

## 10.1. Мова HTML. Поняття, стандарти, теги та атрибути

*Концепція мови розмітки гіпертексту. Сучасні HTML-стандарти: HTML4/5, XHTML. Елементи HTML та їх різновиди, структура тегів. Основні атрибути тегів, класифікація.*

1. Поняття мови розмітки HTML

2. Сучасні стандарти HTML:

- HTML4
- XHTML
- HTML5

3. Консорціум Всесвітньої мережі

4. HTML теги. Структура тегу.

5. HTML теги. Парні та одиночні теги.

6. HTML теги. Блокові та малі теги.

7. HTML теги.

8. Семантичні теги. HTML теги. Класифікація атрибутів:

- Універсальні атрибути
- Унікальні атрибути
- Специфічні атрибути
- Подієві атрибути

### *Поняття мови розмітки HTML*

HTML (HyperText Markup Language) – це не мова програмування (як, наприклад, Python), а так звана мова розмітки. Вона необхідна для повідомлення браузеру способу відображення відвідуваної веб-сторінки. Замість того, щоб показувати суцільний текст, HTML дозволяє обернути його у спеціальні оператори (теги). Це забезпечує особливу поведінку інформації, що передається: вона може виділятися курсивом, жирним текстом, транслюватися у вигляді таблиць, списків, малюнків та інших мультимедіа-даних.

Теги здатні нести як структурний зміст (наприклад, показати, що перед нами таблиця деякої розмірності із заголовком), так і семантичний (виділити інформацію певним чином для пошукових роботів, щоб ті краще індексували сайт, або браузерів).

HTML розробив британець Тім Бернес-Лі наприкінці 1980-х, а як офіційний стандарт HTML функціонує з 1993 року. Ним же опубліковано перший у світі сайт - <http://info.cern.ch/hypertext/WWW/TheProject.html> - працюючий до цього дня. На ньому міститься інформація про технологію World Wide Web, що базується на протоколі HTTP, адресації URI, розмітці HTML.

Сьогодні Тім Бернес-Лі очолює W3-Консорціум (World Wide Web Consortium), основне завдання якого: розробка та впровадження стандартів Інтернету з врахуванням сучасних потреб.

Спочатку HTML був кросплатформним, що створювало певні труднощі. Різні браузери відображали по-своєму теги, деякі елементи могли сприйматися не всіма програмами. Причина: відсутність єдиного стандарту, який підтримує розробник. Особливу проблему доставляв Internet Explorer (Microsoft довгий час наслідувала свої напрацювання в розумінні HTML), що вимагало від розробників врахування різних видів браузерів, щоб вміст сторінок виглядав більш-менш однаковим у користувачів.

В останні роки ця проблема фактично вирішена, оскільки великі гравці на ринку браузерів дійшли згоди. На просторах Інтернету ще можна зустріти сайти, які вважають на старі браузери, але це вже не обов'язкова вимога до верстальників. Найчастіше просто просять оновити свій браузер, щоб побачити сайт у задуманому вигляді.

## *Сучасні стандарти HTML*

Протягом курсу ми дотримуватимемося актуального стандарту HTML5. Він завершений у 2014 році та рекомендований для використання у сучасній веб-розробці. До нього додано нові елементи, що враховують специфіку нинішнього сайтобудування та потребу в мультимедійних компонентах (аудіо, відео, інтерактивність).

До цього 17 років головним стандартом був HTML 4 (з 1997 року). Він зустрічається і сьогодні на низці сайтів. Визначити його на підставі так званого доктайпу.

### Доктайп HTML4

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

Це перший рядок будь-якого HTML-документа, що дозволяє браузерам зрозуміти спосіб роботи з вмістом сторінки. Хоча стандарт і є застарілим, він містить основний набір тегів, якими користуються при створенні сайтів. Важливо врахувати, що частина елементів вже не рекомендована до застосування та згодом буде прибрана.

Незважаючи на строгу типізацію в мові HTML, ряд браузерів досить лояльно ставиться до документів. Так, тег абзацу (позначається `<p></p>`) розробник цілком може написати на свій розсуд:

- `<p>Абзац</p>` (написання відповідно до стандарту);
- `<P>Абзац</P>` (браузери рідко враховують регістр тегів);
- `<p>Абзац` (деякі верстальники можуть забути поставити кінцевий тег, що допускають нові версії практично будь-якого браузера).

Практика показує, що потрібно дотримуватись єдиного стилю. Щоб обмежити вільності програмістів, впровадили стандарт XHTML. Він поєднує HTML та XML формати. Першочергове завдання цього стандарту – заміна HTML 4. Однак після появи HTML5 ця потреба дещо знизилася. Тим не менш, багато розробників використовують його і сьогодні, щоб гарантувати відсутність помилок у написанні коду.

### **Ключові особливості стандарту XHTML:**

- Всі елементи пишуться строго в нижньому регістрі;
- Закриваються навіть одиночні теги (наприклад, `<img />`).
- Основне кодування - UTF-8.

- Всі атрибути записуються тільки в розгорнутому вигляді (наприклад, якщо в HTML5 відключити можливість вибору елемента з випадного списку можна вказати простий атрибут disabled, то в XHTML обов'язково писати disabled="disabled").

Верстати код сайту на основі стандарту XHTML не обов'язково, оскільки HTML5 і сучасні браузери не такі вимогливі, але він може зустрітися (і цього не потрібно лякатися).

#### **Специфіка HTML5:**

- Додано семантичні елементи (<nav>, <section>, <article> та ін), які полегшують читання структури сторінки як розробникам, так і пошуковим машинам.
- Впроваджено підтримку векторної графіки та спеціальних форматів (svg, canvas, MathML).
- Представлено нові елементи управління (dates, email, tel).
- Видалено застарілі теги (big, center, isindex).

### **Консорціум Всесвітньої мережі**

Кількість сайтів у мережі настільки велика, що вже бракує закладеного обсягу IPv4-адрес для їхньої ідентифікації. З кожним роком кількість зростає, що логічно призводить до необхідності стандартизації їх написання та відображення кінцевим користувачем.

Некомерційна організація W3-Консорціум ([www.w3.org](http://www.w3.org)) розробляє єдині принципи та рекомендації для виробників Інтернет-контенту, щоб забезпечити максимальну сумісність між продуктами, платформами, програмами. Це дозволяє робити Всесвітню мережу максимально зручною та універсальною.



Рис.1. W3C-Консорціум

Крім безпосередньо стандартів HTML компанія займається твердженням та інших пов'язаних положень:

- CSS (каскадні таблиці стилів).
- DOM (Об'єктна модель документа).
- PNG (формат зберігання растрових зображень).
- HTTP (протокол передачі).
- RDF (модель уявлення метаданих).
- XPath (мова для доступу до частин XML-документів) та ін.

На сайті консорціуму завжди можна перевірити відповідність коду до стандартів за допомогою валідаторів. У мережі їх є багато і для різних цілей: HTML, CSS, JavaScript. Сервіси дозволяють валідувати свій чи чужий код. Для цього потрібно ввести в пошуковій системі необхідний (наприклад, HTML5 валідатор), вставити туди код і провести перевірку. Усі порушення будуть підсвічені відповідним кольором та підказками. Ними користуються усі розробники для верифікації якості програмістів. Всі сучасні IDE дають можливість валідувати код (у більшості випадків для цього потрібно встановити плагін).

## HTML теги. Структура тега

Тег – окремий HTML елемент, що дозволяє вкладати чи обертати частини документа для певного відображення чи поведінки. За допомогою тегів можна виділяти текст, вставляти зображення та відео, передавати службову інформацію, позначати посилання.

Теги можуть бути доповнені атрибутами, властивостями. З їхньою допомогою можна модифікувати вміст елемента або уніфікувати його.

На даний момент налічується близько 120 тегів, а якщо прибрати застарілі, то й того менше. Їх не так складно запам'ятати, та й кілька десятків, що найчастіше використовуються.

### Типова структура веб-елемента :

Початковий тег (теги подаються в кутових дужках, у початковому тезі, якщо потрібно, зазначаються атрибути).

Кінцевий тег (є не у всіх тегах, ідентифікується косою рисою).

Атрибут, властивість (назва властивості та її значення переважно у подвійних лапках, можливо використовувати і одинарні).

Вміст (внутрішній вміст тегу, зазвичай як текст чи інші вкладені теги).

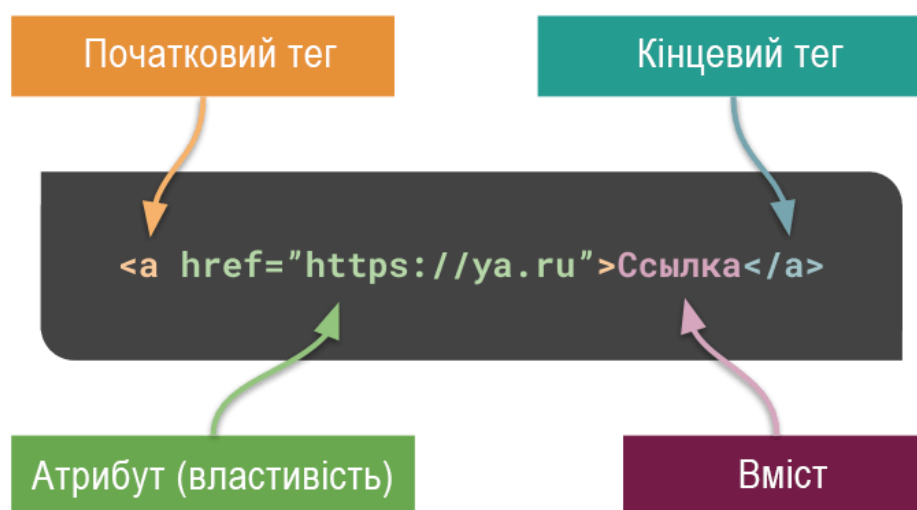


Рис.2. Структура веб-елемента

Подібно виглядає будь-який елемент веб-сторінки. За наявності кількох атрибутів вони поділяються пробілом. Теги не завжди можуть мати властивості або вміст.

Первинне знайомство з різноманітністю HTML-елементів можна здійснити на будь-якому сайті. Якщо ви знаходитесь у браузері Chrome, то прямо зараз можна натиснути клавіші CTRL + U (відкриються Інструменти розробника) і переглянути код будь-якої сторінки. Фактично весь цей

набір тегів і властивостей у поєднанні з JavaScript обробляється браузером, щоб зручно користуватися поточним сайтом.

Сайт є сукупністю таких сторінок, пов'язаних між собою внутрішніми та зовнішніми гіперпосиланнями.

## *HTML теги. Парні та одиночні теги*

Теги бувають парними чи одиночними.

### **1. Одиночні теги**

Одиночних елементів налічується 16 штук. Найчастіше використовувані:

- `<!doctype>` Якась інструкція браузерам для визначення того, до якого типу відноситься документ, яку версію HTML використовувати.
- `<meta>` Необхідний для завдання додаткової інформації про сторінку вказується в заголовку HTML-документа. Як і вищезгаданий тег, не впливає на зовнішній вигляд сторінки, оскільки має службовий характер.
- `<hr>` Задає горизонтальну лінію візуального відділення елементів сторінки, застосовується відділення заголовка від основного тексту, виділення логічних блоків. За умовчанням є сіра лінія товщиною в 1 піксель.
- `<br>` Перенесення нижченаведеного вмісту на новий рядок. Може знадобитися, щоб уникнути довгих полотен тексту. Невеликі шматки інформації сприймаються краще ніж величезний абзац.
- `<img>` Щоб вставити графічний елемент на сторінку. Малюнки можна демонструвати в одному з таких форматів: jpg, png, gif. Саме зображення в документ не вставляється безпосередньо, вказується посилання на нього (локальне або з іншого джерела), а браузер вже завантажує картинку і демонструє користувачеві.
- `<input>` Застосовується у формах і визначає тип полів (можуть бути текстовими, числовими, дозволяють вибирати дату, файл, відзначати ті чи інші елементи списку тощо).

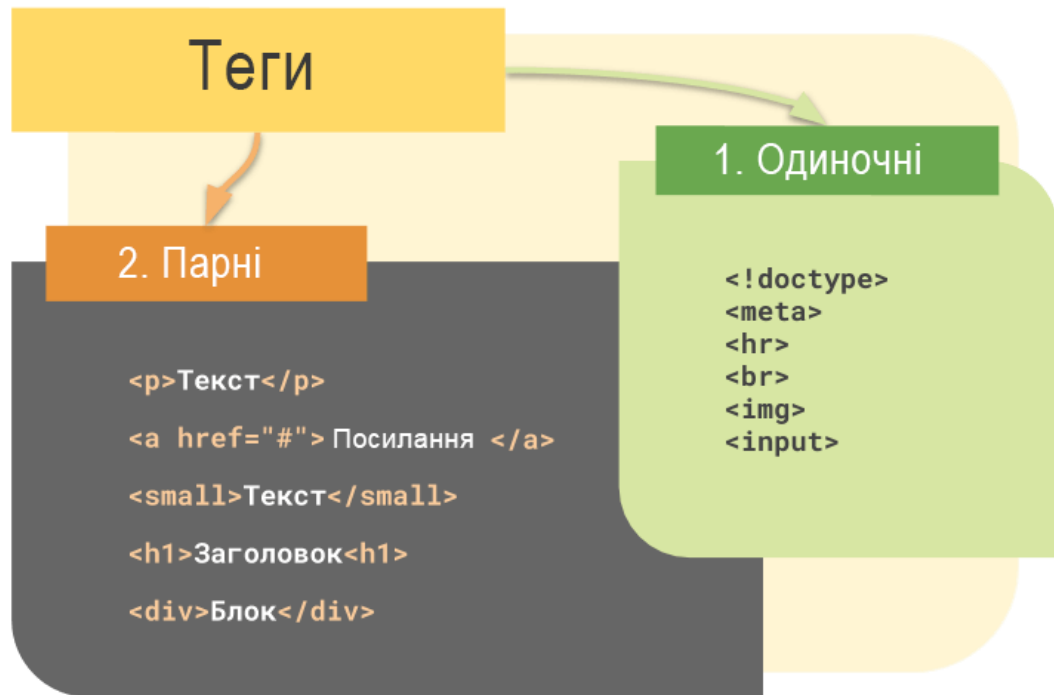


Рис.3. Парні та одиночні теги

## 2. Парні теги

Парні теги обов'язково мають кінцевий тег, і вміст (може бути й порожній). У HTML їх найбільше (близько 100). Як приклад наведемо деякі:

- `<p>Текст</p>` Представляє укладений усередині текст у вигляді окремого абзацу. За умовчанням він має відступ і відокремлюється від попереднього та наступного елементів додатковим розривом (ця поведінка можна змінити).
- `<a href="#">Текст посилання</a>` Використовується для завдання та подання в документі посилання (вона може вести на іншу сторінку сайту, певну ділянку на самій сторінці або будь-яку адресу в Інтернеті).
- `<small>Текст</small>` Робить текст меншого масштабу, ніж основний.
- `<mark>Текст</mark>` Виділяє текст на зразок маркера (аналог того, як на аркушах паперу підкреслюють найважливіше фломастером) жовтим кольором.

За допомогою CSS можна змінювати поведінку практично всіх тегів.

### *HTML теги. Блокові та рядкові теги*

Поведінка будь-якого тегу сьогодні (в стандарті HTML5) легко змінити на власний розсуд. Розподіл елементів на блокові та рядкові склався історично (саме тому вони спочатку відносяться до одного з цих типів). Нині таких типів більше, та їх визначення пов'язані з каскадними таблицями стилів, про які піде детальна мова у одному з наступних уроків.

Блокові теги займають всю ширину сторінки чи батьківського елемента. Якщо ви поставите два таких теги поспіль, то кожен з них буде починатися з нового рядка, абзацу.

Рядкові теги не мають строго розміру. Він залежить від того, скільки символів у них міститься. Декілька поспіль рядкових елементів не будуть розділятися новими рядками, а будуть шикуватися один за одним.

У CSS-властивості відображення тегів можна змінювати за такими типами:

- Блокові (display: block)- отримує властивість блоку, що займає всю ширину сторінки.
- Рядкові (display: inline)- тег стає малим.
- Флекс (display: flex)– елемент набуває властивість особливого типу блоків – флекс (поводиться як блок, але внутрішній вміст гнучко налаштовується).
- Грід (display: grid)– ще один блоковий тип з особливостями внутрішньої поведінки елементів (будуються на підставі «сітки», що складається з рядків та колонок).
- Таблиця (display: table)- тег та його вміст будуть успадковувати властивості таблиць.
- Рядково-блоковий блок (display: inline-block)- блоковий елемент поводить як рядок, але зберігає частину властивостей (можна задати розмір, межі).
- Прихований (display: none)– робить елемент невидимим і повністю видаляє відображення його зі сторінки.

Існують й інші властивості відображення, багато з яких вузько спрямовані та найчастіше експериментальні.

На прикладі двох найяскравіших представників (1 блоковий і 1 рядковий відповідно) подивимося на їхню поведінку.

### **Блоковий тег <div>**

Найпопулярніший блоковий тег часів HTML 4. Основне його завдання – відобразити вміст як блок. Він і сьогодні входить до списку часто використовуваних. Достатньо відкрити код будь-якої сторінки в мережі та проаналізувати його. Майже скрізь застосовується максимально широко. Подивимося на простий приклад.

Приклад - HTML

```
<div>Цей текст займає всю ширину сторінки</div>
```

```
<div>Цей текст розпочнеться з нового рядка</div>
```

Якщо зберегти наведений код у вигляді html-файлу і відкрити в браузері, побачимо, що фрагменти тексту починаються з різних рядків. Причина - по суті блокових елементів (вони прагнуть зайняти всю доступну ширину, яка їм надана).

### **Інші блокові теги:**

- <table></table> Дозволяє створювати таблиці.
- <p>Параграф</p> Позначає параграф тексту.
- <h1>Великий заголовок</h1> Для оформлення заголовка найвищого рівня (в ролі такого зазвичай вибирається основна стаття сторінки).
- <form></form> Відображає на сайті форму (авторизації, реєстрації, опитування та ін.).

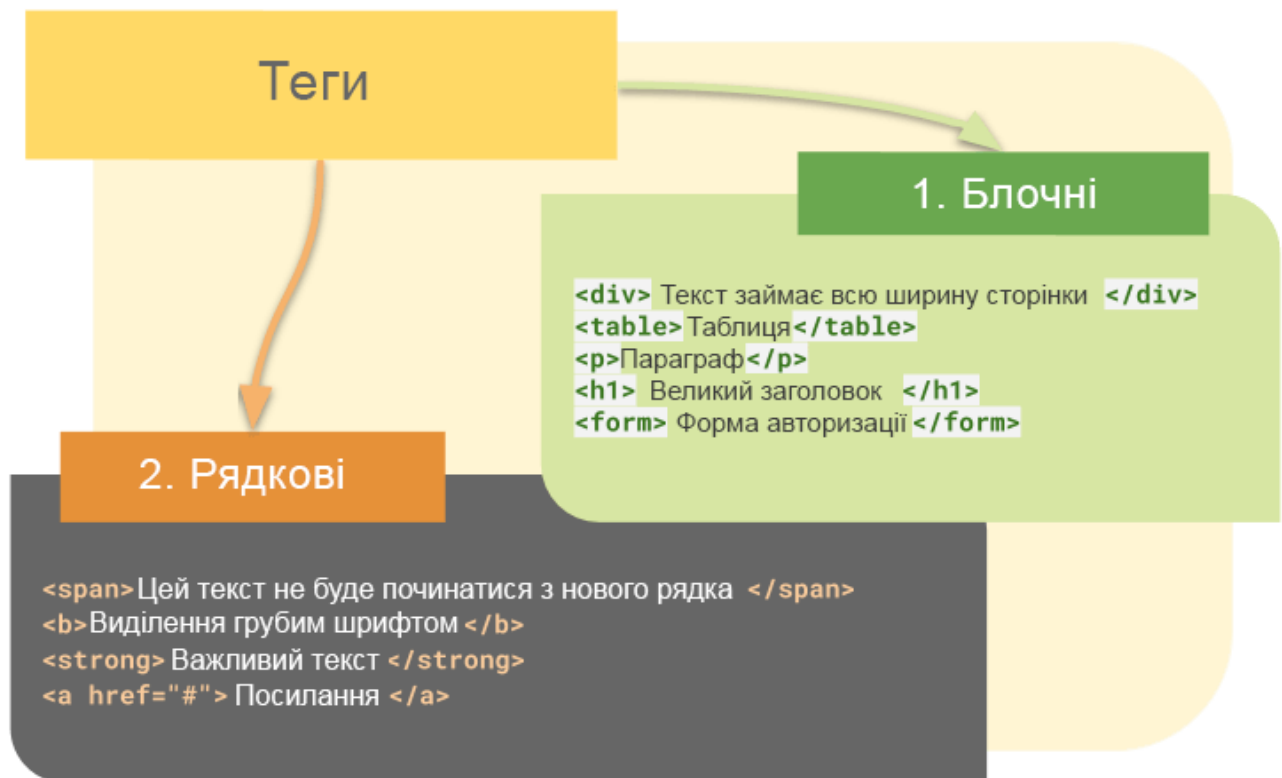


Рис.4. Блокові та рядкові теги

## 2. Рядковий тег <span>

Часто використовуваний рядковий елемент. Фактично він нічого не робить з текстом, але за допомогою атрибутів його вміст можна певним чином виділяти. Важливо й те, що інформація не перериватиметься, не переноситься на новий рядок і не виділятиметься абзацом.

Приклад - HTML

```
<span>Звичайний текст без особливостей.</span>
```

```
<span>Цей текст не почнеться з нового рядка.</span>
```

Друга частина тексту йде за першою без будь-яких розривів. Рядкові елементи займають стільки місця, скільки необхідно для їхнього відображення.

На практиці часто використовуються такі рядкові теги:

- `<b>Виділення жирним шрифтом</b>` Виділяє ділянку вмісту жирним шрифтом без особливої важливості. Тег застосовують з метою візуалізації, щоб інформація сприймалася краще
- `<strong>Важливий текст</strong>` Позначення значної ділянки тексту. Найчастіше виглядає як виділений жирним шрифтом, але сприймається пошуковими системами як такий, що має особливу важливість у наведеному контексті.
- `<a href="#">Посилання </a>` Посилання займає той обсяг на сторінці, який потрібний для демонстрації тексту
- `<sub>`, `<sup>` Необхідні для представлення ступенів та індексів.

Розглянемо приклад:

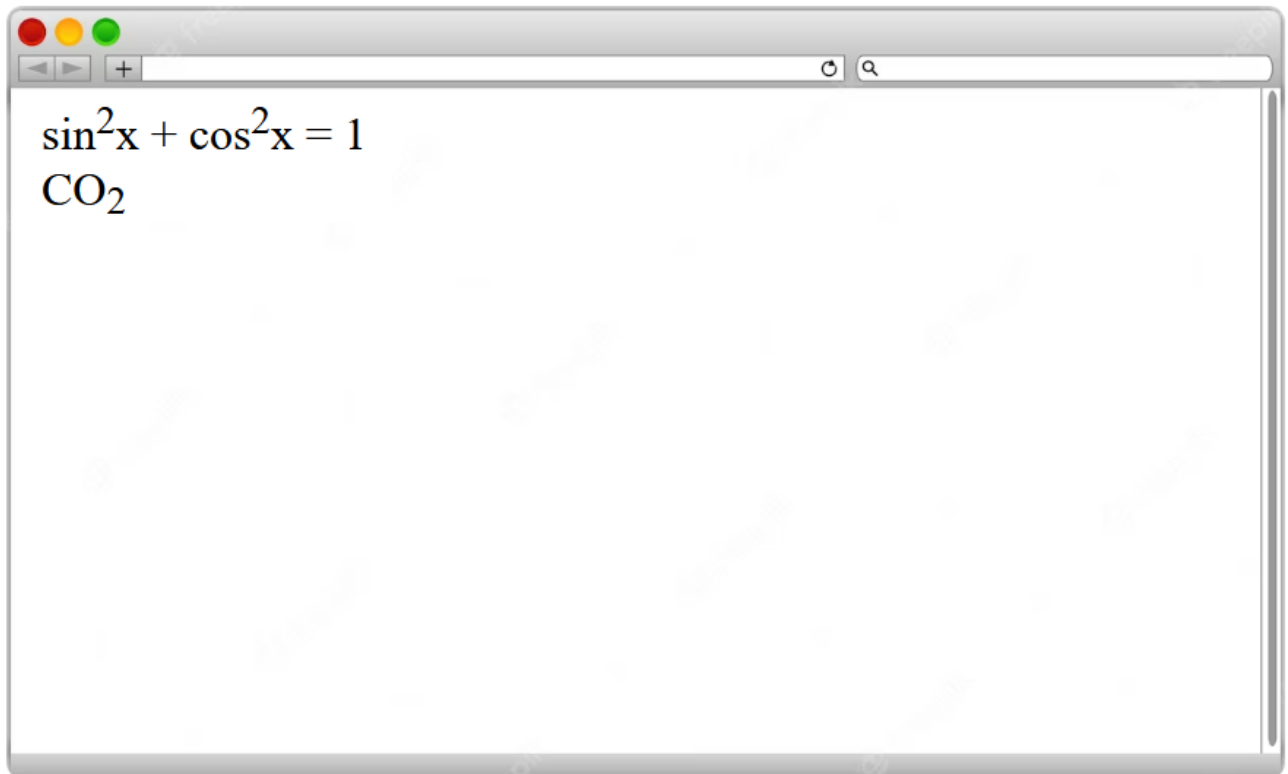


### Приклад - HTML

```
<span>sin<sup>2</sup>x + cos<sup>2</sup>x = 1</span>
```

```
<br>
```

```
<span>CO<sub>2</sub></span>
```



У першому випадку отримуємо тригонометричну тотожність, а в другому – формулу вуглекислого газу.

У HTML5 прямого поділу тегів на блокові та рядкові фактично немає. Зараз говорять про категорії контенту, що міститься всередині елемента.

Контентна модель HTML5 дозволяє більш точно описати типи вмісту, з яким стикаються автори при розробці сайтів. Вона містить такі типи:

- Потік (flow) – відображає основний вміст сторінки (більшість тегів відноситься саме сюди. Наприклад: a, button, header, nav, ol, section).
- Метадані (metadata) - застосовуються в заголовку документа і визначають поведінку іншого вмісту і зв'язок з іншими даними (Приклад тегів: title, meta, script, style).
- Текст (phrasing) – містить безпосередньо текст сторінки та використовувані для форматування теги. Фактично, це рядкові елементи: a, button, i, img, span, textarea.
- Вбудований контент (embedded) - частина текстового, рядкового вмісту, що містить імпортований контент (Приклад тегів: audio, video, img, svg, canvas ).
- Інтерактивні елементи (interactive) – дозволяють відвідувачу сайту взаємодіяти з ресурсом (Приклади тегів: button, select, video, a, input, textarea).
- Заголовки (heading) – для визначення заголовків сайту, статей, підрозділів (теги від h1 до h6).

- Секції (sectioning) – виділення ізольованого контенту на блоки. Містить теги: article, aside, nav, section.

Незважаючи на це, всі елементи з початковими властивостями є рядковими або блочними за своєю суттю.

## *HTML теги. Семантичні теги*

Поява HTML5 дозволила підійти до питання розміщення контенту на сайтах по-новому. До цього вирішувалося лише технічне питання: як зробити сайт, який нормально відображатиметься у різних користувачів з різними пристроями та системами. Так як дана проблема сьогодні не така актуальна (розробники браузерів більш-менш домовилися і прийшли до єдиного стандарту), виникла інша: величезна кількість сайтів ускладнює демонстрацію користувачам максимально релевантних. Немаловажним стало завдання забезпечення доступності ресурсів відвідувачам з обмеженими можливостями (наприклад, з обмеженим зором чи руховим апаратом).

Звичайного набору тегів забракло, знадобився семантичний набір. Звичайний користувач особливої різниці не побачить: сторінки виглядають так само, мають ту саму швидкість завантаження. Однак вирішено важливу проблему: пошукові системи стали краще розуміти інтернет-ресурси, а люди з обмеженими можливостями тепер можуть зручніше користуватися сайтами (так, спеціалізований софт для таких громадян розуміє семантику тегів: заголовки будуть озвучені як заголовки, побічну інформацію з того ж тегу <aside > можна пропустити і т.п.).

Семантичні теги допомагають браузерам, пошуковикам, розробникам. Тепер із проблемою однозначності трактування практично покінчено. Розглянемо найпростіший приклад.

### **Приклад - HTML**

```
<div>Щоб створювати сайти необхідно знати HTML, CSS, JavaScript та серверну технологію (наприклад, Node.js, Python або php)</div>
```

```
<div>Перший сайт створив Тім Бернес-Лі</div>
```

Аналіз вмісту такої сторінки не складний. Багато хто, швидше за все, зрозуміє, що перший блок тексту зачіпає основний вміст розділу, тоді як другий – доповнення, цікавий факт, побічна, але не обов'язкова інформація. А якщо йдеться про більш комплексну ситуацію? Як з'ясувати, яке змістовне навантаження має той чи інший <div>? Фактично ніяк.

Якщо людина ще хоч якось здатна логічно зіставити дані, то машині це не під силу. Та й замість того, щоб оцінювати код, його якість та ефективність, програмісту потрібно витратити час на розуміння смислового навантаження використовуваних елементів. Це дуже незручно.

Семантичні елементи дають змогу спростити таку ситуацію. Розробник допомагає не лише своїм колегам, а й браузерам та пошуковикам. Та й сам сайт суттєво швидше потрапить до цільової аудиторії (адже який-небудь Яндекс показуватиме вашу сторінку не тим, кому цікава біографія засновника W3-Консорціуму, а тим, кому хочеться навчитися створювати сайти. Саме в цьому полягає суть зазначеного вище HTML-коду ).

До основних тегів такого типу відносять:

**1. <header>Вступ</header>**

Призначений для вступного змісту сторінки сайту або статті. Зазвичай тут вказують заголовок, підзаголовок, деякі вступні дані.

## 2. `<nav>Розділи сайту</nav>`

Основне меню сайту містить тег `nav`. Містить посилання розділи сервісу.

## 3. `<footer>Завершення</footer>`

Дозволяє виділити нижній колонтитул статті або розділу сайту. Тут часто підбивають підсумки, пишуть інформацію про автора, резюмують написане.

## 4. `<main>Основний вміст</main>`

Основний контент документа. Якщо йдеться про статтю, то саме тут знаходиться її тіло за винятком вступної та заключної частин.

## 5. `<section>Якийсь блок сайту</section>`

Окремий розділ документа, не обов'язково незалежний. Представляє певну тему з вмістом, які потрібно виділити ізольований блок.

## 6. `<article>Майбутня стаття </article>`

Для виділення незалежного самодостатнього контенту. Тег означає вміст, який несе цінність в загальному контексті сайту. Хорошим прикладом такого виду є пост на форумі або в блозі, окрема стаття сайту, новина.

## 7. `<aside>Побічна інформація</aside>`

Деякі додаткові дані, опосередковано пов'язані з основним змістом сторінки. Наприклад, у блоці `aside` можна розмістити посилання на джерела інформації чи додаткові ресурси (рекламні банери, архів новин тощо).

## 8. `<details>Додаткові відомості</details>`

Довідкова інформація, яку користувач за своїм бажанням може не дивитися. Це блок, що розкривається з коротким текстом (за замовчуванням). Тут можуть містити відомості про автора, деякі уточнення.

## 9. `<summary></summary>`

Надає коротку характеристику вмісту блоку `details`.

### **Приклад - HTML**

```
<details>
```

```
  <summary>W3-Консорціум</summary>
```

```
  Очолюється Тимом Бернес-Лі, розробником мови розмітки HTML.
```

```
</details>
```

Замість кнопки `Детальніше` користувач побачить напис `W3-Консорціум`, натиснувши на який отримає додаткові відомості.

## 10. `<figcaption>Опис</figcaption>`

Коментар до картинки, пояснення, назва;

## 11. <figure></figure>

Виділяє зображення та його опис у єдиний блок.

### Приклад - HTML

```
<figure>
  
  <figcaption>Мій кіт сьогодні задоволений</figcaption>
</figure>
```

У тезі <figure> розміщено картинку кота з поясненнями.

## 12. <time></time>

Дозволяє браузеру та пошуковим системам зрозуміти, що укладені в тег дані є датою.

### Приклад - HTML

Ми працюємо до <time>22:00</time> щодня

Візуально цей елемент себе ніяк не виявляє, проте передбачає семантичний контекст.

## 13. <mark>Виділені дані</mark>

Містить інформацію, яка потребує виділення або підкреслення.

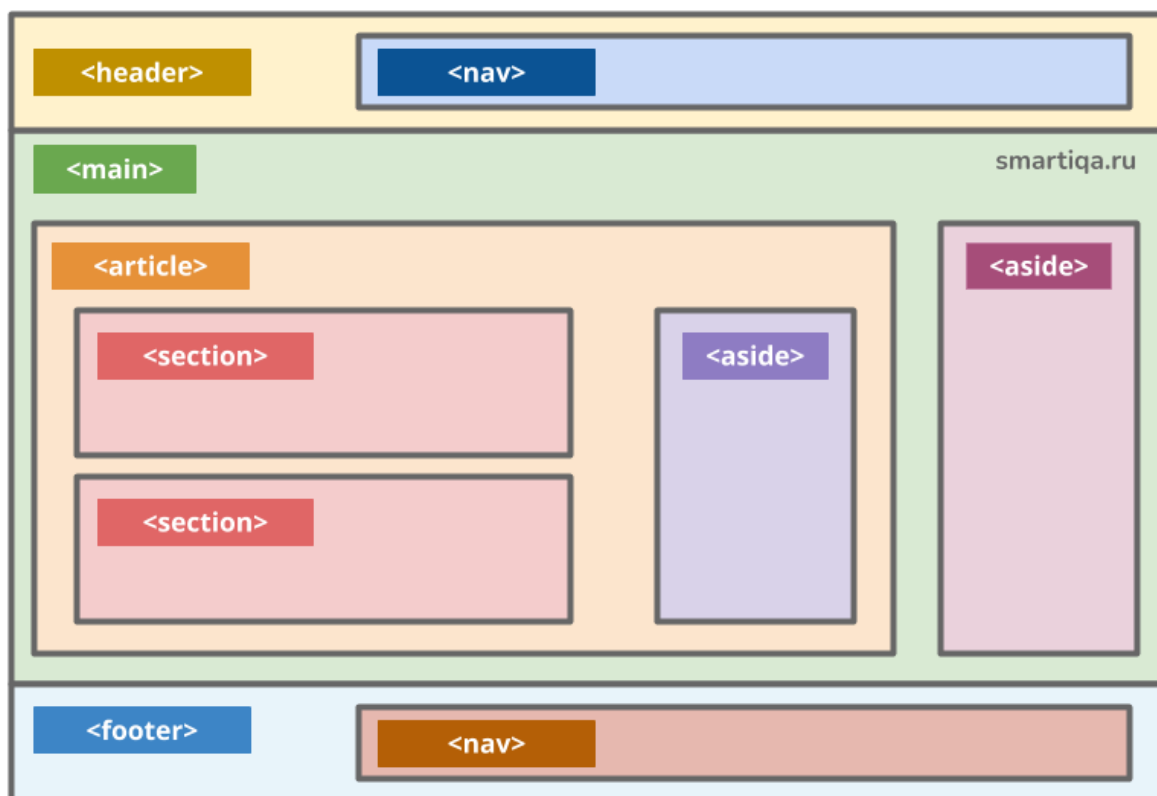


Рис.5. Структура веб-сторінки, побудованої на семантичних тегах

Семантичні теги є новими, введенними в HTML5. Важливо розуміти, що багато старих елементів ставляться по суті до семантичних (<h1>, <strong>, <table>, <form> та інших.).

Ніхто не забороняє створювати сайт без використання семантичних елементів, але його ефективність і рейтинг будуть істотно нижчими від тих, які застосовують нові можливості.

## HTML теги. Класифікація атрибутів

Атрибути або властивості тегів дозволяють розширити функціонал HTML-елементів або надати додаткові відомості про них. Коли йдеться про атрибути, важливо розуміти таке:

- Усі теги мають атрибути (універсальні, унікальні, дієві чи специфічні).
- Вони необхідні доповнення елемента новими особливостями.
- Вони наводяться у тезі, що відкриває, через пробіл.
- Типовий вид атрибута - пара ключ = "значення".

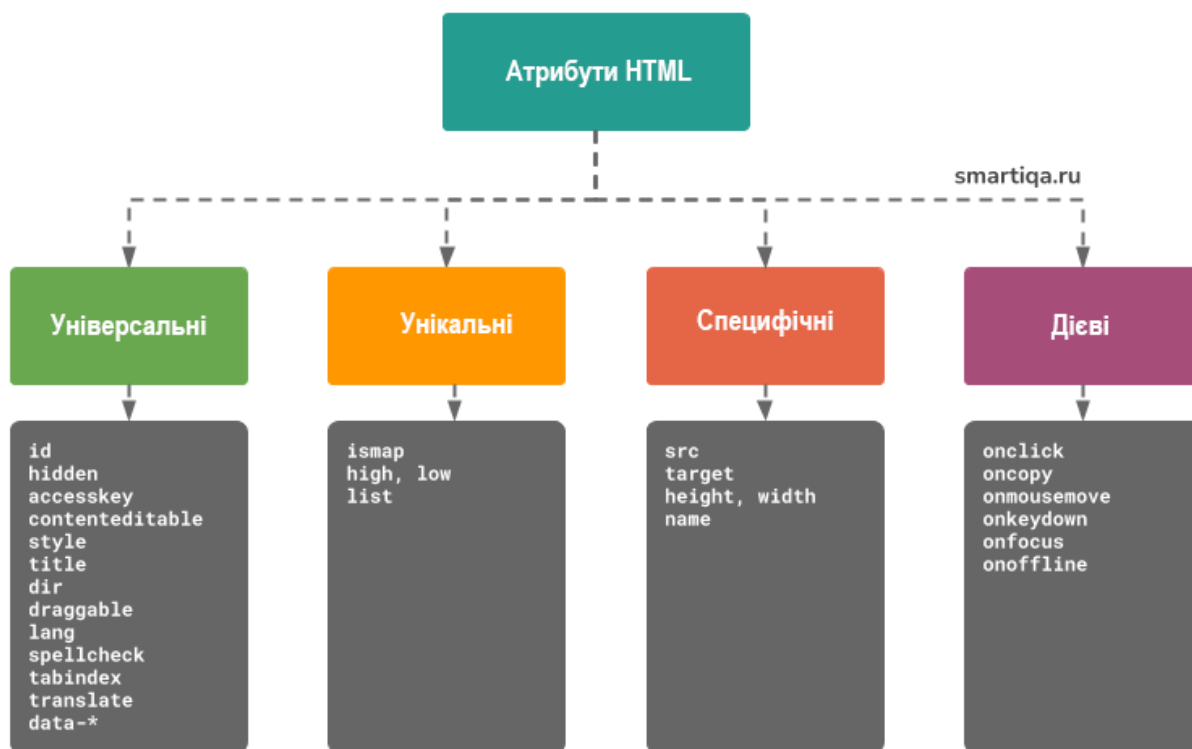


Рис.6. Класифікація атрибутів HTML тегів

### Універсальні атрибути

Виходячи з назви такого роду властивості притаманні кожному тегу в HTML. Ніхто не забороняє «навішувати» їх ліворуч і праворуч, але це не завжди розумно. У будь-якому випадку, якщо універсальний атрибут застосований «не до місця», він не матиме жодної дії, і, що добре, не призведе до помилки.

Атрибут	Опис
<i>class</i>	Вказує на ім'я класу елемента, яким можна звертатися через каскадні таблиці стилів (або через js-скрипт). Класова властивість з тим самим ім'ям може бути у декількох тегів.
<i>id</i>	Вказує на ідентифікатор імені елемента, через який можна отримати доступ до тега. На сторінці може бути лише одне унікальне ім'я ідентифікатора (дозволяє унікалізувати елемент і отримати можливість впливати на нього скриптами, не торкаючись інших тегів).

<i>hidden</i>	Якщо вказати таку властивість, то елемент не відображатиметься на сторінці (інакше він бачить).
<i>accesskey</i>	Як значення вказується кнопка на клавіатурі. Дозволяє отримати доступ до web-елемента – взяти його у фокус або активувати – за певною кнопкою. Працює поки що не завжди добре і має особливості у різних браузерях та операційних системах. Проте активно використовується в пошукових системах.
<i>contenteditable</i>	При виставленому значенні "true" дозволяє редагувати вміст елемента.
<i>style</i>	Застосовується для завдання CSS-властивостей у тезі без використання CSS-файлу.
<i>title</i>	Додаткові відомості про елемент. Якщо навести на нього мишкою, то виводиться підказка з вмістом атрибуту title.
<i>dir</i>	Актуальний для мов з іншим напрямом листа (наприклад, арабської). За умовчанням усі теги мають значення «ltr», тобто зліва направо.
<i>draggable</i> (приймає значення true чи false)	Дозволяє переміщати елемент на сторінці за допомогою миші. Додатково необхідне завдання скрипта, що переміщується. Посилання та картинки мають цю властивість за умовчанням.
<i>Lang</i>	Задає мову вмісту елемента (для англійської, наприклад, має такий вигляд: lang="en").
<i>spellcheck</i> (приймає значення true або false)	Для перевірки орфографії тексту, що вводиться. Актуально для форм і тегів з властивістю contenteditable. Важливо, щоб у браузера була включена ця опція також.
<i>tabindex</i>	Для навігації по елементах сторінки можна скористатися клавішею TAB. Змінити порядок фокусування об'єктів можна через властивість tabindex. Воно набуває числові значення. Якщо перший тег стоїть «2», а другий «1», то при натисканні табуляції спочатку фокус переміститься до другого тегу.
<i>translate</i> (приймає значення yes або no)	Повідомляє браузеру, перекладати чи ні вміст елемента. Сучасні програми дозволяють перекладати вміст сторінки на потрібну вам мову. Це можна примусово заборонити частині контенту, якщо є така потреба. data-*
	Особлива властивість, що дозволяє створювати власні атрибути. Їх можна використовувати як об'єкти js-скриптів. Наприклад, ніхто не забороняє створити такі атрибути: data-lang="python", data-lang="c++", data-lang="javascript".

## Унікальні атрибути

Характерні лише для одного елемента. Наведемо кілька прикладів:

### 1. ismap

Є тільки тег <img> (дозволяє отримувати координати кліка по картинці);

### 2. high, low

Унікальні для тега `<meter>` – для завдання верхньої та нижньої межі певного діапазону в рамках елемента (наприклад, ми показуємо користувачеві відсоток прочитання статті. Він може змінюватись від 0 до 100 %).

### 3. list

Належить до елемента `<input>` – вказує на ID списку, з елементів якого можна вибирати значення для вибраного поля форми.

## Специфічні атрибути

Значна частина атрибутів відноситься не до одного тегу, а до кількох. До HTML5 їх було ще більше, але зараз їх заборонили, оскільки краще задавати через таблиці стилів. Озвучимо ряд з них, які актуальні в HTML5:

### 1. src

Посилання на джерело медіа-файлу (застосовують до тегів: `<audio>`, `<embed>`, `<iframe>`, `<img>`, `<input>`, `<script>`, `<source>`, `<track>`, `<video>` ).

### 2. target

Вказує джерело цільової адреси посилання або валідації форми. Актуально для елементів: `<a>`, `<area>`, `<base>`, `<form>`. Може приймати такі значення:

- `_blank` (відкриється нова вкладка у браузері),
- `_self` (залишимося на поточній вкладці),
- `_parent` (завантаження в батьківський кадр, якщо сторінка на них розділена),
- `_top` (попри наявність кадрів вони всі закриються, а посилання відкриється на все вікно браузера)

### 3. height, width

Визначають висоту та ширину елемента. Застосовні для тегів `<canvas>`, `<embed>`, `<iframe>`, `<img>`, `<input>`, `<object>`, `<video>`. Значення - позитивні числа (трактуються як пікселі). Можна використовувати й відсотки – вважаються щодо батьківського елемента.

### 4. name

Для завдання імені елемента. Застосовується для наступних тегів: `<button>`, `<fieldset>`, `<form>`, `<iframe>`, `<input>`, `<map>`, `<meta>`, `<object>`, `<output>`, `<param>`, `<select>`, `<textarea>`. За цією назвою до об'єкта можна звертатися через JavaScript, наприклад.

## Дієві атрибути

Пов'язані з JavaScript - подіями. При взаємодії користувача з елементами ці події можуть спрацьовувати та змінювати поведінку об'єктів. Їх практичне використання буде розглянуто в темі JavaScript. Зараз лише озвучимо деякі з них для розуміння:

### 1. onclick

Скрипт спрацьовує при натисканні мишкою по елементу (можна викликати спливаюче вікно, змінити розмір або колір, скерувати до іншої сторінки тощо).

### 2. oncopy

Викликає скрипт при спробі скопіювати вміст елемента (в такий спосіб часто забороняють копіювати вміст або додають посилання на першоджерело, коли ви вставляєте скопійований текстовий редактор).

### **3. onmousemove**

ефект при русі миші (дозволяє створювати цікаві ефекти на сторінці: зміщення, збільшення, тряску тощо).

### **4. onkeydown**

Обробка натискання певної кнопки.

### **5. onfocus**

Зміна поведінки об'єкта при фокусі на ньому (наприклад, коли поставили курсор у полі для введення логіну, саме поле може змінити колір, стати більше, дати якусь підказку).

### **6. onoffline**

Поведінка сторінки в момент відсутності активного підключення до мережі (зручно, коли під рукою немає Інтернету, а сайт продовжує функціонувати, хоч і в обмеженому режимі).

Як видно, атрибути тегів істотно підвищують можливості HTML. Вони дозволяють не просто додавати нові характеристики та змінювати властивості елементів, але й дають можливість використовувати фактично необмежений потенціал JavaScript.

## ***Контрольні запитання***

### **1. Який стандарт HTML сьогодні є найактуальнішим?**

З 2014 діє рекомендований стандарт HTML5. Остання його версія 5.2 представлена наприкінці 2017 року. Версія 5.3 знаходиться у розробці та вже доступна для ознайомлення.

### **2. Чим відрізняється XHTML від HTML? Наведіть приклад.**

XHTML – суворіша версія HTML4. Основні її особливості, що призводять до відмінності зі стандартом HTML:

обов'язкове закриття навіть одиночних тегів; всі атрибути записують у повній формі, без скорочень; імена елементів і властивостей обов'язково представлені в нижньому регістрі; основне кодування UTF-8 (в HTML – ISO 8859-1); – облік будь-яких помилок у коді (браузеру заборонено «додумувати» та виправляти промахи розробників).

### **3. Які медіаформати можна використовувати на веб-сторінках?**

В даний час HTML підтримує багато медіа-форматів. Основні :

- jpg; - png; - jpg; - svg;
- потокове відео;
- mp3; - wav; - ogg; - mp4; - WebM.



**4. Наведіть та опишіть будь-які 2 рядкових теги. Для цього скористайтесь офіційною документацією.**

Рядкові теги:

`<b>` – виділяє текст жирним шрифтом. Приклад: `<b>Текст</b>` ;

`<s>` – робить текст перекресленим. Приклад: `<s>Текст</s>`.

**5. Наведіть приклад 2 одиночних тегів та опишіть їх.**

Одиночні теги:

`<base>` – базова адреса сторінки, від якої будуються всі відносні посилання. Приклад:  
`<base href="https://smartiqa.ru/">` ;

`<link>` – для завантаження зовнішніх файлів, найчастіше css. Приклад: `<link href="main.css">`.

**6. Опишіть 3 будь-які дієві атрибути, не розглянуті в теоретичній частині.**

Наведемо приклад трьох нерозглянутих дієвих атрибутів:

`onresize` - спрацьовує при зміні розмірів вікна браузера;

`onblur` - Викликає скрипт в момент втрати фокусу елементом;

`ondblclick` – обробляє подвійний клік мишею.

## **Контрольні завдання**

Знайдіть та виправте всі помилки в коді (використано лише вивчені елементи). Знати структуру иуи-сторінки поки не обов'язково – завдання більше на уважність.

### **Завдання – IDE**

```
<footer>Успіхів у освоєнні</footer>
<section>
  <H1>Мови програмування</H1>
  <figure>
    
    <figcaption> Логотип Python </figcaption>
  </figure>
  <article>
    <p>Вивчати Python дуже весело</p>
    <p>Але потрібно читати та практикуватися</p>
    <p>I сильно-сильно старатися</p>
  </article>
</section>
<header>Дізнайся на всі 100%!</header>
<navigation>
  <a href="#">Головна сторінка </a><br>
```

```
<a href="#">Розділ 1</a><br>
    Розділ 2
</navigation>
```

### **Відповідь на завдання**

Оскільки ми ще не вивчали структуру html-документа, завдання полягало в уважності та логіці. Отже, в коді можна побачити такі помилки :

1. Тег <h1> написано у верхньому регістрі. Хоч браузер і зрозуміє вас, але стилістика коду не повинна страждати.
2. Тег <p> в одному з абзаців не закрито.
3. Теги <header> та <footer> – семантичні. Перший належить до вступу, а другий – нижньому колонтитулу. Їх необхідно поміняти місцями.
4. Тегу <navigation> немає, є тег <nav>. Зазвичай його розміщують вгорі сторінки, але ніхто не забороняє розміщувати його і внизу.
5. Тег <img> вказує не на картинку, а на сайт, що не так. Та й підказка нижче переконує нас у помилці.
6. Перші 2 розділи в меню навігації повернуті на посилання, а третій ні, що потрібно підкоригувати.

### **Рішення - HTML**

```
<header>Дізнайся на всі 100%!</header>

<section>

    <h1>Мови програмування</h1>

    <figure>

        <figcaption>Логотип Python</figcaption>

    </figure>

    <article>

        <p>Вивчати Python дуже весело</p>

        <p>Але потрібно читати та практикуватися</p>

        <p>І сильно-сильно старатися</p>

    </article>

</section>

<nav>

    <a href="#">Головна сторінка</a><br>

    <a href="#">Розділ 1 </a> <br>

    <a href="#">Розділ 2 </a> <br>

</nav>
```

<footer>Успіхів у освоєнні</footer>

### *Джерела інформації*

1. HTML: Поняття, стандарти, теги і атрибути <https://smartiga.ru/courses/web/lesson-2-html>