

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування
імені адмірала Макарова

О. І. ЛИТВИНЕНКО

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт
з курсу «WEB-програмування. HTML»

Рекомендовано Методичною радою НУК



2026

УДК 004.42(076.5)

Л64

Автор О. І. Литвиненко, канд. техн. наук, доцент кафедри теплотехніки ХННІ НУК

Рецензенти: О. М. Дудченко, канд. техн. наук, професор;

М. Б. Літвінова, д-р пед. наук, канд. фіз.-мат. наук, професор

Рекомендовано Методичною радою НУК

Литвиненко О. І.

Л64 Методичні вказівки до виконання лабораторних робіт з курсу
«WEB-програмування. HTML» / О. І. Литвиненко. – Миколаїв : НУК,
2026. – 80 с.

У вказівках подано завдання до лабораторних робіт і приклади їх виконання.

Призначено для студентів ХННІ НУК, які навчаються за спеціальністю 121 «Інженерія програмного забезпечення», а також для студентів усіх інших споділених спеціальностей, що вивчають основи WEB-програмування.

УДК 004.42(076.5)

© О. І. Литвиненко, 2026

© Національний університет кораблебудування
імені адмірала Макарова, 2026

ЗМІСТ

ВСТУП.....	4
<i>Робота 1. HTML. Основні теги. Заголовки.</i>	
Абзаци. Посилання та атрибути.....	7
<i>Робота 2. Мнемоніки. Логічні розділи. Зображення.</i>	
Графічні формати. Блоки.....	20
Зразки оформлення звітів до лабораторних робіт.....	40
Довідка А. Основи HTML.....	50
Рекомендована література.....	79

ВСТУП

Дисципліна «WEB-програмування» призначена для набуття студентами базових та прикладних знань основних засобів вебпрограмування мовою розмітки гіпертексту HTML та набуття практичних навиків вебпрограмування та вебдизайну при створенні цілісних вебпроектів.

Програма навчальної дисципліни «WEB-програмування» розрахована на студентів, які вивчили «Основи програмної інженерії», «Об'єктно-орієнтоване програмування» та «Бази даних».

Дисципліна «WEB-програмування» носить міждисциплінарний характер, вона забезпечує підготовку студентів до вивчення навчальних дисциплін «Організація та технології передачі даних у комп'ютерних мережах», «Проектування інформаційних систем промислових підприємств».

Зміст лабораторної роботи студентів з дисципліни визначається навчальною програмою дисципліни та робочою навчальною програмою вивчення дисципліни.

Навчально-методичні засоби виконання для лабораторної роботи:

- основна література (підручник, конспект лекцій, навчальні та методичні посібники);
- додаткова література (наукова, фахова, періодична);
- методичні матеріали.

Вимоги до лабораторної роботи студента:

- робота має бути виконана особисто студентом або групою студентів, де кожен її член самостійно виконує свою частку колективної роботи;
- робота повинна являти собою закінчену розробку (або її етап), де розкриваються й аналізуються актуальні проблеми з певної теми чи її окремих аспектів;
- робота має демонструвати достатню компетентність автора (авторів) у розкритті питань, що досліджуються;
- робота повинна мати навчальну, наукову та (або) практичну спрямованість і значимість, містити певні елементи новизни (при виконанні науково-дослідної роботи).

Оформлення звітів з лабораторної роботи студентів здійснюється відповідно до вимог, розроблених кафедрою, та інших нормативних документів, що стосуються виконання та оформлення наукових, навчально-методичних та інших робіт.

Формами контролю студентів за якістю оволодіння навчальним матеріалом є: самоконтроль за допомогою контрольних-тестових завдань і контроль з боку викладача, який здійснюється за допомогою методів поточного та підсумкового контролю.

Методами поточного контролю є:

- усне опитування студентів на лабораторних заняттях;
- перевірка практичних завдань, виконаних студентами індивідуально;
- співбесіда.

Формами підсумкового контролю є:

- проведення контрольних робіт в аудиторії (за тематичними модулями);
- екзамен (залік) за передбаченими програмою курсу питаннями та практичними завданнями.

Облік успішності студентів з виконання лабораторної роботи здійснюють викладачі у журналах обліку успішності.

Метою дисципліни є формування у студентів здатностей:

- створювати прості вебсторінки;
- організовувати механізми взаємодії з користувачем;
- розв'язувати практичні задачі проектування структури вебсайтів.

Кожна лабораторна робота складається з коротких теоретичних відомостей, завдань для самостійного виконання та контрольних питань.

ЛАБОРАТОРНА РОБОТА 1

Для виконання лабораторних робіт можна використовувати будь-який безкоштовний HTML&CSS&JS редактор.

Що таке HTML

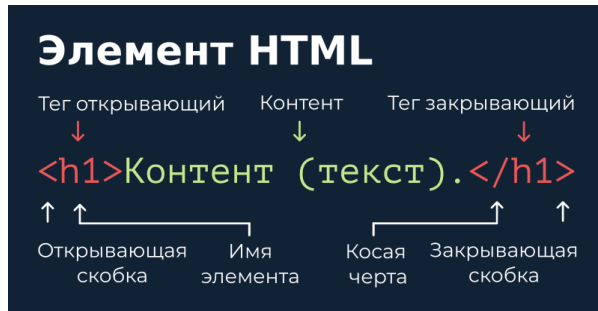
Теорія

HTML – це спеціальний текст, який розуміє браузер. Браузер читає HTML-текст і показує його в зрозумілому і зручному для людини вигляді.

Окремі частини такого тексту мають назву **HTML-теги**.

Теги – це метод дати можливість браузеру, як виділити і відобразити контент.

Синтаксис тегу



На вебсторінках **основний заголовок h1** використовують лише один раз, як головний заголовок сторінки. Для його створення використовують тег `<h1>`, наприклад:

Код	Результат відображення у браузері
<code><h1>Заголовок всієї сторінки</h1></code>	Заголовок всієї сторінки



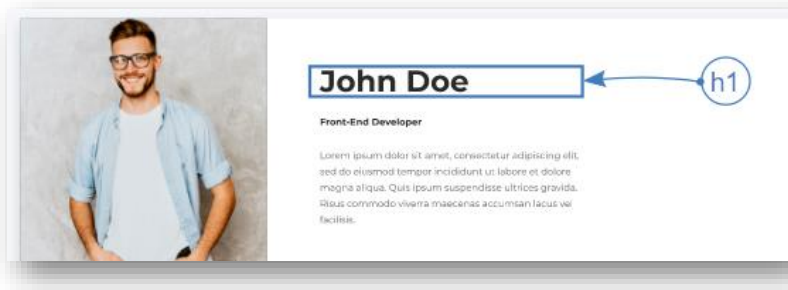
ЗАВДАННЯ 1

Напишіть заголовок `<h1>` зі своїм ім'ям і прізвищем.

Критерії прийому завдання

- Код повинен мати тег-заголовок з ім'ям `<h1>`.
- Текст заголовка не повинен бути порожнім.
- Інших тегів у тексті не повинно бути.

Макет 1



Абзац (параграф)

Теорія

Абзац (параграф) – універсальний контейнер для групування тексту.

За замовчуванням абзац починається з нового рядка і має вертикальні відступи, які можна буде змінити в CSS.

Окремі абзаци в термінах HTML мають назву *параграфи*. Текст кожного параграфа обертається в тег `<p>`.

Приклад параграфа:

Код
<pre><p> My first paragraph. Lorem ipsum dolor sit amet, consectetur adipisicing elit. </p> <p> My second paragraph. Neque eligendi, a eaque, esse itaque porro non exercitationem odio earum quos perferendis! </p></pre>
Результат відображення у браузері
My first paragraph. Lorem ipsum dolor sit amet, consectetur adipisicing elit. My second paragraph. Neque eligendi, a eaque, esse itaque porro non exercitationem odio earum quos perferendis!



ЗАВДАННЯ 2

Додайте два теги абзаців. Один для назви професії, а другий для біографії. Не видаляйте існуючий тег **<h1>**.

Критерії прийому завдання

- В коді повинні бути тег **<h1>** завдання 1.
- Текст першого параграфа – **Front-end developer**.
- Текст другого параграфа не повинен бути порожнім.
- Інших тегів у тексті не повинно бути.

Макет 2



Заголовки

Теорія

Заголовки – це теги, які позначають важливість логічних розділів різного рівня. Тегами заголовків контенту додають потрібну структуру.

Код
<pre><h1>Заголовок всієї сторінки</h1> <h2>Заголовок всієї статті</h2> <p> Lorem ipsum dolor sit amet consectetur adipisicing elit. Unde voluptas necessitatibus, doloremque vero pariatur aut delectus neque sint maxime! </p></pre>
Результат відображення в браузері
Заголовок всієї сторінки Заголовок всієї статті Lorem ipsum dolor sit amet consectetur adipisicing elit. Unde voluptas necessitatibus, doloremque vero pariatur aut delectus neque sint maxime!

Код після виконання завдання	Результат відображення у браузері
<code><h2>Заголовок другого рівня</h2></code>	Заголовок другого рівня



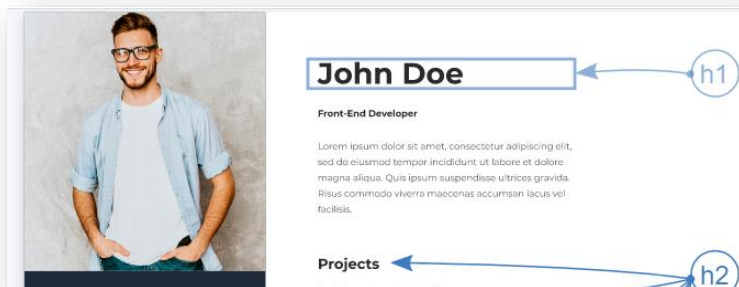
ЗАВДАННЯ 3

Напишіть заголовок `<h2>` з с текстом **Projects**.

Критерії прийому завдання

- У коді повинні бути теги завдань 1-2.
- Код повинен мати заголовок другого рівня `<h2>`.
- Текст заголовка другого рівня повинен дорівнювати **Projects**.
- Інших тегів у тексті не повинно бути.

Макет 3



Теги ol та li

Списки дозволяють впорядкувати набір елементів і записати їх в зручному вигляді. Список – це контейнер. Всередині його розташовуються тільки елементи списку – теги `<li>`.

Нумерований (впорядкований) список – це список, в якому кожен елемент **li** починається з числа. Браузер автоматично нумерує елементи за порядком. **Впорядкований** список використовують для перелічення дій у певному порядку. Приклад:

Код
<pre><h1>Як заварити чай</h1> <p>Покрокова інструкція, просто повтори і в тебе все вийде! </p> Закип'ятити воду Всипати чай до чашки Налити в чашку окропу Чекати 10 хвилин, після чого можна пити! </pre>
Результат відображення у браузері
Як заварити чай Покрокова інструкція, просто повтори і в тебе все вийде! 1. Закип'ятити воду 2. Всипати чай до чашки 3. Налити в чашку окропу 4. Чекати 10 хвилин, після чого можна пити!



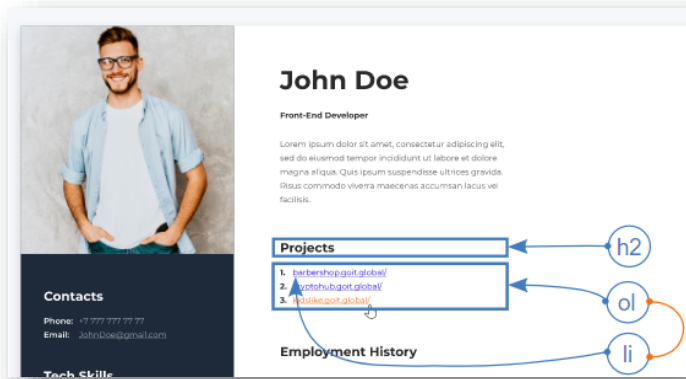
ЗАВДАННЯ 4

Використайте тег `<ol>` для розмітки списку проєктів.

Критерії прийому завдання

- У коді повинні бути теги завдань 1-3.
- Код повинен мати нумерований список `<ol>`.
- Всередині списку `<ol>` повинно бути три елементи `<li>`.
- Значення кожного елемента `<li>` не повинно бути порожнім.

Макет 4



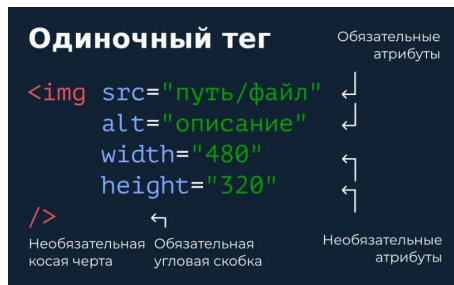
Посилання та атрибути

Теорія



Атрибути – це додаткові параметри тегів, за допомогою яких можна змінювати властивості та поведінку елемента. Кожен атрибут має ім'я та значення.

Атрибути записуються всередині тегу, що відкриває, а їх значення розташовуються всередині лапок. Атрибутів може бути кілька або зовсім не бути.



Деякі теги мають обов'язкові атрибути.

```
<!--обов'язковий атрибут href -->
<!--необов'язковий атрибут class -->
<a href="https://google.com" class="link"> link </a>
```

Посилання

Тег **<a>** призначений для створення посилань. При натисканні на посилання ми переходимо на інші сторінки, завантажуюємо файли. Браузер підкреслює посилання та виділяє їх синім кольором.

Адреса посилання задається в обов'язковому атрибуті **href="адреса"**. Адреса – це URL-адреса, яка вказує на іншу вебсторінку або файл.

Наприклад, ось посилання на Google:

Код
<code> https://google.com </code>
Результат відображення у браузері
https://google.com



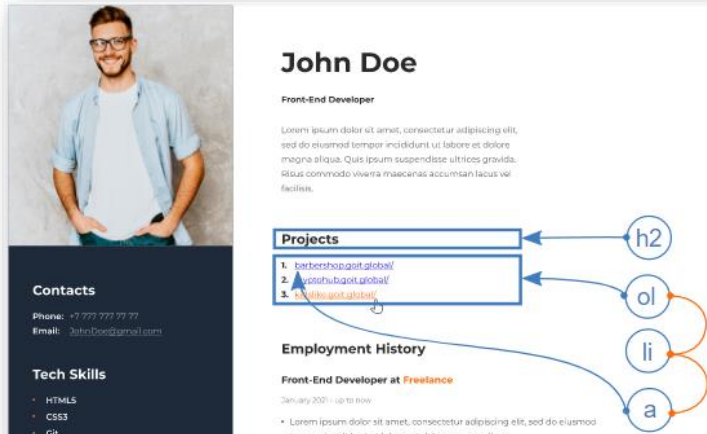
ЗАВДАННЯ 5

Обгорніть текст всередині кожного `<li>` посиланням. Значення атрибуту `href` і текст посилання мають бути однаковими. Не видаляйте існуючі теги.

Критерії прийому завдання

- В коді повинні бути теги завдань 1-4.
- Всередині кожного елемента `<li>` повинен бути тег `<a>`.
- Текст кожного тега `<a>` повинен бути рівним атрибуту `href` посилання.
- Атрибут кожного посилання `href` повинен бути URL (починатися з `https://`).

Макет 5



Заголовок h3

Теорія

Тег `<h3>` – заголовок третього рівня.

Усього існує 6 тегів заголовків від `<h1>` до `<h6>`. Браузер за замовчуванням показує найбільший заголовок `h1` і далі зменшує розмір аж до `h6` – найменшого за розміром. Використовувати відповідний заголовок потрібно за змістом.



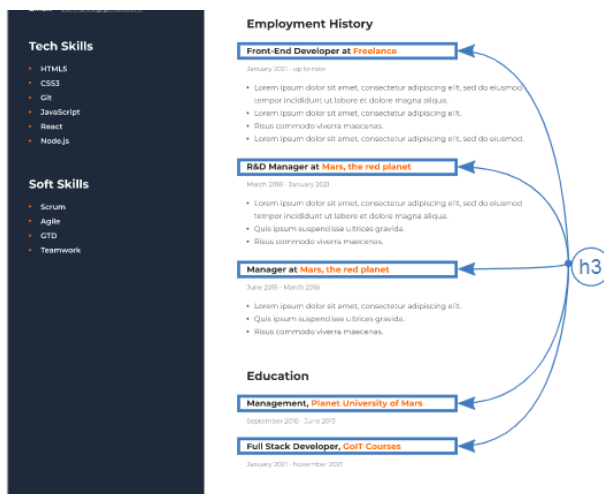
ЗАВДАННЯ 6

Додайте розмітку для посади (заголовок третього рівня) і періоду роботи (абзац). Не видаляйте існуючі теги.

Критерії прийому завдання

- У коді повинні бути теги завдань 1–5.
- Код повинен мати заголовок третього рівня <h3>.
- Тег <h3> повинен мати текст **Front-end developer at Freelance**.
- Код повинен мати рівно один параграф <p>.
- Текст параграфа не повинен бути порожнім.
- У коді не повинно бути інших тегів, крім <h2>, <h3>, <p>.

Макет 6



Тег ul та li

Теорія

Тег створює маркований (невпорядкований) список, кожен елемент якого починається з невеликого символу – маркера.

Наприклад, так виглядає список країн, що є популярними серед туристів.

Код
<pre> <h1>Країни</h1> Туніс Туреччина Греція Єгипет </pre>
Результат відображення у браузері
Країни • Туніс • Туреччина • Греція • Єгипет



ЗАВДАННЯ 7

Додайте розмітку списку обов'язків, використовуючи маркований список `<ul>`. Не видаляйте існуючі теги.

Критерії прийому завдання

- У коді повинні бути теги завдань 1-6.
- Код повинен мати рівно один маркований список `<ul>`.
- У маркованому списку `<ul>` повинно бути рівно три теги `<li>`.
- Всі теги `<li>` повинні мати текст.
- У коді не повинно бути інших тегів, крім `<h2>`, `<h3>`, `<p>`, `<ul>`, `<li>`.

Maker 7



John Doe

Front-End Developer

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Quis ipsum suspendisse ultrices gravida. Risus commodo viverra maecenas accumsan lacus vel facilisis.

Projects

1. [barbershop.git.global](#)
2. [cryptohub.git.global](#)
3. [lidlake.git.global](#)

Employment History

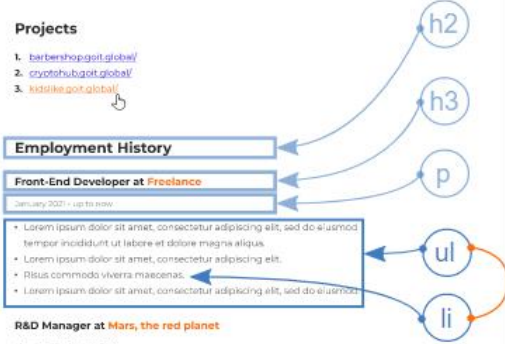
Front-End Developer at **Freelance**

January 2021 - up to now

- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Risus commodo viverra maecenas.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod

R&D Manager at **Mars, the red planet**

March 2018 - January 2021



```
graph TD; h2((h2)) --> Projects; h3((h3)) --> Projects; p((p)) --> Employment[Employment History]; ul((ul)) --> Freelance[Front-End Developer at Freelance]; li((li)) --> Freelance;
```

ЛАБОРАТОРНА РОБОТА 2

Мнемоніки

У макеті є символ **&**, для відображення якого використовують мнемоніки.

Мнемоніки – деякі символи неможливо набрати в **html** для відображення, тому що вони належать до синтаксису тега. Наприклад, знаки **>**, **<** і **/**. Інші символи часто використовуються, але не належать до літер алфавіту (нерозривний пробіл, математичні символи і багато інших). Проте всі вони можуть бути записані у вигляді латинських літер і відображатися у потрібному вигляді. Для цього використовують запис, який починається з **&** і закінчується **;**. Такі сполучення символів називають мнемоніками.

Код 1

```
<ul>
<li>&amp; амперсанд</li>
<li>&lt;&nbsp;&gt; менше, неразрывный пробел,
большее</li>
<li>&copy; знак охраны авторского права</li>
<li>&plusmn; &frac12; ... математические символы</li>
<li>&uarr; &rarr; &darr; &larr;</li>
<li>&#8226; небольшой черный круг</li>
<li>&rdquo; &ldquo; кавычки</li>
</ul>
```

Результат відображення у браузері

- & амперсанд
- < > менше, нерозривний пробіл, більше
- © знак охорони авторського права
- $\pm \frac{1}{2}$... математичні символи
- $\uparrow \rightarrow \downarrow \leftarrow$
- • невеличкий чорний круг
- ” “ лапки

Виходить, що ми повинні використовувати мнемоніку для символу амперсанд.

У макеті можуть бути схожі групи елементів. Для таких груп ми застосовуємо однакову HTML розмітку.



ЗАВДАННЯ 8

Додайте розмітку ще двох місць роботи з макета по аналогії з уже написаним прикладом. Не видаляйте існуючі теги.

Критерії прийому завдання

- Код повинен мати рівно три маркованих списки `<ul>`.
- У кожному маркованому списку `<ul>` повинно бути рівно три теги `<li>`.
- Усі теги `<li>` повинні мати непустий текст.
- Перед кожним списком повинен бути не порожній параграф `<p>`.

- Перед кожним параграфом повинен бути не порожній заголовок третього рівня `<h3>`. Один або два із заголовків `<h3>` повинні мати мнемоніки. Наприклад: `&` амперсанд, `"` " лапки або інші.
- У коді не повинно бути інших тегів, крім `<h2>`, `<h3>`, `<p>`, `<ul>`, `<li>`.

Текст макета збільшений для можливості читання:

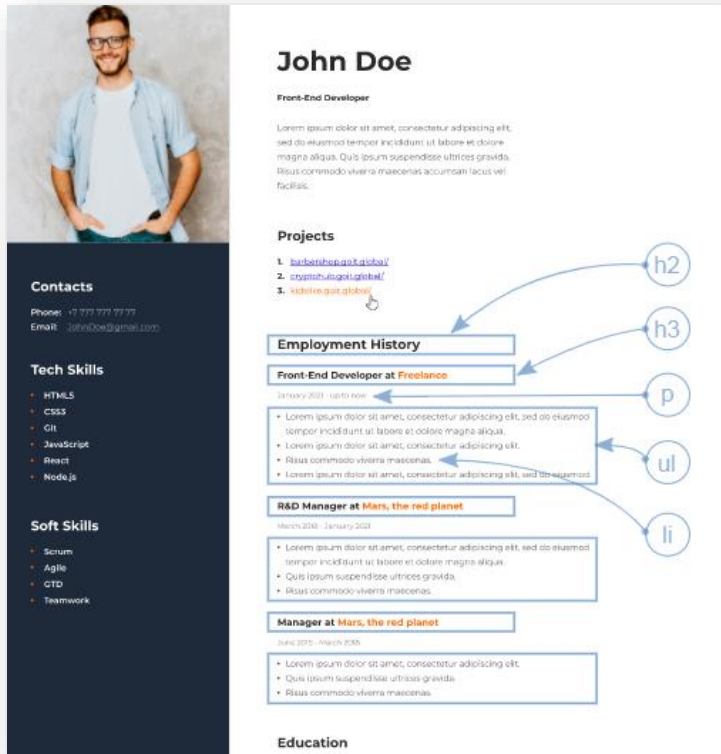
Front-end developer at Freelance

January 2021 - up to now

- Lorem ipsum dolor sit amet...
- Risus commodo viverra maecenas.
- Lorem ipsum dolor sit amet...

R&D Manager at Mars, the red planet March 2018 - January 2021 Lorem ipsum dolor sit amet... Quis ipsum suspendisse ultrices gravida. Risus commodo viverra maecenas. Manager at Mars, the red planet June 2015 - March 2018 Lorem ipsum dolor sit amet... Quis ipsum suspendisse ultrices gravida. Risus commodo viverra maecenas.

Макет 8



Схожість за виглядом і семантиці. Семантичні теги

Теорія

Якщо в макеті ми можемо виділити візуально групи елементів, які повторюються, зі схожою семантикою, то до цих груп застосовуємо однакову **html** розмітку документа.



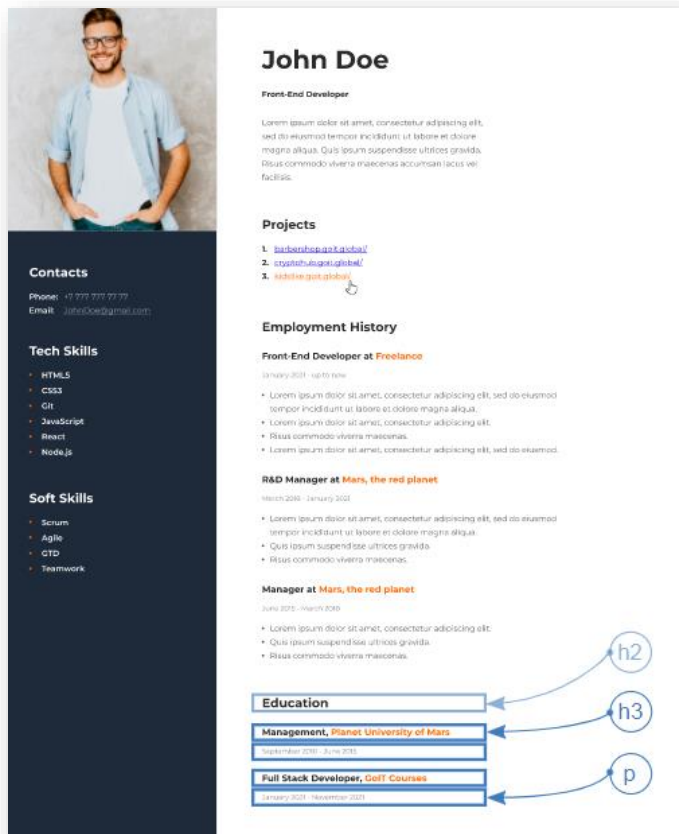
ЗАВДАННЯ 9

Напишіть розмітку блока освіти по аналогії з тим, як Ви робили розмітку місць роботи. Не видаляйте існуючі теги.

Критерії прийому завдання

- Код повинен мати один заголовок другого рівня **<h2>** з текстом **Education**.
- Код повинен мати рівно два заголовка третього рівня **<h3>** з текстом (назви закладів, де Ви навчалися – школа, ліцей, виш, курси...).
- Після кожного заголовка **<h3>** повинен бути параграф **<p>** з не порожнім текстом (дати початку і закінчення терміну навчання).
- В коді не повинно бути інших тегів, крім **<h2>**, **<h3>**, **<p>**.

Макет 9



Текст макета збільшений для можливості читання:

Education

Management, Planet University of Mars September 2010 - June 2015 Full Stack Developer, GoIT Courses January 2021 - November 2021

Виділення логічних розділів

Теорія

Існують спеціальні теги для розмітки логічних розділів. Розмічаючи сторінку такими тегами, ми робимо її більш зрозумілою для комп'ютера.

Тег `<section>` використовується для групування зв'язаних елементів.

Секція списку товарів, блок особистої інформації у профілі користувача, розділ контактної інформації – це все можна представити окремою секцією.



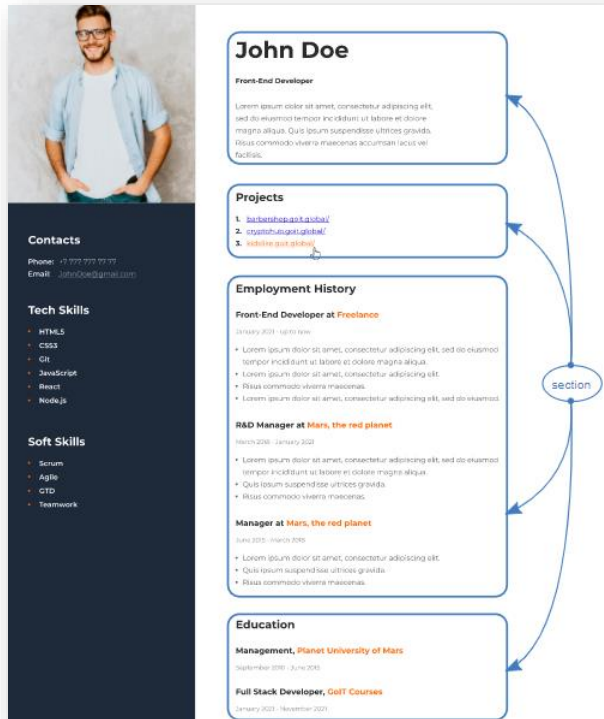
ЗАВДАННЯ 10

Оберніть логічні блоки (секції): біографія, проекти, досвід роботи і освіта тегами `<section>`, як у макеті.

Критерії прийому завдання

- Код повинен мати 4 секції `<section>`.
- Всередині кожної секції повинні бути відповідні частини коду, як на макеті.

Макет 10



Зображення. Тег `img`

Теорія

Тег `<img>` використовується для додавання картинок на вебсторінку. Це одиночний тег, йому не потрібна закриваюча частина. Достатньо написати просто ``.

Щоб картинка відобразилася, у значенні атрибуту `src` картинки необхідно указати маршрут до зображення.

Маршрут до зображення може бути **абсолютним**. Він однозначно вказує, де знаходиться зображення. Наприклад:

<https://images.pexels.com/photos/583842/pexels-photo-583842.jpeg>

Або ж маршрут може бути **відносним**. Тоді картинка повинна бути на нашому сервері у відповідному місці **відносно** поточної сторінки. Саме тому він називається **відносним**.

Графічні формати

Картинки бувають у різних форматах. Найбільш популярні – це **JPEG** і **PNG**.

- **JPEG** застосовується для великих файлів, які не вимагають прозорого фону або анімації. Цей формат ідеальний для фотографій, тому що вони можуть мати мільйони різних кольорів.
- **PNG**, на відміну від JPEG, має додатковий параметр для опису прозорості – альфа-канал. Цей формат підходящий для зображень складної форми, яка не вкладається у прямокутник.

Ми будемо використовувати **JPEG** формат.

Ось приклад застосування тега `<img>` (зверніть увагу, що відсутня закриваюча частина тега):

```

```



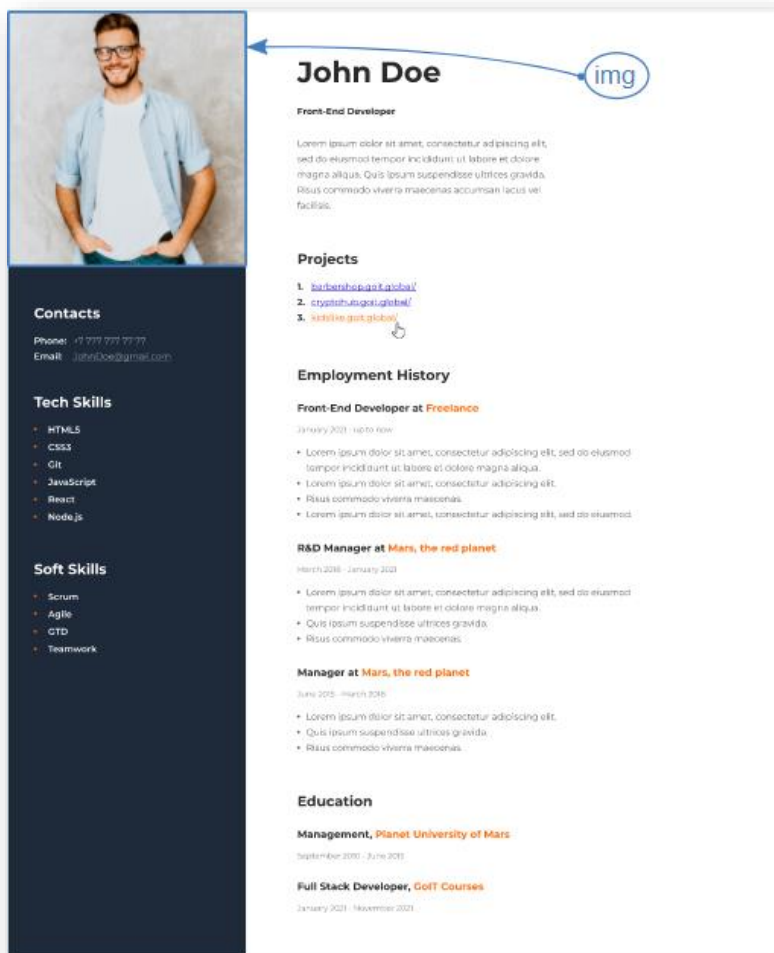
ЗАВДАННЯ 11

Використовуючи тег `<img>`, створіть розмітку зображення з наступними значеннями атрибутів:

атрибут	значення
src	photo.jpg
alt	My photo
width	370

Критерії прийому завдання

- Код повинен мати рівно один тег ****.
- У тегу **** повинні бути атрибути **src**, **width** і **alt**.
- Атрибут **src** повинен мати текст **photo.jpg**.
- Атрибут **width** повинен мати число **370**.
- Атрибут **alt** повинен мати текст **My photo**.
- У коді не повинно бути інших тегів.



Ще раз про схожість по виду і семантиці

Теорія

Якщо у макеті ми можемо виділити елементи зі схожою семантикою, але такі, що відрізняються за зовнішнім виглядом, то до цих елементів можна застосовувати однакову **html** розмітку.

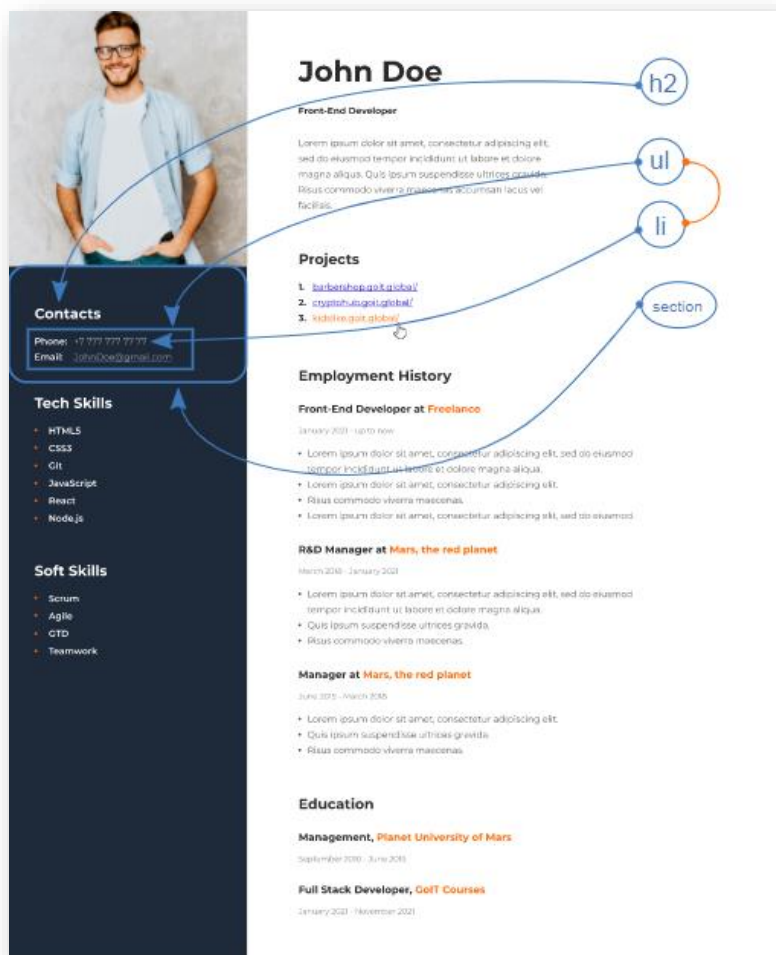


ЗАВДАННЯ 12

Додайте розмітку секції контактів з заголовком і маркованим списком з двох елементів: телефон і мейл.

Критерії прийому завдання

- Код повинен мати рівно один тег `<section>`.
- У середині тега `<section>` повинен бути рівно один заголовок `<h2>` з текстом `Contacts`.
- У середині тега `<section>` повинен бути рівно один маркований список `<ul>`.
- У середині списку повинно бути два не порожніх теги `<li>` з номером телефона та e-mail.
- У коді не повинно бути інших тегів.



Значення атрибута href

Теорія

У посилань існує можливість не тільки переходити на інші сторінки, але і телефонувати, відправляти повідомлення або дзвонити по скайпу.

Наприклад:

```
<!-- Посилання на телефонний номер -->  
<a href="tel:+14251234563">  
+1 (425) 123-45-63  
</a>  
  
<!-- Посилання на адресу електронної пошти -->  
<a href="mailto:info@goit.ua">  
info@goit.ua  
</a>  
  
+1 (425) 123-45-63 info@goit.ua
```



ЗАВДАННЯ 13

У середині елементів списку оберніть телефонний номер та **e-mail** у посилання. Використовуйте правильні значення для атрибутів **href**. Не видаляйте існуючі теги.

Критерії прийому завдання

- У кожному тегу **** повинне бути посилання **<a>** з атрибутом **href**.
- Текст кожного посилання не повинен бути порожнім.

- ## Макет 13



Розмітка блоків Tech Skills i Soft Skills



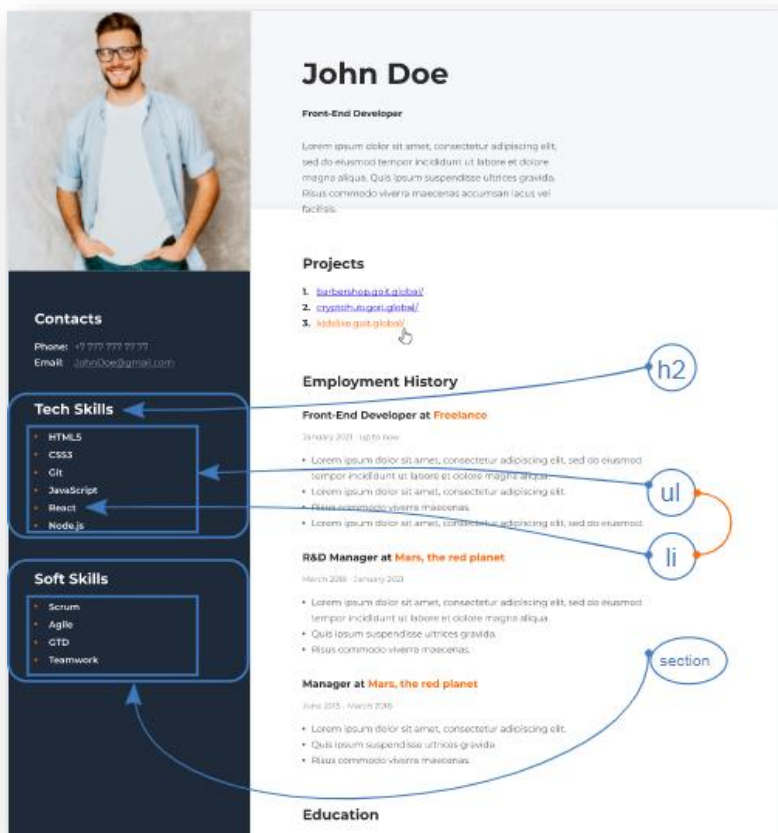
ЗАВДАННЯ 14

Додайте розмітку секцій **Tech Skills** і **Soft Skills**. Не видаляйте існуючий HTML код.

Критерії прийому завдання

- Код повинен мати рівно два теги `<section>`.
- У середині першої секції заголовок `<h2>` повинен мати текст **Tech skills**.
- У середині першої секції у списку `<ul>` повинно бути рівно 6 не порожніх тегів `<li>`.
- У середині другої секції заголовок `<h2>` повинен мати текст **Soft skills**.
- У середині другої секції у списку `<ul>` повинно бути рівно 4 не порожніх тегів `<li>`.
- У коді не повинно бути інших тегів.

Maker 14



Приблизний текст макета:

Tech skills

- HTML
- CSS
- Python
- JavaScript
- Swift
- C++

Soft skills

- Attentive
- Initiative
- Erudite
- Creative

Завершення розмітки всієї сторінки

Теорія

Тег `<div>` – універсальний контейнер без семантичного значення. Використовується як блок-обертка для наступної стилізації (застосування CSS).

У подальшому ми зможемо розташувати результат нашого коду як в макеті. Для цього нам знадобиться розташувати кожен частину зі світлим і темним дизайном всередині тегів-контейнерів. А потім обидва ці контейнери розташувати всередині ще одного загального контейнера.

Такий спосіб називають **обгорткою** і він застосовується при верстці. Тег `<aside>` використовують на сторінці для розташування допоміжної інформації. Зазвичай вміст `<aside>` складається з посилань, контактів і т. ін.

Коментар – фрагмент коду, який не обробляється браузером. Він призначений допомагати у розробці додатку. Коментар може повідомити **навіщо** написаний той чи інший фрагмент коду або відмітити **місце вставки** ще не написаного коду.



ЗАВДАННЯ 15

Розташуйте всередині тега `<div>` з класом `container` два сусідніх елементи-контейнери з класами `left-side` і `right-side`.

Критерії прийому завдання

- У тегу `<div class="container">` повинні бути теги `<aside>` і `<div>`.
- Тег `<aside>` повинен мати атрибут `class` зі значенням `left-side`.
- Вкладений тег `<div>` повинен мати атрибут `class` зі значенням `right-side`.
- Всередині кожного тега `<div>` повинен бути відповідний HTML-коментар. Наприклад:

`<!-- Right-side-->` та `<!-- Left-side-->`.

- У коді не повинно бути інших тегів.



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТА КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова

Кафедра інформаційних технологій
та фізико-математичних дисциплін

Звіт до лабораторної роботи № 1

Тема: **«HTML. Основні теги. Заголовки. Абзаци.
Посилання та атрибути»**

Варіант № ____

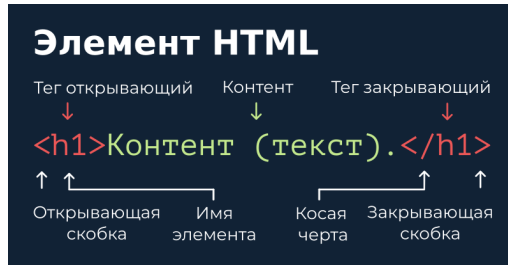
Виконав
студент групи _____

Херсон – 2026

Теоретичні відомості

Теги – це спосіб підказати браузеру, як виділити і відобразити контент.

Синтаксис тега



**Записати перелік тегів, які використовуються
в даній лабораторній роботі:**

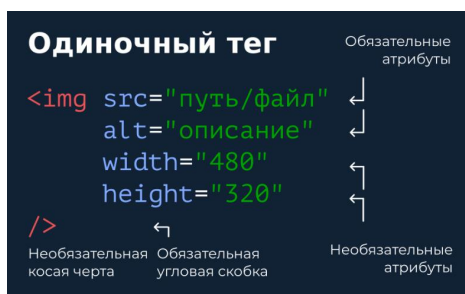
Опис тега	Синтаксис тега
Основний заголовок Тег h1	Заголовок всієї сторінки
Параграф Тег p	Зміст параграфа
Теги для списків Теги ol та li : Теги ul та li :	 текст текст
Теги для посилань Тег a	 https://google.com
Теги для атрибутів href i class:	 link

Посилання та атрибути



Атрибути – це додаткові настройки тегів, за допомогою яких можна змінювати властивості і поведінку елемента. У кожного атрибута є **ім'я** та **значення**.

Атрибути записуються всередині відкриваючого тега, а їх значення розташовуються всередині лапок. Атрибутів може бути кілька або не бути зовсім.



У деяких тегів є обов'язкові атрибути.

```
<!-- обов'язковий атрибут href -->
<!-- необов'язковий атрибут class -->
<a href="https://google.com" class="link"> link </a>
```

Посилання

Тег **<a>** призначений для створення посилань. При кліку на посилання ми переходимо на інші сторінки, скачуємо файли. Браузер підкреслює посилання та виділяє їх синім кольором.

Адреса посилання задається у *обов'язковому атрибуті* **href="адреса"**. Адреса – це URL, який вказує на іншу вебсторінку або файл.

Наприклад, от посилання на Google:

Результат відображення у браузері

<https://google.com>



ЗАВДАННЯ 1

Напишіть заголовок **h1** зі своїм ім'ям та прізвищем.

Критерії прийому завдання

- Код повинен мати тег-заголовок з ім'ям **h1**.
- Текст заголовка не повинен бути порожнім.
- Інших тегів у тексті не повинно бути.

ЗАВДАННЯ 1

Готовий код після виконання завдання 1

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<title></title>
</head>
<body>
<h1>John Doe</h1>
</body>
</html>
```

Результат відображення у браузері
John Doe
Готовий код після виконання завдання 1
Результат відображення у браузері

Параграф



ЗАВДАННЯ 2

Додайте два теги абзаців. Один для назви професії, а другий – для біографії. Не видаляйте існуючий тег **h1**.

Критерії прийому завдання

- У коді повинен бути тег завдання 1.
- Текст першого параграфа – **Front-end developer**.
- Текст другого параграфа не повинен бути порожнім.
- Інших тегів у тексті не повинно бути.

ЗАВДАННЯ 2

Готовий код після виконання завдання 2

```
<h1>John Doe</h1>
<p>Front-end developer</p>
<p>I was born in New York in 1995. My parents are employees.
In 2014 he graduated from the Physics and Mathematics Lyceum.
In 2015 he entered Columbia University.
</p>
```

Результат відображення у браузері
John Doe Front-end developer I was born in New York in 1995. My parents are employees. In 2014 he graduated from the Physics and Mathematics Lyceum. In 2015 he entered Columbia University.
Готовий код після виконання завдання 2
Результат відображення у браузері

Заголовки



ЗАВДАННЯ 3

Напишіть заголовок **h2** з текстом **Projects**.

Критерії прийому завдання

- Код повинен починатися з тегів завдань 1–2.
- Код повинен мати заголовок другого рівня **h2**.
- Текст заголовка другого рівня повинен дорівнювати **Projects**.
- Інших тегів у тексті не повинно бути.

ЗАВДАННЯ 3

Готовий код після виконання завдання 3
<pre><body> <h1>John Doe</h1> <p>Front-end developer</p> <p> I was born in New York in 1995. My parents are employees. In 2014 he graduated from the Physics and Mathematics Lyceum. In 2015 he entered Columbia University. </p> <h2>Projects</h2> </body></pre>

Результат відображення у браузері
John Doe Front-end developer I was born in New York in 1995. My parents are employees. In 2014 he graduated from the Physics and Mathematics. In 2015 he entered Columbia University. Projects
Готовий код після виконання завдання 3
Результат відображення у браузері

Нумерований список. Теги `ol` та `li`



ЗАВДАННЯ 4

Використайте тег `<ol>` для розмітки списку проєктів. Не видаляйте існуючі теги.

Критерії прийому завдання

- Код повинен починатися з тегів завдань 1–3.
- Код повинен мати нумерований список `<ol>`.
- Всередині списку `<ol>` повинно бути три елементи `<li>`.
- Значення кожного елемента `<li>` не повинно бути порожнім.
- У коді не повинно бути інших тегів, крім `<ol>`, `<li>`, `<h2>`.

ЗАВДАННЯ 4

Готовий код після виконання завдання 4
Результат відображення у браузері

Посилання та атрибути



ЗАВДАННЯ 5

Обгорніть текст усередині кожного `<li>` посиланням. Значення атрибуту `href` і текст посилання мають бути однаковими. Не видаляйте існуючі теги.

Критерії прийому завдання

- Код повинен починатися з тегів завдань 1–4.
- Всередині кожного елемента `<li>` повинен бути тег `<a>`.
- Текст кожного тега `<a>` повинен бути рівним атрибуту `href` посилання.
- Атрибут кожного посилання `href` повинен бути URL (починатися з `https://`).

ЗАВДАННЯ 5

Готовий код після виконання завдання 5
Результат відображення у браузері

Заголовок h3



ЗАВДАННЯ 6

Додайте розмітку для посади (заголовок третього рівня) і періоду роботи (абзац). Не видаляйте існуючі теги.

Критерії прийому завдання

- Код повинен починатися з тегів завдань 1–5.
- Код повинен мати заголовок третього рівня `<h3>`.
- Тег `<h3>` повинен мати текст **Front-end developer at Freelance**.
- Код повинен мати рівно один параграф `<p>`.
- Текст параграфа не повинен бути порожнім.
- У коді не повинно бути інших тегів, крім `<h2>`, `<h3>`, `<p>`.

ЗАВДАННЯ 6

Готовий код після виконання завдання 6
Результат відображення у браузері

Маркований список.

Теги ul та li



ЗАВДАННЯ 7

Додайте розмітку списку обов'язків, використовуючи маркований список `<ul>`. Не видаляйте існуючі теги.

Критерії прийому завдання

- Код повинен починатися з тегів завдань 1-6.
- Код повинен мати рівно один маркований список `<ul>`.
- У маркованому списку `<ul>` повинно бути рівно три теги `<li>`.
- Всі теги `<li>` повинні мати текст.
- У коді не повинно бути інших тегів, крім `<h2>`, `<h3>`, `<p>`, `<ul>`, `<li>`.

ЗАВДАННЯ 7

Готовий код після виконання завдання 7
Результат відображення у браузері

Довідка А. 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 веброзробника.

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 аббревіатурою 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 тега використовуються атрибути, що вказують текстові чи числові значення.

<table width="50%" align="center">...</table>

У 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 **
** – перенесення на новий рядок, **<hr>** – розділова лінія, **** – вставляння зображення. Властивості для одиночного тегу вказуються всередині через атрибути, наприклад:

```

```

У наведеному прикладі в одиночному тезі **** прописано атрибути, які вказують на місце розташування і назву графічного файлу, ширину зображення в 200 пікселів і наявність рамки навколо зображення шириною в 1 піксель.

Певний час вимагалось наприкінці одиночного тега вказувати слеш (**
, **), натеper 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 елемента: рядка, блоку, таблиці, заголовка.

Наприклад, для того, щоб певний текст відобразився в браузері як рядок, слід застосувати теги, що позначають елемент-параграф (рис. 2.1):

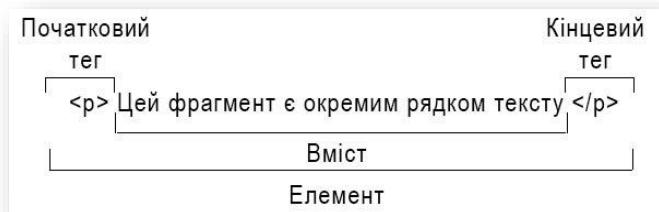


Рис. 2.1. Склад HTML-елемента

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 елемента. Закінчення в парних тегах є обов'язковими, інакше браузер може невірно відобразити вміст.
- **Вміст:** у 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 окремим рядком 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 вебсторінки. 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 **doctype**, яким має слідувати HTML-сторінка, також вказівка на автоматичну перевірку помилок та 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 тегу **<html>**, утворюють дерево документа, так звану об'єктну модель документа, **DOM (document object model)**. Елемент **<html>** є кореневим елементом, **body** і **head** – структурними елементами.

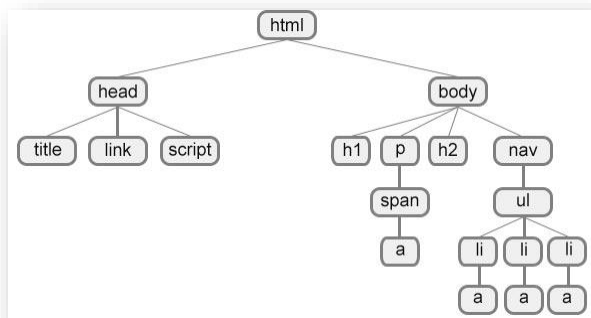


Рис. 2.2. Найпростіша структура вебсторінки

У взаємодії елементів вебсторінки 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. 2.2 **<html>** є батьківським тільки для **<head>** і **<body>**. Елемент **<body>** є батьківським для всіх елементів, що містяться в ньому: **<h1>**, **<p>**, ****, **<nav>** і т. д. Тег **<p>** є батьківським тільки для ****.

- **Дочірній елемент** – елемент, що розташований всередині іншого елемента. На рис. 2.2 елементи **<h1>**, **<h2>**, **<p>**, **<nav>** є дочірніми по відношенню до **<body>**.
- **Сестринський елемент** – елемент, що має загальний батьківський елемент з даним, так звані елементи одного рівня. На рис. 2.2 **<head>** і **<body>** – елементи одного рівня, так само як і елементи **<h1>**, **<h2>**, **<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 веббраузер 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">**.
- **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 у зовнішньому файлі, то він під'єднується за допомогою атрибутів елемента, наприклад: **<script src=" scripts.js"></script>**

Елемент <body>

Усередині тега **<body>** міститься інформація, яка відображається в браузері. В таблиці <http://html5book.ru/html-tags/> наведено повний перелік тегів, що підтримуються HTML4 та 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 HTML атрибути, які повідомляють браузеру, яким чином має відображатися той чи інший елемент сторінки та особливості відображення даного елемента. Значення атрибута завжди заключають у 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 використати для будь-якого HTML-елемента, та **власні**, які притаманні лише цьому тегу.

Універсальні атрибути

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 елемента, 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.
-

Теги для блокових елементів

Загальні характеристики блокових тегів

- На початку і в кінці блокових тегів, автоматично ставиться перенесення рядка.
- У блокових тегах можна керувати шириною і висотою: **width**, **height**.

- У блокових тегах можна керувати зовнішніми і внутрішніми відступами.
- За замовченням блокові теги займають всю ширину батьківського елемента.
- Об'єктами, що знаходяться всередині блокового тега, можна керувати за допомогою горизонтального та вертикального вирівнювання.
- Всередині блокових тегів можна розміщувати інші блокові теги, а також рядкові теги.

Типові блокові теги

- **<div>** – універсальний блоковий елемент.
- **<article>**, **<aside>**, **<header>**, **<footer>** – семантичні блоки для позначення структурних ділянок сторінки.
- **<p>**, **<h1>**-**<h6>**, **<blockquote>** – теги для HTML-тексту.
- **<hr>** – горизонтальна лінія.
- ****, **** – встановлює нумерований (маркований) список.
- **<table>** – створює таблицю.
- **<form>** – встановлює форму на вебсторінці.

Теги для рядкових елементів

Загальні характеристики рядкових тегів

- На відміну від блокових, рядкові теги слідують один за одним і НЕ переносяться на новий рядок.
- У рядкових тегах не можна керувати шириною і висотою: **width**, **height**.

- В рядкових тегах не можна встановлювати зовнішні відступи (**margin**), лише внутрішні (**padding**).
- Рядкові теги НЕ займають всю ширину батьківського елемента.
- Оскільки рядкові теги НЕ займають усю ширину батьківського елемента, то елементами, що знаходяться всередині рядкових тегів, не можна керувати за допомогою горизонтального чи вертикального вирівнювання.
- Усередині рядкових тегів можна розміщувати інші рядкові теги, але блокові розміщувати не можна.

Типові рядкові теги

<a> – є одним з важливих елементів HTML і призначений для створення посилань.

**** – універсальний рядковий елемент.

**
** – розірвання рядка.

**** – уставляння зображення.

HTML текст

Теги 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.

Правильно відформатований текст надає для пошукових систем відомості, які фрагменти тексту мають важливий зміст, за якими з них слід ранжувати вебсторінку в пошуковій видачі.

Тег абзаца <p>

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

Теги заголовків <h1>-<h6>

Заголовки є важливим елементом вебсторінки, 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 теги, 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 допускається вкладення інших тегів в теги <h1> ... <h6>.

Теги для форматування тексту

- **** – задає грубіше накреслення шрифту, вказуючи браузеру на важливість тексту.
- **** – відображає текст нахиленим шрифтом (курсивом), надаючи для тексту певної значимості.

- **<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.
- **
** – переносить текст на наступний рядок, створюючи розрив рядка.
- **<hr>** – використовується для поділу контенту на вебсторінці. Відображається у вигляді горизонтальної лінії.

HTML-посилання

HTML посилання складають основу Веб, вони пов'язують між собою різні вебсторінки.

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

<p>Замовити книжки у

магазини</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 показник ресурсів (URL) визначає місцезнаходження ресурсу. В загальному вигляді має наступний формат:

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

Найбільш поширені методи доступу: **http, mailto, file, ftp.**

- **Метод http** надає доступ до вебсторінки за протоколом HTTP, наприклад:
http://www.site.ua/
- **Метод mailto** запускає сеанс поштової зв'язку із зазначеним адресатом і хостом, наприклад:
mailto:admin@gmail.com
- **Метод file** забезпечує читання файлу з локального диска, наприклад:
file://gallery/pictures/summer.html
- **Метод ftp** здійснює запит до FTP-сервера на отримання файлу, наприклад:
ftp://pgu/directory/library

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).

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,

`http://www.site.ua/folder/file.html`

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

`http://www.site.ru/file.html`

У разі відсутності назви файлу буде завантажуватися вебсторінка, яка задана за замовченням у налаштуваннях вебсервера:

`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 вебсторінки, 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 вебсторінка знаходиться в тій же папці, що і вебсторінка, 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/folder2/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>
```

4. 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 вебсторінки інших сайтів є зовнішніми, для них завжди вказують Абсолютний URL, наприклад:

```
<a href="http://english-films.com" >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 тега **<a>** з різницею в тому, що атрибут **href** містить назву покажчика (якір), 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:

```
  
<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 **http://**, 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">
```

Гіпертекстові посилання

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

```
<a href="styling.html"> Гіпертекстові посилання  
</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>
    
    <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);
- _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 вебсторінки візуально привабливими, зображення допомагають краще передати суть і зміст вебдокумента. Також, за допомогою HTML-тегів можна робити карти-зображення з активними областями.

Уставляння зображень в HTML-документ

Тег

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

Тег **** має обов'язковий атрибут **src**, значенням якого є адреса втіленого зображення, і рекомендований атрибут **alt**, наприклад:

```

```

Для тега **** 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:

```

```

Або через відносний шлях від документа або кореневого каталогу сайту:

```
 – відносний шлях  
від документа,
```

```
 – 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 рисунка.

Формати графічних файлів

- **.JPEG (.JPG)**. Зображення **JPEG** є ідеальними для фотографій, вони містять мільйони різних кольорів.
- **.GIF**. GIF-файли дозволяють встановити один з кольорів 100 % прозорим, завдяки чому фон вебсторінки 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.
- **.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.
- **.APNG**. Формат зображення, що засновано на форматі **PNG**. Дозволяє зберігати анімацію, а також підтримує прозорість.
- **.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.

- **.ICO**. Формат зберігання фавіконок, що відображаються на закладці в браузері або поруч з адресою сайту у сторінці видачі результатів пошукової системи. Один **ICO**-файл містить одну або кілька іконок, розмір кожної з яких задається окремо. Найбільш вживані розміри фавіконок: 16, 32, 48, 57, 72, 114, 256 пікселів.

HTML-списки

HTML списки представляють набір згрупованих абзаців тексту, помічених значками (маркований список) або цифрами (нумерований список). Елементи списку додаються за допомогою тега **** (**List Item** – елемент списку). Для тега **** доступні різні атрибути, що дозволяють змінити нумерацію або маркування обраного елемента списку.

Маркований список

Маркований список представляє невпорядкований перелік елементів (**Unordered List**). Створюється за допомогою парного тега ****. На початку кожного елемента встановлюється маркер, наприклад, зафарбований кружечок.

<code></code>	Відображення в браузері
<code>Microsoft</code>	
<code>Google</code>	
<code>Apple</code>	• Microsoft
<code>IBM</code>	• Google
<code></code>	• Apple
	• IBM

Нумерований список

Нумерований список поміщають усередину пари тегів `<ol>`. Кожен пункт списку потрібно помістити в тег `<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>
```

Відображення

в браузері

1. Microsoft

2. Google

3. Apple

4. IBM

Список визначень

Списки визначень описуються за допомогою тега **<dl>**, який містить пари термін-визначення. Для додавання терміна застосовується блоковий тег **<dt>**, а для уставляння визначення – блочний парний тег **<dd>**.

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

<dl>	
<dt>Національний	Відображення
університет «Львівська	в браузері
політехніка»:</dt>	
<dd>ІКНІ</dd>	Національний
<dt>Кафедри</dt>	університет
<dd>САПР</dd>	«Львівська
<dd>ІСМ</dd>	політехніка»:
<dd>АСУ</dd>	ІКНІ
</dl>	Кафедри
	САПР
	ІСМ
	АСУ

Вкладений список

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

<pre> Пункт 1. Пункт 2. Підпункт 2.1. Підпункт 2.2. Підпункт 2.2.1. Підпункт 2.2.2. Підпункт 2.3. Пункт 3. </pre>	Відображення в браузері • Пункт 1. • Пункт 2. ○ Підпункт 2.1. ○ Підпункт 2.2. ▪ Підпункт 2.2.1. ▪ Підпункт 2.2.2. ○ Підпункт 2.3. • Пункт 3.
--	--

Рекомендована література

1. Матвієнко О. В., Бородіна І. Л. Internet-технології: проєктування Web-сторінки : навч. посіб. Київ : Центр навчальної літератури, 2004. 154 с.
2. Глушков С. В., Жакін І. А., Хачиров Т. С. Програмування Web-сторінок. Харків : Фоліо, 2005. 390 с.
3. Пономаренко В. С., Мінухін С. В., Торохтій І. О. Основи технологій Internet : навч. посіб. Харків : Видання ХДЕУ, 2001. 256 с.
4. Мельник Р. А. Програмування ВЕБ-застосувань (фронт енд та бек-енд). Львів : Львівська політехніка, 2018. 248 с.
5. Myers M. A Smarter Way to Learn HTML & CSS: Learn it faster. Remember it longer. (Volume 2). CreateSpace Independent Publishing Platform, 2015. P. 258.
6. Вікіпедія. HTML: офіційний сайт. URL: <https://uk.wikipedia.org/wiki/HTML> (дата звернення 22.03.2026).
7. HTML.IN.UA : електронний ресурс. URL: <https://html.in.ua/html> (дата звернення 17.03.2026).
8. Пре-курс GO IT : електронний ресурс. URL: <https://gofrontend.com.ua/?token=oad5d17e94b9145c5360fe93de564b1875263d2ff5e05710b51cd36d7c7a9d9e&block=645hg4y7> (дата звернення 28.03.2026).
9. Довідник HTML та CSS : електронний ресурс. URL: <https://html-css.co.ua/dovidnuk-css-atrubytiv/> (дата звернення 17.02.2026).
10. Вікіпідручник. HTML : електронний ресурс. URL: <https://uk.wikibooks.org/wiki/HTML> (дата звернення 15.03.2026).

Навчальне видання

ЛИТВИНЕНКО Олена Іванівна

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт
з курсу «WEB-програмування. HTML»

Комп'ютерна правка й коректура *О. Є. Вакула*
Комп'ютерне верстання *Ю. С. Семенченко*

Формат 60×84/16. Ум. друк. арк. 4,65. Вид. № 7. Зам. № 2105-12.
Видавець і виготівник Національний університет кораблебудування
імені адмірала Макарова
просп. Героїв України, 9, м. Миколаїв, 54007
E-mail : publishing@nuos.edu.ua
Свідоцтво суб'єкта видавничої справи ДК № 6402 від 19.09.2018 р.