

1 ЛАБОРАТОРНА РОБОТА № 1

ЗАГАЛЬНА СТРУКТУРА HTML-ДОКУМЕНТА

1.1 Мета роботи

Ознайомитись із загальною структурою HTML-документа. Створити базову web-сторінку згідно з поточним стандартом мови HTML.

1.2 Короткі теоретичні відомості

1.2.1 Перед початком роботи. Середовище розробки.

На відміну від PHP, Java, C++ та інших мов, HTML (*HyperText Markup Language*) є не мовою програмування, а мовою розмітки тексту, або, точніше, гіпертексту.

Незважаючи на це, мова HTML має власний синтаксис, який декларується стандартами W3C HTML 4.01 (прийнятий 24 грудня 1999 р. Наразі вважається застарілим) та W3C HTML 5 (прийнятий 28 жовтня 2014 р.).

Фактично HTML документ є текстовим файлом з певною структурою тексту та розширенням .html або .htm, тобто створити його можна у будь-якому текстовому редакторі, навіть у стандартному Блокноті. Але для спрощення процесу розробки рекомендується використовувати програмне забезпечення з функціями підсвічування синтаксису, перевірки орфографії, автодоповнення введення тексту тощо. До таких програм відносяться як більш професійні Adobe Dreamweaver, Microsoft Visual Web Developer, PHP Storm, так і простіші безкоштовні програми з відкритим програмним кодом, наприклад Notepad++.

Щоб створити новий HTML документ у Notepad++ відкрийте програму та оберіть «Файл – Створити», або натисніть комбінацію клавіш Ctrl+N. Після цього ви побачите порожній документ без розширення. Оберіть «Файл – Зберегти як» та назвіть файл, наприклад, index.html, або оберіть тип документа з відповідного випадаючого списку (рис. 1.1).

Після збереження файла, згідно з налаштуваннями Notepad++ за замовчуванням, весь код, що буде додано до цього файла, буде підсвічено за стандартом HTML.

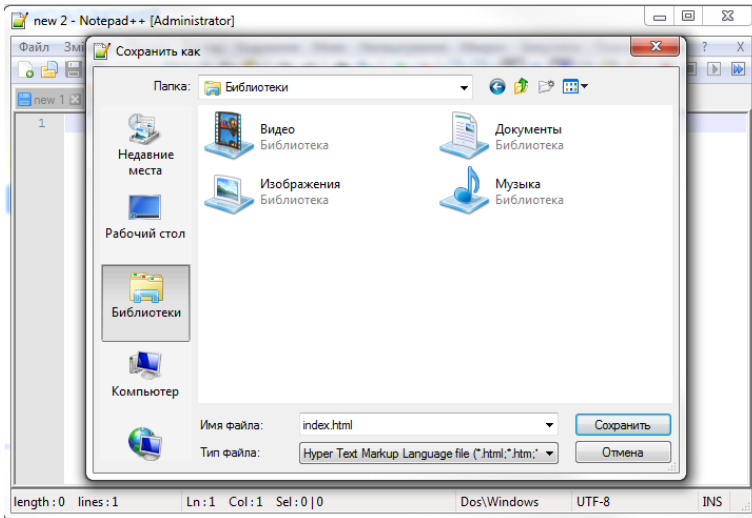


Рисунок 1.1 – Створення та збереження нового HTML-документа

Другим способом задати синтаксис документа є вибір відповідної мови у пункті меню «Мова» (рис. 1.2).

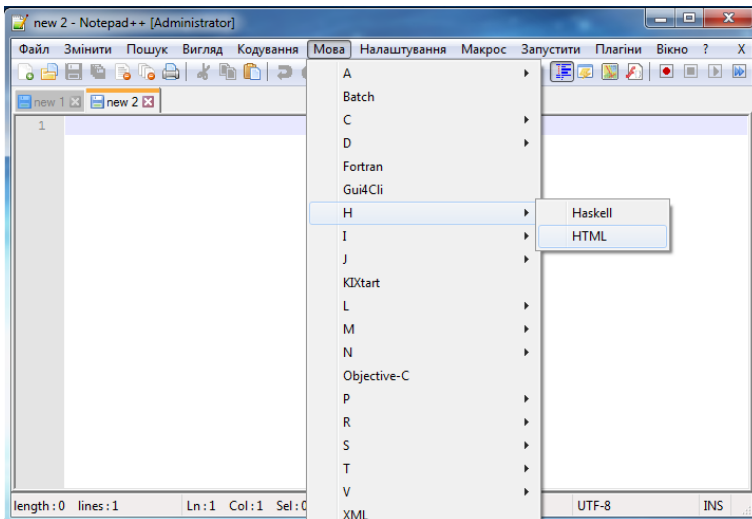


Рисунок 1.2 – Вибір мови документа

1.2.2 Елементи HTML. Теги

Елементи – це базові компоненти розмітки HTML. Кожен елемент має дві основні властивості: атрибути та зміст. Існують певні настанови щодо кожного атрибута та змісту елемента, які треба виконувати для того, щоб HTML-документ був визнаний валідним (міг бути коректно обробленим та відображеним браузером).

У елемента є початковий тег, який має вигляд `<tag>`, та кінцевий тег, який має вигляд `</tag>`. Атрибути елемента записуються в початковому тегу одразу після назви елемента, зміст елемента записується між його двома тегами. Наприклад:

```
<tag attribute="value">Зміст</tag >
```

Деякі елементи (br, hr, img) не містять змісту, тож не мають і кінцевого тега. В стандарті HTML 4.01 такі теги записуються як відкриваючий тег без кінцевого:

```
<br>
```

В стандарті HTML 5 такі теги позначають як самозакриті:

```
<br />
```

1.2.3 Базова структура документа

Окрім видимої користувачу інформації HTML документ включає певну частину службової інформації, яка підказує браузеру, як коректно відобразити даний документ.

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Заголовок документа</title>
  <meta http-equiv="content-type" content="text/HTML; charset=utf-8" />
</head>
<body>
  Зміст документа
</body>
</html>
```

HTML

Першим рядком документа зазвичай є `<!DOCTYPE>`, який вказує стандарт, згідно з яким браузеру слід інтерпретувати дану сторінку. Стандарти HTML 5 та HTML 4.01 мають деякі відмінності, тож для коректної роботи слід визначити, який з них використовується у даному документі.

Для HTML 5 `<!DOCTYPE>` має наступний запис:

```
<!DOCTYPE html>
```

А для строгого синтаксису HTML 4.01 використовують наступний формат:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/HTML4/strict.dtd">
```

Тег `<html>` визначає початок документа, а `</html>` зазвичай є останнім тегом документа.

Всередині тега `<html>` документ розділено на дві частини: заголовок документа, який огорнуто тегами `<head>` `</head>`, та тіло документа, яке визначається парою тегів `<body>` `</body>`.

Заголовок документа включає в себе службову інформацію, яка необхідна для коректної обробки сторінки браузером, пошуковими машинами тощо, і яка не показується користувачу у браузері, окрім змісту тега `<title>`. Текст, що розміщено всередині парного тега `<title>`, вважається заголовком сторінки і показується, наприклад, в заголовку вікна браузера, у видачі пошукової машини, тощо.

Весь текст, що розміщено всередині тега `<body>`, буде показано користувачу у вікні браузера. Це – основний зміст сторінки.

1.2.4 Meta-теги

Тег `<meta>` визначає групу тегів, що використовуються для збереження службової інформації для браузера або пошукової машини. Найважливішими його атрибутами є *name* та *content*. *Name* визначає, яку саме службову інформацію містить мета-тег, а *content* – зміст цієї інформації.

Нижче приведено декілька найбільш популярних мета-тегів, хоча стандарт передбачає значно більше можливих значень.

```
<meta name="description" content="опис сторінки" />  
<meta name="keywords" content="ключові, слова, через, кому" />  
<meta name="author" content="автор" />  
<meta name="robots" content="index, follow" />  
<meta http-equiv="content-type" content="text/HTML; charset=utf-8" />
```

Description – включає опис сторінки у вигляді зв'язного тексту.

Keywords – перелік ключових слів та словосполучень через кому, за якими бажано знаходити цю сторінку у пошукових системах.

Author – визначає ім'я автора сторінки.

Robots – дає вказівку пошуковим ботам, чи слід сканувати зміст сторінки (index/noindex) та чи переходити до інших сторінок, на які є посилання в даному документі (follow/nofollow).

Останній наведений тег є прикладом того, як вказати браузеру створити http заголовок сторінки. Даний приклад вказує, яке кодування використовувати для сторінки.

Якщо створена сторінка некоректно відображається у браузері (порожні клітинки або спеціальні символи замість літер), перевірте чи відповідає цей мета-тег кодуванню створеної сторінки. Notepad++ вказує кодування документа в правому нижньому кутку (ANSI або UTF-8). Щоб змінити кодування сторінки, скористайтесь пунктом меню «Кодування – Перетворити в UTF-8 без BOM».

1.2.5 Основні теги для оформлення тексту

1.2.5.1 Заголовки тексту

В HTML передбачено до 6 рівнів заголовків, які дозволяють відображати структуру документа. Для того щоб відобразити заголовок на web-сторінці, необхідно використати тег `<hx></hx>`, де *x* – ціле число від 1 до 6, що позначає номер рівня заголовка. Найвищим рівнем прийнято вважати 1. Пропускати рівні не рекомендується, тобто після `<h1>` не повинен бути тег `<h3>`, якщо між ними немає `<h2>`.

1.2.5.2 Абзаци (параграфи)

Для поділу тексту на змістовні фрагменти використовується тег `<p></p>`.

Браузери за замовчуванням ігнорують невидимі символи, окрім одинарного пробілу. Тобто табуляції та переноси, які буде додано в самому HTML коді буде пропущено та замінено на звичайний пробіл.

Елемент `<p>` повідомляє браузеру про те, що весь текст, вміщений між його відкриваючим і закриваючим тегами, – це єдиний абзац. Коли інтерпретатор браузера зустрічає в коді відкриваючий тег `<p>`, він вставляє порожній рядок, позначаючи тим самим початок нового абзацу. За замовчуванням кожен абзац має відступи, що візуально відділяють його від сусідніх елементів.

Якщо треба перенести частину тексту на новий рядок в межах одного абзацу, можна використати тег переносу на нову строку `
`. На відміну від тега `<p>` він не додає додатковий відступ до тексту.

1.2.5.3 Елементи фізичного форматування

Текст може бути візуально виділений напівжирним, курсивом або підкресленням. Ці елементи є абсолютними, а отже, будь-який браузер зобов'язаний відображати їх однаково. Основні елементи фізичного форматування наведені в таблиці 1.1.

Таблиця 1.1 – Елементи фізичного форматування

Елемент	Призначення
<code> </code>	Жирний
<code><i> </i></code>	<i>Курсив</i>
<code><u> </u></code>	<u>Підкреслений</u>
<code><sub> </sub></code>	Нижній <small>індекс</small>
<code><sup> </sup></code>	Верхній <small>індекс</small>

1.2.5.4 Елементи логічного форматування

Логічним є стиль, конкретні параметри якого задаються браузером. Браузер робить текст жирніше, виділяє його курсивом, розфарбовує в зелений колір або вимовляє голосніше (голосові браузери) залежно від можливостей конкретної програми. Будь-який елемент логічного стилю має відкриваючий і закриваючий тег, що формує контейнер для тексту, що виділяється. В таблиці 1.2 наведені елементи логічного форматування.

Зазвичай `` відповідає курсиву, а `` – жирному шрифту. Але це не завжди так. Іноді `` може означати підкреслення, а `` – виділення яскравістю. У голосових браузерях текст, позначений ``, вимовляється голосніше, ніж звичайний, а `` – ще голосніше. Основна ідея полягає в тому, що `` робить текст виділеним, а `` – посилено виділеним.

Таблиця 1.2 – Елементи логічного форматування

Елемент	Призначення
<code> </code>	Виділення
<code> </code>	Посилене виділення
<code><cite> </cite></code>	Цитата або посилання на зовнішнє джерело
<code><dfn> </dfn></code>	Вихідний код програми
<code><samp> </samp></code>	Приклад роботи програми, часто відображається так само, як і попередній
<code><kbd> </kbd></code>	Текст, що вводиться з клавіатури
<code><var> </var></code>	Змінна або значення

Інші стилі використовуються для деяких специфічних цілей, найчастіше пов'язаних з набором технічної або наукової літератури.

Може здатися, що логічні елементи є надлишковими. Насправді це не так, більше того, саме їм і віддають перевагу, тому що не всі браузери можуть відображати фізичні стилі, наприклад виділення жирним шрифтом. Якщо, наприклад, браузер, убудований у мобільний телефон, не вміє цього, то він просто проігнорує тег ``. Якщо ж використати логічні елементи, то телефон поставиться виділити це місце в тексті якимось по-іншому.

1.2.5.5 Елементи списків

Список – це структурований перелік подібних об'єктів. У списку в HTML завжди повинно бути два елементи, один із яких задає тип списку, а інший відповідає за один конкретний пункт. Цими пунктами можуть бути слова, речення й інші елементи HTML, наприклад зображення. У більшості списків використовується наступний формат:

```
<тип_списку>
  <li>Перший пункт</li>
  <li>Другий пункт</li>
  <li>Третій пункт</li>
</тип_списку>
```

Кожен елемент `` – це пункт списку, що за замовчуванням відображається з нового рядка. З чого починається цей рядок, залежить від того, маркований список або нумерований.

Нумерований/упорядкований список задається елементом-обгорткою ``, а маркований – елементом ``. Але пункти кожного з них позначаються за допомогою ``. При цьому якщо список упорядкований (``), то в початок кожного рядка вставляється його порядковий номер у списку (1,2,3...); якщо він неупорядкований (``), то вставляється позначка маркування (•).

1.2.5.6 Базові атрибути елементів оформлення тексту

Важливим атрибутом, який мають теги заголовків, абзаців та списків є атрибут *align*. Він визначає, який тип вирівнювання буде застосовано до змісту тега. Наприклад

```
<h1 align="center"> Заголовок </h1>
```

означає, що слово "Заголовок", яке є заголовком першого рівня, вирівнюється посередині сторінки.

Можливі значення атрибута *align* наведено у таблиці 1.3.

Таблиця 1.3 – Значення атрибута *align*

Значення	Пояснення
left	Вирівнювання змісту за лівим краєм
center	Вирівнювання змісту за центром
right	Вирівнювання змісту за правим краєм
justify	Вирівнювання змісту за лівим і правим краєм одночасно

Слід зазначити, що цей атрибут вважається застарілим в стандарті HTML 5 і нерідко замінюється аналогічним рішенням за допомогою CSS стилів.

1.3 Завдання на лабораторну роботу

1.3.1 Ознайомитися з теоретичними відомостями, необхідними для виконання роботи.

1.3.2 Створити HTML-шаблон сторінки, що містить теги, які визначають структуру HTML-документа.

1.3.3 Наповнити створену сторінку текстом, що представляє собою резюме студента. Резюме має містити:

- назву документа;
- персональні дані, такі як прізвище, ім'я, по батькові, дату народження;
- дані про освіту у зворотному хронологічному порядку;
- інформацію про досвід роботи (якщо є);
- професійні навички (загальний рівень володіння комп'ютером, перелік відомих офісних програм та середовищ розробки та рівень знайомства з ними), знання мов;
- контактну інформацію (email, телефон).

1.3.4 Оформити резюме, використовуючи якомога більше тегів форматування тексту: <h1>, <h2>, <h3>, <p>, , <u>, <i>, , та ін.

Резюме має бути добре структурованим, візуально розділеним на розділи і секції за допомогою заголовків та підзаголовків. Професійні навички та рівень володіння ПЗ слід оформити у вигляді нумерованого списку, а дані про освіту – у вигляді маркованого.

1.3.5 Перевірити створену сторінку за допомогою online-валідатора <https://validator.w3.org/> (розділ Validate by Direct Input), та виправити знайдені помилки, якщо такі є.

1.3.6 Оформити звіт з роботи.

1.3.7 Відповісти на контрольні питання, наведені в кінці роботи.

1.4 Зміст звіту

1.4.1 Тема та мета роботи.

1.4.2 Завдання до роботи.

1.4.3 Результати виконання роботи у вигляді знімка екрана, що відображає зовнішній вигляд розробленої сторінки у вікні браузера.

1.4.4 Код HTML-документа.

1.4.5 Висновки, що містять відповіді на три контрольних запитання на вибір, а також відображають результати виконання роботи.

1.5 Контрольні запитання

1.5.1. Що таке HTML?

1.5.2. Якими стандартами декларується синтаксис мови HTML?

1.5.3. Наведіть три приклади коду, що відрізняється у стандартах HTML 4.01 та HTML 5.

(Наприклад HTML 4.01:
, HTML 5:
)

1.5.4. Які корисні функції пропонує спеціалізоване ПЗ для розробки коду на HTML?

1.5.5. Як увімкнути підсвічування синтаксису HTML в програмі Notepad++?

1.5.6. Які розширення можуть мати файли з HTML кодом?

1.5.7. Які компоненти може мати елемент коду HTML?

1.5.8. Що таке тег?

1.5.9. Де розміщується зміст елемента?

1.5.10. Чому деякі елементи мають кінцевий тег, а деякі – ні?

1.5.11. Наведіть як мінімум три одинарних тега (таких, що не мають кінцевого тега).

1.5.12. Для чого потрібен тег <!DOCTYPE>?

1.5.13. Які структурні теги ви знаєте?

1.5.14. Для чого потрібен тег <html>?

1.5.15. Які частини змісту тега <html> побачить кінцевий користувач у браузері?

1.5.16. Для чого потрібен тег <head>?

1.5.17. Що включає тег <title>?

1.5.18. Для чого потрібен тег <body>?

1.5.19. Що таке мета-теги? Назвіть декілька основних мета-тегів та їхнє призначення.

1.5.20. Що таке кодування сторінки? Як воно впливає на відображення документа у браузері?

1.5.21. Скільки рівнів заголовків передбачено в HTML?

1.5.22. Які текстові теги ви знаєте?

1.5.23. Які теги можуть виконати перенос тексту на новий рядок? Чим вони відрізняються?

1.5.24. Які дії виконує тег `<hr />`?

1.5.25. Які елементи фізичного форматування вам відомі?

1.5.26. Які елементи логічного форматування вам відомі?

1.5.27. Що спільного і чим відрізняються елементи фізичного та логічного форматування?

1.5.28. Що таке списки? Які види списків передбачає мова HTML?

1.5.29. Наведіть приклад коду маркованого списку.

1.5.30. Наведіть приклад коду нумерованого списку.