

Тема 3: Таблиці, мультимедіа та форми в коді HTML

3.1 Таблиці в коді HTML

Таблиці на веб-сторінках використовуються як з метою показу табличних даних, так і для підтримки макету сторінки. Довший час таблиці були основним засобом верстки веб-сторінок. Зараз цей спосіб здебільшого замінюється стильовим блоковим оформленням.

Таблиця у коді html описується по рядках, у кожному з яких описуються стовпці, утворюючи клітинки.

Таблиці

Таблицю охоплює тег **<table>**, рядки описує тег **<tr>**, а клітинки - тег **<td>**. Перший рядок таблиці може містити заголовки стовпців – це тег **<th>**.

Наприклад:

123
456
789

Код:

```
<table>
<tbody><th>
<td> 1</td>
<td> 2</td>
<td> 3</td>
</th>
<tr>
<td> 4</td>
<td> 5</td>
<td> 6</td>
</tr>
<tr>
<td> 7</td>
<td> 8</td>
<td> 9</td>
</tr>
</tbody></table>
```

Межі таблиці

За замовчанням таблиця не має меж. Атрибут **border** дозволяє вказати ширину меж клітинок.

Наприклад:

1	2	3
4	5	6
7	8	9

Код:

```
<table border=3>
<tbody><th>
<td> 1</td>
<td> 2</td>
<td> 3</td>
</th>
<tr>
<td> 4</td>
<td> 5</td>
<td> 6</td>
</tr>
<tr>
<td> 7</td>
<td> 8</td>
<td> 9</td>
</tr>
</tbody></table>
```

Колір

Атрибут **bgcolor** може задавати колір як тла усієї таблиці, так і окремих рядків та клітинок.

Наприклад:

1	2	3
4	5	6
7	8	9

Код:

```
<table border=3>
<tbody bgcolor="yellow"><th bgcolor="lightgreen">
<td> 1</td>
<td> 2</td>
<td> 3</td>
</th>
<tr>
<td bgcolor="lightblue"> 4</td>
<td> 5</td>
<td> 6</td>
</tr>
<tr>
<td> 7</td>
<td> 8</td>
<td> 9</td>
</tr>
</tbody></table>
```

Об'єднання клітинок

Клітинки таблиці можна об'єднувати за допомогою атрибутів **rowspan** та **colspan**, вони вказують на скільки рядків чи стовпців відповідно простягаються клітинки.

Наприклад:

1	2	
4	5	6
	8	9

Зверніть увагу на об'єднання клітинок: фактично об'єднання клітинок 2 та 3 означає, що клітинка 3 стає непотрібною, тож її описувати не потрібно. Аналогічно клітинки 4 та 7.

Код:

```
<table border=3>
<tbody><th>
<td> 1</td>
<td colspan="2"> 2</td>
</th>
```

```

<tr>
<td rowspan="2"> 4</td>
<td> 5</td>
<td> 6</td>
</tr>
<tr>
<td> 8</td>
<td> 9</td>
</tr>
</tbody></table>

```

Ширина

Ширина таблиці та її окремих клітинок визначається вмістом клітинок. Атрибут **width** дозволяє вказати ширину клітинки (чи таблиці) в пікселях чи у відсотках.

Наприклад:

1	2	3
4	5	6
7	8	9

Таблиця загалом має ширину 300 пікселів, перший стовпець – 100 пікселів, а другий – 50% від загальної ширини таблиці.

Код:

```

<table border=3 width="300">
<tbody><th>
<td width="100"> 1</td>
<td width="50%"> 2</td>
<td> 3</td>
</th>
<tr>
<td> 4</td>
<td> 5</td>
<td> 6</td>
</tr>
<tr>
<td> 7</td>
<td> 8</td>
<td> 9</td>
</tr>
</tbody></table>

```

Вирівнювання

Вміст клітинок та таблицю загалом можна вирівнювати через атрибут **align** (center, left, right, justify) та по вертикалі **valign** (top, bottom, middle).

Наприклад,

1	2	3
4	5	6
7	8	9

У поданій таблиці другий рядок вирівняно за правим краєм, клітинки 1 та 8 – за центром. Сама таблиця теж вирівнюється за центром сторінки.

Код:

```
<table border=3 width="300" align="center">
<tbody><th>
<td width="100"> 1</td>
<td width="50%"> 2</td>
<td> 3</td>
</th>
<tr align="right">
<td> 4</td>
<td> 5</td>
<td> 6</td>
</tr>
<tr>
<td> 7</td>
<td align="center"> 8</td>
<td> 9</td>
</tr>
</tbody></table>
```

Таблична розмітка

Оскільки таблицю можна розмістити всередині іншої таблиці, і при цьому достатньо гнучко задавати вирівнювання вмісту в них, такий метод досить широко використовувався для верстки макетів сторінок.

Наприклад,

Меню 1	Інформація сайту
Меню 2	
Меню 3	
Меню 4	

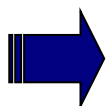
На сторінці ліворуч передбачена область меню, яка має ширину 100 пікселів, а решту простору займає інформаційне наповнення сторінки. Оскільки межі для вкладеної таблиці не задавались, користувач може навіть не помітити цієї таблиці, а використання колірної заливки дозволяє створити довільний дизайн такої сторінки.

Код:

```
<table border=3 width="300">
<tbody>
<tr>
<td width="100">

<table width="100%">
<tbody>
<tr><td>Меню 1</td></tr>
<tr><td>Меню 2</td></tr>
<tr><td>Меню 3</td></tr>
<tr><td>Меню 4</td></tr>
</tbody>
</table>

</td>
<td> Інформація сайту </td>
</tr>
</tbody>
</table>
```



Довідник тегів та налаштувань: <https://css.in.ua/html/tag/table>.

3.2 Мультимедіа в коді HTML5

Мультимедіа на веб-сторінках тривалий час завдавало значних труднощів веб-розробникам через різноманіття форматів та їх некоректну підтримку браузерами. Із появою стандарту HTML5 ці труднощі дещо зменшились, оскільки запропоновано кілька стандартних універсальних тегів для додавання мультимедійних об'єктів.

Це теги **audio** та **video**.

Вони мають таку структуру:

```
<video width="400" controls>  
<source src="mov_bbb.mp4" type="video/mp4">  
<source src="mov_bbb.ogg" type="video/ogg">  
Ваш браузер не підтримує теги відео HTML5. </video>
```

Такий тег додає на сторінку відео-програвач, шириною 400 пікселів, з елементами керування (кнопки play, pause та volume). Можна також задати атрибут **autoplay** для того, щоб відтворення відео розпочиналось автоматично.

Варто зауважити, що у тезі вказано кілька файлів для відтворення – спочатку **mov_bbb.mp4**, а тоді **mov_bbb.ogg**. Браузер спробує відтворити перший вказаний файл, і у випадку невдачі (наприклад, через відсутність кодеків), відтворюватиметься другий і т.д. Якщо ж засобів відтворення мультимедіа поданих типів не знайдеться, на місці відео-плеєра буде текст, наведений перед закриваючим тегом **</video>**.

Аналогічно працює тег для звуку:

```
<audio controls>  
<source src="horse.ogg" type="audio/ogg">  
<source src="horse.mp3" type="audio/mpeg">  
Ваш браузер не підтримує теги аудіо-елементів.</audio>
```

Існують також теги для додавання на сторінку інших об'єктів.

Так, тег **object** можна використовувати для додавання флеш-роликів, інших веб-сторінок чи зображень.

```
<object width="400" height="50" data="bookmark.swf"> </object>  
<object width="100%" height="500px" data="snippet.html"> </object>  
<object data="audi.jpg"></object>
```

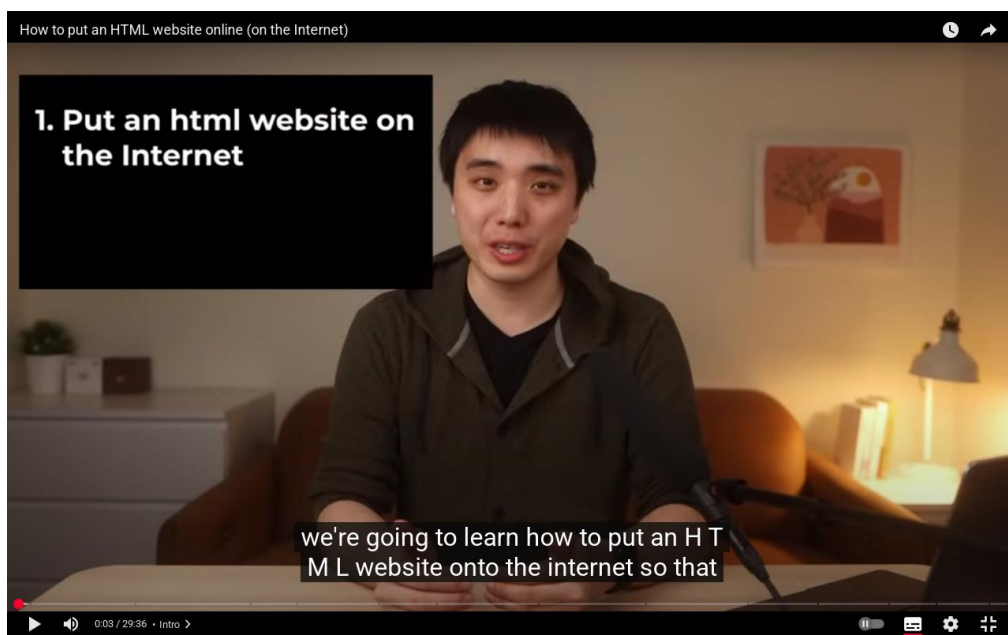
Тег **embed** використовується з аналогічною метою.

```
<embed width="400" height="50" src="bookmark.swf">  
<embed width="100%" height="500px" src="snippet.html">  
<embed src="audi.jpg">
```

Наприклад, відео-ролик з YouTube можна додати на сторінку одним із наведених способів:

```
<iframe width="420" height="315" src="https://youtu.be/p1QU3kLFPdg"> </iframe>
```

```
<object width="420" height="315" data="
```



```
"> </object>
```

```
<embed width="420" height="315" src="https://youtu.be/ p1QU3kLFPdg ">
```

При цьому різні браузери можуть відображати різні теги, зокрема мультимедійні, по-різному, тож остаточний результат необхідно тестувати у різних браузерах та передбачати відображення відповідних замінників при потребі.

3.3 Форми

Форми на веб-сторінках є засобом організації зворотного зв'язку із відвідувачем сторінки.

Форма містить запитання, поля, які можна заповнити та надіслати для подальшої обробки на певну електронну скриньку чи у базу даних на веб-сервері. Власник сайту згодом може опрацювати результати відповідей користувачів.

Форми розміщуються між тегами **<form>** та **</form>**, і можуть містити довільну кількість полів-запитань.

Кожне із питань оформлюється тегом **<input>** певного типу: текст **text**, пароль **password**, електронна скринька **email**, веб-адреса **url**, число **number** (існує також варіант діапазону **range**), поле пошуку **search**, вибір кольору **color**, прикріплений файл **file** тощо. Для введення дати передбачено цілий ряд типів: **date**, **month**, **week**, **time**, **datetime**, **datetime-local**. Існують питання із вибором відповіді з поданого переліку: це типи прапорців **checkbox**, перемикачів **radio**, а також випадне меню **<select><option>**.

Перевагою такого різноманіття типів питань є те, що відбувається так звана **валідація**, тобто перевірка даних, що вводяться у поле відповіді. Так, якщо передбачене введення електронної скриньки, то запис без символу **@** не буде прийнятий при надсиланні форми.

Дія, яку потрібно виконати над заповненою формою вказується у атрибуті тегу **form action**.

Найчастіше там можна зустріти адресу скрипту, який буде обробляти дані форми, або адресу електронної скриньки, на яку слід надіслати ці дані. Наприклад, при такому записі, заповнену форму буде передано у файл **handler.php**

<form action="handler.php" method="post">

Варто зазначити, що передача даних може відбуватись одним із двох методів: **get** та **post**.

Метод **get** застосовується для передачі невеликих текстових даних та пошуку на сайті. В такому разі зазвичай у рядку адреси відбуваються зміни і можна побачити доданий знак питання з подальшими символами. Це власне запит до сервера, який надсилається формою. Такий запит у явному вигляді доступний відвідувачеві сайту.

Метод **post** використовується для пересилки файлів, коментарів, додавання повідомлень на форумах тощо. Адреса у відповідному рядку браузера не змінюється, запит передається у більш захищеному режимі.



Дію стосовно відправлення заповненої форми можна також записати в кнопці - це тип поля **input submit** :

<input type="submit" value="Надіслати" formaction="handler.php" formmethod="post">

Приклад форми

Розглянемо приклад форми.

Вкажіть ваше ім'я		<input type="text">
Вкажіть пароль		<input type="password">
Вкажіть електронну скриньку	<input type="email"/>	<input type="email">
Вкажіть адресу веб-сайту	<input type="url"/>	<input type="url">
Вкажіть число	<input type="number"/>	<input type="number">
Вкажіть число з діапазону	<input type="range"/>	<input type="range">
Вкажіть пошуковий запит	<input type="search"/>	<input type="search">
Вкажіть колір	<input type="color"/>	<input type="color">
Вкажіть дату	<input type="date"/>	<input type="date">
		<select>
		<option>Варіант 1</option>
		<option>Варіант 2</option>
		<option>Варіант 3</option>
		</select>
Виберіть з переліку:	Варіант 1	

Обравши для форми бажані елементи, залишається налаштувати спосіб та метод її надсилання (у тезі <**form**> чи у кнопці <**input type="submit"**>). Саме опрацювання відповідей з форми відбувається на серверній стороні веб-сайту, і не розглядається у цьому курсі.

Якщо розмістити веб-сторінку із формою на хостингу, то наступний скрипт надсилатиме інформацію з форми на електронну скриньку `newbie0@ukr.net` (вказано у змінній `sendto`). Із форми береться інформація `firstname`, `lastname`, `quantity` (передані методом POST). Крім того, задається тема повідомлення у змінній `subject`.

Скрипт:

```
<?php
$sendto = 'newbie0@ukr.net';
$subject = 'Email from web-site';
$firstname = $_POST['firstname'];
$lastname = $_POST['lastname'];
$quantity = $_POST['quantity'];
$content = "Name: " . $firstname . "\n";
$content .= "Email: " . $lastname . "\n\n";
$content .= $quantity;
mail($sendto,$subject,$content);
?>
```

Для того, щоб подібна інформація передавалась коректно, елементам форми необхідно надавати змістових імен (**id** – для керування елементом форми за допомогою локального скрипту JavaScript та **name** – для передачі на сервер). Крім того кожен елемент форми може мати атрибут обов'язкового заповнення **required** та значення за умовчанням **value**. Атрибут **title** означає підказку, яка з'являється при наведенні вказівника миші на елемент форми або при спробі надіслати некоректні дані.

<input type="text" name="firstname" id="firstname" value="Mickey" title="Потрібно вказати ім'я" required>

тип назва ідентифікатор значення за умовчанням підказка обов'язковість

Для таких елементів форми як прапорець (**checkbox**) та перемикач (**radio**) потрібно вказувати однакову назву **name** для варіантів, котрі стосуються одного запитання. Так, у першому прикладі назви перемикачів різні: `name="variant"` та `name="variant1"` і можна обрати обидва варіанти відповіді.

```
<input type="radio" name="variant1" id="variant1" value="variant1">Варіант 1  
<input type="radio" name="variant2" id="variant2" value="variant2">Варіант 2
```

- ☐ Варіант 1
- ☐ Варіант 2

У другому прикладі назва в обох елементів однакова name="variant", тож можна обрати лише один варіант відповіді

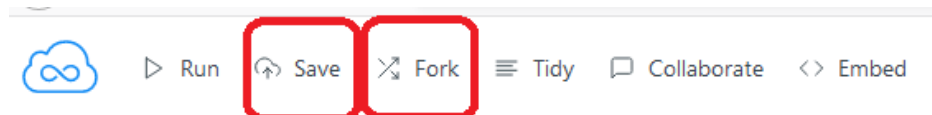
```
<input type="radio" name="variant" id="variant1" value="variant1">Варіант 1  
<input type="radio" name="variant" id="variant2" value="variant2">Варіант 2
```

- ☐ Варіант 1
- ☐ Варіант 2

ВПРАВИ ТА ПРАКТИЧНІ ЗАВДАННЯ ДО ВИКОНАННЯ

1. Практичне завдання «Табель»:

Заготовка для завдання доступна за посиланням: <https://jsfiddle.net/vleonteva/1qbgau7v/>. Натиснувши **Fork** на панелі інструментів можна продовжити роботу із заготовкою на власному комп'ютері.



У цьому проєкті потрібно створити таблицю за зразком:

Прізвище та ім'я учня	1 семестр		2 семестр	
	Тема 1	Тема 2	Тема 3	Тема 4
Іваненко Іван	11.5	10.85	9.56	10.25
Петренко Софія	8.75	9.45	7.58	8.27
Лісничка Олена	10.67	9.58	8.38	9.43
Ковалюк Василь	9.12	7.89	8.67	8.94
Горинь Антон	7.58	8.15	9.62	8.37

Запишіть прізвища та імена студентів своєї групи. Бали можна проставити довільні.

Після виконання завдання клацніть **Save** і завантажте посилання (з рядка адреси) на результат роботи.

Корисні посилання:

Відео з курсу Prometheus: Цифрові комунікації в глобальному просторі
https://edx.prometheus.org.ua/courses/course-v1:Prometheus+ITArts101+2017_T1