

Списки і таблиці

План:

1. Визначення та види списків, їх оформлення
2. Оформлення, використання і особливості таблиць.
3. Створення форм засобами HTML
4. Використання фреймів у HTML

Визначення та види списків, їх оформлення

Списки – це взаємопов'язаний набір даних, які можуть бути представлені у вигляді впорядкованого списку (ordered list) або неупорядкованого списку (unordered list).

Вони використовуються для систематизації та структурування контенту на веб-сторінках, щоб зробити його більш легким для сприйняття користувачами.

HTML списки можуть містити різні типи елементів, такі як текст, зображення, посилання, таблиці тощо, і можуть бути вкладені один в одного, що дозволяє створювати більш складні структури.

У HTML є три типи списків:

Неупорядкований список (unordered list) – це список, що складається з маркованих елементів, які можуть бути позначені символами, наприклад крапкою, кружечком або квадратиком.

Ці елементи не мають порядкових номерів, тому вони відображаються на сторінці у вигляді маркованого списку. Для створення невпорядкованого списку використовують тег `ul`, в який додають елементи списку за допомогою тегу `li`.

```
<ul>
```

```
<li>Перший елемент</li>
```

```
<li>Другий елемент</li>
```

```
<li>Третій елемент</li>
```

```
</ul>
```

Впорядкований список (ordered list) – це список, в якому кожен елемент має порядковий номер. Зазвичай для відображення порядкових номерів використовуються арабські цифри, але можуть бути використані також римські цифри, літери або рядкові числа.

Для створення впорядкованого списку використовують тег `ol`, в який додають елементи списку за допомогою тегу `li`.

```
<ol>
```

```
<li>Перший елемент</li>
```

```
<li>Другий елемент</li>
```

```
<li>Третій елемент</li>
```

```
</ol>
```

Список з описом (description list) – це список, що складається з термінів та їхніх описів. Термін – це слово або фраза, що потребує пояснення, а опис – пояснення до терміну.

Для створення списку з описом використовують тег `dl`, в який додають терміни за допомогою тегу `dt`, а їхні описи – за допомогою тегу `dd`.

```
<dl>
```

```
<dt>HTML</dt>
```

```
<dd>Мова розмітки гіпертексту.</dd>
```

```
<dt>CSS</dt>
```

```
<dd>Мова опису стилів, що використовується для оформлення веб-сторінок.</dd>
```

```
</dl>
```

Оформлення, використання і особливості таблиць

Таблиця – це розташування даних у вигляді сітки з рядків і стовпців. Таблиці використовуються для представлення даних, які потребують організації у вигляді табличної форми, наприклад, графіки, переліки товарів і їхніх характеристик, розклади та інше.

Основною метою таблиць в HTML є відображення даних в наочному та структурованому вигляді. Таблиці можуть містити будь-який тип даних, такий як текст, числа, зображення, гіперпосилання та інше. Вони дозволяють користувачам легко знаходити необхідну інформацію та порівнювати дані.

Таблиці складаються з рядків і стовпців. Рядки в таблиці представляють собою елемент `tr` (table row), а стовпці – елемент `td` (table data).

Таблиці можуть містити заголовки, які розміщуються в елементі th (table header).

Таблиці можуть мати межі (border), фоновий колір (background color) та інші атрибути, які визначають їх вигляд.

Для оформлення таблиць можна використовувати CSS стилі. Таблиці можуть бути вкладені одна в одну.

Приклад коду таблиці в HTML:

```
<table border="1">

  <thead>

    <tr>

      <th>Назва товару</th>

      <th>Ціна</th>

      <th>Кількість</th>

    </tr>

  </thead>

  <tbody>

    <tr>

      <td>Яблука</td>

      <td>20 грн</td>

      <td>10 шт</td>

    </tr>

    <tr>

      <td>Груші</td>
```

```
<td>25 грн</td>  
<td>5 шт</td>  
</tr>  
</tbody>  
</table>
```

Назва товару	Ціна	Кількість
Яблука	20 грн	10 шт
Груші	25 грн	5 шт

Заголовок таблиці (<caption>) – це елемент, який дозволяє додавати заголовок до таблиці. Заголовок таблиці розміщується над самою таблицею і використовується для зазначення її загального змісту.

Основною метою використання заголовка таблиці є забезпечення додаткової інформації про зміст таблиці. Він може бути корисним для тих користувачів, які використовують скрінрідери або переглядають сторінку на мобільних пристроях.

Заголовок таблиці додається за допомогою елемента <caption> та розміщується після тегу <table> і перед першим рядком таблиці.

Заголовок таблиці може містити будь-який текст або HTML-код, який може бути відформатований за допомогою CSS стилів.

Заголовок таблиці може містити інші HTML-елементи, такі як зображення, посилання та інше.

Заголовок таблиці може бути відсутнім. У такому випадку перший рядок таблиці вважається заголовком.

Атрибути таблиць

Атрибут	Опис
border	Встановлює товщину межі між комірками таблиці.
cellpadding	Встановлює відстань між вмістом комірки і її межею.
cellspacing	Встановлює відстань між комірками таблиці.
width	Встановлює ширину таблиці.
height	Встановлює висоту таблиці.
align	Встановлює горизонтальне вирівнювання таблиці.
bgcolor	Встановлює фоновий колір таблиці.
summary	Надає опис таблиці.
frame	Встановлює вид ліній, що відображають межі таблиці.
rules	Встановлює вид ліній, що відображають межі комірок таблиці.
caption	Додає заголовок до таблиці.
class	Додає клас до таблиці для застосування стилів CSS.
id	Додає ідентифікатор до таблиці для застосування стилів CSS або для взаємодії з JavaScript.
style	Додає стилі CSS до таблиці.
dir	Встановлює напрямок тексту таблиці.
lang	Встановлює мову тексту таблиці.
title	Додає підказку до таблиці.
cols	вказує кількість стовпців в таблиці

Атрибути комірок таблиць

Атрибут	Опис
colspan	Встановлює кількість комірок, які має охоплювати даний елемент.

rowspan	Встановлює кількість рядків, які має охоплювати даний елемент.
headers	Вказує ідентифікатори заголовків таблиці, які стосуються даних у комірці.
scope	Встановлює зону видимості заголовка для даних у комірці.
abbr	Вказує скорочення для даних у комірці.
valign	Встановлює вертикальне вирівнювання даних у комірці.
align	Встановлює горизонтальне вирівнювання даних у комірці.
nowrap	Відключає перенесення тексту в комірці на новий рядок.
bgcolor	Встановлює фоновий колір для даних у комірці.
class	Додає клас до комірки для застосування стилів CSS.
id	Додає ідентифікатор до комірки для застосування стилів CSS або для взаємодії з JavaScript.
style	Додає стилі CSS до комірки.
dir	Встановлює напрямок тексту у комірці.
lang	Встановлює мову тексту у комірці.
title	Додає підказку до даних у комірці.

Приклад використання атрибутів таблиці:

```
<table border="1" cellpadding="10" cellspacing="0" width="100%">
  <caption>Приклад таблиці</caption>
  <thead>
    <tr>
      <th>Ім'я</th>
      <th>Прізвище</th>
      <th>Вік</th>
    </tr>
  </thead>
```

```
<tbody>

<tr>

<td>Анна</td>

<td>Петренко</td>

<td>25</td>

</tr>

<tr>

<td>Олексій</td>

<td>Іваненко</td>

<td>32</td>

</tr>

</tbody>

</table>
```

Створення форм засобами HTML

Форма є взаємодіючим елементом, що дозволяє користувачам взаємодіяти з веб-сторінками, надсилаючи дані на сервер для обробки. Форма складається з одного або кількох полів вводу, які можуть містити різні типи даних, такі як текст, числа, дати, вибрані значення зі списку, файлів та ін.

Основна мета форм в HTML полягає у зборі інформації від користувачів та надсилання її на сервер для обробки або зберігання.

Форми можуть використовуватися для збору різних типів даних, від коментарів користувачів до замовлень товарів або послуг.

Форма має атрибут `action`, який вказує адресу сторінки на сервері, до якої будуть надіслані дані після відправки форми.

Форма може мати різні методи відправки даних, такі як `GET` або `POST`, які вказують, як будуть надіслані дані.

Поля форми можуть мати різні типи, такі як `text`, `password`, `number`, `email`, `checkbox`, `radio`, `select`, `textarea` та інші.

Кожен елемент поля форми має свій власний атрибут `name`, який ідентифікує його при надсиланні даних на сервер.

Форма може мати кнопки для відправки даних, відміни або скидання полів форми. Форми можуть мати різні атрибути для визначення обов'язковості заповнення поля, валідації даних, встановлення мінімального та максимального значення та ін.

Приклад форми в HTML:

```
<form action="/submit" method="post">
```

```
<label for="name">Ім'я:</label>
```

```
<input type="text" id="name" name="name" required>
```

```
<label for="email">Email:</label>
```

```
<input type="email" id="email" name="email" required>
```

```
<label for="message">Повідомлення:</label>
```

<textarea id="message" name="message" required></textarea>

<input type="submit" value="Ві

Теги форм та їхні атрибути

Тег	Атрибути	Опис
<form>	action, method, name, target, autocomplete, enctype, novalidate	Визначає властивості форми
<input>	type, name, value, placeholder, required, min, max, step, readonly, disabled, checked, autofocus	Визначає властивості поля введення
<label>	for	Вказує на ідентифікатор пов'язаного елемента форми
<textarea>	name, rows, cols, placeholder, required, readonly, disabled	Визначає властивості поля для введення довгого тексту
<select>	name, size, multiple, required, disabled	Визначає властивості списку для вибору значення
<option>	value, selected, disabled	Визначає властивості елемента у списку <select>
<optgroup>	label, disabled	Визначає властивості групи елементів у списку <select>
<button>	type, name, value, form, disabled, autofocus	Визначає властивості кнопки для надсилання форми або виконання скрипту

<fieldset>	disabled	Визначає, чи є група елементів форми доступною для користувача
<legend>	-	Визначає заголовок для групи елементів у <fieldset>
<datalist>	id	Визначає ідентифікатор списку з варіантами для поля <input> з атрибутом list
<keygen>	name	Визначає ім'я поля для генерації пар ключів для шифрування даних форми
<output>	for, name	Вказує на ідентифікатор пов'язаного елемента форми або визначає ім'я поля для виведення результатів форми або результатів скрипту.

Приклад використання форми з декількома атрибутами:

```
<form action="/submit-form" method="POST" name="my-form"
target="_blank" autocomplete="on" novalidate>
```

```
<label for="name-input">Ім'я:</label>
```

```
<input type="text" name="name" id="name-input" required>
```

```
<label for="email-input">Email:</label>
```

```
<input type="email" name="email" id="email-input" required>
```

```
<label for="message-input">Повідомлення:</label>
```

```
<textarea name="message" id="message-input" rows="5"
cols="30"></textarea>
```

```
<label for="subscribe-input">Підписка на новини:</label>
```

```
<input type="checkbox" name="subscribe" id="subscribe-input"
checked>
```

```
<button type="submit" name="submit-btn"
form="my-form">Надіслати</button>
</form>
```

Використання фреймів у HTML

Фрейми – це технологія, яка дозволяє розділити веб-сторінку на кілька окремих вікон або фреймів, кожен з яких містить свій вміст. Фрейми дозволяють вбудовувати одну веб-сторінку в іншу і створювати складні макети веб-сайту, які складаються з різних частин.

Основна мета фреймів – це полегшити створення складних веб-сторінок, що містять багато різних елементів, наприклад, меню, навігацію, контент, бокові панелі тощо. За допомогою фреймів можна розділити сторінку на окремі області, які можна заповнити різним вмістом.

Фрейми оголошуються за допомогою тега <frameset>, який вказує, як розбити сторінку на окремі фрейми.

Кожен фрейм визначається за допомогою тега <frame>, який вказує, який вміст потрібно відображати в кожному фреймі.

Інший спосіб використання фреймів полягає в тому, щоб вбудовувати одну веб-сторінку в іншу за допомогою тега <iframe>. Він дозволяє

вбудовувати зовнішній вміст в поточну веб-сторінку, наприклад, відео, карти, форми тощо.

Фрейми дозволяють динамічно оновлювати вміст окремих фреймів без перезавантаження всієї сторінки.