміють посилання на місце всередині поточного html-документа.

Сам текст URL не відображається браузером, він використовується для виконання запропонованих дій при активації посилання (зазвичай, клацанням миші).

Найбільш поширені методи доступу: http (https), mailto, tel, viber.

* **Метод http (https)** надає доступ до веб-сторінки за протоколом HTTP, наприклад:
<a href="https://lpnu.ua/">НУ "Львівська політехніка"</a>
* **Метод mailto** запускає сеанс поштового зв'язку із зазначеним адресатом і хостом, наприклад:
<a href="mailto:Iryna.Y.Yurchak@lpnu.ua">Iryna.Y.Yurchak@lpnu.ua</a>
* **Метод tel** забезпечує з’єднання з програмою для здійснення телефонних дзвінків. Такий метод спрацьовує в мобільних пристроях і користувач може відразу позвонити за вказаним номером, наприклад: <a href="tel:0679476121">(067)947-61-21</a>
* **Метод viber** здійснює з’єднання з месенжером Viber, наприклад:
<a href="viber://chat?number=+380679476121">

### Абсолютний URL

Абсолютний шлях містить всю інформацію, що необхідна браузеру для знаходження файлу: назву протоколу, доменну або IP-адресу комп'ютера (хосту), папку (шлях до файлу) та ім'я файлу. Наприклад,

https://www.site.ua/folder/file.html

Якщо файл знаходиться в кореневій папці, то він вказується відразу:

https://www.site.ru/file.html

У разі відсутності назви файлу буде завантажуватися веб-сторінка, яка задана за замовченням у налаштуваннях веб-сервера (зазвичай, це index.\* ):

http://www.site.ua/

http://www.site.ua/folder/

Наявність закриваючого слешу / означає, що звернення йде до папки, а якщо його немає - то безпосередньо до файлу.

### Відносний URL

Відносна URL-адреса описує шлях до зазначеного документа відносно поточного, вона визначається з врахуванням місця розташування веб-сторінки, на якій знаходиться посилання. Відносні посилання використовуються при створенні посилань на інші документи на одному сайті. У такому випадку не вказується протокол або домен, а тільки сам шлях до файлу.

1. Наприклад, у разі переходу зі сторінки <http://site.ua/folder1/folder2/file1.html> на сторінку <http://www.site.ua/folder1/folder2/file2.html>, тобто потрібна веб-сторінка знаходиться в тій же папці, що і веб-сторінка, що містить посилання, то посилання буде мати наступний запис:

<a href="file2.html">Текст посилання</a>

2. Якщо зі сторінки <http://www.site.ua/folder1/flder2/file1.html> потрібно перейти на сторінку <http://www.site.ru/folder1/folder2/folder3/file2.html>, то посилання буде таким:

<a href="folder3/file2.html">Текст посилання</a>

3. При переході зі сторінки <http://www.site.ua/folder1/folder2/file1.html> на сторінку <http://www.site.ua/folder1/file2.html> посилання буде мати наступний вигляд:

<a href="../file2.html">Текст посилання</a>

3. У разі переходу з <http://www.site.ua/folder1/folder2/folder3/file1.html> на сторінку <http://www.site.ua/folder1/file2.html> посилання буде таким:

<a href="../../file2.html">Текст посилання</a>

### Зовнішні посилання

Посилання на веб-сторінки інших сайтів є зовнішніми, для них завжди вказують Абсолютний URL, наприклад:

<a href="https://english-films.co/">Фільми англійською мовою</a>

### Посилання на розділи поточної сторінки

Внутрішні посилання посилаються на різні частини поточної сторінки, що сприяє швидкому переміщенню, коли на сторінці багато тексту.

Внутрішні посилання також вставляються за допомогою тегу <a> з різницею в тому, що атрибут href містить назву покажчика (ідентифікатор), а не URL-адресу. Перед назвою покажчика ставиться знак #.

Наприклад, наступний запис описує код зі швидкими переходами на відповідні розділи.

<h1>Пори року</h1>

<h2>Зміст</h2>

 <a href="#p1">Літо</a> <!--Задаємо якір через id елемента-->

 <a href="#p2">Осінь</a>

 <a href="#p3">Зима</a>

 <a href="#p4">Весна</a>

<p id="p1"> ... </p> <!--Додаємо відповідний id елементу-->

<p id="p2"> ... </p>

<p id="p3"> ... </p>

<p id="p4"> ... </p>

Якщо потрібно зробити посилання з однієї сторінки сайту на певний розділ іншої сторінки сайту, то необхідно задати id для цього розділу сторінки, а потім додати його до абсолютної або відносної адреси:

<img src=”picture.png” id="picture1">

<a href="http://html5.ua/page.html#picture1">Вигляд картинки</a>

Або

<a href="page.html#picture1" target="\_blank"> Вигляд картинки</a>

### Посилання на сторінки одного сайту

При створенні посилань на сторінки власного сайту задаються відносні посилання, які звертаються до певної сторінці на одному сервері.

Коли браузер не знаходить у посиланні протокол https://, він виконує пошук зазначеного документа на тому ж сервері. Відносні посилання описують шлях до потрібного файлу відносно розташування поточного документа.

У випадку, якщо сторінки знаходяться в одному каталозі, то замість запису

<a href="http://html5book.ua/styling.html">Оформлення гіпертекстових посилань</a>

досить буде вказати ім'я файлу з розширенням (якщо воно задане в адресі сторінки)

<a href="styling.html">Оформлення гіпертекстових посилань</a>.

Якщо документи знаходяться в різних каталогах, і кінцевий файл знаходиться у вкладеному каталозі відносно поточного, необхідно вказати назву каталогу і назву сторінки через слеш, наприклад,

<a href="lessons/lesson1.html">Урок 1</a>

При створенні посилання на файли з батьківського каталогу, шлях до файлу повинен починатися з ../, що дослівно означає: повернутися на рівень вище і слідувати за зазначеним шляхом, аналогічно з кнопкою "Назад".

Можна задавати шлях до документів відносно кореневого каталогу. Звернення до кореневого каталогу здійснюється через символ /. Далі вказуються каталоги, в які браузер повинен перейти, щоб дістатися до потрібного документа.

<a href="/lessons/lesson1.html">Урок 1</a>

### Елементи, до яких застосовують посилання

Засобами HTML можна створювати посилання за допомогою тексту, зображень та блокових елементів, тобто, можна об'єднувати в одному посиланні текст, зображення та блоки:

 <a href="http://summer.ua">

 <h1>ЛІТО</h1>

 <img src="logo.png" alt="ЛІТО">

 <div>

 <ul>

 <li>Пропозиції по відпочинку</li>

 <li>Акції та знижки</li>

 </ul>

 </div>

 </a>

### Атрибути посилань

Окрім глобальних атрибутів, елемент <a> підтримує власні атрибути.

* **href** - значенням атрибута є URL-адреса документа, на яку вказує посилання.
* **nofollow** - забороняє роботам пошукової системи переходити за посиланнями на даній сторінці або за конкретними посиланнями.
* **target** - вказує на те, в якому вікні повинна відкриватися сторінка або документ, до якого веде посилання. Приймає наступні значення:

**\_self** - сторінка завантажується до поточного вікна (вкладки). Значення за замовченням;

**\_blank** - сторінка відкривається в новому вікні (вкладці) браузера;

**\_parent** - сторінка завантажується у фрейм-батько;

**\_top** - сторінка завантажується в повне вікно браузера.

## HTML зображення

Використання графіки робить веб-сторінки візуально привабливими, зображення допомагають краще передати суть і зміст веб-документа. Також, за допомогою HTML-тегів можна робити карти-зображення з активними областями.

### Уставляння зображень в HTML-документ

#### Елемент <img>

Зображення додаються на сторінку за допомогою одинарного тега <img>. Оскільки елемент <img> є рядковим, то рекомендується розташовувати його всередині блокового елемента, наприклад, <p> або <div>.

Елемент <img> має обов'язковий атрибут **src**, значенням якого є адреса втіленого зображення, і рекомендований атрибут alt, наприклад:

<img src="image.png" alt="Приклад коду">

Для елементу <img> доступні наступні атрибути:

* **src** - вказує URL-адресу зображення (де знаходиться зображення).
* **alt** - зазначає альтернативний текст для зображення. Виводиться на місці появи зображення до його завантаження або при відімкнутій графіці, в деяких браузерах виводиться спливаючою підказкою при наведенні вказівника на зображення. Цей атрибут є важливим для пошукових систем.
* **width** - задає ширину зображення. Прийняті значення: пікселі або відсотки.
* **height** - задає висоту зображення. Прийняті значення: пікселі або відсотки.

#### Адреса зображення

Адреса зображення може бути вказана повністю (абсолютний URL), наприклад:

<img src="http://site.ua/images/photo.png">

Або через відносний шлях від документа або кореневого каталогу сайту:

<img src="../images/photo.png"> - відносний шлях від документа,

<img src="/images/photo.png"> - відносний шлях від кореневого каталогу.

#### Розміри зображення

Без завдання розмірів зображення відображається на сторінці в реальному розмірі. Відредагувати розміри зображення можна за допомогою атрибутів width і height. Якщо буде задано лише один з атрибутів, то інший буде обчислюватися автоматично для збереження пропорцій малюнка.

#### Формати графічних файлів

* Растровий формат **JPEG (.JPG).** Зображення JPEG є ідеальними для фотографій, вони містять мільйони різних кольорів.
* Растровий формат **GIF.** GIF-файли дозволяють встановити один з кольорів 100% прозорим, завдяки чому фон веб-сторінки буде видно через частину зображення (напівпрозорі ділянки зафарбовуються в суцільний колір). GIF-файли можуть містити просту покадрову анімацію. Основний недолік у використанні індексованої палітри кольорів, GIF-зображення містять всього лише 256 кольорів, через що зображення виглядають плямистими і нереалістичного кольору.
* Растровий формат **PNG.** Містить кращі властивості GIF- і JPEG-форматів. Містить мільйони кольорів і надає можливість зберігати напівпрозорі та прозорі ділянки. Недоліком є значно більший розмір файлу, ніж при JPEG-форматі.
* Векторний формат **SVG.** SVG-зображення, що складається з набору геометричних фігур, які описані у форматі XML: лінія, еліпс, багатокутник тощо. Підтримується як статична, так і анімована графіка. Набір функцій містить різні перетворення, альфа-маски, ефекти фільтрів, можливість використання шаблонів. Зображення у форматі SVG можуть змінюватися в розмірі без зниження якості.
* Растровий формат **WebP**. Веб-формат від Google, що створено в 2010 році. Використовує потужні способи оптимізації зображень для зменшення ваги файлу, підтримує прозорість і анімацію. На сьогоднішній день формат підтримують популярні браузери (Chrome, Firefox, Edge), у застарілих браузерів зображення в цьому форматі не відобразиться.
* Растровий формат **APNG.** Формат зображення, що засновано на форматі PNG від Mozilla в 2008році. Дозволяє зберігати анімацію, а також підтримує прозорість. Дозволу з боку Portable Network Graphics формат APNG не отримав, тому його неможна віднести ані до веб-стандарт, ані до стандарту PNG та libpng. Тому, робота з даним форматом може бути зі збоями.
* Растровий формат **ICO.** Формат зберігання фавіконок, що відображаються на закладці в браузері або поруч з адресою сайту у сторінці видачі результатів пошукової системи. На сьогоднішній день формат вважається застарілим, а фавіконки застосовують у форматах png та svg. Найбільш вживані розміри фавіконок: 16, 32, 48, 57, 72, 114, 256 пікселів.

## HTML списки

HTML списки представляють набір згрупованих абзаців тексту, помічених значками (маркірований список) або цифрами (нумерований список). Елементи списку додаються за допомогою елементу <li> (List Item - елемент списку). Для елементу <li> доступні різні атрибути, що дозволяють змінити нумерацію або маркірування обраного елемента списку.

#### Маркірований список

Маркірований список представляє невпорядкований перелік елементів (Unordered List). Створюється за допомогою парного елементу <ul>. На початку кожного елемента встановлюється маркер, наприклад, зафарбований кружечок.

|  |  |
| --- | --- |
| <ul> <li>Microsoft</li> <li>Google</li> <li>Apple</li> <li>IBM</li></ul> | * Microsoft
* Google
* Apple
* IBM
 |

#### Нумерований список

Нумерований список поміщають всередину пари тегів <ol>. Кожен пункт списку потрібно помістити в елемент <li>. Браузер автоматично проставляє номери елементів по порядку і якщо видалити один або кілька елементів такого списку, то інші номери будуть автоматично змінені.

Для тегу <ol> доступні наступні атрибути:

* **start** - задає початкове значення, від якого починається відлік нумерації, наприклад, конструкція <ol start = "10"> першому пункту привласнить порядковий номер "10". Також можна одночасно задавати тип нумерації, наприклад, <ol type = "I" start = "10">.
* **type** - задає тип нумерації 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 start=”5”> <li>Microsoft</li> <li>Google</li> <li>Apple</li> <li>IBM</li> </ol> | 1. Microsoft
2. Google
3. Apple
4. IBM
 |

#### Список визначень

Списки визначень описуються за допомогою елементу <dl>, який містить пари термін-визначення. Для додавання терміну застосовується блоковий елемент <dt>, а для уставляння визначення - <dd>.

Для елементів <dl>, <dt> і <dd> доступні глобальні атрибути.

|  |  |
| --- | --- |
| <dl> <dt>Національний університет «Львівська політехніка»:</dt> <dd>ІКНІ</dd> <dt>Кафедри</dt> <dd>САПР</dd> <dd>ІСМ</dd> <dd>АСУ</dd></dl> | **Національний університет «Львівська політехніка»:** ІКНІ**Кафедри** САПР ІСМ АСУ |

#### Вкладений список

Іноді можливостей простих списків бракує, наприклад, при створенні змістів потрібно застосувати вкладені пункти. Код для створення вкладеного списку буде мати наступний вигляд:

|  |  |
| --- | --- |
| <ul> <li>Пункт 1.</li> <li>Пункт 2. <ul> <li>Підпункт 2.1.</li> <li>Підпункт 2.2. <ul> <li>Підпункт 2.2.1.</li> <li>Підпункт 2.2.2.</li> </ul> </li> <li>Підпункт 2.3.</li> </ul> </li> <li>Пункт 3.</li></ul> | * Пункт 1.
* Пункт 2.
	+ Підпункт 2.1.
	+ Підпункт 2.2.
		- Підпункт 2.2.1.
		- Підпункт 2.2.2.
	+ Підпункт 2.3.
* Пункт 3.
 |

Більш докладну інформацію про HTML елементи можна дізнатися у специфікації мови та довідниках для веб-розробників.

## Контрольні питання

1. Що собою представляють теги HTML?
2. Навести відмінності між одинарними та парними тегами.
3. Навести найпростішу структуру веб-сторінки.
4. Перелічити універсальні атрибути для тегів HTML.
5. Пояснити різницю між блоковими та рядковими елементами HTML.
6. Навести приклади посилання з абсолютним та відносним шляхом.
7. Які атрибути застосовують до посилань в HTML-документі?
8. Які атрибути застосовують до зображень в HTML-документі?
9. Які графічні формати підтримує HTML?
10. Перелічити відомі HTML-списки.

## Використані джерела

1. Вступ у HTML - [https://developer.mozilla.org/ru/docs/Learn/HTML/Введение\_в\_HTML](https://developer.mozilla.org/ru/docs/Learn/HTML/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B2_HTML)
2. HTML, HTML5 - <http://html5book.ru/html-html5/>
3. HTML5 Семантичні елементи - <https://html5css.ru/html/html5_semantic_elements.php>
4. Блокові і рядкові елементи - <https://html5book.ru/block-inline-elements/>
5. Абсолютні і відносні посилання - <https://htmlacademy.ru/blog/boost/frontend/links>
6. Зображення в HTML - <https://developer.mozilla.org/ru/docs/Learn/HTML/Multimedia_and_embedding/Images_in_HTML>
7. Все про HTML списки - <https://guruweba.com/html/sozdanie-spiskov-vse-o-html-spiskakh/>