

## Практична робота № 2

### Введення в мову HTML

**Мета роботи:** Ознайомитись з основами мови розмітки гіпертексту.

**Завдання роботи:**

1. Ознайомитись з синтаксисом мови розмітки гіпертексту.
2. Оволодіти технікою організації структури Web-документа.

**Забезпечуючі засоби:** Notepad++ 5.8.6; FireFox 3.6.

### Теоритичні відомості

Автономні Web-документи використовують мову HTML (HypertextMarkup Language - мова розмітки гіпертексту). Гіпертекст, тобто розширений текст, включає додаткові елементи: ілюстрації, посилання, вставні об'єкти. Під розміткою розуміється використання спеціальних кодів, легко відділяємих від смислового змісту документа і використовуваних для реалізації гіпертекста. Застосування цих кодів підпорядковується суворим правилам, що визначаються специфікацією мови HTML.

HTML-документ - це файл, що містить звичайний текст зі спеціальними командами.

Такий файл може бути підготовлений у довільному текстовому редакторі (існують, проте, спеціальні програми-конвертори і HTML-редактори).

HTML-документ складається з вмісту, тобто власне корисної інформації і команд, які задають структуру.

Кожна команда (керуюча конструкція) HTML-документа (тег) повинна розміщуватись в кутових дужках - ось так: <тег>. Найчастіше в документі зустрічаються парні теги (відкриваючи і відповідний йому закриваючий), так як браузеру необхідно знати область дії тега. Існують і одиночні теги, проте, використовуються вони тільки там, де область дії очевидна і додаткової інформації не потрібно (зрозуміло, наприклад, що якщо ми зустріли тег "початок абзацу" (<P>), то попередній абзац вже закінчився). У сумнівному ж випадку краще перестраховатися і поставити закриваючий парний тег, інакше документ може виявитися нечитабельним. Відкриваючий і закриваючий теги називаються однаково і відрізняються один від одного тільки символом "похила риса" або "слеш" - "/", який ставиться відразу після першою кутової дужки закриваючого тега. Закриття парних тегів виконується так, щоб дотримувалися правила вкладення.

**<b> <i> На цей текст впливають два тега </ i> </ b>**

Крім того, тег може містити атрибут, що дає додаткову інформацію браузеру. Наприклад, за допомогою атрибута можна попросити браузер змінити величину шрифту, орієнтацію зображення по відношенню до рядка наступного за ним тексту, поміняти колір фону документа і т. д. У парних тегах атрибути додаються лише до відкриваючого тегу. Атрибути представляють собою додаткові ключові слова, відокремлювані від ключового слова, що визначає тег, і від інших атрибутів пробілами і розміщуються до завершального тег символу ">". Спосіб застосування деяких атрибутів вимагає вказівки значення атрибута. Значення атрибута відокремлюється від ключового слова атрибута символом "=" (знак рівності) і розміщується в лапках.

**<H1 ALIGN="LEFT">**

Мова HTML в більшості випадків байдужа до регістру, в якому набираються теги. Скажімо, браузеру абсолютно все одно, наберете ви тег, який служить для малювання горизонтальної лінії, як <HR> або <hr> - ефект буде один і той же.

HTML не визнає ніякого додаткового форматування тексту, окрім як за допомогою тегів. У результаті текст, який чудово виглядає в текстовому редакторі, у вікні браузера зіллється в єдину нечитабельним масу. Так, на місці кількох прогалін буде лише один пробіл. Зникнуть всі заголовки, порожні рядки, поділи тексту на абзаци. Без HTML-тегів браузер просто ігнорує всі елементи форматування.

Визначення HTML як мови розмітки ґрунтується на тому, що при видаленні з документа всіх тегів виходить текстовий документ, абсолютно еквівалентний за змістом вихідного гіпертекстових документів. Таким чином, при відображенні документа HTML самі теги не відображаються, але впливають на спосіб відображення решти частини документу.

Якщо відкрити будь-яку веб-сторінку, то вона буде містити в собі типові елементи, які не змінюються від виду та спрямованості сайту.

### **Приклад 1.**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8"> <!-- windows-1251 -->
    <title>Приклад веб-сторінки</title>
  </head>
  <body>
    <h1>Заголовок</h1>
```

```

<!-- Коментар -->
<p>Перший абзац.</p>
<p>Другий абзац.</p>
</body>
</html>

```

Розглянемо поетапно наш приклад

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

```

Елемент `<!DOCTYPE>` призначений для вказівки типу поточного документа - DTD (document type definition, опис типу документа). Це необхідно, щоб браузер розумів, як слід інтерпретувати поточну веб-сторінку, адже HTML існує в декількох версіях, крім того, є XHTML (EXtensible HyperText Markup Language, розширена мова розмітки гіпертексту), схожа на HTML. Щоб браузер «не плутався» і розумів, згідно з яким стандартом відображати веб-сторінку і необхідно в першому рядку коду задавати `<!DOCTYPE>`.

Існує кілька видів `<!DOCTYPE>`, вони різняться в залежності від версії HTML, на яку орієнтовані.

DOCTYPE	Опис
<b>HTML 4.01</b>	
<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"&gt;</code>	Строгий синтаксис HTML.
<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;</code>	Перехідний синтаксис HTML.
<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"&gt;</code>	У HTML-документі застосовуються фрейми.
<b>HTML 5</b>	
<code>&lt;!DOCTYPE html&gt;</code>	У цій версії HTML тільки один доктайп.
<b>XHTML 1.0</b>	
<code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"&gt;</code>	Строгий синтаксис XHTML.
<code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-</code>	Перехідний синтаксис XHTML.

transitional.dtd">	
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">	У XHTML -документі застосовуються фрейми.
<b>XHTML 1.1</b>	
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">	

Різниця між строгим і перехідним описом документа полягає в різному підході до написання коду документа. Строгий HTML вимагає жорсткого дотримання специфікації HTML і не прощає помилок. Перехідний HTML більш «спокійно» ставиться до деяких огріхів коду, тому цей тип у певних випадках використовувати переважно.

### <html>

Тег <html> визначає початок HTML-файла, всередині нього зберігається заголовок (<head>) і тіло документа (<body>).

### <head>

Заголовок документа, як ще називають блок <head>, може містити текст і теги, але вміст цього розділу не показується прямо на сторінці, за винятком контейнера <title>.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

Тег <meta> є універсальним і додає цілий клас можливостей, зокрема, за допомогою метатегів, як узагальнено називають цей тег, можна змінювати кодування сторінки, додавати ключові слова, опис документа і багато іншого. Щоб браузер розумів, що має справу з кодуванням UTF-8 (Unicode transformation format, формат перетворення Юнікоду) і додається даний рядок.

### <title> Приклад веб-сторінки </ title>

Тег <title> визначає заголовок веб-сторінки, це один з важливих елементів призначений для вирішення безлічі завдань. В операційній системі Windows текст заголовка відображається в лівому верхньому кутку вікна браузера.

Тег <title> є обов'язковим і повинен неодмінно бути присутнім в коді документа.

### </ head>

Обов'язково слід додавати закриває тег </ head>, щоб показати, що блок заголовка документа завершено.

### <body>

Тіло документа <body> призначено для розміщення тегів і змістовної частини веб-сторінки.

## **<h1> Тема </ h1>**

HTML пропонує шість текстових заголовків різного рівня, які показують відносну важливість секції, розташованої після заголовка.

### **<! - Коментар ->**

Деякий текст можна сховати від показу в браузері, зробивши його коментарем. Хоча такий текст користувач не побачить, він все одно буде передаватися в документі, так що, подивившись вихідний код, можна виявити приховані нотатки.

Коментарі потрібні для внесення в код своїх записів, які не впливають на вигляд сторінки.

Починаються вони тегом <! - і закінчуються тегом ->. Все, що знаходиться між цими тегами відображатися на веб-сторінці не буде.

### **<p> Перший абзац. </ p>**

Тег <p> визначає абзац (параграф) тексту. Якщо закриває тега немає, вважається, що кінець абзацу збігається з початком наступного блокового елемента.

### **<p> Другий абзац. </ p>**

Тег <p> є блоковим елементом, тому текст завжди починається з нового рядка, абзаци що йдуть один за одним розділяються між собою відбиттям (так називається порожній простір між ними).

### **</ body>**

Слід додати закриваючий тег </ body>, щоб показати, що тіло документа завершено.

### **</ html>**

Останнім елементом у коді завжди йде закриваючий тег </ html>.

Більшість елементів мови HTML описує частини змісту документа і розміщується між тегами <BODY> і </ BODY>, тобто, всереді структурного елемента BODY.

Кожен тег HTML належить до певної групи тегів, наприклад, табличні теги спрямовані на формування таблиць і не можуть застосовуватися для інших цілей.

Умовно теги діляться на наступні типи:

- \* Теги верхнього рівня;
- \* Теги заголовка документа;
- \* Блокові елементи;
- \* Вбудовані елементи;
- \* Універсальні елементи;
- \* Списки;
- \* Таблиці.

Слід враховувати, що один і той же тег може одночасно належати різним групам, наприклад, теги `<ol>` і `<ul>` відносяться до категорії списків, але також є і блочними елементами.

Засіб, що використовується для відображення Web-документа, зробить все можливе, щоб витлумачити розмітку найбільш розумним чином. Тим не менш, гарантію правильного відтворення документу дає тільки неухильне дотримання вимогам специфікації мови.