

## МОДУЛЬ I. Мова HTML 4.01. Каскадні таблиці стилів CSS

### Лабораторна робота 1.1.

#### Структура HTML-документа. Форматування тексту веб-сторінки. Гіпертекстові посилання. Використання глобальних і локальних посилань в HTML-документах. Створення навігаційних карт

**Мета:** ознайомитися з мовою HTML і створенням найпростішого документа; набуття навичок форматування веб-сторінки; ознайомитися з принципами створення гіперпосилань, способами додавання графіки на веб-сторінку та технологією створення навігаційних карт.

#### Вимоги до обладнання та програмного забезпечення

Лабораторна робота виконується на ПК з використанням програми Microsoft Expression Web 4 та браузерів Google Chrom, Opera, Mazila Firefox.

#### Основні теоретичні відомості

Елементи розмітки бувають двох типів - одиночні й парні (теги). Одиночний тег використовується самостійно, а парний може включати в себе інші теги або текст. Елемент розмітки складається з трьох компонентів: початкового елемента розмітки, вмісту і кінцевого елемента. У деяких елементах може не бути кінцевого елемента розмітки.

Всі елементи можна умовно розбити на декілька категорій:

- Структурні, які задають структуру документа (наприклад, *html*, *head*, *body* і *title*);
- блокові елементи, призначені для форматування цілих текстових блоків (наприклад, *div*, *h1*, *p*, *pre*);
- текстові елементи, які задають розмітку тексту (*em*, *dfn*, *code*, *samp*);
- спеціальні елементи порожнього рядка (*br*, *hr*, *nobr*), якірний елемент *a*, вбудовані елементи (*embed*, *uimg*, *map*), елементи форм (*input* *select*), елементи таблиць (*table*) та інші.

Часто елементи розмітки містять додаткові елементи – атрибути. Допустимі різні атрибути, які розділяються між собою пробілом. Втім, є теги без будь яких додаткових атрибутів. Умовно атрибути можна розділити на обов'язкові, неодмінно повинні бути, і необов'язкові (їх додавання залежить від мети застосування тега). Атрибут записується після імені елемента розмітки перед закриваючою дужкою і складається, як правило, з пари «ім'я атрибут та – значення».

Правила сумісності описує стандарт XHTML:

- всі імена елементів розмітки і атрибутів пишуться маленькими літерами;
- всі елементи розмітки обов'язково мають закриватися: парні – звичайним способом, непарні: `<... />` (пробіл перед косою рисою обов'язковий з міркувань оберненої сумісності із старими браузерами);
- значення атрибутів завжди існують і завжди записуються в лапках;
- використовувати застарілі і специфічні для конкретного браузера елементи не можливе;
- можливе оформлення виноситься у таблиці стилів.

HTML-документ складається з трьох частин:

1. оголошення типу документа,
2. заголовка документа,
3. тіла документа.

#### Приклад 1.1:

```
<!DOCTYPE HTML >
<html>
<head>
  <title>Задаємо назву документа</title>
</head>
<body>
  Основна частина документа
</body>
</html>
```

Елементи, що відносяться до заголовка документа:

`<head>` – містить технічну інформацію про сторінку: заголовок, опис, ключові слова для пошукових машин, кодування і т.д.;

`<title>` – текст, розміщений всередині цього тега, відображається в рядку заголовка веб-браузера. Довжина заголовка повинна бути не більше 60 символів, щоб повністю поміститися в заголовок.

`<style>` – використовується для вставлення в документ таблиці стилів CSS. Таких елементів на сторінці може бути декілька;

`<link>` – описує взаємозв'язок документа з іншими документами на сайті, вказуючи його місце в ієрархічній структурі сайту. Елемент не має кінцевого елемента розмітки. У заголовку може містити кілька елементів `<link>`;

`<meta>` – технічна інформація про документ. З його допомогою можна задати опис вмісту сторінки і ключові слова для пошукових машин, автора HTML-документа й інші властивості метаданих. Елемент не має кінцевого елемента розмітки;

`<script>` – дозволяє приєднувати до документа різні сценарії. Тег обов'язково закривається, при цьому текст сценарію може розташовуватися або всередині цього елемента, або в зовнішньому файлі. Якщо текст сценарію розташований у зовнішньому файлі, то він підключається за допомогою атрибутів елемента;

елемент `<body>` призначений для розміщення даних основної частини веб-сторінки. Вона включає блокові і текстові елементи. Блокові елементи відносяться до частин тексту абзацу. Текстові елементи описують властивості окремих фраз і ще більш дрібних частин тексту.

Правила вкладення елементів:

- елементи не повинні перетинатися. Іншими словами, якщо відкриваючий елемент розташовується всередині елемента, то і відповідний закриваючий елемент повинен розташовуватися в межах цього самого елемента;
- блокові елементи можуть містити вкладені блокові і текстові елементи;
- текстові елементи можуть містити вкладені текстові елементи;
- текстові елементи не можуть містити вкладені блокові елементи.

Мова HTML допускає два підходи до виділення фрагментів тексту. Розробник може прямо вказати, що шрифт на якійсь ділянці тексту повинен бути жирним або курсивом, тобто змінити фізичний стиль тексту. Або може виділити деякий фрагмент тексту як такий, що має відмінний від звичайного логічний стиль, залишивши інтерпретацію цього стилю браузеру.

При використанні логічних стилів автор документа не може знати заздалегідь, що побачить на екрані читач. Різні браузери тлумачать одні й ті самі мітки логічних стилів по-різному. Деякі браузери ігнорують деякі мітки взагалі і показують нормальний текст замість виділеного логічним стилем.

Логічні елементи наведено в табл. 1.1.

Таблиця 1.1

<code>&lt;em&gt; ... &lt;/em&gt;</code>	Від англійського <i>emphasis</i> – наголос, виділяє текст, за замовчуванням при відображенні на екрані виводить його курсивом
<code>&lt;strong&gt; ... &lt;/strong&gt;</code>	Від англійського <i>strong emphasis</i> – сильний акцент
<code>&lt;code&gt; ... &lt;/code&gt;</code>	Призначений для відображення однієї або декількох рядків тексту, який являє собою програмний код
<code>&lt;samp&gt; ... &lt;/samp&gt;</code>	Використовується для відображення тексту, який є результатом виведення комп'ютерної програми або скрипта
<code>&lt;kbd&gt; ... &lt;/kbd&gt;</code>	Від англійського <i>keyboard</i> – клавіатура. Виділяє текст, що вводиться з клавіатури
<code>&lt;var&gt; ... &lt;/var&gt;</code>	Від англійського <i>variable</i> – змінна. Використовується для виділення змінних комп'ютерних програм

Елемент `<font>`, хоча і скасований, залишається досить популярним. Цей елемент керує зовнішнім виглядом шрифту і повинен містити хоча б один з трьох атрибутів:

- *face* – вказує гарнітуру шрифту. Наприклад, *face="arial"*. Якщо такий шрифт не встановлено на комп'ютері, то через кому вказують кілька шрифтів у порядку переваги;
- *size* – встановлює розмір шрифту у власних умовних одиницях від 1 до 7 (найбільший). Значення за замовчуванням – 3 (близько 12 пунктів);
- *color* – вказує колір шрифту в системі RGB.

Під фізичним елементом прийнято розуміти пряму вказівку браузеру на модифікацію поточного шрифту. Наприклад, все, що розташовується між елементами `<b>` і `</b>`, буде написано жирним шрифтом. Текст між елементами `<i>` і `</i>` буде написаний курсивним шрифтом. Дещо окремо стоїть елемент розмітки `<tt>`.

Фізичні елементи наведено в табл. 1.2.

Таблиця 1.2

<code>&lt;b&gt;</code>	жирний шрифт
<code>&lt;i&gt;</code>	курсив
<code>&lt;tt&gt;</code>	шрифт фіксованої ширини (як на друкарській машинці)

<u>	підкреслений текст
<strike>	закреслений шрифт
<s>	теж закреслений шрифт
<big>	шрифт більшого розміру
<small>	шрифт меншого розміру

Шрифти можуть комбінуватися. Наприклад, використовуючи такий текст `<b><i>приклад</i></b>`, ми отримаємо комбінацію жирного і курсивного шрифтів.

Блочні елементи використовують для розбивання тексту документа на блоки. Прикладами текстових блоків є параграфи, абзаци і заголовки. Для відділення однієї частини тексту від іншої також використовують розділові горизонтальні лінії і символи повернення каретки.

Елементи `h1, h2, ..., h6` використовують для створення заголовків тексту. Їх слід використовувати тільки для виділення заголовків нового розділу або підрозділу. При використанні заголовків потрібно враховувати їх ієрархію, тобто за `<h1>` повинен піти `<h2>` і т.д. Також не допускається вкладення інших тегів у теги `<h1> ... <h6>`. Вони мають атрибут `align` – який визначає спосіб вирівнювання заголовка по горизонталі. Можливі значення: *left, right, center, justify*.

Елемент розмітки `p` розбиває текст на окремі абзаци.

Елемент розмітки `<div>` використовують для логічного виділення блоку HTML-документа. Він призначений для виділення фрагмента документа з метою зміни виду вмісту. Як правило, вид блока управляється за допомогою стилів. Розміщуючись між початковим і кінцевим елементами розмітки, текст або HTML-елементи за замовчуванням оформляються як окремий параграф.

Елемент розмітки `<address>` призначений для зберігання інформації про автора або адреси і може включати в себе будь-які елементи HTML на зразок посилань, тексту, виділень і т.д. Оформлення виражається у виділенні рядка адреси курсивом.

#### Лістинг 1.2:

```
<address>
```

Ця сторінка розроблена веб-дизайнером Петренком В.І.

```
</address>
```

Елемент розмітки `<blockquote>` оформляє текст що розташований між початковим і кінцевим елементами як цитату. Використовується для довгих цитат (на відміну від елемента `<cite>`). Цитований текст відображається окремим абзацом зі збільшеним відступом.

Елемент розмітки `<br>` – здійснює перехід до нового рядка, аналогічний натисканню *Enter* у текстовому редакторі.

Елемент розмітки `<hr>` додає в текст горизонтальну лінію.

#### Приклад 1.3:

```
<hr noshade width="90%" color="red">
```

`<pre>` – виводить текст без форматування зі збереженням пробілів і переносів тексту. Може бути використаний для відображення комп'ютерного коду, повідомлення електронної пошти і т.д. Бажано уникати використання символу горизонтальної табуляції всередині `<pre>`, оскільки він може бути неадекватно інтерпретований деякими браузерами. Замість символу табуляції рекомендується використовувати число пробілів, кратне чотирьом.

Списки в HTML бувають двох видів:

- впорядковані (пронумеровані);
- неупорядковані (не пронумеровані).

Відрізняються вони лише способом оформлення. Перед пунктами неупорядкованих списків зазвичай ставляться маркери (*bullets*), наприклад, крапки, квадрати та інше, тоді як у пунктах впорядкованих списків використовуються номери.

Елемент розмітки `<ul>` створює неупорядкований список. Між початковим і кінцевим елементами повинні бути один або кілька елементів `<li>`, що позначають окремі пункти списку.

#### Приклад 1.4:

```
<ul>
```

```
<li> Системне програмування </li>
```

```
<li> Бази даних </li>
```

```
<li> Інтернет-технології </li>
```

```
</ul>
```

Елемент розмітки `<ol>` створює впорядкований список. Між початковим і кінцевим елементами повинні бути один або кілька елементів `<li>`, що позначають окремі пункти списку. Має атрибути:

`start` – визначає перше число, з якого починається нумерація пунктів (лише цілі числа)

`type` – визначає стиль нумерації пунктів. Може мати значення:

"A" – заголовні букви A, B, C ...

"a" – рядкові букви a, b, c ...

"I" – великі римські числа I, II, III ...

"i" – маленькі римські числа i, ii, iii ...

"1" – арабські числа 1, 2, 3 ...

За замовчуванням `<ol type="1">`.

#### Приклад 1.5:

```
<ol type="i" start="2">
```

```
<li> Другий пункт впорядкованого списку </li>
```

```
<li> Третій пункт впорядкованого списку </li>
```

```
<li> Четвертий пункт впорядкованого списку </li>
```

```
</ol>
```

Елемент розмітки `<li>` створює пункт у списку. Розташовується усередині елементів `<ol>` або `<ul>`.

Елемент розмітки `<dl>` відкриває і закриває список визначень (термінів та їх описів). Визначення задаються з допомогою елементів `<dt>` та `<dd>`.

#### Приклад 1.6:

```
<dl>
```

```
<dt>Кава</dt><dd>батьорий гарячий напій</dd>
```

```
<dt>Молоко</dt><dd>корисний холодний напій</dd>
```

```
</dl>
```

Елемент розмітки `<dt>` створює термін у списку визначень всередині елемента `<dl>`

Елемент розмітки `<dd>` створює визначення терміна всередині елемента `<dl>`.

Гіперпосилання – посилання на інші документи в HTML, які створюються за допомогою елемента розмітки `<a>`. Текст посилання відображається в браузері з підкресленням, колір шрифту – синій, при наведенні на посилання курсор миші змінює вид. Походить від англійського слова *anchor* – «якір».

Елемент розмітки `<a>` застосовується, якщо посиланням планується зробити частину тексту або ціле зображення. Посилання - інструмент, що дозволяє зв'язувати між собою різні документи.

HTML означає «мова маркування гіпертекстів». Гіпертекст, на відміну від звичайного тексту, який можна читати лише від початку до кінця, дозволяє здійснювати миттєвий перехід від однієї частини тексту до іншої. Системи допомоги багатьох популярних програмних продуктів влаштовані саме за гіпертекстовим принципом. Натиснувши лівою кнопкою миші на деякий виділений фрагмент документа, можна перейти до деякого заздалегідь призначеного документа або фрагмента документа.

Елемент розмітки `a` має атрибути:

- `href` – визначає розташований між початковим і кінцевим елементами текст або зображення як гіпертекстове посилання (URL, або лінк) на документ (і/або область документа), зазначений у значенні даного атрибута. Можливі значення:
- `http://...` – надає доступ до веб-сторінки за протоколом HTTP;
- `ftp://...` – здійснює запит до FTP-сервера на отримання файлу;
- `mailto:...` – запускає сеанс поштового зв'язку із зазначеним адресатом і хостом. Якщо після адреси поставити знак питання, то можна вказати додаткові атрибути, розділені знаком "&";
- `name` – позначає розташовану між початковим та кінцевим елементами область документа як можливий об'єкт для посилання. Як значення потрібно латиною написати будь-яке слово-показник, унікальне для даного документа.  
Наприклад: `<a name="part">Розділ</a>`. Тепер можна посилатися на позначену область простим зазначенням її імені після імені документа. Наприклад, лінк `<a href="document.html#part"> Розділ</a>` відправить в розділ "part" файла document.html, а лінк `<a href="#bottom">У кінець документа</a>` – в розділ "bottom" поточного документа.
- `target` – визначає вікно (фрейм), на яке вказує гіпертекстове посилання. Цей атрибут використовується лише разом з атрибутом `href`. Як значення потрібно задати ім'я одного з наявних фреймів або одне з таких зарезервованих імен: `_self`, `_parent`, `_top`, `_blank`.

#### Приклад 1.7:

```
<!-- Використання атрибута name: -->
```

<a name="history">Історія спорту</a>

...

<a name="now">Новини спорту</a>

...

Повернутися до розділу<a href="#history">Історія</a>

#### Приклад 1.8:

<!-- Використання атрибуту href: -->

<a href="ftp://ftp.cdrom.com" target="\_blank">FTP-site</a>

<a href="http://opengl.rdc.ru">Російський проект з OpenGL</a>

#### Приклад 1.9:

<!-- Створимо посилання для листа -->

<a href="mailto:green@igf.ua?subject=Запрошення &cc=bg@microsoft.com&body=Чекаємо на вечірку.">

Відправити запрошення </a>.

<!-- або просто листа : -->

<a href="mailto:green@igf.ua?subject=Вітаю">авторам</a>.

Елемент *a* не може бути вкладеним у собі подібні, тобто неприпустимі конструкції на зразок

<a href="link1.html">

Перше посилання

<a href="link2.html">Друге посилання </a>

Продовжуємо перше посилання

</a>

Описуються посилання так: <a href="шлях/ім'я файлу">Посилання на документ</a>. Може містити атрибут *title*, що описує посилання. Якщо користувач наведе курсор миші на посилання, то з'явиться його опис – вміст цього атрибута.

Розрізняють відносні й абсолютні посилання. Приклади відносного посилання:

<a href="index.html"> – на документ "index.html", який міститься в одному каталозі з поточним документом;

<a href="./index.html"> – те саме;

<a href="folder/index.html"> – перехід на файл "index.html" у підкаталозі "folder" поточного каталогу;

<a href="../folder/index.html"> – піднятися вгору на один крок по дереву каталогів, перейти у каталог "folder" та перейти на "index.html".

В абсолютних посиланнях вказується повний шлях до файлу:

<a href="http://www.maineprage.com/business/139/index.htm">

Відносні посилання зручніші. Вони не прив'язані до конкретної структури каталогів, тобто можна створити свій сайт на комп'ютері, тестувати і виправляти помилки не підключаючись до Інтернету. І лише коли він буде готовий перенести його на сервер, він працюватиме так само.

Можна робити посилання всередині документа. Це зручно для переходу до різних глав однієї великої статті. Для цього потрібно створити якусь опорну точку – анкер (*anchor* – якір, англ.), на яку і відбуватиметься перехід.

Щоб зробити графічне посилання, слід набрати той самий HTML-код, що і для текстового посилання. Але в тому місці, де був текст, потрібно вставити графічне зображення.

<a href="filename.html"> </a>

Для розміщення зображення на сторінці слід скористатись елементом розмітки <img>, обов'язковий атрибут якого *src* визначає адресу графічного файлу.

Найбільш популярні графічні формати, які підтримуються усіма сучасними графічними браузером, це:

- **Формат JPEG** (*Joint Photographic Experts Group*). Зображення JPEG ідеальні для фотографій, вони можуть містити мільйони різних кольорів. Стискають зображення краще за формат GIF, але текст і великі площі із суцільним кольором можуть покритися плямами.
- **Формат PNG** (*Portable Network Graphics*). Включає в себе кращі риси GIF- і JPEG-форматів. Містить 64 000 кольорів і дає можливість зробити один із кольорів прозорим, при цьому стискає зображення в менший розмір, ніж GIF-файл.
- **Формат GIF** (*Graphics Interchange Format*). Ідеальний для стискання зображень, у яких є області із суцільним кольором, наприклад, логотипів. GIF-файли дають змогу встановити один із кольорів прозорим, завдяки чому фон веб-сторінки може проявлятися через частину зображення. Також GIF-файли можуть включати в себе просту анімацію. GIF-зображення містять лише 256 відтінків, тому зображення виглядають плямистими і нереалістичного кольору, як плакати.

#### Приклад 1.10:

 – малюнок "pic.gif" знаходиться в тому ж каталозі, що і поточний документ;

`` – перейти в папку "images" поточного каталогу і взяти файл звідти;  
`` – піднятися вгору на один крок, перейти у каталог "images" і взяти малюнок звідти;  
`` – зазначення повного шляху до файлу.  
Зазвичай застосовується, якщо малюнок розміщується на іншому сервері, або використовується картинка з іншого сайту.

Елемент розмітки `<img>` має атрибути:

- `alt="пояснювальний текст"` – альтернативний текст. Текст, який з'являється замість зображень при перегляді сторінки браузером із відключеним автоматичним завантаженням зображень (відключають – для економії часу на повільних лініях зв'язку);
- `border="n"` – рамка навколо малюнка, де  $n$  – її товщина в пікселях. При  $n = "0"$  рамка відсутня;
- `width="n"` – задається ширина зображення в пікселях або у відсотках від ширини вікна браузера (тоді після  $n$  ставлять знак %);
- `height="n"` – висота зображення у пікселях або у відсотках від висоти вікна браузера.

Тег `<map>` служить для подання графічного зображення у вигляді карти з активними областями. Існує два типи зображень-карт:

- клієнтські (*client-side*) – коли користувач клікає по малюнку, браузер сам інтерпретує координати кліка. Він вибирає посилання, визначені для даної області, і переходить по ньому (або виконує задану дію);
- серверні (*server-side*) – координати кліка передаються для інтерпретації на сервер, і вже він робить відповідні дії (наприклад, повертає браузеру URL для переходу).

Перший тип простіший і доступніший, тому далі будемо розглядатимемо лише його.

Для створення зображення-карти використовуються елементи розмітки `<map>` та `<area>`.

Елемент `<map>` повинен обов'язково мати атрибут `name`. Це дає змогу вказати браузеру, до якого саме малюнка на сторінці належить ця карта.

Елемент `<area>` має такі атрибути:

- `shape` – описує форму виділеної області, можливі значення:
  - `rect` – прямокутник;
  - `circle` – коло;
  - `poly` – багатокутник;
  - `default` – визначає всю область.
- `coords` – координати, що визначають розміри та розташування області на зображенні. Кількість і порядок значень залежить від значення атрибута `shape`:
  - `rect`: – лівий- $X$ , верхній- $Y$ , правий- $X$ , нижній- $Y$  (тобто спочатку координати лівого верхнього кута, потім правого нижнього);
  - `circle`: – центр- $X$ , центр- $Y$ , радіус (тобто горизонтальна і вертикальна координати центру кола і радіус);
  - `poly`: –  $X_1, Y_1, X_2, Y_2, \dots, X_n, Y_n$  (просто перераховуються координати всіх вершин багатокутника).
- `target` – значення цього атрибута ("`_top`", "`_blank`", "`_self`" або "`_parent`") визначає, в якому вікні буде відкритий документ.

### Приклад 1.11:

```
<!-- Створюємо карту з ім'ям ImageMap: -->
<map name="ImageMap">
<area href="something.html" shape="rect" coords="0,0,70,140" alt="Ліва частина">
<area href="anything.html" shape="rect" coords="71,0,140,140" alt="Права частина">
</map>
<!-- Підключаємо зображення -->
...
```

### Завдання до самостійної роботи

1. Створіть HTML-сторінку, в якій буде інформація про ваше улюблене заняття (хобі). При створенні сторінки повинні використовуватися всі досліджені в лабораторній роботі елементи та атрибути (заголовки, абзаци, списки...).
2. Створену сторінку про хобі доповніть малюнками, спробуйте встановити фонове зображення. Використовуйте різні способи розміщення (вирівнювання) малюнків. Також створіть усі види посилань.
3. Створіть навігаційну карту, яка показує розклад занять.

### Контрольні запитання та завдання

1. Назвіть категорії елементів, які використовуються для створення веб-сторінок.
2. Назвіть елементи, що відносяться до заголовка документа.
3. Чим відрізняються логічні стилі від фізичних?
4. Які існують види гіперпосилань?
5. Назвіть основні графічні формати.
6. Опишіть призначення навігаційних карт.

### **Лабораторна робота 1.2.**

#### **Елементи створення таблиць HTML. Таблична верстка. Створення форм в HTML-документах.**

**Мета:** Набуття навичок створення таблиць.; ознайомитися з використанням таблиць для більш ефективного розміщення тексту й графіки на веб-сторінці; навчитися створювати HTML-форми; вивчити основні елементи форм.

## Вимоги до обладнання та програмного забезпечення

Лабораторна робота виконується на ПК з використанням програми Microsoft Expression Web 4 та браузерів Google Chrom, Opera, Mazila Firefox.

### Основні теоретичні відомості

Таблиці складаються з рядків, стовпчиків та комірок, які можуть містити текст і малюнки. Зазвичай таблиці використовують для впорядкування та представлення даних, однак можливості таблиць цим не обмежуються. За допомогою таблиць зручно верстати макети сторінок, розташувавши потрібним чином фрагменти тексту і зображень.

Для додавання таблиці на веб-сторінку використовується елемент розмітки `<table>`.

Для додавання рядків використовуються елементи `<tr>` і `</tr>`. Щоб розділити рядки на стовпці, застосовуються елементи `<td>` і `</td>`.

Для створення комірок із заголовками застосовуються елементи `<th>` і `</th>`.

За замовчуванням таблиця відображається без рамки, а розмітка здійснюється автоматично залежно від обсягу інформації.

#### Приклад 1.12:

```
<table border="1">
<caption align="right"><strong>Таблиця №1</strong> </caption>
<tr>
<th>Заголовок 1 </th>
<th>Заголовок 2 </th>
<th>Заголовок 3 </th>
</tr>
<tr>
<td>Дані в комірці 1 першого рядка</td>
<td>Дані в комірці 2 першого рядка</td>
<td>Дані в комірці 3 першого рядка</td>
</tr>
<tr>
<td>Дані в комірці 1 другого рядка</td>
<td>Дані в комірці 2 другого рядка</td>
<td>Дані в комірці 3 другого рядка</td>
</tr>
<tr>
<td>Дані в комірці 1 третього рядка</td>
<td>Дані в комірці 2 третього рядка</td>
<td>Дані в комірці 3 третього рядка</td>
</tr>
</table>
```

Основними атрибутами таблиць є:

- *align* – визначає спосіб горизонтального вирівнювання таблиці. Можливі значення: *left*, *center*, *right*. Значення за замовчуванням – *left*;
- *valign* – повинен визначати спосіб вертикального вирівнювання таблиці. Можливі значення: *top*, *bottom*, *middle*;
- *border* – визначає ширину рамки таблиці (у пікселях). При *border="0"* або при відсутності цього атрибута рамка не відображатиметься;
- *cellpadding* – визначає відстань (у пікселях) між рамкою кожної клітинки таблиці і матеріалом що міститься в ній;
- *cellspacing* – визначає відстань (у пікселях) між межами сусідніх комірок;
- *width* – визначає ширину таблиці. Ширину задається або в пікселях, або у відсотковому відношенні до ширини вікна браузера. За замовчуванням цей атрибут визначається автоматично залежно від обсягу матеріалу що міститься в таблиці;
- *height* – визначає висоту таблиці. Висоту задається або в пікселях, або у відсотковому відношенні до висоти вікна браузера. За замовчуванням цей атрибут визначається автоматично залежно від обсягу матеріалу що міститься в таблиці;



- *bgcolor* – визначає колір тла комірок таблиці. Задається або RGB-значенням у шістнадцятковій системі, або одним із 16 базових кольорів;
  - *background* – дозволяє заповнити тло таблиці малюнком. Як значення потрібно вказати URL малюнка.
- Елемент `<caption>` задає заголовок таблиці. Зміст заголовка повинен складатися тільки з тексту. Використання блокових елементів у цьому випадку неприпустимо.

Межі комірок відображаються лише в тому разі, коли вони мають якийсь зміст. Щоб отримати порожню комірку з межами, досить помістити в неї спеціальний символ `&nbsp;`, або маленьку прозору gif-картинку.

#### Приклад 1.13:

```
<table border="1" width="100%" cellpadding="10">
<caption align="right"><b>Приклад таблиці!</b></caption>
<tr>
<th rowspan="3">Об'єднана комірка по вертикалі</th>
<th colspan="3">Об'єднана комірка по горизонталі</th>
</tr>
<tr align="center">
<td>Комірка №1</td>
<td>Комірка №2</td>
<td>Комірка №3</td>
</tr>
<tr align="center">
<td>Комірка №4</td>
<td>Комірка №5</td>
<td>Комірка №6</td>
</tr>
</table>
```

Використання таблиць із невидимою межею відомий спосіб верстки, який застосовують для створення сайтів. Така таблиця фактично являє собою модульну сітку, в якій зручно розміщувати окремі елементи веб-сторінки.

Ширина таблиці, якщо вона явно не вказана, встановлюється браузером автоматично, виходячи з вмісту комірок. Коли таблиця застосовується для створення опорної сітки на сторінці, такий підхід неприйнятний, оскільки залежить від розміру даних. Тому ширину таблиці вказують завжди - у відсотках, якщо використовується «гумовий» макет, або в пікселях для макета фіксованої ширини.

Ширина комірок визначається атрибутом *width* елемента `<td>`. Переваги та недоліки табличної верстки наведено в табл. 1.3.

Таблиця 1.3

Основні переваги табличної верстки	Недоліки табличної верстки
Просте створення колонок «Гумова» верстка «Склейка» зображень Однозначне трактування браузером	тривале завантаження Громіздкий код Погана індексація пошуковими системами Немає поділу вмісту та оформлення Невідповідність стандартам Неможливість конвертації в мобільну версію

Зразок гумової табличної верстки сайта наведено в прикладі 1.14.

#### Приклад 1.14

```
<!DOCTYPE html>
<head>
<meta charset="windows-1251" />
<title>«Гумова» таблична верстка сайта</title>
</head>
<body>
<table cellpadding="0" cellspacing="0" width="90%" align="center" border="1">
<tr>
<td width="20" align="center">Логотип</td>
```

```

<td colspan="2" width="80%" height="60" align="center"><h1>Мій сайт</h1></td>
</tr>
<tr>
<td width=20% height="460" valign="top"><ul>
<li><a href="index.html">Посилання 1</a></li>
<li><a href="index.html">Посилання 2</a></li>
<li><a href="index.html">Посилання 3</a></li>
</ul></td>
<td width="70%" valign="top">Зміст</td>
<td valign="top">Посилання</td>
</tr>
<tr>
<td colspan="3" width="100%" align="left">&copy; Всі права захищені</td>
</tr>
</table>
</body>
</html>

```

Приклад жорсткої табличної верстки сайту наведено в прикладі 1.15.

### Приклад 1.15

```

<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html"; charset="windows-1251" />
<title>Жорстка таблична верстка сайту</title>
</head>
<body>
<table cellpadding="20" cellspacing="0" width="860" align="center" border="1">
<tr>
<td width="30">Логотип</td>
<td colspan="2" width="100%" height="60" align="center"><h1>Мій сайт</h1></td>
</tr>
<tr>
<td width="30" height="460" valign="top"><ul>
<li><a href="index.html" title="Посилання">Посилання</a></li>
<li><a href="index.html" title="Посилання">Посилання</a></li>
<li><a href="index.html" title="Посилання">Посилання</a></li>
</ul></td>
<td valign="top">Зміст</td>
<td width="20" valign="top">Посилання</td>
</tr>
<tr>
<td colspan="3" width="100%" align="center">&copy; Всі права захищені</td>
</tr>
</table>
</body>
</html>

```

За допомогою наведених нижче елементів можна створювати анкети, опитувальники і різні поля для введення тексту користувачем із можливістю подальшої відправки заповненої форми на сервер.

Елемент розмітки *<form>* використовується для створення заповнюваної форми. Обов'язково має бути початковий і кінцевий елементи. У середині елемента *<form>* можна використовувати HTML-елементи. Має атрибути:

- *name* – визначає ім'я форми, унікальне для даного документа. Використовується, якщо в документі є декілька форм;
- *action* – обов'язковий атрибут. Визначає адресу URL, за якою буде відправлено вміст форми – шлях до скрипта сервера, який обслуговує цю форму;
- *method* – визначає спосіб надсилання вмісту форми. Можливі значення *get* (за замовчуванням) та *post*;
- *enctype* – визначає спосіб кодування даних;

- *autocomplete* – Включає автозаповнення полів форми.

#### Приклад 1.16:

```
<!-- Створюємо форму -->
<form action="/cgi-bin/thanks.pl" method=get name="TestForm">
<!-- Всередині форми створюємо поле введення: -->
Прізвище:
<input type="text" name="lastname" size="20" value="Петренко"><br>
<!-- Кнопка " Відправити ": -->
<input type="submit" value="Відправити">
</form>
```

Під час налагодження скрипта, який приймає дані, частіше використовують метод *get*. Метод *get* не дозволяє передати скрипту великий обсяг даних. Якщо передбачається, що користувач заповнюватиме дуже велику форму або вводитиме об'ємні текстові дані, пересилатиме файл, використовують *method="post"*.

Елемент розмітки *<textarea>* створює поле для введення декількох рядків тексту. У текстовому полі допустимо робити переноси рядків, вони зберігаються при відправці даних на сервер. Елемент *<textarea>* повинен розташовуватися усередині елемента *<form>*. Він має такі атрибути:

- *name* – обов'язковий атрибут. Застосовується для програмування форм;
- *rows* – визначає кількість рядків тексту, видимих на екрані;
- *cols* – визначає ширину текстового поля – у друкованих символах;
- *wrap* – параметри перенесення рядків.

#### Приклад 1.17:

```
<form action="receive.html" method=post>
<textarea name="address" wrap=virtual" cols="40" rows="3">Ваша адреса...</textarea><br>
<input type="submit" value="ok">
</form>
```

Якщо у формі поле *<textarea>*, передача дані передаються за допомогою методу POST.

Елемент розмітки *<select>* створює в заповнюваній формі меню типу «Вибір одного пункту з багатьох» або «Вибір кількох пунктів із багатьох». Повинен розташовуватися усередині елемента *form* і мати як початковий, так і кінцевий елементами розмітки. Містить кілька елементів *<option>*. Має атрибути:

- *multiple* – дає можливість вибрати декілька пунктів меню при утриманні клавіші *Ctrl*. За замовчуванням можна вибрати лише один пункт меню;
- *name* – визначає ім'я меню, унікальне для форми, яке використовуватиметься при передачі даних на сервер. Кожен виділений пункт меню при передачі на сервер матиме вигляд: *name/value*. Значення (*value*) формується елементом *<option>*;
- *size* – визначає кількість видимих пунктів меню. Якщо значення цього атрибута більше одиниці, то результатом буде список пунктів.

#### Приклад 1.18:

```
<form action="receive.cgi">
<select name="os" multiple>
<option value="iOS">iOS
<option value="win">MS Windows 10
<option value="unix" selected>UNIX
<option value="linux">Linux
</select>
<input type="submit" value="послати">
</form>
```

Елемент розмітки *<option>* використовується лише з елементом *select*. Елемент *<option>* створює окремі пункти меню. Не має кінцевого тега. Має атрибути:

- *selected* – визначає пункт меню, який буде обрано спочатку при завантаженні документа. Якщо меню має тип "один з багатьох", то командою *selected* може бути помічений лише один з пунктів меню;
- *value* – задає цим пунктом значення, яке буде використано поряд з іншими відомостями про вміст заповненої форми. При наданні інформації на сервер це значення буде об'єднано зі значенням атрибута *name* в елементі *<select>*.

#### Приклад 1.19:

```
<form action="script.cgi">
```

```

<select name="gender">
<option value="male" selected>Не одружений
<option value="female">Одружений
<option value="not_yet">Визначаюся
</select>
<input type="submit" value="ok">
</form>

```

Елемент розмітки *<input>* створює поле форми (кнопку, поле введення, чекбокс тощо), зміст якого може бути змінено або активізовано користувачем. Елемент не має кінцевого елемента. Елемент *input* повинен розташовуватися у середині елемента *form*.

Має атрибути:

- *name* – визначає ім'я, яке використовуватиметься для доступу до елемента форми, наприклад, у таблицях стилів CSS. Цей атрибут необхідний для більшості типів елемента *<input>*;
- *type* – визначає тип поля для введення даних. За замовчуванням – це *"text"*. Можливі значення:
  - *text* – створює поле введення одного рядка тексту. Зазвичай використовується спільно з атрибутами *size* і *maxlength*;
  - *file* – дає можливість користувачеві додати файл до поточної форми. Можливе використання спільно з атрибутом *accept*;
  - *password* – створює поле введення пароля, текст що вводиться користувачем, відображається у вигляді піктограм "\*", щоб приховати введений пароль;
  - *checkbox* – прапорці. Дають змогу вибрати більш одного варіанта із запропонованих. Поле цього типу обов'язково повинно мати атрибути *name* і *value*, а також необов'язковий атрибут *checked*, який вказує на те, що поле активізовано;
  - *radio* – перемикачі. Використовуються, коли потрібно вибрати один варіант із декількох запропонованих. Як і для полів *checkbox*, атрибут *checked* необов'язковий; він може бути використаний для визначення виділеної кнопки у групі кнопок (*radio button*);
  - *submit* – створює кнопку, при натисканні якої заповнена форма надсилається на сервер. Атрибут *value* у даному випадку змінює напис на кнопці, зміст якої задано за замовчуванням залежно від браузера. Якщо атрибут *name* вказаний, то при натисканні цієї кнопки до інформації, що посилається на сервер, додається пара *name/value*, зазначена для атрибута *submit*, інакше пара не додається;
  - *image* – створює графічну кнопку-картинку, що запускає передачу даних на сервер. Місцезнаходження графічного зображення можна задати за допомогою атрибута *src*. При передачі даних серверу повідомляються координати *X* та *Y* тієї точки на зображенні, де було виконано клацання клавішею миші. Координати вимірюються з верхнього лівого кута зображення. При цьому інформація про поле типу *image* записується у вигляді двох пар значень *name/value*. Значення *name* утворюється за допомогою додавання до назви відповідного поля суфіксів *"X"* (абсциси), та *"Y"* (ординати);
  - *reset* – створює кнопку, що очищає значення полів форми до їх початкових значень. При натисканні кнопки дані на сервер не відправляються. Напис на кнопці можна змінити за допомогою атрибута *value*. За замовчуванням напис на кнопці залежить від версії браузера;
  - *hidden* – поля цього типу не відображаються на екрані монітора, що дає змогу розмістити «таємну» інформацію в межах форми. Вміст цього поля надсилається на сервер у вигляді *name/value* разом з іншою інформацією форми. Цей тип полів зручно використовувати для передачі даних від скрипта скрипту непомітно для користувача;
  - *button* – дозволяє створити кнопку у HTML документі, що, при використанні мови JavaScript, додає формі функціональності. Атрибут *name* дозволяє задати ім'я цій кнопці, яке може бути використане для будь-якої функції в скрипті. Атрибут *value* дозволяє задати текст, який буде відображений на кнопці в документі;
- *value* – задає текстовий заголовок для полів будь-якого типу, зокрема і кнопок. Для таких полів, як *checkbox* або *radio*, буде повернуто значення, задане в атрибуті *value*;
- *checked* – вказує, що поля типів *checkbox* і/або *radio* активізовані;
- *size* – визначає розмір поля в символах. Наприклад, щоб визначити поле з видимою шириною 24 символи, треба вказати *size="24"*;
- *maxlength* – визначає максимальну кількість символів, які можна ввести в текстове поле. Вона може бути більшою, ніж кількість символів, зазначених в атрибуті *size*. За замовчуванням кількість знаків не обмежена;
- *src* – задає URL-адресу зображення, що використовується при створенні графічної кнопки. Використовується спільно з атрибутом *type="image"*;

- *align* – визначає спосіб вертикального вирівнювання для зображень. Використовується спільно з атрибутом *type="image"*. Повністю аналогічний атрибуту *align* елемента *img*. За замовчуванням має значення *bottom*;
- *accept* – конкретизує тип файлу. Використовується лише спільно з параметром *type="file"*. Значення задається у вигляді *MIME*-типу.

#### Приклад 1.20:

```
<form name="form1" action="http://www.css.ua/cgi-bin/test.pl">
<input type="hidden" name="info" value="Реєстрація студентів">
<input type="radio" name="pol" value="male" checked> Не одружений <br>
<input type="radio" name="vol" value="female"> Одружений<br>
Ім'я:<br>
<input type="text" name="textfield" value="ім'я" size="30" maxlength="60"><br>
Пароль:<br>
<input type="password" width="10" name="passwd"><br><br>
<input type="submit" value="Відправити">
</form>
```

#### Приклад 1.21:

```
Хочу отримувати такі видання:<br>
<form name="form2" action="http://www.igf.ru/cgi-bin/magazines.pl">
<input type="checkbox" name="m1">Сьогодні<br>
<input type="checkbox" name="m2">Факти та коментарі <br>
<input type="checkbox" name="m3" checked>Урядовий кур'єр<br>
<input type="image" src="/img/button.gif" width="60" height="30">
</form>...
```

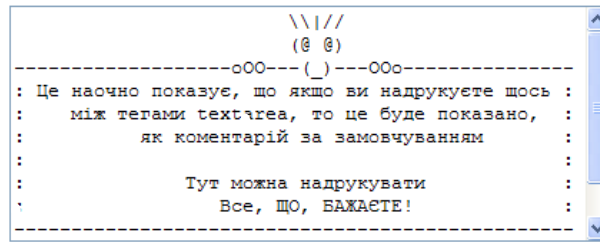
#### Завдання до самостійної роботи

1. Створіть таблицю за зразком:

№ з/п	Прізви- ще та ініціал- ли студен- та	Модульна рейтингова оцінка									
		Модуль І					Модуль ІІ				
		Пото- чна	Контро- льна		Підсум- кова		Пото- чна	Контро- льна		Підсум- кова	
			Ба- ли	Нац. шка ла	Ба- ли	Нац. шка ла		Ба- ли	Нац. шка ла	Ба- ли	Нац. шка ла

2. Використовуючи табличну верстку, створіть сторінку про свою навчальну групу.
3. Створіть форму, в якій повинно розміщуватися:
  - список перемикачів: синій, бузковий, блакитний, зелений, причому слова повинні відображатися відповідним кольором;
  - список прапорців повинен розміщуватися в центрі сторінки і містити назви восьми геометричних фігур;
  - дані про себе користувач повинен вводити в текстові поля (Прізвище, Ім'я), а кафедра у (КСУ або КСМ) і номер групи (320-326) вибираються за допомогою списків;
  - при натисканні кнопки «Відправити» повинна запускатися програма для відправки такого повідомлення:

```
cvet= синій
figura= еліпс
figura =прямокутник
fam= Шевченко
imyа= Максим
fakultet= КСУ
gruppa= 325
```
  - після натискання кнопки «Очистити поля форми» всі поля форми мають набути початкового вигляду;
  - в кінці сторінки створіть в точно таке оголошення (оголошення підказка для виконання цього завдання):



### Контрольні запитання та завдання

1. Назвіть елементи побудови таблиць.
2. Поясніть особливості «гумової» табличної верстки сайту.
3. Поясніть особливості «жорсткої» табличної верстки сайту.
4. Назвіть призначення форм.
5. Як відправляються дані форми?
6. Які атрибути використовуються з елементом розмітки `<input>`?