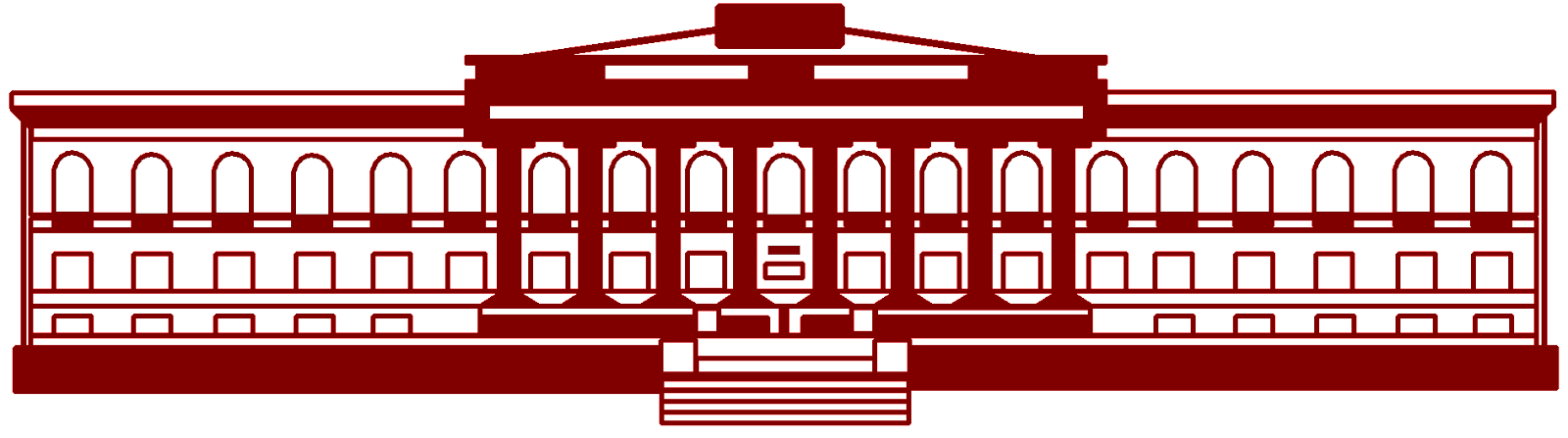
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

**Кафедра прикладних інформаційних систем**



**Звіт**

до виконання лабораторної роботи №1

з дисципліни «**Веб-технології і програмування сайтів**»

на тему:

« Знайомство з HTML тегами, Верстка елементів »

**Виконано:**

студ. групи ПП- 31, підгрупа 2

Шкандюк Анною Леонідівною

**Перевірено:**

Пирог М.В.

**Київ – 2023**

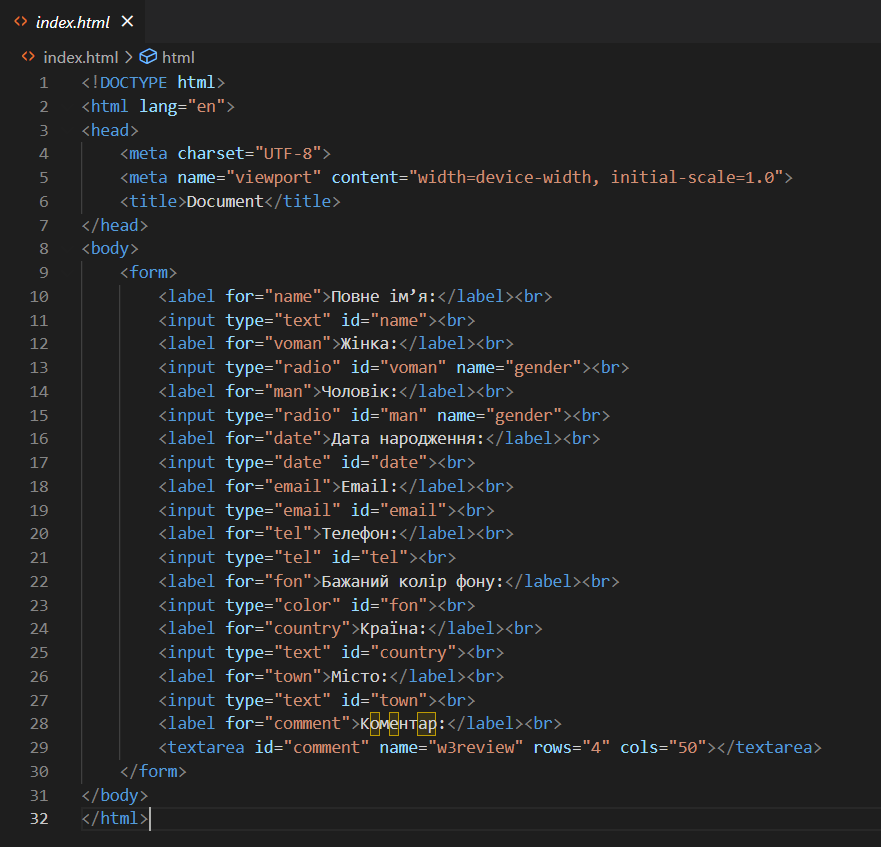
1. **Мета роботи:**

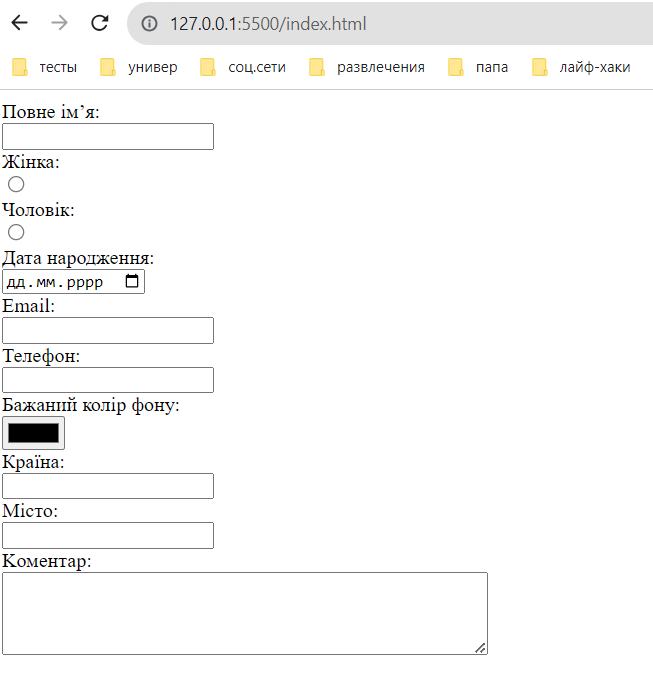
Отримати навички створення простих HTML блоків, детально вивчити теги HTML 5.

1. **Завдання та їх виконання:**

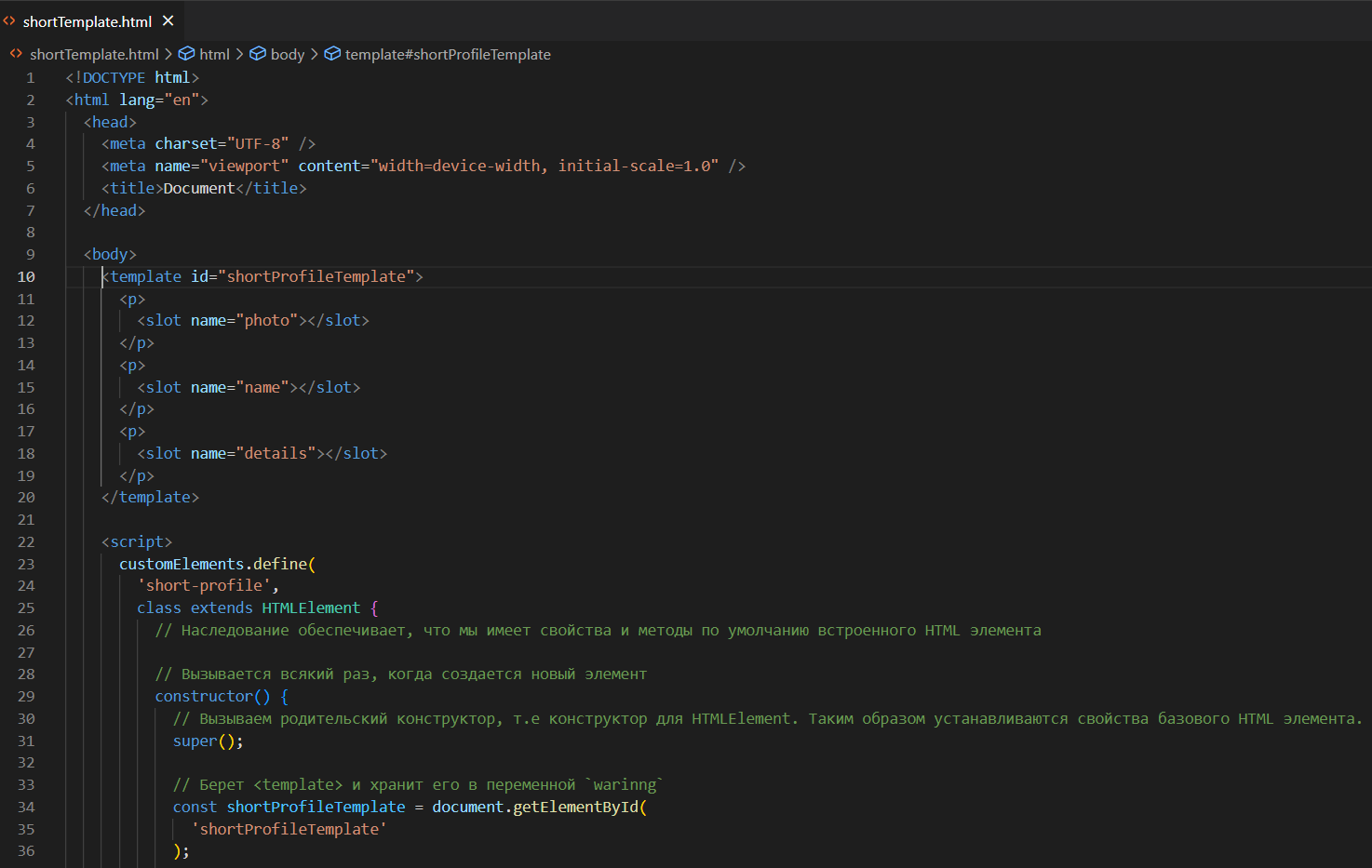
*(примітка: завдання кожного з пунктів оформлюйте окремим html файлом без використання css)*

**Завдання 1**. Створити html файл з формою для додавання викладача. Форма повинна містити наступні поля: Повне ім’я, стать, дата народження, email, телефон, бажаний колір фону, країна, місто, коментар.



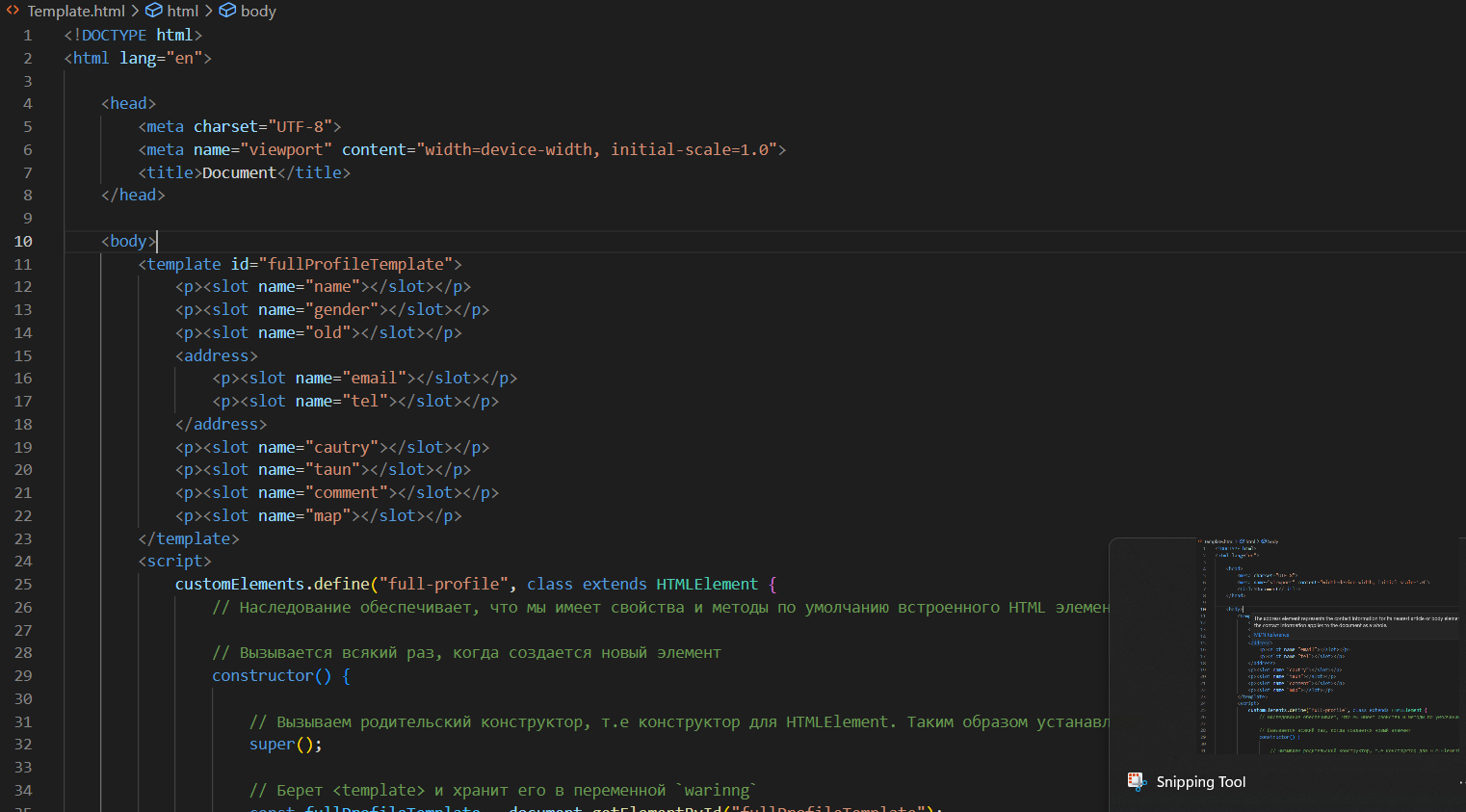
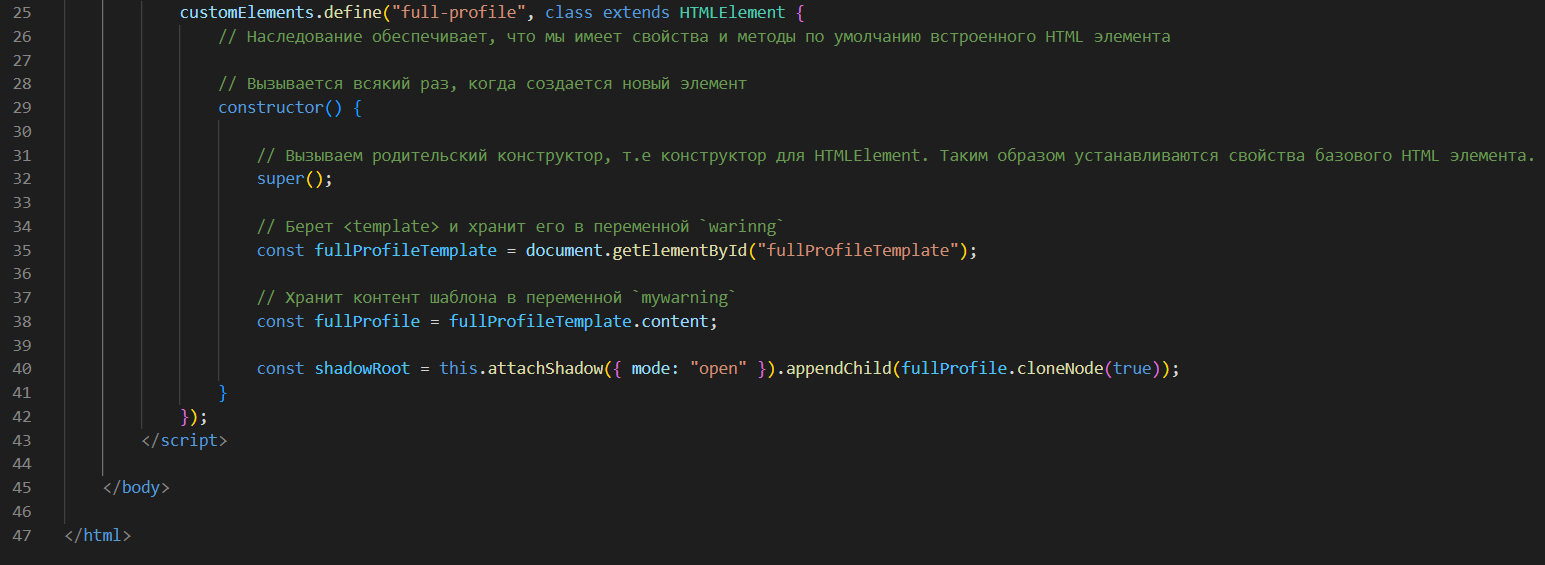


**Завдання 2**. Створити html файл з компонентом «компактний варіант картки викладача». Компонент повинен містити лише зображення та повне ім’я + "докладніше".

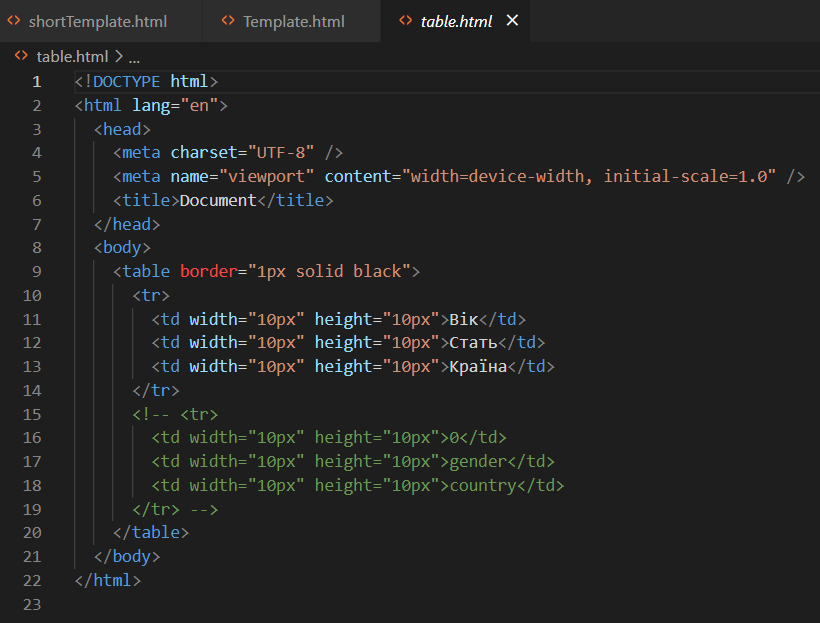
 

**Завдання 3**. Створити html файл з компонентом «повна версія картки викладача» з полями: Повне ім’я, стать, вік, email, телефон, країна, місто, коментар та карта. Додати до компонента data-attribute з кольором фону.

Виконуючи завдання використовувати відповідні семантичні теги.

**Завдання 4.** Створити html файл з таблицею статистики по наступним полям: вік, стать, країна



1. **Висновки до роботи**

За результатами проробленої роботи можемо сказати, що отримали навички створення простих HTML блоків та детально вивчили теги HTML 5.

1. **Контрольні запитання**
2. Гіпертекст. Мова HTML. Стандарти HTML. Структура HTML документу.

*Гіпертекст* - це система посилань між різними елементами інформації на веб-сторінках. Він дозволяє користувачам переходити з одного документу на інший, а також з однієї частини документу на іншу.

*Мова HTML* (HyperText Markup Language) - це стандартна мова розмітки для створення структури веб-сторінок. HTML використовується для визначення різних елементів на сторінці, таких як заголовки, абзаци, зображення, посилання та інші.

*Стандарти HTML* - це правила та специфікації, які визначають, яким чином має бути написаний код HTML для того, щоб браузери могли правильно інтерпретувати і відображати веб-сторінки. Найвідоміші стандарти HTML - це HTML5, HTML4, XHTML тощо.

*Структура HTML документу* - це спосіб, за яким розташовані різні елементи в HTML файлі. Зазвичай структура HTML документу виглядає наступним чином:

<!DOCTYPE html>

<html>

<head>

<title>Заголовок сторінки</title>

</head>

<body>

<header>

<h1>Заголовок сайту</h1>

</header>

<nav>

<ul>

<li><a href="#">Посилання 1</a></li>

<li><a href="#">Посилання 2</a></li>

</ul>

</nav>

<main>

<article>

<h2>Заголовок статті</h2>

<p>Текст статті...</p>

</article>

</main>

<footer>

<p>&copy; 2023 Ваша компанія</p>

</footer>

</body>

</html>

Ця структура включає основні елементи, які розташовані на більшості веб-сторінок, такі як заголовок, меню, основний контент та підвал.

1. Поняття HTML тегу та атрибутів тегів.

*HTML теги* - це основні будівельні блоки веб-сторінки, які визначають, як браузер повинен відображати різні частини сторінки. Кожен тег має відкриваючий та закриваючий елемент (зокрема, за винятком тегів, які не мають контенту, таких як `<img>`).

Наприклад:

<p>Це параграф з текстом.</p>

<a href="https://www.example.com">Посилання на прикладний сайт</a>

У цьому прикладі `<p>` та `<a>` - це HTML теги.

*Атрибути* - це додаткова інформація, яка може бути додана до тегів для зміни їхньої поведінки або вигляду. Кожен тег може мати один чи кілька атрибутів. Атрибути завжди вказуються в парі з назвою тегу та мають значення в подвійних або одинарних лапках.

Приклад з атрибутом:

<img src="image.jpg" alt="Опис зображення">

У цьому прикладі `src` та `alt` - це атрибути тега `<img>`. `src` вказує шлях до зображення, а `alt` надає альтернативний текст для випадку, якщо зображення не може бути завантажене.

Розуміння використання тегів та їхніх атрибутів є ключовим для створення вірного вигляду та функціональності веб-сторінок.

1. <! DOCTYPE> декларація. Різниця між типами документів.

`<!DOCTYPE>` декларація в HTML вказує браузеру, який тип документа відображається на сторінці і як слід інтерпретувати код HTML на сторінці. Ця декларація розміщується в самому початку HTML-документа, перед будь-яким іншим кодом.

Різні типи документів (DOCTYPE) визначають стандарти та правила, за якими браузери повинні відображати сторінку. Основні типи документів включають:

HTML5:

<!DOCTYPE html>

HTML5 є останньою версією HTML та включає нові теги, атрибути та можливості для веб-розробки.

HTML 4.01:

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

HTML 4.01 включав різні варіанти (строгий, перехідний, рамковий), які вказували на різні рівні стандартів.

XHTML 1.0:

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

XHTML (Extensible HyperText Markup Language) був більш суворим відносно правил розмітки, та пов'язаним з XML.

Кожен тип документа має свої властивості, правила та обмеження. Зазвичай, використання `<!DOCTYPE>` декларації з поточними стандартами допомагає забезпечити коректне відображення веб-сторінки у різних браузерах.

HTML5 став стандартом веб-розробки, і для більшості проектів рекомендується використовувати `<!DOCTYPE html>` для вказання типу документа.

1. Теги логічного та фізичного форматування тексту.

Логічне форматування вказує на змістовні аспекти тексту, тоді як фізичне форматування зосереджується на зовнішньому вигляді тексту. Ось декілька прикладів тегів для обох видів форматування:

Логічне форматування:

1. `<em>` - Наголошений текст (зазвичай відображається нахиленим шрифтом): <p>This is <em>emphasized</em> text.</p>

2. `<strong>` - Важливий текст (зазвичай відображається жирним шрифтом):

<p>This is <strong>strong</strong> text.</p>

3. `<cite>` - Цитата джерела (зазвичай відображається курсивним шрифтом):

<p><cite>Book Title</cite> by Author Name</p>

Фізичне форматування:

1. `<b>` - Жирний текст: <p>This is <b>bold</b> text.</p>

2. `<i>` - Курсивний текст: <p>This is <i>italic</i> text.</p>

3. `<u>` - Підкреслений текст: <p>This is <u>underlined</u> text.</p>

1. Типи списків HTML. Аттрибути тегів списку.

HTML надає декілька типів списків, які можна використовувати для структурування і представлення інформації на веб-сторінці. Основні типи списків - це невпорядкований список (`<ul>`), впорядкований список (`<ol>`) та список визначення (`<dl>`). Кожен з цих типів має свої атрибути та використання.

*Невпорядкований список* (`<ul>`):

Використовується для створення списку, в якому пункти не мають конкретного порядку.

<ul>

<li>Перший пункт</li>

<li>Другий пункт</li>

<li>Третій пункт</li>

</ul>

*Впорядкований список* (`<ol>`):

Використовується для створення списку з упорядкованими пунктами.

<ol>

<li>Перший пункт</li>

<li>Другий пункт</li>

<li>Третій пункт</li>

</ol>

*Список визначення* (`<dl>`):

Використовується для створення списку, в якому кожен пункт має визначення або опис.

<dl>

<dt>Термін 1</dt>

<dd>Опис терміну 1</dd>

<dt>Термін 2</dt>

<dd>Опис терміну 2</dd>

</dl>

Атрибути для тегів списку допомагають додатково налаштовувати їх вигляд та поведінку. Основні атрибути включають:

1. `type` для `<ol>` - визначає тип нумерації (наприклад, `type="1"`, `type="a"`, `type="A"`, `type="i"`, `type="I"`).

2. `start` для `<ol>` - вказує початкове число для нумерації списку.

3. `reversed` для `<ol>` - вказує, що нумерація має бути в зворотньому порядку.

4. `value` для `<li>` - встановлює конкретне значення для елемента списку.

1. Зображення та посилання в HTML.

Зображення та посилання - це важливі елементи в HTML, які дозволяють візуально збагачувати та надавати функціональності веб-сторінкам.

*Зображення* (`<img>`):

Для вставки зображення використовується тег `<img>`. Головний атрибут для цього тега - це `src`, який вказує шлях до зображення.

<img src="image.jpg" alt="Опис зображення">

- `src` - вказує шлях до зображення.

- `alt` - надає альтернативний текст, який буде відображений, якщо зображення не може бути завантажене.

*Посилання* (`<a>`):

Для створення посилання використовується тег `<a>`. Головний атрибут цього тега - це `href`, який вказує URL-адресу, на яку веде посилання.

<a href="https://www.example.com">Текст посилання</a>

- `href` - вказує URL-адресу, на яку буде веде посилання.

Посилання зі зображенням:

Можна поєднати зображення та посилання, створивши посилання зі зображенням.

<a href="https://www.example.com">

<img src="image.jpg" alt="Опис зображення">

</a>

Це створить зображення, яке буде функціонувати як посилання, переводячи користувача на задану URL-адресу після кліку на зображення.

1. Таблиці в HTML.

Таблиці в HTML використовуються для організації даних у вигляді рядків і стовпців. Таблиці можуть містити дані, текст, зображення та інші елементи. Основними тегами для створення таблиць є `<table>`, `<tr>`, `<th>` та `<td>`. Ось як це працює:

Основна структура таблиці:

<table>

<tr>

<th>Заголовок стовпця 1</th>

<th>Заголовок стовпця 2</th>

<th>Заголовок стовпця 3</th>

</tr>

<tr>

<td>Дані 1</td>

<td>Дані 2</td>

<td>Дані 3</td>

</tr>

<tr>

<td>Дані 4</td>

<td>Дані 5</td>

<td>Дані 6</td>

</tr>

</table>

- `<table>` - основний контейнер для таблиці.

- `<tr>` - рядок таблиці.

- `<th>` - заголовок стовпця (зазвичай відображається жирним шрифтом).

- `<td>` - комірка даних (звичайний текст).

Об'єднання комірок:

<table>

<tr>

<th>Заголовок 1</th>

<th colspan="2">Заголовок 2</th>

</tr>

<tr>

<td>Дані 1</td>

<td rowspan="2">Дані 2</td>

<td>Дані 3</td>

</tr>

<tr>

<td>Дані 4</td>

<td>Дані 5</td>

</tr>

</table>

- `colspan` - об'єднання горизонтальних комірок.

- `rowspan` - об'єднання вертикальних комірок.

1. Meta-теги в HTML.

*Мета-теги* (`<meta>`) в HTML використовуються для надання додаткової інформації про веб-сторінку. Вони зазвичай розташовуються в розділі `<head>` вашого HTML-документу. Мета-теги надають метадані, які впливають на відображення, поведінку та інші аспекти сторінки. Ось деякі загально використовувані мета-теги:

1. Задання коду символу та кодування: Ці мета-теги використовуються для вказання кодування символів вашої сторінки. <meta charset="UTF-8">

2. Опис сторінки: Цей мета-тег надає короткий опис сторінки, який може використовуватися пошуковими системами.

<meta name="description" content="Опис вашої сторінки">

3. Ключові слова: Цей мета-тег допомагає вказати ключові слова, які відображають зміст сторінки.

<meta name="keywords" content="ключові слова, ваша тема, інші слова">

4. Автор: Цей мета-тег дозволяє вказати автора сторінки.

<meta name="author" content="Ім'я автора">

5. Авторські права: Цей мета-тег дозволяє вказати авторські права сторінки.

<meta name="copyright" content="© 2023 Ваша компанія">

6. Viewport: Цей мета-тег визначає відображення сторінки на різних пристроях і розмірах екрану.

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Це лише декілька прикладів мета-тегів. Вони дозволяють вам надати додаткові відомості про вашу веб-сторінку для пошукових систем, браузерів та інших інструментів.

1. HTML форми.

HTML форми використовуються для збору від користувачів інформації, яка може бути надіслана на сервер для обробки. Форми дозволяють створювати різні типи полів введення, вибору та інші елементи, які користувач може заповнити та відправити. Ось основні елементи форми та їхні атрибути:

Основна структура форми:

<form action="обработчик.php" method="POST">

<!-- Елементи форми тут -->

<input type="text" name="username" placeholder="Введіть ім'я">

<input type="submit" value="Відправити">

</form>

- `action` - вказує URL-адресу або шлях до обробника форми на сервері.

- `method` - визначає метод передачі даних (зазвичай `POST` або `GET`).

Елементи форми:

1. `<input>` - використовується для створення різних полів введення, таких як текстові, пароль, чекбокси тощо.

<input type="text" name="username" placeholder="Введіть ім'я">

<input type="password" name="password" placeholder="Введіть пароль">

<input type="checkbox" name="subscribe" id="subscribe">

<label for="subscribe">Підписатися на розсилку</label>

2. `<textarea>` - використовується для великих поля введення тексту.

<textarea name="comments" placeholder="Ваші коментарі"></textarea>

3. `<select>` та `<option>` - використовується для створення випадаючого списку.

<select name="country">

<option value="us">Сполучені Штати</option>

<option value="ca">Канада</option>

</select>

4. `<button>` - використовується для створення кнопок управління формою.

<button type="submit">Відправити</button>

5. `<label>` - використовується для створення текстової мітки для полів введення.

<label for="username">Ім'я:</label>

<input type="text" name="username" id="username">

1. Стандарт a11y

Стандарт a11y, також відомий як "Web Content Accessibility Guidelines" (WCAG), є набором рекомендацій та стандартів, які визначають, як зробити веб-сайти та веб-додатки більш доступними для людей з різними видами обмежень. Цей стандарт розробляється і підтримується консорціумом W3C (World Wide Web Consortium) з метою поліпшення доступності веб-ресурсів.

WCAG містить рекомендації та критерії для створення веб-сайтів, які більш дружні до користувачів з різними видами обмежень, такими як відчуття, моторика, когнітивні здібності тощо. Стандарт включає чотири основних принципи доступності, які можна абревіювати як POUR:

1. *Perceivable* (Загальновідомий): Інформація та контент повинні бути доступні для всіх користувачів, незалежно від їхніх здібностей. Це означає, що контент повинен бути сприйняттям для різних відчуттів (зору, слуху, дотику тощо).

2. *Operable* (Придатний до користування): Веб-сайт повинен бути придатним для взаємодії різних користувачів, включаючи тих, хто використовує допоміжні технології. Це включає навігацію, фокус та взаємодію елементів.

3. *Understandable* (Зрозумілий): Інформація та функціонал повинні бути представлені зрозуміло та логічно, щоб користувачі могли легко розуміти, як взаємодіяти з сайтом.

4. *Robust* (Стійкий): Веб-сайт повинен бути стійким до змін, технологічних оновлень та допоміжних технологій, які використовуються користувачами з різними потребами.

WCAG поділено на три рівні доступності: A, AA та AAA, де AAA - найвищий рівень доступності. Керуючись стандартами WCAG, веб-розробники можуть створювати веб-сайти, які доступні для більш широкого кола користувачів, забезпечуючи рівні доступу та взаємодії для всіх.