

Internet- технології

Мова розмітки HTML.
Тіло документу

Зміст

```
<!DOCTYPE html>
<html>
<head>
  <title>Назва сторінки</title>
  <meta charset="utf-8">
</head>
<body>
  <h2><i>Зміст</i></h2>
  <ul>
    <li>Блокові та строкові елементи</li>
    <li>Форматування тексту</li>
    <li>Списки</li>
    <li><table border="1"><tr><th>Таблиці</th><th>Табличне та семантичне
верстання</th></tr></table></li>
    <li><a href="#">Гіперпосилання</a></li>
    <li>Фрейми</li>
  </ul>
</body>
</html>
```

Зміст

- Блокові та строкові елементи
- Форматування тексту
- Списки
- | | |
|---------|----------------------------------|
| Таблиці | Табличне та семантичне верстання |
|---------|----------------------------------|
- [Гіперпосилання](#)
- Фрейми

Рекомендовані джерела

- Покроковий розбір типового HTML-документа <https://habr.com/ru/company/macloud/blog/555082/> та <https://github.com/joshbuchea/HEAD>
- Аналіз табличного верстання: <https://web-standards.ru/articles/css-techniques-p1-tables/>
- Про семантичну верстку <https://habr.com/ru/company/htmlacademy/blog/546500>
- Using ARIA: Roles, states, and properties https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques
- Перевірка гіперпосилань: <https://validator.w3.org/checklink>
- Документація щодо елементів HTML <https://developer.mozilla.org/en-US/docs/Web/HTML>

Блокові та строкові елементи

Строковими (рядковими) називаються елементи, які є частиною іншого елемента і використовуються, як правило, для зміни відображення частини тексту або його логічного форматування. Строкові елементи можна створити, наприклад, за допомогою тегів `<a>`, ``, `<code>`.

Блоковий елемент займає всю доступну ширину, висота елемента визначається його вмістом, і він завжди починається з нового рядка.

Блокові елементи можна вкладати один в одного, а також можна в блоковий елемент вкласти строкові.

До блокових елементів відносяться теги-контейнери `<div>`, заголовки `<h1>`-`<h6>`, форми та інші.

```
<div> <p> Some text </p> </div>
```

```
<h2> <a href="#"> Link </a> </h2>
```

Блокові та строкові елементи

Важливо враховувати, із чого складається розмір блоку.

Ширина блоку обчислюється автоматично і займає весь доступний простір, тобто 100%.

Якщо висота блочного елемента не задана явно, то обчислюється виходячи з обсягу вмісту.

Розміри блоку включають розміри самого блоку, зовнішніх відступів, внутрішніх відступів (полів) і границь блоку.

Якщо помістити один тег `<div>` всередину іншого, ширина внутрішнього тега починає розраховуватися відносно його «батька».



Блокові та строкові елементи

Різниця між блоковими та строковими елементами:

- Рядкові елементи можуть містити лише дані та інші рядкові елементи.
- Блокові елементи розміщуються з нового рядка, строкові такого ефекту не мають.
- Ширина рядкових елементів визначається за їх вмістом, відступами та межами.

Існують також елементи, які не можна віднести до блокових або рядкових елементів – рядково-блокові.

В даний момент різниця між рядковими та блоковими елементами зазвичай достатньо умовна, оскільки за допомогою CSS (каскадних таблиць стилів) можна цю характеристику змінювати.

Форматування тексту

Текст може бути форматований таким чином:

- `bold`,
- `bold and important`,
- `<i>italics</i>`,
- `italics and important`,
- `<small>small text</small>`,
- `deleted text`,
- `<ins>inserted text</ins>`,
- `<u>underlined</u>`,
- `<s>crossed out</s>`,
- `^{text above line}`,
- `_{text under line}`.

bold,
bold and important,
italics,
italics and important,
small text,
~~deleted text,~~
inserted text,
underlined,
~~crossed out,~~
text above line,
text under line.

Списки

Елементи списку є простими наборами **блокових** елементів, що мають додатковий блок (маркер), розташований збоку, який не бере участі в компонуванні. HTML дозволяє створювати марковані, нумеровані списки та списки визначень.

Нумеровані – для перерахування елементів, що йдуть у строго визначеному порядку. Якщо доповнити існуючий список новими значеннями, то браузер автоматично перерахує нумерацію його елементів.

Марковані – для переліку елементів списку, які наведені у довільному порядку.

Список визначень – використовується для форматування словників та подібного контенту.

Нумерований список

Для створення нумерованих списків використовуються теги `` та ``. Тегом `` відзначається початок/закінчення всього списку. Тегом `` позначають початок/кінець окремого елемента списку.

За допомогою атрибута **type** тега `ol` можна змінити стиль нумерації:

- `type="A"` – в алфавітному порядку великими латинськими літерами (A, B, C).
- `type="a"` – нумерація великими латинськими літерами (a, b, c).
- `type="I"` – нумерація великими римськими цифрами (I, II, III).
- `type="i"` – нумерація маленькими римськими цифрами (i, ii, iii).

За замовчуванням елементи списку нумеруються за порядком 1, 2, 3... За допомогою атрибутів `start` та `value` можна змінити порядок нумерації списку. `start` – служить для завдання початкового номера списку, відмінного від 1. `value` – дає змогу призначити довільний номер будь-якому елементу списку. Атрибут `reversed` задає отображення списка в обратном порядке.

Нумерований список. Приклад

```
<ol>
```

```
  <li>List item 1</li>
```

```
  <li>List item 2</li>
```

```
  <li>List item 3</li>
```

```
</ol>
```

```
<ol type="A">
```

```
  <li>List item 1</li>
```

```
  <li>List item 2</li>
```

```
  <li>List item 3</li>
```

```
</ol>
```

```
<ol start="5">
```

```
  <li>List item 1</li>
```

```
  <li value="9">List item 2</li>
```

```
  <li>List item 3</li>
```

```
</ol>
```

1. List item 1

2. List item 2

3. List item 3

A. List item 1

B. List item 2

C. List item 3

5. List item 1

9. List item 2

10. List item 3

Маркований список

Для створення маркованих списків використовуються теги `` та ``. Тегом `` (Unordered List) відзначається початок/закінчення всього списку. Тегом `` позначають початок/кінець окремого елемента списку.

За замовчанням елементи списку маркуються чорним кружком. За допомогою застарілого атрибуту `type` можна змінити стиль маркування. У межах одного списку можна використовувати різне маркування елементів списку.

```
<ul type="circle">  
  <li>List item 1</li>  
  <li>List item 2</li>  
  <li type="disc">List item 3</li>  
  <li type="square">List item 4</li>  
</ul>
```

- List item 1
- List item 2
- List item 3
- List item 4

Вкладений маркований список

```
<ul>
<li>List item 1.</li>
<li>List item 2.
<ul>
<li>List item 2.1.</li>
<li>List item 2.2.
<ul>
<li>List item 2.2.1.</li>
<li>List item 2.2.2.</li>
</ul>
</li>
<li>List item 2.3.</li>
</ul>
</li>
<li>List item 3.</li>
</ul>
```

Багаторівневі списки використовують для конкретизації інформації певних елементів. Для створення багаторівневих списків можна використовувати як марковані, так і нумеровані списки та їх поєднання. Багаторівневий список утворюється шляхом вкладення одного списку в тіло елемента іншого.

- List item 1.
- List item 2.
 - List item 2.1.
 - List item 2.2.
 - List item 2.2.1.
 - List item 2.2.2.
 - List item 2.3.
- List item 3.

Список визначень

Кожен пункт складається із двох частин: термін та його визначення.

Для створення списків визначень використовують три теги:

`<dl>` – початок/кінець списку.

`<dt>` – початок/кінець конкретного терміна.

`<dd>` – початок/кінець пояснюючої статті терміна.

Теги `<dt>` та `<dd>` не обов'язково чергувати. Тобто, можна «прив'язати» до одного визначення кілька термінів і навпаки.

`<dl>`

`<dt>Термін 1</dt>`

`<dd>Анотація1 до терміну 1</dd>`

`<dd>Анотація2 до терміну 1</dd>`

`<dt>Термін 2</dt>`

`<dd>Анотація1 до терміну 2</dd>`

`<dd>Анотація2 до терміну 2</dd>`

`</dl>`

Термін 1

Анотація1 до терміну 1

Анотація2 до терміну 1

Термін 2

Анотація1 до терміну 2

Анотація2 до терміну 2

Таблиці

Таблиця містить рядки, рядки містять комірки, у яких може бути довільний контент. Для додавання таблиці на веб-сторінку використовується тег `<table>`, який служить контейнером для рядків та комірок, які створюються відповідно за допомогою тегів `<tr>` та `<td>`. Для створення заголовних комірок замість тега `<td>` використовується тег `<th>`. Текст у такій комірці відображається браузером зі шрифтом жирного креслення і вирівнюється по центру комірки таблиці.

Тег `<caption>` призначений для створення заголовка до таблиці і може розміщуватись лише всередині контейнера `<table>`, причому відразу після початкового тега таблиці.

Атрибути тегів `<td>` і `<th>`:

- `colspan` – визначає об'єднання комірок по горизонталі.
- `rowspan` – визначає вертикальне об'єднання комірок.

Основна функція таблиць в HTML – представлення табличних даних, але також таблиці інколи використовуються для визначення структури тіла документа. Табличне верстання вважається застарілим та не рекомендується.

Таблиці. Приклад

```
<table width="500" border="1">  
  <caption>Заголовок таблиці, необов'язковий.</caption>  
  <tr>  
    <th colspan="2">Об'єднана (по горизонталі) заголовочна комірка</th>  
  </tr>  
  <tr>  
    <th>Заголовок 1</th><th>Заголовок 2</th>  
  </tr>  
  <tr align="center">  
    <td rowspan="2" align="center">Комірка, об'єднана по вертикалі.<br>Зверніть увагу на  
структуру цієї таблиці з точки зору браузеру.</td>  
    <td>Комірка 1-2</td>  
  </tr>  
  <tr>  
    <td>Комірка 2-2</td>  
  </tr>  
</table>
```

Заголовок таблиці, необов'язковий.	
Об'єднана (по горизонталі) заголовочна комірка	
Заголовок 1	Заголовок 2
Комірка, об'єднана по вертикалі. Зверніть увагу на структуру цієї таблиці з точки зору браузеру.	Комірка 1-2

Табличне верстання

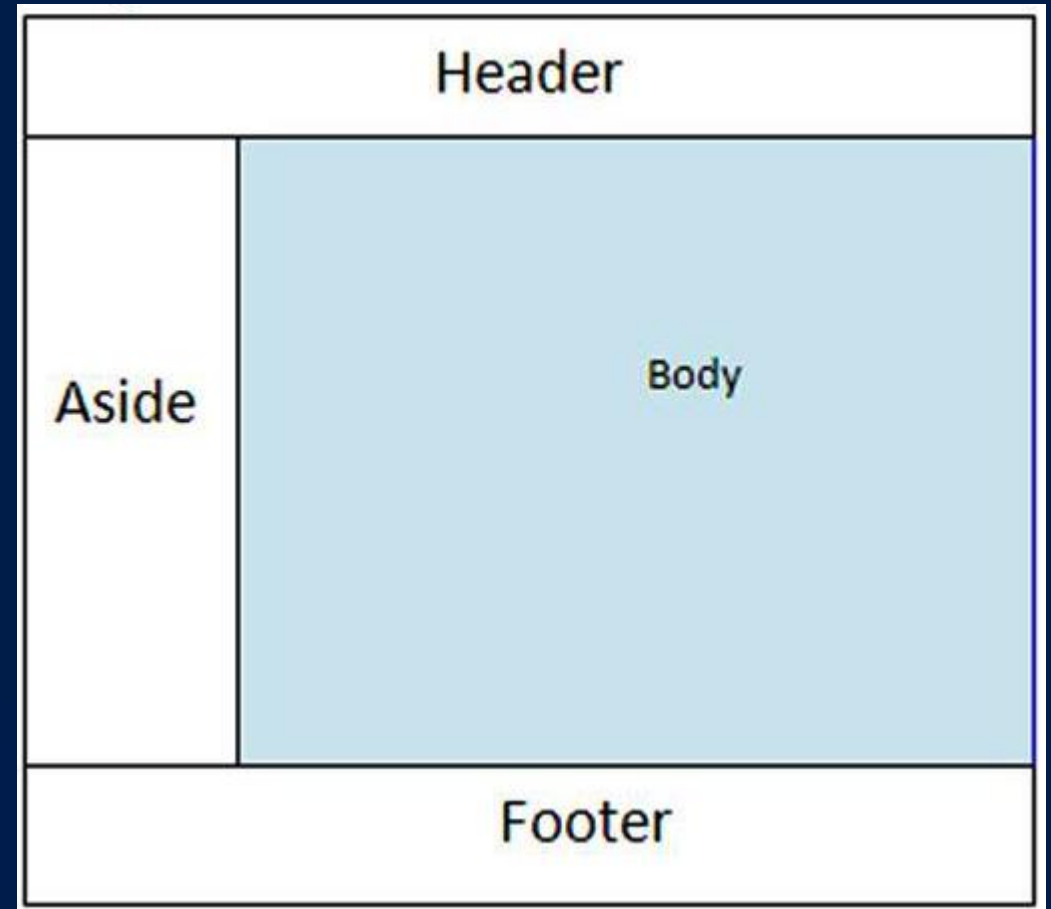
Використання таблиці у ролі невидимої модульної сітки, у яку додається текст, зображення та інші елементи.

Переваги:

- проста організація багатостовпкової сторінки;
- "гумовий" макет;
- однаковість обробки браузером.

Недоліки:

- довге завантаження;
- зайвий код;
- поділ інформації, що виводиться – це погано для SEO;
- складність оформлення;
- невідповідність стандартам.



Семантичне верстання

Підхід до розмітки, що спирається змістове призначення кожного блоку сторінки та логічну структуру документа.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Заголовок сторінки</title>
  </head>
  <body>
    <header>
      <!-- «Шапка» сайту -->
    </header>
    <main>
      <!-- Основний зміст сторінки -->
    </main>
    <footer>
      <!-- «Підвал» сайту -->
    </footer>
  </body>
</html>
```

Всі спеціальні семантичні теги – це блокові контейнери з визначеними **ARIA**-ролями, які можна призначити елементу і вручну: наприклад, замість `<header>` зробити `<div role="banner">`

Семантичне верстання

Елемент `<section>` поділяє вміст великих змістовних розділів (`<main>` та інших) на частини чи секції. Якщо контент поділяється на декілька логічних частин, то кожен можна загорнути в окремий `<section>`. У середині кожної секції слід використовувати заголовок.

`<footer>` є довідковим блоком, наприклад, з датою публікації, посиланнями на інформацію про автора. Якщо у підвалі сайту багато інформації, можливий його поділ на секції.

```
<h1>Обід</h1>
<section>
  <h2>Перші страви</h2>
</section>
<section>
  <h2>Другі страви</h2>
</section>
```

```
<footer>
  <section>
    <h2>Пошук</h2>
  </section>
  <section>
    <h2>Довідка</h2>
  </section>
</footer>
```

Семантичне верстання

Процес розмітки при семантичній верстці можна розділити на кілька кроків з різним ступенем деталізації:

- Виділити великі смислові блоки на кожній сторінці сайту, використовуючи теги `<header>`, `<main>`, `<footer>`.
- Великі смислові розділи в блоках визначити за допомогою тегів: `<nav>`, `<section>`, `<article>`, `<aside>`.
- Використовувати заголовок всього документа та заголовки у смислових розділах. Теги: `<h1>`-`<h6>`.
- Виділити дрібні структурні елементи у смислових розділах. Списки, таблиці, параграфи та переноси, форми, контактна інформація.
- Фразові елементи всередині дрібних структурних елементів, таких як зображення, посилання, кнопки та дрібні текстові елементи.

Семантична верстка не передбачає бездумної заміни всіх елементів `div` у документі на теги семантичної розмітки.

Гіперпосилання

Гіперпосилання дозволяють переходити з однієї веб-сторінки на іншу:

`текст посилання (або можна зображення)`

Атрибут `href` визначає URL (Universal Resource Locator, універсальний вказівник ресурсу) – адреса документа, до якого слід здійснити перехід. В адресах гіперпосилань слід використовувати лише латинські символи, без пропусків:

Наприклад, таке посилання не рекомендується:

`Посилання`

Посилання може вести не тільки на HTML-файли, але у вікні браузера ресурс, відповідний посиланню, відкривається лише якщо браузер підтримує тип цього ресурсу.

Посилання може бути не на файл взагалі, наприклад:

`Пошта`

Абсолютні гіперпосилання

Гіперпосилання, залежно від виду адреси (значення атрибуту **href**), можуть бути **абсолютними** та **відносними**.

Адреси **абсолютних** гіперпосилань повинні починатися із зазначення протоколу (зазвичай **http** або **https**), тобто потрібно вказувати **повний** шлях до ресурсу, до якого здійснюється перехід:

https://en.wikipedia.org/wiki/Web_design

Можливі гіперпосилання на ресурси, що розміщені фізично на цьому ж ПК, тоді використовується псевдопротокол **file** (без цього деякі браузері не працюють зі шляхами у файловій системі):

file:///D:/about.html

file:///D:/Education/%D0%9F%D1%80%D0%B5%D0%BF%D0%BE%D0%B4%D0%Bo%D0%B2%D0%Bo%D0%BD%D0%B8%D0%B5/Trening_2014/Lb1.html

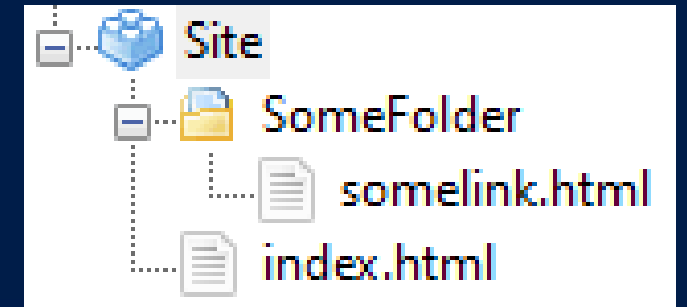
Але на практиці такі гіперпосилання не використовуються. Абсолютні посилання зазвичай застосовуються для вказівки документа на **іншому** Інтернет-ресурсі. Робити абсолютні посилання на інші файли нашого ж ресурсу можливо, але не має сенсу.

Відносні гіперпосилання

Відносні посилання використовуються для переходу на інші сторінки у складі цього ж самого ресурсу. Адреси відносних гіперпосилань будуються відносно знаходження поточного документа або кореня сайту. Така поведінка може бути змінена за допомогою тега `base`.

Правило «трьох кліків» – намагайтеся формувати «дерево посилань» таким чином, щоб відвідувач сайту міг з будь-якої його сторінки потрапити до будь-якого місця сайту за мінімальну кількість переходів за допомогою гіперпосилань. Не більше трьох переходів.

Наприклад, поточний файл (`index.html`) знаходиться в корені сайту, а файл, на який необхідно посилання (`somelink.html`) – у папці (`SomeFolder`):



`Текст посилання`

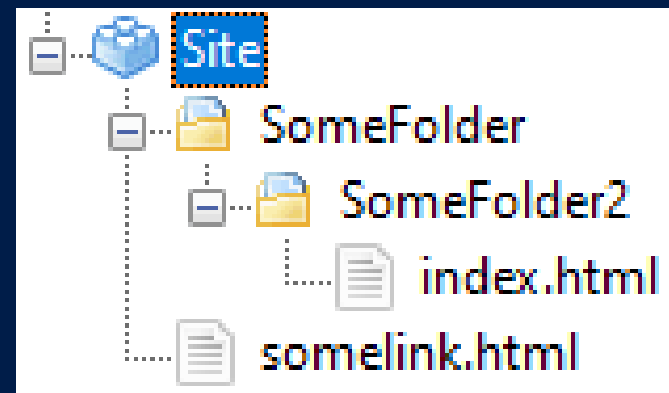
Відносні гіперпосилання

Приклади, у яких поточний файл і той, на який веде посилання, розміщуються у різних папках:

`Текст посилання`

`Текст посилання`

Дві крапки в цих прикладах означають вихід з поточної папки на рівень вище.



Гіперпосилання. Мітки

При переході за гіперпосиланням вказаний ресурс зазвичай відкривається з самого початку сторінки. Але у випадках, коли потрібно перейти до якогось фрагменту сторінки («проскролити» її), або перейти до певного місця сторінки, а не просто до її початку, необхідно використати **мітки** – «якоря», до яких буде здійснюватися перехід, замість початку сторінки.

Для створення якоря слід спочатку зробити мітку у відповідному місці сторінки та дати їй «ім'я» для ідентифікації. Для цього використовується атрибут **id**, який можливий у більшості тегів. Для гіперпосилання на мітку потрібно вказувати у адресі (в атрибуті href) символ **#** та значення атрибуту **id**, що було використано для створення мітки.

Гіперпосилання можливе як до мітки, що розташована на поточній сторінці (тоді у атрибуті **href** гіперпосилання вказується лише **#** та мітка), так і до мітки, що розташована на будь-якій іншій, тоді **#** та мітка дописуються до адреси (немає різниці, відносної чи абсолютної).

Гіперпосилання. Мітки

```
<body>
  <h1 id="top">Назва сторінки</h1>
  <p>Мітки доцільно використовувати на
сторінках з дуже великим обсягом
контенту, для спрощення навігації.</p>
  <p id="section1">Наприклад, для
створення змісту сторінки, який
дозволяє переходити без скролу до
потрібного розділу.</p>
  <a href="#section1">Повернутися до
початку розділу</a> <br>
  <a href="#top">Повернутися до початку
сторінки</a>
</body>
```

для спрощення навігації.

Наприклад, для створення змісту сторінки, який дозволяє переходити без скролу до потрібного розділу.

[Повернутися до початку розділу](#)
[Повернутися до початку сторінки](#)

Гіперпосилання. Стани

Стани гіперпосилань на сторінці:

Невідвідане – зображується синього кольору та з підкресленням.

Активне – посилання позначається як активне в момент його відкриття. Колір такого посилання червоний. Оскільки час між натисканням на посилання та початком завантаження нового документа досить малий, активний стан посилання помітити важко. Активним посилання стає також при його фокусі за допомогою клавіатури.

Відвідано – після здійснення переходу по гіперпосиланню, при повторних відображеннях такого гіперпосилання користувачу воно змінює свій колір на фіолетовий.

Вказано відображення гіперпосилань з різними станами за замовчуванням. Але зазвичай оформлення посилань, залежне від їх стану, визначається засобами каскадних таблиць стилів (CSS).

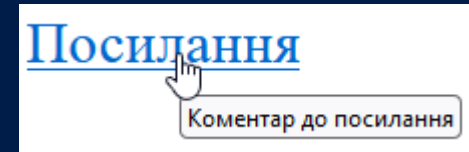
Невідвідане, активне та відвідане посилання.

Гіперпосилання. Атрибути

Обов'язкових (для валідації) атрибутів у тега **a** **нема**. Деякі атрибути:

- **target** – у якому вікні/вкладці відкривати. Значення **_blank**, **_self** (за замовчуванням), **_parent**, **_top**.
- **title** – пояснювальний текст до посилання у вигляді спливаючої підказки.

```
<a href="new.html" target="_blank" title="Коментар до посилання">Посилання</a>
```



Вказувати відкриття посилання у поточній чи новій вкладці? Рекомендується в залежності від розташування сторінок, що адресуються, в дереві ресурсу. Якщо перехід за посиланням відбувається до сторінки-батька або дочірньої сторінки (верх-вниз по гілці), то відкривати рекомендується у поточній вкладці. Якщо посилання веде до сторінки на іншій гілці дерева в рамках сайту, або, тим паче, адреса веде за межі поточного ресурсу – то посилання має відкриватися в новій вкладці.

Гіперпосилання. Атрибут href

Атрибут href є необов'язковим, тому, якщо його не додати до тега a, цей елемент семантично залишиться гіперпосиланням, але неактивним. Щоб зробити його знову активним, достатньо повернути атрибут href (наприклад, за допомогою JavaScript). Такий підхід можна використовувати при вказівці поточної сторінки у розмітці навігації.

Не замінюйте тег посилання:

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Projects</a></li>
    <li><span>About me</span></li>
  </ul>
</nav>
```

Краще керуйте атрибутом href:

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Projects</a></li>
    <li><a href="#">Projects</a></li>
  </ul>
</nav>
```

Контрольні запитання

- У чому різниця між блоковими та строковими елементами? Наведіть приклад елементів, що є блочно-строковими.
- Які теги використовуються для семантичної верстки?
- Види списків. Як змінити порядок нумерації у списку?
- Теги для створення таблиць. Як об'єднати комірки по вертикалі?
- Навіщо використовуються гіперпосилання? У чому різниця між абсолютними та відносними гіперпосиланнями?
- Які атрибути тега `a` є обов'язковими (без яких валідацію не буде пройдено)?
- Як вказати відкриття гіперпосилання у новій вкладці?
- Як забезпечити перехід до іншого розділу поточного документа?
- Можливі стани гіперпосилань.