

Web-технології та web-дизайн

2024/2025

Лекція №2 HTML

Лекції: ст. викладач каф. Штучного інтелекту
Гриньова Олена Євгенівна
olena.hrynova@nure.ua

Зміст

- Мови розмітки
- Базові основи та визначення HTML
- Базові теги форматування та розміщення тексту
- Атрибути тегів
- Заголовки та списки
- Гіперпосилання та зображення
- Таблиці
- Форми

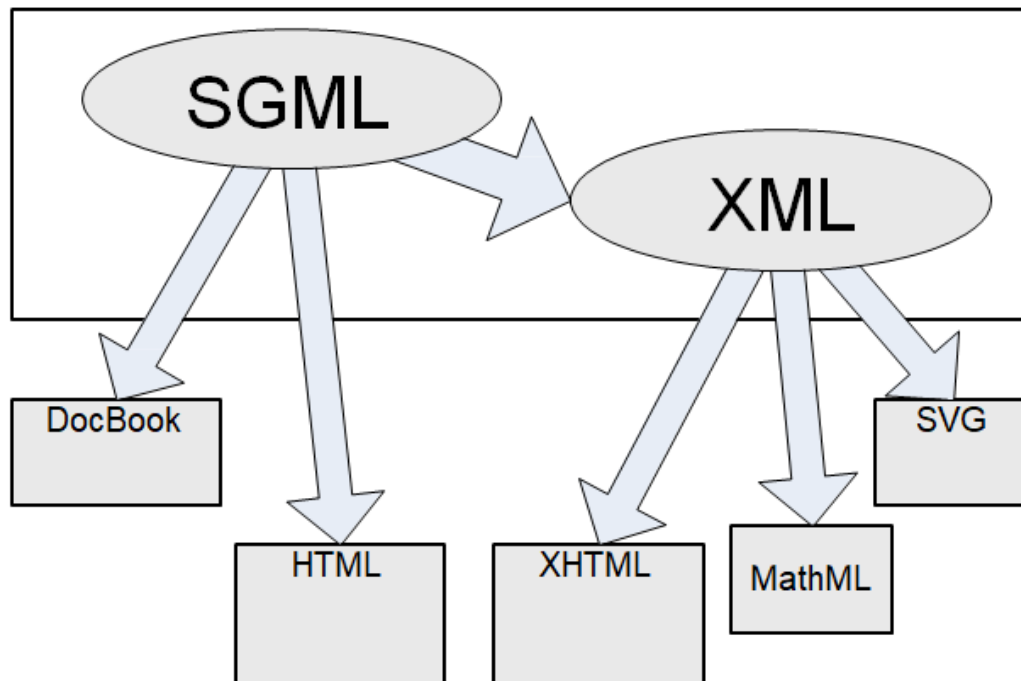
Що необхідно?

Браузер, текстовий редактор, ідеї

Дерево мов розмітки

1969. **Generalized Markup Language** (GML) - IBM text formatter, SCRIPT

ISO 8879:1986 Information processing -Text and office systems - **Standard Generalized Markup Language** (SGML)



HyperText Markup Language

Розробка 1986—1991, Tim Berners-Lee, CERN

HTML 2.0 – 22.09.1995,

ETF RFC 1866, RFC 1867, RFC 1942, RFC 1980, RFC 2070

HTML 3.2 – 14.01.1997 W3C Recommendation

HTML 4.0[2] – 18.12.1997;

HTML 4.01 – 24.12.1999;

ISO/IEC 15445:2000 – (на основі HTML 4.01 Strict) – 15.05.2000

HTML 5 – 28.10.2014 W3C Recommendation

HTML 5.1 – 01.11.2016 W3C Recommendation

HTML 5.2 – 14.12.2017 W3C Recommendation

HTML 5.3 був представлений 24 грудня 2018 року.

<https://www.w3.org/TR/html52/>

HTML

Що таке HTML?

- HTML є стандартною **мовою розмітки** для створення веб-сторінок.
- HTML розшифровується як **Hyper Text Markup Language**
- HTML описує **структуру** веб-сторінки
- HTML складається із серій **елементів**
- HTML-елементи повідомляють браузеру, як відображати **контент**
- HTML-елементи представлені **тегами**
- HTML-теги позначають **фрагменти контенту**, такі як «заголовок», «абзац», «таблиця», «зображення» і т. д.
- Браузери не відображають теги HTML, але використовують їх для відображення контенту сторінки

XML. <http://www.w3.org/XML>

XML (англ. eXtensible Markup Language — **розширювана мова розмітки**)

```
<?xml version="1.0" encoding="UTF-8"?>  
<note>  
  <to>ІВАНОВ</to>  
  <from>ПЕТРОВ</from>  
  <heading>Нагадування</heading>  
  <body>Завтра нарада о 10 годині!</body>  
</note>
```

MathML

MathML (від англ. Mathematical Markup Language — «мова математичної розмітки») — це мова розмітки на основі XML для представлення математичних символів та формул в документах WWW. MathML рекомендований математичною групою W3C.

XHTML

XHTML (англ. Extensible Hypertext Markup Language – розширювана мова розмітки гіпертексту)

- XHTML практично ідентичний HTML
- XHTML більш суворий, ніж HTML
- XHTML - це HTML, визначений як додаток XML
- XHTML підтримується усіма основними браузерами

Відмінності XHTML від HTML

- Елементи XHTML повинні бути правильно вкладені
- Всі елементи повинні бути закриті: `
`.
- Булеві атрибути записуються у повній формі.
- Імена тегів і атрибутів тільки малими літерами:
- Кодуванням за замовчуванням є UTF-8
- Документи XHTML повинні мати один кореневий елемент
- `<!DOCTYPE>` Обов'язковий
- Атрибут `xmlns` в `<html>` повинен вказувати простір імен `xml` для документа
- Всі значення атрибутів пишуться в лапках

Структура HTML/XHTML-сторінки

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
  "http://www.w3.org/TR/html4/strict.dtd">  
<html>  
  <head>  
    <title>!DOCTYPE</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
  </head>  
  <body>  
    <p> ...</p>  
  </body>  
</html>
```

Document Type Definition

<!DOCTYPE> призначений для вказівки **типу** поточного документа.
<!DOCTYPE> оголошення повинно бути найпершим у вашому HTML-документі, перед тегом **<html>**.
<!DOCTYPE> - це інструкція для веб-браузера про те, **в якій версії HTML** написана веб-сторінка.

HTML5:

<!DOCTYPE html>

HTML 4.1 (Перехідний синтаксис HTML):

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

Transitional XHTML 1.0 (Перехідний синтаксис XHTML):

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

Strict XHTML 1.0 (Суворий синтаксис XHTML):

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

Основні поняття

- Документ
- Елемент
- Атрибут
- Тег

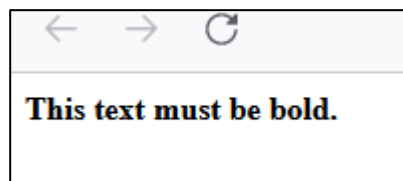
Що таке HTML-теги

Tag – тег, мітка, команда розмітки тексту

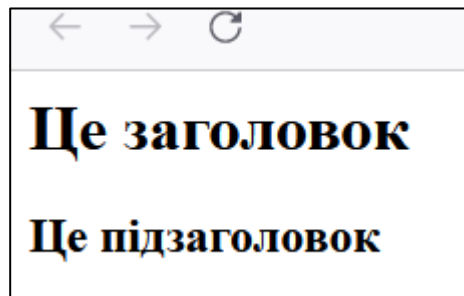
- Всі теги починаються з **<** і закінчуються **>**
- Відкриваючий та закриваючий тег: `<html>` и `</html>`
- Поодинокі теги, `
` або `<hr />`

Приклади перших тегів

- `This text must be bold.`



- `<h1>Це заголовок</h1>`
`<h2>Це підзаголовок</h2>`

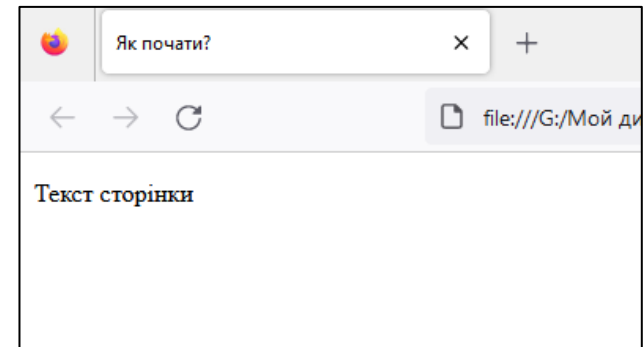


Регістрозалежність

- <HTML>, <html> або <HtMl>
- звикайте друкувати теги в **нижньому** регістрі (W3C **рекомендує** використовувати малі літери в HTML, і **вимагає** малих літер для більш строгих типів документів, таких як XHTML)

Як почати?

```
<!DOCTYPE html>
<html>
  <head>
    <title>Як почати?</title>
  </head>
  <body>
    <p>Текст сторінки</p>
  </body>
</html>
```



наполегливо рекомендуємо структурувати ваш HTML за допомогою розриву рядків і відступів

- у браузері відображається тільки **вміст** розділу **<body>**

У розділі body

<p>Параграф</p>

Текст жирним шрифтом

<h1>Заголовок</h1>

<h2>Підзаголовок</h2>

<h3>Під-підзаголовок</h3>

<i>Похилий шрифт</i>

Комбінації тегів

<i>Текст bold та italic</i>

Текст bold та italic

виключено перекривання тегів:

<i>Текст bold та italic</i>

Поодиночні теги

- Деякий текст **
** та ще текст, але вже на новому рядку
- **<hr/>** - малює горизонтальну лінію

Елементи HTML

Елементи HTML – це окремі блоки, з яких будується вся сторінка. Вони визначають, як браузер має відображати текст, зображення, відео та інші елементи на екрані.

Елементи HTML

Кожен елемент HTML складається з **відкриваючого тегу** (наприклад, `<p>`) та **закриваючого тегу** (наприклад, `</p>`).

Між відкриваючим та закриваючим тегами розміщується **вміст елемента**.

`<p>Текст сторінки</p>`

Елементи HTML **структурують** контент вебсторінки.

Семантичні елементи роблять код більш зрозумілим для браузерів і пошукових систем.

Основні елементи HTML

Заголовки: `<h1, <h2, ..., <h6>` - використовуються для позначення заголовків різного рівня.

Абзаци: `<p>` - визначає абзац тексту.

Списки: `` (unordered list) для неупорядкованих списків, `` (ordered list) для впорядкованих списків, `` (list item) для елементів списку.

Посилання: `<a>` - створює посилання на інший документ або ресурс.

Зображення: `` - вставляє зображення на сторінку.

Таблиці: `<table>`, `<tr>`, `<td>` - використовуються для створення таблиць.

Форми: `<form>` - створює форму для збору даних від користувача.

Контейнер: `<div>` - універсальний контейнер для групування інших елементів.

Різниця між елементом та тегом

Хоча **теги** та **елементи** тісно пов'язані, вони виконують різні функції.

Теги визначають **тип** елемента, а елементи є **конкретними частинами веб-сторінки**.

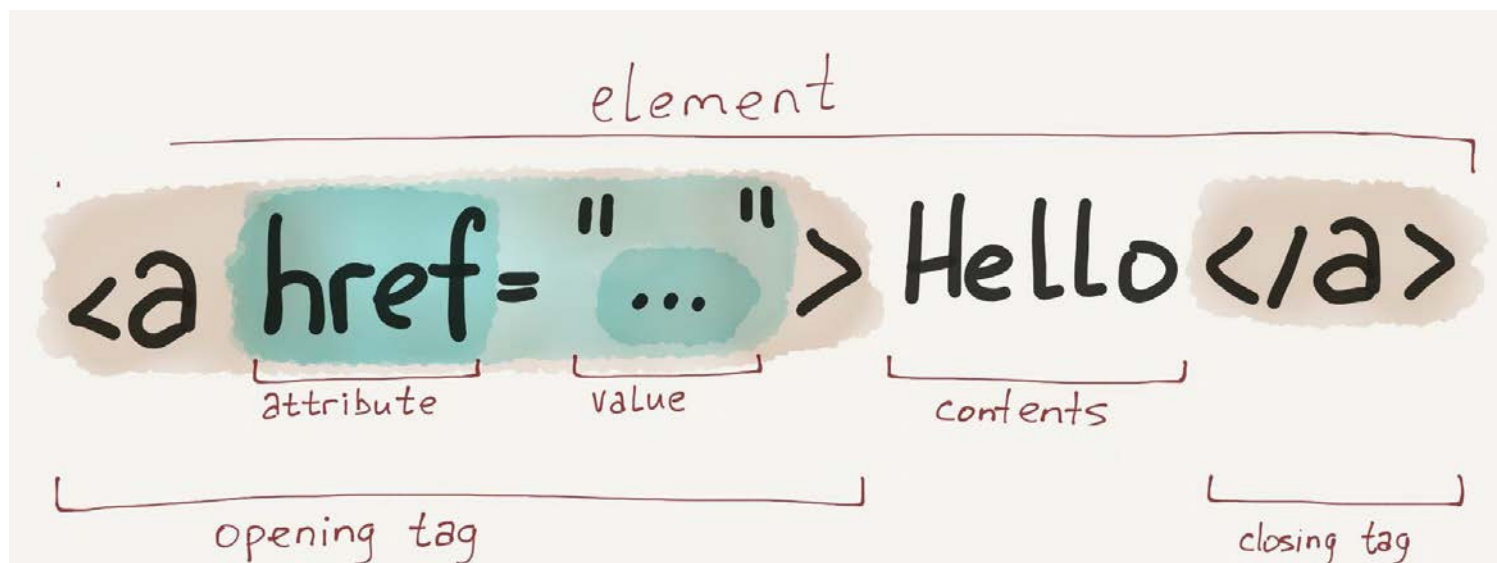
Тег (Tag) - це **ключове слово** у мові HTML розмітки. Означає початок чи кінець якогось елемента. Зазвичай пишуться у кутових дужках.

Елемент (Element) - це **структурна одиниця вебсторінки**. Складається з відкриваючого тегу, вмісту (тексту, інших елементів) і закриваючого тегу. Представляє певну частину документа.

Різниця між елементом та тегом

Тег – це будівельний блок, а елемент – це конструкція, створена з цих блоків.

Тег вказує на тип елемента, а елемент є конкретною реалізацією цього типу.



Атрибути тегів

- **Атрибути** використовуються для надання додаткової інформації про елементи HTML.
- Всі елементи HTML можуть мати атрибути
- Атрибути завжди вказуються в **початковому** тегу
- Атрибути, зазвичай, входять в пари **ім'я/значення**, такі як: `name="value"`
- Атрибути завжди записуються всередині тегу, після них розміщується **знак рівності** і деталі атрибута, які поміщують у подвійні лапки. **Крапка з комою** після атрибута служить для поділу команд різних стилів. У деяких ситуаціях, коли саме значення атрибута містить **подвійні лапки**, необхідно використовувати одинарні лапки `<p style="color:red;">`
- пишіть імена атрибутів в **нижньому** регістрі
- Зверніть увагу, що деякі теги та атрибути використовують **американський різновид англійської мови**, тобто **color** замість **colour**.

Атрибути тегів

Атрибути можуть застосовуватися для більшості тегів.

У таких тегах, як **body**, ви будете часто використовувати атрибути, а в **br** - рідко, оскільки перенесення рядка це зазвичай перенесення рядка, без будь-яких уточнюючих параметрів.

Є безліч тегів, є і багато атрибутів. Є атрибути, призначені спеціально для якогось певного тегу (**специфічні**), а інші можна використовувати в різних тегах (**універсальні**).

Деякі теги можуть мати тільки один якийсь певний атрибут, в той час як інші теги - кілька атрибутів.

З яких ще частин складається тег? Основна частина тегу називається **елемент**. Тег складається з **елемента** (наприклад, `<p>`), або з **елемента** плюс один або більше **атрибутів**. Наприклад,

`<p style="background-color:#ff0000;">`

Формат атрибутів

Кожний атрибут тегу відноситься до певного **типу даних** (наприклад: текст, число, шлях до файлу та ін.), який **обов'язково** повинен враховуватися при написанні атрибута.

Важливий також і **домен припустимих значень**.

Тег **** додає на веб-сторінку малюнок, а його атрибут **width** задає ширину зображення у **пікселях**. Якщо поставити не число, а щось інше, то значення буде проігноровано і виникне помилка при валідації документа.

Атрибути тегів

Як буде відображатися елемент, якщо не задані значення атрибутів тегу?

Значення за замовчуванням атрибутів тегів

Для атрибутів тегів використовуються **значення за замовчуванням**.

Коли для тегу не доданий будь-який допустимий атрибут, це означає, що браузер в цьому випадку буде підставляти **значення, задане за замовчуванням**.

Якщо ви очікували отримати інший результат на веб-сторінці, перевірте, можливо, слід явно вказати значення деяких атрибутів.

Атрибути без значень

Допустимо використовувати деякі атрибути тегів, які не привласнюють їм **ніякого значення**.

`<p>`

`<input type="submit" disabled>`

`</p>`

У даному прикладі використовується атрибут **disabled**, для якого явно не задано значення. Подібний запис називається «**скорочений атрибут тегу**».

Порядок атрибутів в тегах

Порядок атрибутів в будь-якому тегу **не має значення** і на результат відображення елемента не впливає.

Тому теги виду:

```

```

та

```

```

за своєю дією рівні.

Універсальні атрибути тегів

Універсальні атрибути застосовуються практично до всіх тегів, тому виділені в окрему групу, щоб не повторювати їх для всіх тегів:

- accesskey
- **class**
- contenteditable
- contextmenu
- dir
- hidden
- **id**
- lang
- spellcheck
- **style**
- tabindex
- title

Подієві атрибути тегів

Атрибути подій пов'язані з JavaScript-подіями. При взаємодії користувача з елементами ці події можуть спрацьовувати та змінювати поведінку об'єктів.

onclick

oncopy

onmousemove

onfocus

...

Атрибути, приклади

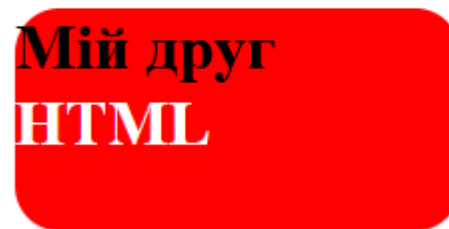
<Ім'яТегу Ім'яАтрибута="ДеталіАтрибута;">

Наприклад:

```
<h1>Мій друг  
  <span style="color:red; size:10; face:Arial;"> HTML  
  </span>  
</h1>
```

```
<div style="border-radius: 20px; width: 200px; height:  
100px; background-color: #ff0000; ">  
  <h1>Мій друг  
    <span style="color:white; size:10; face:Arial;">HTML  
    </span>  
  </h1>  
</div>
```

Мій друг HTML



Колір, атрибут color

Значення властивості кольору можна встановити трьома основними способами:

- за **назвою** кольору:

red, green, blue, gold, olive, magenta, cyan, brown і так далі

- **шістнадцяткове число (hexadecimal number):**

від #000000 до #ffffff

- за кодом кольору в палітрі **RGB:**

від rgb(0, 0, 0) до rgb(255, 255, 255)

Колір, атрибут color

Шістнадцятковий код коліру / HEX код

xx **xx** **xx**

Білий: #ffffff

Значення **00** - найнижча інтенсивність, **ff** - найвища. Наприклад, для отримання білого кольору потрібно змішати три основних кольора з максимальною інтенсивністю: #ffffff. Для отримання чорного кольору, відповідно, інтенсивність усіх кольорів повинна бути мінімальною: #000000.

216 Web Safe Colour Chart ([таблиця кольорів](#)):

Колір, атрибут color

216 Web Safe Colour Chart

#FFFFFF	#FFCCFF	#FF99FF	#FF66FF	#FF33FF	#FF00FF
#FFFFCC	#FFCCCC	#FF99CC	#FF66CC	#FF33CC	#FF00CC
#FFFF99	#FFCC99	#FF9999	#FF6699	#FF3399	#FF0099
#FFFF66	#FFCC66	#FF9966	#FF6666	#FF3366	#FF0066
#FFFF33	#FFCC33	#FF9933	#FF6633	#FF3333	#FF0033
#FFFF00	#FFCC00	#FF9900	#FF6600	#FF3300	#FF0000
#CCFFFF	#CCCCFF	#CC99FF	#CC66FF	#CC33FF	#CC00FF
#CCFFCC	#CCCCCC	#CC99CC	#CC66CC	#CC33CC	#CC00CC
#CCFF99	#CCCC99	#CC9999	#CC6699	#CC3399	#CC0099
#CCFF66	#CCCC66	#CC9966	#CC6666	#CC3366	#CC0066
#CCFF33	#CCCC33	#CC9933	#CC6633	#CC3333	#CC0033
#CCFF00	#CCCC00	#CC9900	#CC6600	#CC3300	#CC0000
#99FFFF	#99CCFF	#9999FF	#9966FF	#9933FF	#9900FF
#99FFCC	#99CCCC	#9999CC	#9966CC	#9933CC	#9900CC
#99FF99	#99CC99	#999999	#996699	#993399	#990099
#99FF66	#99CC66	#999966	#996666	#993366	#990066
#99FF33	#99CC33	#999933	#996633	#993333	#990033
#99FF00	#99CC00	#999900	#996600	#993300	#990000
#66FFFF	#66CCFF	#6699FF	#6666FF	#6633FF	#6600FF
#66FFCC	#66CCCC	#6699CC	#6666CC	#6633CC	#6600CC
#66FF99	#66CC99	#669999	#666699	#663399	#660099
#66FF66	#66CC66	#669966	#666666	#663366	#660066
#66FF33	#66CC33	#669933	#666633	#663333	#660033
#66FF00	#66CC00	#669900	#666600	#663300	#660000
#33FFFF	#33CCFF	#3399FF	#3366FF	#3333FF	#3300FF
#33FFCC	#33CCCC	#3399CC	#3366CC	#3333CC	#3300CC
#33FF99	#33CC99	#339999	#336699	#333399	#330099
#33FF66	#33CC66	#339966	#336666	#333366	#330066
#33FF33	#33CC33	#339933	#336633	#333333	#330033
#33FF00	#33CC00	#339900	#336600	#333300	#330000
#00FFFF	#00CCFF	#0099FF	#0066FF	#0033FF	#0000FF
#00FFCC	#00CCCC	#0099CC	#0066CC	#0033CC	#0000CC
#00FF99	#00CC99	#009999	#006699	#003399	#000099
#00FF66	#00CC66	#009966	#006666	#003366	#000066

Колір, атрибут color

Палітра RGB кольорів HTML

У палітрі RGB колір додається по такому принципу: задається інтенсивність червоного (red), зеленого (green) і синього (blue) кольорів. Від сюди і абревіатура RGB. Тільки інтенсивність зазначається тут числами від **0** до **225**, де 0 - найнижча інтенсивність, 225 - найвища. Для прикладу, зелений колір буде виглядати наступним чином:

`rgb(0, 225, 0)`

Заголовки. Форматування тексту за допомогою <h1>...<h6>

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
  </head>
```

```
  <body>
```

```
    <p>Текст</p>
```

```
    <h1>Глава 1</h1>
```

```
    <p>Текст, який відноситься до глави 1. </p>
```

```
    <h2>Глава 1.1</h2>
```

```
    <p>Текст, який має відношення до глави 1.1</p>
```

```
    <h2>Глава 1.2</h2>
```

```
    <p>Текст, який має відношення до глави 1.2</p>
```

```
  </body>
```

```
</html>
```


<h1>...<h6>

- Браузери автоматично додають пробіл (поле) до і після заголовка.
- Пошукові системи використовують заголовки для індексації структури і змісту ваших веб-сторінок.
- Важливо використовувати заголовки, щоб показати структуру документа.
- Використовуйте заголовки HTML тільки для заголовків. Не використовуйте заголовки, щоб зробити текст ВЕЛИКИМ або жирним .

Текст

Глава 1

Текст, який відноситься до глави 1.

Глава 1.1

Текст, який має відношення до глави 1.1

Глава 1.2

Текст, який має відношення до глави 1.2

Контейнери в HTML

Контейнери в HTML - це фундаментальний інструмент для структурування та організації вмісту на веб-сторінках. Вони дозволяють групувати різні елементи (текст, зображення, інші контейнери) в логічні блоки, що полегшує управління їхнім зовнішнім виглядом та поведінкою.

```
<div class="container">  
  <h1>Заголовок</h1>  
  <p>Абзац тексту.</p>  
    
</div>
```

Контейнери в HTML

Контейнери об'єднують різні елементи HTML в єдине ціле, дозволяючи застосовувати до них стилі та маніпулювати ними як одним об'єктом.

Структурування сторінки: За допомогою контейнерів можна створювати чітку ієрархію елементів на сторінці, що полегшує розуміння її структури як для розробника, так і для браузера.

Застосування стилів: До контейнерів можна застосовувати стилі CSS для зміни їхнього зовнішнього вигляду (фон, розмір, відступи, вирівнювання тощо).

JavaScript-маніпуляції: Контейнери часто використовуються як мішені для JavaScript-скриптів, дозволяючи динамічно додавати, видаляти або змінювати їхній вміст.

Контейнер <div>

Універсальний контейнер **<div>** - для групування інших елементів.

Хоча сам по собі елемент **div** не має вбудованого стилю або семантики, він служить універсальним блоком для групування інших елементів на веб-сторінці. Елемент **<div>** не має жодної прив'язки до конкретного типу контенту, що робить його ідеальним для створення гнучких і багатофункціональних блоків.

div є скороченням від англійського слова '**division**', що означає розділ, секція.

Контейнери

Окрім універсального `<div>`, у веб-розробці використовується цілий спектр контейнерів, кожен з яких має свою специфічну мету та семантику. Обираючи тип контейнера, важливо розуміти, яку роль він відіграватиме у вашому макеті та як він вплине на структуру сторінки.

- Семантичні контейнери HTML5
- Контейнери для створення макетів
- Контейнери для форм
- Спеціалізовані контейнери
- Універсальний контейнер для інлайн-елементів

Спеціалізовані контейнери в HTML

Спеціалізовані контейнери в HTML

Таблиці: `<table>`, `<tr>`, `<td>` - для структурування даних у табличному вигляді.

Списки: `` (неупорядкований), `` (упорядкований), `` (елемент списку) - для представлення списків.

Блоки цитування: `<blockquote>`, `<q>` - для цитування тексту

Спеціалізовані контейнери в HTML

Тег **span** є одним з найчастіше використовуваних елементів у HTML. Він є універсальним контейнером для **інлайн-елементів**, який не має власної семантики. Це означає, що він не несе в собі жодної конкретної інформації про зміст, а служить виключно для угруповання елементів та застосування до них стилів.

<p>Це звичайний текст. А <**span** style="color: blue;">це виділене слово</**span**>.</p>

Це звичайний текст. А це виділене слово.

Списки в HTML

Елементи ``, `` є контейнерами.

`` - скорочення від **unordered list** (невпорядкований / ненумерований / маркований список)

Атрибути:

type = disk | circle | square | none

`` - скорочення від **ordered list** (нумерований)

Атрибути:

- **type** = 1 | A | a | I | i
- **start** – початок нумерації (за замовчуванням упорядкований список починає відлік з 1. Якщо ви хочете почати підрахунок з зазначеного числа, ви можете використовувати **start** атрибут)

`` - скорочення від **list item** (елемент списку)

Нумеровані списки

Перший елемент списку

Другий елемент списку

1. Перший елемент списку
2. Другий елемент списку

Маркований список

```
<ul>
  <li>маркер за замовчуванням</li>
  <li>маркер за замовчуванням</li>
</ul>
<ul type="disk">
  <li>маркер "disk"</li>
  <li>маркер "disk"</li>
</ul>
<ul type="circle">
  <li>маркер "circle"</li>
  <li>маркер "circle"</li>
</ul>
<ul type="square">
  <li>маркер "square"</li>
  <li>маркер "square"</li>
</ul>
<ul type="none">
  <li>маркер "none"</li>
  <li>маркер "none"</li>
</ul>
```

- маркер за замовчуванням
- маркер за замовчуванням

- маркер "disk"
- маркер "disk"

- маркер "circle"
- маркер "circle"

- маркер "square"
- маркер "square"

- маркер "none"
- маркер "none"

Багаторівневі списки

Nested lists

Спочатку потрібно створити список першого рівня, а потім всередину будь-якого елемента цього списку, між тегамі `` і `</ li>`, додати список другого рівня. При цьому необхідно акуратно закривати всі теги.

Кількість рівнів в списках **не обмежена**. У багаторівневому списку можна використовувати як впорядковані, так і невлпорядковані списки.

Багаторівневі списки

```
<!DOCTYPE html>
<html>
<body>
  <ul>
    <li>маркер за замовчуванням
      <ul type="circle">
        <li>маркер "circle"</li>
        <li>маркер "circle"</li>
      </ul>
    </li>
    <li>маркер за замовчуванням
      <ul type="square">
        <li>маркер "square"</li>
        <li>маркер "square"</li>
      </ul>
    </li>
  </ul>
</body>
</html>
```

- маркер за замовчуванням
 - маркер "circle"
 - маркер "circle"
- маркер за замовчуванням
 - маркер "square"
 - маркер "square"

Списки визначень

<dl>

<dt>Термін 1</dt>

<dd>Визначення терміну 1</dd>

<dt>Термін 2</dt>

<dd>Визначення терміну 2</dd>

</dl>

Термін 1

Визначення терміну 1

Термін 2

Визначення терміну 2

dl - **D**escription **L**ist (список визначень)

dt - **D**efinition **T**erm (термін визначень)

dd - **D**escription **D**etails (опис визначень)

Посилання

- HTML-посилання є гіперпосиланнями – елемент **a**.
- **Текст посилання** є видимою частиною
- **Посилання** не обов'язково повинно бути текстовим. Це може бути зображення або будь-який інший елемент HTML

`Це посилання на HTML.net`

[Це посилання на HTML.net](http://www.html.net/)

Елемент **a** – скорочення від **anchor** (якір)

Атрибут **href** – скорочення від **hypertext reference** (гіпертекстове посилання). Адреса посилання може бути **абсолютною** і **відносною**. Абсолютні адреси працюють повсюди незалежно від імені сайту або веб-сторінки, де прописане посилання. Відносні посилання, як випливає з їх назви, побудовані щодо поточного документа або кореня сайту.

W3C рекомендує **лапки** (") навколо **значень** атрибутів в HTML

Приклади відносних посилань

- ``
- ``
- ``
- `<h1 id="heading1">heading1</h1>`
`heading1`
- ``

Поєднання `../` вказує на папку, розташовану на один рівень вище від даної позиції файлу, з якого робиться посилання. Можна також вказати на два рівня вище `../../` або більше.

Альтернативно ви завжди можете вказати повну адресу файлу (URL).

Посилання. Атрибути

accesskey	Активация посилання за допомогою комбінації клавіш.
coords	Встановлює координати активної області.
download	Пропонує завантажити вказаний за посиланням файл.
href	Задає адрес документу, на який треба перейти.
hreflang	Ідентифікує мову тексту за посиланням.
name	Встановлює ім'я якоря усередині документа.
rel	Відносини між документами, на які посилаються, і поточним.
rev	Відношення між поточними документами та на які посилаються.
shape	Задає форму активної області посилання для зображень.
tabindex	Визначає послідовність переходу між посиланнями при натисканні на кнопку Tab.
target	Ім'я вікна або фрейму, куди браузер буде завантажувати документ.
title	Додає спливаючу підказку до тексту посилання.
type	Вказує MIME-тип документу, на який вказує посилання.
Також для цього тегу доступні універсальні атрибути та події .	

Посилання у середині сторінки.

HTML bookmarks

- Атрибут **id** (id="value") використовується для визначення закладок на сторінці, **елемент HTML може мати тільки один унікальний ідентифікатор** (заборонено різним елементам мати однакові ідентифікатори).
- Значення **id** чутливе до регістру.
- Значення **id** повинне містити щонайменше **один символ і не повинно містити пробілів**
- Значення кожного атрибута **id** має бути **унікальним в межах поточного документу**

```
<body>
```

```
<p><a href="#heading1">Посилання на heading 1</a></p>
```

```
<p><a href="#heading2">Посилання на heading 2</a></p>
```

```
<h1 id="heading1">heading 1</h1>
```

```
<p>Text text text text</p>
```

```
<h1 id="heading2">heading 2</h1>
```

```
<p>Text text text text</p>
```

```
</body>
```

Посилання у середині сторінки

[Посилання на heading_1](#)

[Посилання на heading_2](#)

heading 1

Text text text text

heading 2

Text text text text

Посилання. Атрибут title

Атрибут **title** використовується для короткого опису посилання (інформація найчастіше відображається у вигляді підказки, коли миша переміщується над елементом).

```
<a href="http://www.html.net/" title="Visit HTML.net  
and learn HTML">HTML.net</a>
```

[HTML.net](http://www.html.net/)

..... Visit HTML.net and learn HTML

Посилання. Атрибут target.

Тег base.

Атрибут **target** може приймати наступні значення:

- **_blank** — завантажує сторінку у **нове вікно/вкладинку** браузера;
- **_self** — завантажує сторінку у **поточне вікно**;
- **_parent** — завантажує сторінку до **батьківського фрейму**;
- **_top** — **відмінює усі фрейми** та завантажує сторінку у повному вікні браузеру.

Відкриття документу у новому вікні

```
<a href="http://www.htmlbook.ru" target="_blank">Посилання відкриває нове  
вікно/вкладку на сайт htmlbook.ru</a>
```

Щоб усі посилання відкривались у новому вікні – використовуйте тег **<base>**

```
<head>  
  <base target="_blank">  
</head>  
<body>  
  <p><a href="1.html">Посилання відкриється у новому вікні </a></p>  
  <p><a href="2.html">Посилання відкриється у новому вікні</a></p>  
</body>
```

Відкриття посилання у поточному вікні

Відкриття документу у поточному вікні

`` Посилання відкриється у поточному вікні ``

Щоб усі посилання відкривались у поточному вікні – використовуйте тег `<base>`.

```
<head>
```

```
  <base target="_self">
```

```
</head>
```

```
<body>
```

```
  <p><a href="1.html">Посилання відкриється у поточному вікні</a></p>
```

```
  <p><a href="2.html">Посилання відкриється у поточному вікні</a></p>
```

```
</body>
```

Зображення, тег `img`

Тег **img** не потребує наявності закриваючого тегу

```

```

img – скорочення від **image** (зображення)

src – скорочення від **source** (джерело), може вказувати як на повний, так і відносний шлях

alt - альтернативний опис зображення при відключеному у браузері показі картинок або під час їх завантаження, а також, якщо зображення не може бути відображене (наприклад, якщо воно не існує)

title - короткий опис/назва зображення (не всі браузери відображають альтернативний текст у вигляді підказки)

width - установка ширини зображення у пікселях

height - установка висоти зображення у пікселях

Зображення, тег `img`

`<h2>vertical-align: bottom</h2>`

`<p>This is some text. This is some text.</p>`

`<h2>vertical-align: middle</h2>`

`<p>This is some text. This is some text.</p>`

`<h2>vertical-align: top</h2>`

`<p>This is some text. This is some text.</p>`

`<h2>float: right</h2>`

`<p>This is some text. This is some text.</p>`

`<h2>float: left</h2>`

`<p>This is some text. This is some text.</p>`

vertical-align: bottom

This is some text.  This is some text.


vertical-align: middle

This is some text.  This is some text.


vertical-align: top

This is some text.  This is some text.

float: right

This is some text. This is some text. 


float: left

 This is some text. This is some text.


Зображення, тег `img`

Якщо не
завантажується
малюнок


vertical-align: bottom

This is some text.  Sq This is some text.

vertical-align: middle

This is some text.  Sq This is some text.

vertical-align: top


This is some text.  Sq This is some text.

float: right

This is some text. This is some text.



float: left

 Sq This is some text. This is some text.

Типи зображень

Типи зображень:

- APNG Animated Portable Network Graphics **.apng**
- GIF Graphics Interchange Format **.gif**
- ICO Microsoft Icon **.ico, .cur**
- JPEG Joint Photographic Expert Group image **.jpg, .jpeg, .jfif, .pjpeg, .jpg**
- PNG Portable Network Graphics **.png**
- SVG Scalable Vector Graphics **.svg**

Завжди вказуйте ширину і висоту зображення. Якщо ширина і висота не вказані, при завантаженні зображення сторінка може мерехтіти.

Елементи розмітки таблиць

Елемент **<table>** є **контейнером** для **елементів**, визначаючих контент таблиці.

Усередині контейнеру **<table>** допустимо використовувати наступні елементи: **<caption>**, **<col>**, **<colgroup>**, **<tbody>**, **<td>**, **<tfoot>**, **<th>**, **<thead>** и **<tr>**.

- **<tr>** - скорочення від **table row** (ряд таблиці)
- **<td>** - скорочення від **table data cell** (табличні дані комірок)
- **<th>** - скорочення від **table header cell** (заголовок таблиці), призначений для створення однієї комірки таблиці, яка позначається як заголовна. Текст в такій клітинці відображається браузером зазвичай **жирним** шрифтом. Тег **<th>** повинен розміщуватися всередині контейнера **<tr>**, який в свою чергу розташовується всередині тегу **<table>**.

Таблиці. Атрибути

- **border** - товщина рамки навколо таблиці в пікселях
- **align** - вирівнювання вмісту всієї таблиці по горизонталі, в рядку або в окремій клітинці (**left**, **center** или **right**)
- **valign** - специфікує вирівнювання по вертикалі в комірці (**top**, **middle** или **bottom**)
- **colspan** – скорочення від **column span** (злиття стовпців), використовується в **<td>** для специфікації того, скільки **стовпців** охоплює ця група
- **rowspan** – скорочення від **row span** (злиття рядків), специфікує, скільки **рядів** охоплює ця група

Таблиці. Атрибути

- **background** - задає фоновий малюнок у таблиці
- **bgcolor** – колір фону таблиці
- **bordercolor** – колір рамки
- **cellpadding** – відступ від рамки до вмісту комірки
- **cellspacing** – відстань між комірками
- **cols** - число колонок у таблиці
- **frame** – сповіщує браузер, як відображати границі навколо таблиці
- **height** - висота таблиці
- **width** - ширина таблиці
- **rules** – сповіщує браузер, де відображати границі між комірками
- **summary** – короткий опис таблиці

Таблиці

```
<table>
<tr>
  <td>Cell 1</td>
  <td>Cell 2</td>
</tr>
<tr>
  <td>Cell 3</td>
  <td>Cell 4</td>
</tr>
</table>
```

Cell 1	Cell 2
Cell 3	Cell 4

```
<table>
<tr>
  <th>column1</th>
  <th>column2</th>
</tr>
<tr>
  <td>Cell 1</td>
  <td>Cell 2</td>
</tr>
<tr>
  <td>Cell 3</td>
  <td>Cell 4</td>
</tr>
</table>
```

column1	column2
Cell 1	Cell 2
Cell 3	Cell 4

Spanning Columns

```
<table width="400" cellpadding="10"
cellspacing="5" border="2" bgcolor="#efefef"
bordercolor="red">
```

```
<tr>
```

```
<th width="150"></th>
```

```
<th>07:45-09:20</th>
```

```
<th>09:30-11:05</th>
```

```
<th width="150">11:15-12:50</th>
```

```
<th width="150">13:10- 14:45</th>
```

```
</tr>
```

```
<tr>
```

```
<th>Пн</th>
```

```
<td colspan="2"> Web-t Пз</td>
```

```
<td>ОБД Лк</td>
```

```
<td>ІКС Лк</td>
```

```
</tr>
```

```
<tr>
```

```
<th>Вт</th>
```

```
<td colspan="3">ФВ Пз</td>
```

```
<td>Web-t Лк</td>
```

```
</tr>
```

```
</table>
```

Об'єднані комірки по стовпцям

	07:45-09:20	09:30-11:05	11:15-12:50	13:10-14:45
Пн	Web-t Пз		ОБД Лк	ІКС Лк
Вт	ФВ Пз			Web-t Лк

Spanning Rows

```
<table border="2" bgcolor="#efefef">
  <tr>
    <th></th>
    <th>ABC</th>
    <th>BBC</th>
    <th>CNN</th>
  </tr>
  <tr>
    <th>6pm - 7pm</th>
    <td rowspan="2">Movie</td>
    <td>Comedy</td>
    <td>News</td>
  </tr>
  <tr>
    <th>7pm - 8pm</th>
    <td>Sport</td>
    <td>Current Affairs</td>
  </tr>
</table>
```

Об'єднані комірки по рядкам

	ABC	BBC	CNN
6pm - 7pm	Movie	Comedy	News
7pm - 8pm		Sport	Current Affairs

HTML-коментарі

HTML-коментарі – це частини коду, які **призначені для розробників**, а не для браузера.

Теги коментарів використовуються для вставки коментарів у вихідний код HTML.

`<!-- Write your comments here -->`

Коментарі не відображаються в браузері, але можуть допомогти задокументувати вихідний код HTML.

Тег `<! -- ... -->` може містити будь які інші теги, але заборонено один коментар вкладати в інший!

У тега `<!-- ... -->` відсутні атрибути!

Питання для самоконтролю

Мови розмітки, основні поняття HTML, теги, структура HTML-документу, посилання, форматування, заголовки, списки, таблиці. Стандартні атрибути тегів. Робота з графічними об'єктами та їх розміщенням на сайті.

Дякую за увагу!