ЛАБОРАТОРНА РОБОТА №3.
Ознайомлення з мовою розмічування HTML

ЗАВДАННЯ

1. Створити web-сторінку (на якусь визначену вами тему), що містить декілька абзаців тексту, мінімум 2 рівня заголовків, зображення, таблицю, усі 3 види списків, посилання на другу сторінку (див п.2) та якийсь зовнішній сайт.
2. Створити ще одну web-сторінку (можна використовувати не всі перелічені елементи), пов‘язати її з першою (посилання повинні працювати у обидва боки).

Обов’язкові вимоги: присутній правильно визначений DOCTYPE (XHTML 1.0 Strict (бажано) або HTML5), явно визначене кодування, html-документи (локальні файли!) проходять перевірку валідатором на сайті W3C: http://validator.w3.org/.

ДОПОМІЖНА ІНФОРМАЦІЯ

СИНТАКСИС HTML

Крім звичайного тексту, HTML-файл утримує лише один тип керівних конструкцій – так звані теги (tags). Теги розділяють вихідний неформатований текст на елементи або створюють нові елементи, до яких у тексті немає відповідності (наприклад, графічні вставки). Відповідно, теги бувають двох видів: парні (котрі охоплюють якийсь фрагмент тексту) і непарні.

Приклад парного тега: `<strong>Елемент для парного тега</strong>`

Приклад непарного тега: `<br />`

Теги можуть вкладатися один в один, але без перетинів: **<c><d> … </d></c>**

Якщо браузер зустрічає невідомий для нього тег, то він його просто ігнорує. Отже, треба слідкувати за правильністю написання всіх службових слів.

Більшість тегів можуть мати атрибути, які змінюють і уточнюють дію тега:

```
<тег  атрибут1=”значення1”  атрибут2=”значення2”  ...>
```

У більшості випадків атрибути є необов’язковими і, якщо вони не задані, то інтерпретатор HTML використовує значення за умовчанням.

Правила сумісності із стандартом XHTML:

1. Всі імена тегів і атрибутів пишуться маленькими літерами
2. Всі теги обов’язково мають закриватися: парні – звичайним способом, непарні: **<… />** (пробіл перед косою рискою обов’язковий з міркувань оберненої сумісності із старими браузерами)
3. Значення атрибутів завжди існують і завжди записуються в лапках.
4. Використання застарілих і специфічних для конкретного браузера елементів не дозволяється; все можливе оформлення виноситься у таблиці стилів.

Синтаксис задання коментарів: `<!-- це коментар -->`

Для того, щоб відобразити на сторінці спеціальні символи, слід використовувати синтаксис **&code;** - наприклад: **<** - знак < ; **>** - знак > ; ** ** - нерозривний пробіл; **©** - знак © ; **&** - знак **&**.

Схема DTD (Document Type Definition – визначення типу документа) є головним документом, який ідентифікує і визначає елементи, атрибути та інші засоби даної специфікації. У стандартах HTML 4.01 і XHTML 1.0 існували три схеми DTD – строга (Strict), перехідна (Transitional) і для опису фреймів (Frameset). Сучасний стандарт HTML5 не передбачає таких розбіжностей.

Перевірка на дійсність (validation) – процес **автоматичної** перевірки відповідності документа оголошеній для нього схемі DTD. Деякі браузери не відображають недійсні документи!

СТРУКТУРА ДОКУМЕНТА HTML

Будь-який документ HTML складається з 3 частин: інформації про версію HTML, блока заголовка документа і тіла документа (яке утримує безпосередньо інформацію, що відображається у вікні).

Мінімальний документ HTML5:

```
<!DOCTYPE HTML>
<html lang="uk">
<head>
      <title> Мінімальний документ HTML5 </title>
      <meta charset="utf-8" />
</head>
<body>
        <p> Цей документ є дійсним в HTML5 </p>
</body>
</html>
```

Мінімальний документ за стандартом XHTML 1.0 Strict:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="uk" lang="uk">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
  <title> Мінімальний документ XHTML 1.0 </title>
</head>
<body>
  <p> Цей документ є дійсним в XHTML 1.0 </p>
</body>
</html>
```

Розділ документу HEAD може утримувати інформацію про поточний документ – таку, як назва, основа для локальних посилань, посилання на зовнішні сценарії і таблиці стилів. Більшою мірою, інформація, розміщена у цьому розділі, є службовою, і тому браузери не повинні її відображати (крім назви документу).

Зокрема, елемент meta надає можливість надавати допоміжні інструкції як браузеру, так і серверу, на якому розміщена сторінка.

Приклади.

- визначення кодування у випадку використання кирилиці під Windows (для HTML5-документів такий, довгий, формат теж можна використовувати, він еквівалентний наведеному у прикладі)
  `<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">`
- зазначення дати „терміну використання” документа (після цієї дати документ буде примусово завантажуватися з сервера, а не з внутрішньої пам’яті браузера)
  `<meta http-equiv="Expires" content="Mon, 01 Jan 1990 00:00:01 GMT">`

Розділ документу BODY утримує зміст документа, який виводиться у робочу область браузера. Всі елементи, які містяться всередині цього розділу, поділяються на елементи рівня тексту і елементи рівня блока (утворюють більш глобальну структуру, за умовчанням їхній вміст виводиться з нового рядка, а між такими елементами браузер пропускає вільний рядок).

ФОРМАТУВАННЯ ТЕКСТУ

Коли йдеться про розмічування тексту, слід розуміти різницю між структурним (логічним) і візуальним (фізичним) форматуванням.

Теги логічного форматування тексту призначені для розстановки логічних наголосів, виділення логічних частин і підкреслювання суті сказаного, наприклад (усі наведені тут елементи – рівня тексту, усі теги – парні):

- **<abbr>** - відмічає свій текст як **абревіатуру**;

- **<cite>** - використовується для **цитат** або **посилань на інші джерела**. Зазвичай виводиться курсивом;
- **<code>** - відмічає **невеликий фрагмент програмного коду**. Зазвичай виводиться моноширинним шрифтом. Не плутати з елементом **<pre>** рівня блоку!
- **<dfn>** - позначає текстовий фрагмент як **означення**, використовується при першому введенні терміну. Зазвичай виводиться курсивом;
- **** - для **виділення** (emphasis) важливих фрагментів тексту. Зазвичай виводиться курсивом;
- **<kbd>** - помічає текст як такий, що **вводиться з клавіатури**. Зазвичай виводиться моноширинним шрифтом;
- **<q>** - відмічає **короткі цитати** в рядку тексту. Має атрибут **cite** - джерело цитати;
- **<samp>** - помічає текст як **зразок** (sample). Використовується для тексту, який видається програмами (sample output). Зазвичай виводиться моноширинним шрифтом;
- **** - Для виділення **дуже важливих** фрагментів тексту. Може виводитися напівжирним шрифтом або курсивом;
- **<var>** - позначає **імена змінних** програми. Зазвичай виводиться курсивом.

Теги фізичного форматування тексту (усі наведені тут елементи - рівня тексту, усі теги - парні).

- **<b>** - виводить свій текст **напівжирним шрифтом**. Краще використовувати **.**
- **<i>** - виводить свій текст **курсивом**. Краще використовувати ****, або **<dfn>**, або **<var>**, або **<cite>** - в залежності від контексту.
- **<tt>** - виводить свій текст **моноширинним шрифтом**. Краще використовувати **<code>**, або **<samp>**, або **<kbd>** - в залежності від контексту.
- **<big>** - виводить свій текст **більшим шрифтом**.
- **<small>** - виводить свій текст **меншим шрифтом**.
- **<sub>** - виводить свій текст **як верхній індекс**.
- **<sup>** - виводить свій текст **як нижній індекс**.

У стандарті HTML5 семантичний зміст деяких тегів фізичного форматування тексту переосмислений і стосується не лише візуального відображення.

Деякі теги є **відміненими, використовувати їх заборонено:**

- **<u>** - виводить свій текст підкресленим**.**
- **<strike>** або **<s>** - виводить свій текст закресленим.
- **<blink>** - примушує свій текст миготіти**.**
- **<font>** - застарілий спосіб оформлення тексту.
- **<basefont>** - застарілий спосіб визначення базового шрифта.

НАЙПРОСТІШІ ТЕГИ ДЛЯ ФОРМАТУВАННЯ ДОКУМЕНТА HTML:

- **<p>** (парний)- створює розподіл на абзаци;
- **
** - розрив рядка;
- **<h1>, <h2>, …, <h6>** (усі парні) – заголовки різних рівнів;
- **<hr />** - горизонтальна лінія;
- ** ** (парний, рівня **блоку**) – виділяє фрагмент документа з метою керування його представленням;
- **** (парний, рівня **тексту**) – дія аналогічна ** .**

ПОСИЛАННЯ

Посилання складається з 2 частин. Частина, яку ми бачимо на сторінці, називається **вказівником** (ним може бути слово, група слів або зображення), а друга частина дає інструкції браузеру і називається **адресною частиною** (URL-адресою). За замовчуванням текстові вказівники виділяються кольором і підкреслюються, а графічні беруться в рамку.

URL-адреса може бути **абсолютною** і **відносною** – абсолютна повністю визначає сервер, каталог і файл, а відносна визначає адресу відносно адреси документу із вказівником посилання (на тому ж сервері) або адреси, вказаної в **<base>**.

Для створення вказівників посилань використовується тег **<a>.** Значення його атрибуту **href** визначає адресну частину.

Приклади.

```
<a href=”http://www.univ.kiev.ua/”>Київський університет імені Тараса Шевченка </a>
```

```
<a href=”../../links/index.html”>Корисні посилання</a>
```
(тут .. позначає перехід до каталогу на рівень вище)

```
<a href=”mailto:vasyl@ukr.net”>Написати Василю листа</a>
```

У деяких випадках вказівник посилання може вказувати на частину того самого або іншого документу (як правило, конкретне місце задається значенням атрибуту **id** відповідного елемента)

Приклад. `<a href=”#chapter_5”>Глава 5, що знаходиться у цьому ж документі</a>.`

СПИСКИ

У мові HTML існують 3 види списків: невпорядкований, впорядкований і список означень.

1. Невпорядкований список (Unordered List, UL): для виділення елементів використовуються спеціальні символи – маркери списка.

Приклад.

```
<h1>Знаки Зодіаку</h1>
<ul>
  <li>Овен</li>
  <li>Телець</li>
  ….
  <li>Риби</li>
</ul>
```

2. Впорядкований список (Ordered List, OL): списки цього типу являють собою впорядковану перенумеровану послідовність елементів. Вид нумерації (латинські літери, латинські цифри, арабські цифри) залежить від браузера і може бути змінений.

Приклад.

```
<h1>Розклад занять на понеділок</h1>
<ol>
  <li>Математичний аналіз (лекція)</li>
  <li>Математичний аналіз (семінар)</li>
  <li>Фізкультура</li>
</ol>
```

3. Список означень (Definition List, DL): Кожний елемент списку означень складається з двох частин: у першій частині (Definition Term, DT) задається термін для визначення, у другій (Definition Description, DD) – текст, який визначає цей термін.

Приклад.

```
<dl>
  <dt>Протони, нейтрони</dt>
  <dd>Ті, що входять до складу ядра атома</dd>
  <dt>Ядро атома</dt>
  <dd>Об'єкта, який містить протони та нейтрони</dd>
</dl>
```

Списки можуть бути вкладеними, причому вкладатися один в один можуть списки різних типів (у списку означень внутрішній список можуть містити лише елементи <**dd**>).

ГРАФІЧНІ ЗОБРАЖЕННЯ (ІЛЮСТРАЦІЇ)

Для розміщення зображення на сторінці необхідно скористатись тегом **<img>**, обов'язковий атрибут якого **src** визначає адресу графічного файлу. Для виведення альтернативного тексту на місці зображення застосовують атрибут **alt** тегу **<img>.** При затримці завантаження зображення або не підтримці відображення

зображень браузером альтернативний текст обов'язково буде відображатись, тому використання атрибута **alt** є необхідним.

Приклад. `<img src="photo.jpg" alt=”фото автора” />`

Розміри зображення при відображенні можуть бути задані за допомогою атрибутів **width** або **height** тегу **<img>.** Як правило, значення цих атрибутів задають у пікселях. Якщо значення таких параметрів не співпадають із справжніми розмірами зображення, то браузер здійснює його перемасштабування. У зв'язку з цим необхідно зберігати пропорційність між справжніми розмірами зображення і вказаними розмірами для відображення збільшеної версії зображення (зменшені версії використовувати дуже не бажано, бо це спричиняє безцільне додаткове навантаження на мережу).

Зображення можна використовувати як вказівник посилання:

```
<a href="data.html"><img src="image1.gif"></a>
```

Найбільш популярні графічні формати, які підтримуються усіма сучасними графічними браузерами: JPG, PNG, GIF.

Визначаючи формат графічного файлу, в якому краще підготувати графічні матеріали, необхідно враховувати наступне:

1. Формат JPG більше підходить для підготовки наступних типів зображень:

- фотографії, фотореалістичні зображення, побудовані за допомогою програмних засобів;
- графіки і діаграми, що містять складне поєднання кольорів та відтінків;
- зображення, які потребують використання більш, ніж 256 кольорів.

2. Формат PNG (більш сучасний) або GIF більше підходить для підготовки наступних типів зображень:

- зображення з обмеженою кількістю кольорів;
- зображення, які мають чітку границю та краї, що характерно для більшості зображень елементів меню, кнопок, графіків, діаграм;
- зображення, до складу яких входить текст;

3. Формат GIF – єдиний, який підходить для підготовки зображень, що містять анімаційні графічні елементи (при відображенні серія рисунків змінюють один одного із заданим часовим інтервалом).

Незалежно від формату графічного файлу завжди необхідно слідкувати, щоб справжні геометричні розміри зображення узгоджувалися з максимальними розмірами вікна відображення даних файлу, інакше надлишковість інформації ускладнюватиме процес завантаження графіки чи навіть заважатиме відтворенню окремих елементів зображення.

ПРОСТІ ТАБЛИЦІ

Загальні правила

Ще одним способом представлення даних на Web-сторінках є таблиці. Таблиці HTML складаються з рядків, що містять клітинки з даними, і можуть бути лише прямокутними (тобто базова кількість клітинок у різних рядках повинна співпадати).

Керування засобами HTML параметрами рамок та роздільних ліній таблиці дозволяє візуально приховувати таблицю взагалі, розміщуючи, при цьому, дані у впорядкованому вигляді.

Для роботи з таблицями необхідно володіти методами роботи з основним набором тегів:

1. Тіло таблиці розпочинається тегом **<table>** і закінчується тегом **</table>**.
2. Заголовок таблиці описується всередині блоку **<caption>** … **</caption>**, який необхідно розмістити одразу після тегу **<table>**.
3. Початок і кінець рядка таблиці визначаються тегами **<tr>** та, відповідно **</tr>**. Групу тегів, яка утворює блок рядка таблиці, повторюють необхідну кількість разів, залежно від необхідної кількості рядків. Всі блоки **<tr>** … **</tr>** розміщуються всередині блоку **<table>** … **</table>**.
4. Початок і кінець клітинки таблиці визначаються тегами **<td>** та, відповідно **</td>**. Для формування клітинок першого рядка таблиці, в які поміщуються заголовки колонок можуть бути використані теги

<th> та, відповідно **</th>**. Група тегів початку та кінця клітинок розміщується всередині відповідного блоку опису рядка таблиці.

Приклад.

```
<table border="1">
  <tr>
    <th>Заголовок 1 колонки</th>
    <th>Заголовок 2 колонки</th>
    <th>Заголовок 3 колонки</th>
  </tr>
  <tr>
    <td>Дані в 1 комірці 1 рядка</td>
    <td>Дані в 2 комірці 1 рядка</td>
    <td>Дані в 3 комірці 1 рядка</td>
  </tr>
  <tr>
    <td>Дані в 1 комірці 2 рядка</td>
    <td>Дані в 2 комірці 2 рядка</td>
    <td>Дані в 3 комірці 2 рядка</td>
  </tr>
  <tr>
    <td>Дані в 1 комірці 3 рядка</td>
    <td>Дані в 2 комірці 3 рядка</td>
    <td>Дані в 3 комірці 3 рядка</td>
  </tr>
</table>
```

Об'єднання елементів таблиць

Специфіка поданих на Web-сторінці даних іноді вимагає об'єднання декількох сусідніх клітинок в одну. Для об'єднання сусідніх клітинок по горизонталі необхідно присвоїти кількість комірок атрибуту **colspan** для тегу **<td>** або **<th>**. Для об'єднання сусідніх клітинок по вертикалі необхідно присвоїти кількість клітинок атрибуту **rowspan** для тегу **<td>** або **<th>**.

Приклад. Таблиця з об'єднаними клітинками

```
<table border="1" width="100%">
  <tr>
    <td rowspan="2">Об'єднана комірка по вертикалі</td>
    <td colspan="3">Об'єднана комірка по горизонталі</td>
  </tr>
  <tr>
    <td>К.1</td>
    <td>К.2</td>
    <td>К.3</td>
  </tr>
</table>
```

Результат обробки коду:

<table>
<tr><td rowspan="2">Об'єднана комірка по вертикалі</td><td colspan="3">Об'єднана комірка по горизонталі</td></tr>
<tr><td>К. 1</td><td>К. 2</td><td>К. 3</td></tr>
</table>

При необхідності побудови складних таблиць часто застосовують не об'єднання окремих клітинок, а таблиці, вкладені в клітинки іншої таблиці. Загальні правила побудови зовнішньої та вкладеної таблиці однакові.