# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ Кафедра програмного забезпечення автоматизованих систем

## МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

## до виконання лабораторних робіт з дисципліни «WEB-ПРОГРАМУВАННЯ» частина 1

для здобувачів освітнього ступеня «бакалавр» за спеціальністю 121 – Інженерія програмного забезпечення, освітня програма «Інженерія програмного забезпечення» галузі знань 12 - Інформаційні технології усіх форм навчання усіх форм навчання

Затверджено вченою радою ФІТІС, протокол №\_\_\_\_ від \_\_\_\_\_2024 р., згідно з рішенням кафедри програмного забезпечення автоматизованих систем, протокол №\_\_\_\_ від \_\_\_\_\_2024 р.

## Упорядники: Куницька С.Ю., к.т.н., доцент Метелап В.В., к.т.н., доцент Немченко В.В., к.т.н., доцент

Рецензент: Захарова В.М., к.т.н., доцент кафедри комп'ютерної інженерії та інформаційних технологій ЧДБК

Методичні рекомендації до виконання лабораторних робіт з дисципліни «WEB-ПРОГРАМУВАННЯ» частина 1 для здобувачів освітнього ступеня «бакалавр» за спеціальністю 121 – Інженерія програмного забезпечення, освітня програма «Інженерія програмного забезпечення» галузі знань 12 - Інформаційні технології усіх форм навчання [Електронний ресурс] / [упорядники: С.Ю. Куницька, В.В. Метелап, В.В. Немченко] ; М-во освіти і науки України, Черкас. держ. технол. ун-т. Черкаси : ЧДТУ, 2024. - 96 с. – Назва з титульного екрана.

Предметом вивчення навчальної дисципліни є знання з того, що таке WEB та принципи WEB-програмування, застосування та використання різних підходів та інструментарію для розробки та проектування ПЗ, сучасних WEB-систем та їх взаємодія з базами даних, web-додатків, web-сторінок

Навчальне електронне видання мережного використовування

## методичні рекомендації

## до виконання лабораторних робіт з дисципліни «**WEB-ПРОГРАМУВАННЯ**» **Частина 1**

для здобувачів освітнього ступеня «бакалавр» за спеціальністю 121 – Інженерія програмного забезпечення, освітня програма «Інженерія програмного забезпечення» галузі знань 12 - Інформаційні технології усіх форм навчання усіх форм навчання

> Упорядники: Куницька Світлана Юріївна Метелап Володимир Володимирович Немченко Вадим Вячеславович

> > В авторській редакції

## **3MICT**

ВСТУП	4
ХІД РОБОТИ	5
ЗАВДАННЯ ДО ЛАБОРАТОРНИХ РОБІТ	6
Лабораторна робота №1	6
Лабораторна робота №2	8
Лабораторна робота №3	10
Лабораторна робота №4	12
Лабораторна робота №5	14
Лабораторна робота №6	17
Лабораторна робота №7	20
Лабораторна робота №8	22
Лабораторна робота №9	25
Лабораторна робота №10	29
Лабораторна робота №11	38
Лабораторна робота №12	42
Лабораторна робота №13	48
Лабораторна робота №14	56
Лабораторна робота №15	64
Лабораторна робота №16	73
Лабораторна робота №17	
Лабораторна робота №18	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	95

#### ВСТУП

Програма навчальної дисципліни «**WEB-програмування**» з циклу професійної підготовки обов'язкових дисциплін складена з урахуванням вимог до підготовки здобувачів вищої освіти рівня бакалавр за освітньопрофесійною програмою «Інженерія програмного забезпечення».

Предметом вивчення навчальної дисципліни є знання з того, що таке WEB та принципи WEB-програмування, застосування та використання різних підходів та інструментарію для розробки та проектування ПЗ, сучасних WEB-систем та їх взаємодія з базами даних, web-додатків, webсторінок.

Мета та завдання навчальної дисципліни - забезпечити здатність студентів розуміти та засвоїти основні принципи WEB-розробки для побудови ПЗ, набуття практичних навичок по створенню ПЗ з використанням сучасних засобів програмування, а також детально розглядаються методи і механізми:

– ознайомлення з основними поняттями та принципами WEBпрограмування, засобами і проблемами розробки ПЗ;

– опанування підходами для проектування та розробки складного програмного забезпечення для WEB.

#### ХІД РОБОТИ

Лабораторні роботи виконуються послідовно. Для виконання лабораторної роботи необхідно ознайомитись з темою метою та завданням на роботу і після цього приступити до ознайомлення з теоретичними відомостями і інформацією в електронному вигляді, що вказана нижче в літератури, та будь-якою додатково за необхідності для списку виконанням завдання. Після виконання практичної частини завдання необхілно показати викладачу виконане практичне завдання або відправити йому за вказаною адресою для перевірки і після цього ознайомитись із питаннями до роботи і відповісти на них. Після цього необхідно скласти від 10 особистих питань по темі. Якщо викладач перевірив і затвердив правильність виконання особисто студентом практичної задачі, необхідно скласти звіт за наступною послідовністю:

- 1. Теоретичне завдання
- 2. Ескіз
- 3. Блок-схеми
- 4. Реалізація
- 5. Тестування
- 6. Розгортання
- 7. Література, яка була використана у ході виконання роботи.
- 8. Відповіді на контрольні запитання.

Якщо студент виконав лабораторну під час заняття або вдома та показав її викладачу до кінця наступного заняття, то не обов'язково формувати звіт з роботи. Якщо студент запізнився з виконанням завдання на одне заняття, то необхідно сформувати звіт в електронному вигляді і подати викладачу на перевірку і виконане практичне завдання і звіт з роботи. За умови перевищення термінів здачі роботи на два заняття студент повинен зробити звіт з роботи та затвердити його у викладача і після цього подати в роздрукованому вигляді додатково до електронного.

#### ЗАВДАННЯ ДО ЛАБОРАТОРНИХ РОБІТ

#### Лабораторна робота №1

**Тема роботи:** Основи HTML: Вивчення і застосування ключових елементів і тегів.

Мета роботи: Дослідити та засвоїти основні принципи роботи з HTML та використання ключових елементів та тегів для створення WEBсторінок.

#### Завдання

1. Створіть базовий HTML-документ із заголовком «Моя улюблена страва».

2. Створити сторінку та додати три абзаци тексту з коротким описом вашого хобі.

3. Створити сторінку, вставити зображення вашого улюбленого кіноактора та короткий опис про нього.

4. Створити сторінку та залишити на ній посилання на іншу WEB-сторінку, яка містить інформацію про ваш улюблений фільм.

5. Створіть заголовок четвертого рівня з текстом «Мої улюблені книги».

6. Створити сторінку та вставити зображення логотипу вашого університету або місця, де ви навчаєтеся.

7. На сторінці, яка створена вами додати посилання на ваш профіль в соціальних мережах.

8. Додайте на сторінку зображення аудіофайлу з вашим улюбленим музичним треком та посилання на неї з текстом «Моя улюблена пісня».

9. Створіть WEB-сторінку зі списком книг, які ви хочете прочитати. Кожен пункт списку повинен містити назву книги та інформацію про автора.

10. Вставте на сторінку зображення вашого улюбленого місця для

відпочинку.

11. За допомогою абзаців додайте на сторінку короткий текст про ваші враження від останньої книги, яку ви прочитали.

12. Створіть WEB-сторінку з заголовком «Мої враження від останньої подорожі». Додайте абзаци, в яких коротко описано ваші враження від кожного місця, яке ви відвідали.

13. На створеному сайті додайте заголовок «Мої улюблені цитати». У кожному абзаці наведіть одну з цитат та коротко розкажіть, чому вона вас надихає. Вставте зображення, що ілюструє кожну цитату.

14. Створіть заголовок «Моя улюблена книга». В абзаці розкажіть, чому вона вам так сподобалася. Додайте зображення обкладинки книги та посилання на її огляд.

15. Створити WEB-сторінку, яка містить інформацію про вашого улюбленого героя з книги, фільму чи коміксу. Використовуйте зображення та текст для опису цього персонажу.

#### Контрольні питання

1. Що таке HTML і яка її основна функція?

2. Як починається HTML-документ, і що означає декларація <!DOCTYPE>?

3. Які рівні заголовків існують у HTML, і яка їхня значущість?

4. Які елементи використовуються для створення абзаців тексту на WEB-сторінці?

5. Які атрибути використовуються для створення гіперпосилань на інші WEB-сторінки?

6. Як вставити зображення на WEB-сторінку за допомогою елементу <img> і які атрибути використовуються для цього?

7. Чому важливо використовувати атрибут alt для зображень у HTML?

**Тема роботи:** Вплив атрибутів стилю HTML на візуальне оформлення WEB-сторінок.

**Мета роботи:** Дослідити вплив різних атрибутів стилю HTML, таких як колір фону, колір тексту, шрифти, розмір тексту та вирівнювання тексту, на візуальне оформлення WEB-сторінок.

#### Завдання

1. Додайте атрибут стилю до елементу , щоб задати **червоний** колір тексту.

2. Задайте фоновий колір блоку <div> у вигляді шістнадцяткового коду #FFA500.

3. Додайте атрибут стилю до заголовка <h1>, щоб встановити розмір шрифту 24 ет.

4. Встановіть розмір тексту для елементу <**p**> у відсотках, наприклад, **150%**.

5. Додайте атрибут стилю до списку , щоб встановити розмір шрифту **18 пікселів**.

6. Встановіть шрифт Arial для елементу <div> за допомогою властивості font-family.

7. Додайте атрибут стилю до елементу <**a**>, щоб змінити шрифт на **Verdana**.

8. Задайте фоновий колір для елементу **<body>** у вигляді назви кольору, наприклад, **«lightblue**».

9. Встановіть розмір тексту для посилань <a> у відсотках, наприклад, 120%.

10. Задайте горизонтальне вирівнювання тексту по центру для елементу <div>.

11. Додайте атрибут стилю до елементу <div>, щоб задати

фоновий колір у вигляді назви кольору, наприклад, «lavender».

12. Задайте для елементу **<h3>** колір тексту у вигляді назви кольору, наприклад, **«darkorange»**.

13. Встановіть сірий колір тексту для заголовка другого рівня <h2>.

14. Додайте атрибут стилю до елементу <a>, щоб змінити колір тексту на RGB значення, наприклад, **rgb**(255, 0, 0).

15. Установити колір фону для двох різних елементів <h1> та , один колір повинен бути шістнадцяткового коду, а інший у вигляді rgb.

#### Контрольні питання

1. Які атрибути використовуються для задання кольору фону та тексту в HTML?

2. Як можна встановити кольори фону та тексту безпосередньо в HTML за допомогою атрибуту style?

3. Які можливості є для задання шрифтів в CSS?

4. Як можна встановити шрифт безпосередньо в HTML за допомогою атрибуту style?

5. Які одиниці виміру використовуються для визначення розміру тексту в CSS?

6. Як можна встановити розмір тексту безпосередньо в HTML за допомогою атрибуту style?

7. Які можливі варіанти вирівнювання тексту в CSS?

8. Як можна вирівняти текст по центру на HTML сторінці?

9. Чому використання вирівнювання по обох боках (justify) може бути корисним у деяких випадках?

**Тема роботи:** Вплив елементів форматування тексту HTML на візуальне оформлення контенту WEB-сторінок.

Мета роботи: Дослідити різні елементи форматування тексту HTML та їх вплив на структуру та вигляд WEB-сторінок для покращення ефективності комунікації з користувачами.

#### Завдання

1. Створіть абзац тексту, який містить слово «важливо» у жирному шрифті.

2. Створіть просту WEB-сторінку, де використовуйте тег **<b>** для виділення заголовку.

3. Написати цитату улюбленого письменника та чому вона вам подобається. Цитату виділити курсивним шрифтом за допомогою <i>.

4. Створіть рядок з датою чи іншою важливою інформацією, використовуючи тег **<small>** для зменшення розміру тексту.

Написати коротку розповідь на тему «Моє улюблене свято».
Використовуючи тег <strong>, щоб підкреслити важливість певної інформації.

Створіть три хімічних формул та їх опис, використовуючи тег
<sub> для підрядкових індексів.

7. Покажіть експонент у математичних формулах, використовуючи тег **<sup>** для верхніх індексів. Назви формул повинні бути виділенні курсивом.

8. Створіть будь-який текст та додайте у ваш документ тег **<del>** для показу змін у тексті, які були видалені.

9. Вставте фразу або речення, використовуючи тег **<ins>** для показу змін у тексті, які були додані.

10. Підкресліть важливу у тексті дату за допомогою тегу **<еm>**.

11. Підкресліть технічний термін або іншомовну фразу або назву чогось за допомогою тега **<i>**, у вашому тексті за темою «Мої друзі».

12. Вставте фразу або слово, яке потрібно виділити у тексті, використовуючи тег **«mark»** та зробіть його іншого кольору.

13. Вставте фразу або слово, яке бажаєте підкреслити, використовуючи тег **<ins>** та надайте йому спеціальний стиль (наприклад, курсив або жирний шрифт).

14. Додайте у ваш документ рядок з хімічною формулою, використовуючи тег **<sub>** для підрядкового індексу та тег **<sup>** для надрядкового індексу. Виділіть ключове слово у цьому рядку за допомогою тега **<strong>**.

15. Вставте у ваш документ фразу або речення зі змінами за допомогою тегів **<del>** та **<ins>**, щоб показати, що було видалено та додано. Підкресліть важливу дату у цьому ж абзаці за допомогою тега **<em>**.

#### Контрольні питання

1. Яка різниця між елементами  $\langle b \rangle$  i  $\langle strong \rangle$  в HTML?

2. Для чого призначені елементи <i> i <em> в HTML?

3. Що означає елемент <small> в HTML і як він використовується?

4. Коли варто використовувати елемент <mark> в HTML і як він впливає на текст?

5. Як використовувати елементи <del> i <ins> в HTML для позначения вилученого та вставленого тексту відповідно?

6. Для чого призначений елемент <sub> в HTML і в яких ситуаціях його використання найбільш доцільне?

7. Як використовувати елемент <sup> в HTML і в яких випадках його використання є необхідним?

**Тема роботи:** Використання HTML-тегів для структурування та підвищення інформативності цитат на WEB-сторінках.

**Мета роботи:** Дослідити різні способи використання HTML-тегів, зокрема <blockquote>, <q>, <abbr>, <address>, <cite> та <bdo>, для відображення цитат на WEB-сторінках. Проаналізувати вплив кожного з цих тегів на структуру та вигляд цитат, а також їхню інформативність та зручність для користувачів.

#### Завдання

1. Створіть WEB-сторінку з двома цитатами: одна довша цитата за допомогою **<blockquote>**, а інша – коротка цитата з використанням **<q>**.

2. Додайте абревіатури до вашого WEB-сайту, використовуючи елемент **<abbr>**. Переконайтеся, що кожна абревіатура має атрибут title, який надає пояснення до скорочення.

3. Створіть блок контактної інформації на сторінці вашого WEBсайту, використовуючи елемент **<address>**. Додайте адресу, номер телефону та електронну пошту.

 Використайте елемент <cite> для відображення назви книги, фільму або музичної композиції на вашому WEB-сайті, назва повинна знаходитися під фотографією.

5. Створіть сторінку зі списком абревіатур та їх визначень, де кожна абревіатура буде оформлена за допомогою елемента **<abbr>**.

6. Розмістить на сторінці список цитат різної довжини: короткі цитати в <q> та довші цитати в <blockquote>.

7. Додайте на сторінку перелік назв книг, використовуючи елемент **<cite>**. Кожна назва книги повинна бути оформлена як цитата.

8. Розмістить на сторінці довідковий посібник з абревіатурами, де кожна абревіатура відображається за допомогою елемента **<abbr>**, а при наведенні миші на неї відображається пояснення.

9. Створіть блок з інформацією про автора книги або статті, використовуючи елемент **<address>** для відображення його контактних даних.

10. Створіть галерею книг, фільмів або альбомів, де кожна назва буде відображена як цитата за допомогою <**cite**>.

11. Створіть WEB-сайт з базовими словами арабської мови, де використовується елемент **<bdo>** для відображення арабського тексту зліва направо для полегшення читання користувачем.

12. Розмістить на сторінці посилання на ресурси з довідковою інформацією, де кожне посилання включає абревіатуру та її розшифровку, відображені за допомогою елемента **<abbr>**.

13. Створіть блок з текстом, який містить вірш на арабській мові. Застосуйте елемент **<bdo>** для відображення кожного рядка вірша з право наліво.

14. Створіть сторінку зі списком авторів та назв книг, де кожне ім'я автора та назва книги будуть відображені як цитата за допомогою елемента <cite>.

15. Розмістить на сторінці список контактів для кожного відділу компанії, використовуючи елемент **<address>** для форматування контактної інформації.

#### Контрольні питання

 Які основні відмінності між елементами <blockquote> та <q> для відображення цитат?

2. Що таке елемент  $\langle abbr \rangle$  у HTML і які цілі він виконує?

3. Як можна використовувати елемент <address> для форматування контактної інформації на WEB-сторінці?

4. Для чого призначений елемент <cite> у HTML і як його слід використовувати?

5. Що означає та як використовується елемент <bdo> y HTML?

6. Яка основна функція атрибута title в елементі <abbr>?

**Тема роботи:** Вплив кольорів на WEB-дизайн та сприйняття користувачами.

Мета роботи: Вивчити вплив різних кольорів на фоні, тексті та рамці на сприйняття та емоційний стан користувачів для оптимізації WEBдизайну та покращення їхнього досвіду.

#### Завдання

1. Створіть просту HTML-сторінку з декількома абзацами тексту. Встановити різні кольори тексту для кожного абзацу, використовуючи різні методи визначення кольорів: HTML, RGB, HEX, HSL, RGBA та HSLA.

2. Створіть HTML-сторінку з декількома блоками <div>. Встановити різні кольори фону для кожного блоку, використовуючи різні методи визначення кольорів: HTML, RGB, HEX, HSL, RGBA та HSLA.

3. Створіть HTML-сторінку з декількома елементами, наприклад, блоками <div> або абзацами тексту. Потрібно встановити різні рівні прозорості для кожного елементу, використовуючи методи визначення прозорості: RGBA та HSLA.

4. Створити HTML-сторінку, де кожен блок представляє певний колір з колірної палітри. Колір можна вибрати як заздалегідь визначений назвою HTML, так і власноруч заданий за допомогою HEX, RGB, HSL, RGBA або HSLA.

5. Створити свою унікальну HTML-сторінку, де використовуються кольори для вираження певної теми або настрою. Наприклад, можна створити сторінку для сезону, кольорову тему для фотографій або тематичну сторінку.

6. Створити список кольорів, де кожен елемент списку має назву кольору поряд зі своїм НЕХ або RGB значенням. Вони можуть

використати нумерацію або маркери списку для цього завдання.

7. Створіть WEB-сторінку з різними типами шрифтів, такими як звичайний, курсивний, жирний. Кожен тип шрифту повинен мати власний колір тексту.

8. Розробіть сторінку із списком ресторанів або кафе, де кожен заклад має свій власний колір фону та тексту.

9. Створити HTML-сторінку із зображеннями, де кожне зображення буде мати власний колір рамки. Додайте підпис під кожним зображенням з відповідним кольором тексту.

10. Створіть WEB-сторінку, на якій різні заголовки матимуть власний колір, що буде відрізнятися від кольору тексту звичайного абзацу.

11. Створити WEB-сторінку, на якій будуть різні абзаци тексту. Кожен абзац повинен мати власний колір фону та колір тексту.

12. Створіть WEB-сторінку зі списком посилань на різні сторінки. Кожне посилання має мати власний колір тексту.

13. Створіть WEB-сторінку з меню на день та вставити зображення до нього, де кожна назва страви меню має власний колір тексту, а зображення кольорову рамку.

14. Розробіть сторінку з оглядом книг або фільмів, де кожна назва фільму або книги має фон та жирний шрифт, а огляд має власний колір тексту та рамки.

15. Реалізуйте WEB-сторінку зі списком продуктів, де кожен продукт має своє зображення та власний колір фону.

#### Контрольні питання

1. Скільки стандартних назв кольорів підтримує HTML?

2. Які аспекти роботи з кольором фону важливі в WEB-дизайні?

3. Для чого використовується колір рамки в HTML?

4. Що таке HEX коди кольорів і як вони використовуються?

5. Які можливості надає RGBA значення для встановлення

кольорів в HTML?

6. Що означає абревіатура HSL і як вона використовується для визначення кольорів?

7. Що таке альфа-канал в контексті кольорів RGBA і HSLA?

8. Що означає абревіатура RGB і які ці основні складові?

9. Які є способи вказання кольору в HTML?

10. Як виглядає код для встановлення червоного кольору в значеннях RGB?

11. Як можна встановити чорний колір, використовуючи значення RGB?

12. Що означає альфа-канал у значеннях кольору, і як він використовується у RGBA та HSLA?

13. Що відрізняє значення кольору в форматах HEX, RGB та HSL?

**Тема роботи:** Використання CSS для визначення стилю та вигляду WEB-сторінок.

Мета роботи: Вивчити різні способи використання CSS для надання стилю та оформлення WEB-сторінок, такі як вбудований, внутрішній та зовнішній CSS. Дослідити можливості CSS щодо кольорів, шрифтів та розмірів, визначення границь, відступів та полів для кращого вигляду та організації контенту.

#### Завдання

1. Створіть WEB-сторінку з різними заголовками (<h1>, <h2>, <h3>) і застосуйте до них різні кольори тексту за допомогою CSS.

2. Створіть блок тексту (<div>) і застосуйте до нього властивості внутрішніх відступів та рамки.

3. Змініть фоновий колір сторінки за допомогою CSS.

4. Змініть шрифт усього тексту на сторінці, використовуючи CSS.

5. Створіть WEB-сторінку із заголовком (<h1>) та застосуйте до нього стиль червоного кольору тексту за допомогою атрибута HTML style.

6. Визначте внутрішній CSS для сторінки, де всі параграфи () матимуть шрифт Arial та розмір тексту 16 пікселів.

7. Посилання на зовнішній файл CSS. Створіть зовнішній файл CSS з назвою «styles.css» та зв'яжіть його з вашою WEB-сторінкою за допомогою елемента <link>.

8. Визначте у вашому файлі HTML елемент <head> та вкажіть в ньому загальний стиль для всієї сторінки, де фон буде блакитним.

9. Застосуйте внутрішній стиль до тексту вашої WEB-сторінки, де всі посилання (<a>) матимуть фіолетовий колір тексту.

10. Створіть блок тексту (<div>) і надайте йому рамку шириною 2 пікселі та кольором #333333, а також внутрішній відступ 20 пікселів.

11. Візуалізуйте властивість CSS margin у вашій WEB-сторінці, де

кожен параграф () матиме зовнішній margin по 10 пікселів та власний колір.

12. Створіть блок тексту (<div>) і надайте йому зелений колір фону та синю рамку за допомогою CSS.

13. Створіть зовнішній файл CSS з назвою «styles.css» та зв'яжіть його з вашою WEB-сторінкою за допомогою елемента <link>. В цьому файлі задайте стилі для заголовків (<h1>, <h2>, <h3>), а також для абзаців (), де кожен тип елементу матиме власний колір тексту і розмір шрифту. Розділіть стилі для кожного елементу за допомогою коментарів у файлі CSS, щоб було зрозуміло, який стиль застосовується до кожного елементу.

14. Додайте WEB-сторінці зображення і задайте йому рамку та внутрішній відступ, але зробіть так, щоб margin була встановлена лише знизу (наприклад, 20 пікселів знизу, а інші сторони мають margin 0).

15. Створіть блок тексту <div> і застосуйте до нього margin з усіх сторін 10 пікселів, рамку товщиною 1 піксель і бірюзовим кольором, а також внутрішній відступ 5 пікселів з усіх сторін.

#### Контрольні питання

1. Що таке CSS і яке призначення вона має у WEB-розробці?

2. Назвіть основні концепції CSS та поясніть їх значення.

3. Які способи додавання CSS до документів HTML ви знаєте? Поясніть кожен із них.

4. Що таке вбудований CSS і як його використовувати?

5. Як виглядає внутрішній CSS і де його слід розміщувати в HTML-документі?

6. Як встановити зовнішній CSS і які переваги він має порівняно з іншими методами?

7. Як використовувати CSS для встановлення кольорів тексту та які способи ви знаєте?

8. Які властивості CSS використовуються для встановлення шрифтів і розмірів тексту?

9. Що таке границя в CSS і які основні властивості вона має?

10. Як встановити внутрішні відступи і зовнішні відступи за допомогою CSS?

**Тема роботи:** Використання посилань у HTML для створення зв'язків між WEB-сторінками та іншими ресурсами.

Мета роботи: Дослідити синтаксис та різні типи посилань у HTML, включаючи текстові, зображення та електронні адреси. Вивчити атрибути та їхні можливості, такі як target для визначення вікна відкриття посилання. Вивчити методи створення закладок та використання абсолютних та відносних URL.

#### Завдання

1. Створіть посилання на іншу WEB-сторінку і відкрийте її в новому вікні.

2. Вставте зображення як посилання і перевірте його роботу.

3. Створіть посилання на адресу електронної пошти і перевірте, чи відкривається програма електронної пошти з правильною адресою.

4. Додайте кнопку як посилання і стилізуйте її за допомогою CSS.

5. Вставте посилання на іншу частину тієї ж сторінки і перевірте, чи працює прокрутка.

6. Створіть список посилань на різні розділи однієї сторінки.

7. Використовуйте атрибут title для додаткової інформації про кожне посилання.

8. Створіть посилання, що веде на той самий URL, але з різними якостями відкриття (в новому вікні, у тому ж вікні і т. д.).

9. Створіть абсолютні та відносні посилання та порівняйте їхню роботу.

10. Застосуйте різні кольори посилань за допомогою CSS.

11. Створіть посилання на підсторінки та переконайтеся, що вони працюють на вашому WEB-сайті.

12. Створіть посилання на файли (наприклад, зображення, аудіо

або відео) і переконайтеся, що вони відкриваються правильно.

13. Застосуйте стилі до посилань при наведенні миші, щоб виділити їхню важливість.

14. Створіть посилання на інші WEB-сайти та перевірте їхню працездатність.

15. Створіть кнопку, яка посилається на одну сторінку, але при цьому змінює її вміст або виконує інші дії, наприклад, змінює колір фону або розмір тексту при натисканні.

#### Контрольні питання

1. Що таке HTML-посилання, і яке воно має призначення?

2. Які основні компоненти посилання в HTML?

3. Яким чином визначається адреса, на яку веде посилання?

4. Які значення може мати атрибут target у тегу <a>?

5. Як використовувати зображення як посилання в HTML?

6. Як створити посилання, що відкриє програму електронної пошти користувача?

7. Як можна використовувати кнопку як посилання за допомогою HTML i CSS?

8. Як визначити заголовок для посилання в HTML?

9. В чому відмінності між абсолютними та відносними URLадресами?

10. Як створити закладку на сторінці за допомогою HTML?

**Тема роботи:** Додавання зображень до WEB-сторінки за допомогою HTML.

Мета роботи: Навчитися вставляти зображення на WEB-сторінку за допомогою HTML. Ознайомитися з атрибутами тега <img>, такими як src, alt, width та height. Вивчити, як використовувати зображення як посилання. Розглянути методику розміщення зображень на сторінці, не порушуючи текст. Ознайомитися з поняттям фонового зображення та способами його додавання до елементів HTML.

#### Завдання

#### \*Тему для ваших сторінок обираєте самостійно.

1. Створіть HTML-сторінку з наступним вмістом:

• Заголовок WEB-сторінки

• Короткий текст з описом теми

• Зображення, що ілюструє тему

• Посилання на іншу WEB-сторінку, пов'язану з темою

• Фонове зображення для всієї сторінки

2. Застосуйте до зображень атрибути alt, щоб зробити сторінку доступною для людей з вадами зору.

3. Використовуйте атрибути width та height, щоб встановити розміри зображень.

4. Зробіть одне з зображень посиланням на іншу WEB-сторінку.

5. Додайте фонове зображення до всієї сторінки, використовуючи властивості background-image, background-repeat та background-attachment.

6. Додайте анімоване зображення на сторінку, використовуючи формат GIF.

7. Відкрийте зображення як посилання на іншу сторінку, використовуючи теги <a> i <img>.

8. Використайте CSS float, щоб зображення плавало ліворуч або праворуч.

9. Розтягніть фонове зображення так, щоб воно відповідало розміру елемента, використовуючи background-size.

10. Зробити так, щоб фонове зображення розтягувалося, але залишалося фіксованим при прокручуванні сторінки, використовуючи background-attachment.

11. Встановити фонове зображення, щоб воно не повторювалося за допомогою background-repeat.

12. Встановити повторення фонового зображення по горизонталі та вертикалі за допомогою background-repeat.

13. Створити сторінку зі списком друзів, де кожен з вашого списку друзів представлений зображенням, альтернативним текстом буде їх ім'я.

14. Створити WEB-сторінку, використовуючи тег <img>, щоб вставити зображення з відповідним атрибутом src, який містить шлях до файлу зображення.

15. Створити сторінку, додаючи атрибут width та height, щоб стандартизувати розміри зображень на сторінці. Додайте атрибут alt до зображень з описом. Це покращить доступність вашої сторінки для людей з обмеженими можливостями.

#### Контрольні питання

1. Який тег використовується для вставлення зображення на WEB-сторінку HTML?

2. Який атрибут тега  $\langle img \rangle \in obob'язковим?$ 

3. Яке призначення атрибуту alt тега <img>?

4. Як можна використовувати зображення як посилання на WEBсторінку HTML?

5. Що таке "плаваюче зображення" в HTML?

6. Як можна додати фонове зображення до елемента HTML?

7. Яка властивість CSS використовується для налаштування повторення фонового зображення?

8. Що таке "фонова обкладинка" в CSS?

9. Які переваги використання стилів CSS для стилізації зображень на WEB-сторінці?

Тема роботи: Додавання іконки та заголовка до WEB-сторінки.

Мета роботи: Вивчення впливу іконки та заголовка сторінки на сприйняття користувачами, їхню важливість для WEB-ідентифікації та SEO-показники з метою розробки рекомендацій для покращення WEB-сторінок.

#### Завдання

1. Створити WEB-сторінку на тему «Фітнес та здоров'я» і додати на неї HTML-іконку за допомогою елементу <link>. Також на сторінці повинні бути присутні: заголовок, опис, зображення, посилання можливо на якусь продукцію або інвертар і т.д.

2. Створити WEB-сторінку на тему «Кулінарія та рецепти», додати на неї HTML-іконку за допомогою елементу <link> та змінити розмір іконки сайту за допомогою CSS. Також на сторінці повинні бути присутні: заголовок, опис, зображення, поради щодо приготування страв і т.д.

3. Створити WEB-сторінку на тему «Технології та IT», додати на неї створену власну іконку сайту за допомогою онлайн-генератора Також на сторінці повинні бути присутні: заголовок, опис, зображення, огляди гаджетів і т.д.

4. Створіть WEB-сторінку на тему «Події та розваги» та додайте заголовок сторінки <title> з назвою вашого проекту. Також на сторінці повинні бути присутні: заголовок, опис, зображення, місцеві події, концерти і т.д.

5. Створити WEB-сторінку на тему «Мода та стиль» і додати на неї HTML-іконку за допомогою елементу <link>. Змініть формат файлу іконки на .png або .svg і перевірте, чи відображається вона на вкладці браузера. Також на сторінці повинні бути присутні: заголовок, опис, зображення, стильний одяг, модні тенденції і т.д.

6. Створити WEB-сторінку на тему «Подорожі та туризм» і додати на неї HTML-іконку за допомогою елементу <link>. Використовуючи атрибут sizes у тегу <link>, налаштуйте різні розміри іконок для відображення на різних пристроях. Також на сторінці повинні бути присутні: заголовок, опис, зображення, захоплюючі місця для подорожей і т.д.

7. Створити WEB-сторінку на тему «Огляди фільмів» і додати на неї HTML-іконку за допомогою елементу <link>. Створити папку «images» у вашому проекті та зберегти у ній файли іконок для вашого сайту. Також на сторінці повинні бути присутні: заголовок, опис, зображення, нові та класичні фільми і т.д.

8. Створити WEB-сторінку на тему «Історія кіно» і додати на неї HTML-іконку за допомогою елементу <link>. Змінити ім'я файлу з іконкою на «favicon.png» і встановіть його як іконку сайту. Також на сторінці повинні бути присутні: заголовок, опис, зображення, дослідження історії кінематографу і т.д.

9. Створити WEB-сторінку на тему «Туристичні маршрути» і додати на неї HTML-іконку за допомогою елементу <link>. Перевірте сайт на різних браузерах та пристроях, щоб переконатися, що іконка відображається коректно. Також на сторінці повинні бути присутні: заголовок, опис маршруту, зображення, популярні туристичні маршрути і т.д.

10. Створити WEB-сторінку на тему «Культурні подорожі» і додати на неї HTML-іконку за допомогою елементу <link>. Відредагувати іконку сайту за допомогою графічного редактора і змінити її колір або стиль. Також на сторінці повинні бути присутні: заголовок, опис, зображення, культурні особливості різних країн та міст і т.д.

11. Створіть сторінку з інформацією про використання іконок сайту та їх важливість для користувачів. Додати на сайт HTML-іконку за допомогою елементу <link>.

12. Створити WEB-сторінку на тему «Коктейльні напої» і додати на неї HTML-іконку за допомогою елементу <link>. Також застосуйте тематичні іконки до різних розділів вашої WEB-сторінки (наприклад, іконка келиха для розділу «Маргарита»).

13. Створити WEB-сторінку на тему «Ексклюзивні напої» і додати на неї HTML-іконку за допомогою елементу <link>. Встановити фонове зображення для WEB-сторінки. Також на сторінці повинні бути присутні: заголовок, опис, зображення, рідкісні або ексклюзивні напої і т.д.

14. Створити WEB-сторінку на тему «Фотогалерея тварин» і додати на неї HTML-іконку за допомогою елементу <link>. Додайте анімацію до картин. Також на сторінці повинні бути присутні: заголовок, опис, зображення і т.д.

15. Створити WEB-сторінку на тему «Огляди ігор» і додати на неї HTML-іконку за допомогою елементу <link>. Зробіть заголовки реагуючими на події, наприклад, коли користувач наводить на них курсор миші. Також на сторінці повинні бути присутні: заголовок, опис, зображення, найпопулярніші ігри і т.д.

#### Контрольні питання

1. Що таке HTML-іконка і як вона використовується?

2. Як додати іконку до WEB-сайту за допомогою елемента <link>?

3. Які поради надаються щодо вибору зображення для іконки сайту?

4. Як забезпечити, щоб іконка відображалася на вкладці браузера?

5. Які формати файлів іконок підтримуються для використання на WEB-сайті?

6. Що таке заголовок сторінки HTML і чому він важливий?

7. Які функції виконує елемент <title> у HTML-документі?

8. Які аспекти заголовка сторінки важливі для пошукової оптимізації (SEO)?

9. Як заголовок сторінки впливає на відображення у результатах пошуку?

10. Як можна зробити заголовок сторінки максимально зрозумілим та змістовним?

Тема роботи: Створення та форматування таблиць в HTML.

**Мета роботи:** Навчитися форматувати таблиці в HTML, використовуючи різні методи та атрибути, а також застосувати знання CSS для стилізації та покращення візуального вигляду таблиць.

#### Завдання

## 1. Створення таблиці з інформацією про книги

Створіть HTML-код для таблиці з 5 стовпцями.

- Назва книги
- Автор
- Рік публікації
- Жанр
- Кількість сторінок

Додайте заголовки до кожного стовпця. Заповніть комірки даними про 5 різних книг. Встановіть загальний стиль для таблиці (фон, шрифт, рамки). Зробіть заголовки таблиці жирними та вирівняними по центру. Виділіть перший рядок (заголовки) іншим кольором. Зробіть текст у стовпцях «Автор» та «Жанр» курсивом.

## 2. Створення таблиці розкладу уроків

Створіть HTML-код для таблиці розкладу уроків на один день.

- День тижня (понеділок, вівторок, середа, ...)
- Час початку уроку
- Час закінчення уроку
- Назва предмета
- Ім'я викладача

Додайте заголовки до кожного стовпця. Заповніть комірки даними про розклад уроків для вашої групи. Зробіть кожен день тижня окремим заголовком з жирним шрифтом. Вирівняйте стовпці по центру. Виділіть парні та непарні рядки різними кольорами. Зробіть час уроків жирним шрифтом. Також додайте рамку до вашої таблиці, для цього використайтеся властивістю **border**.

## 3. Створення таблиці з порівнянням цін на продукти

Створіть HTML-код для таблиці порівняння цін на 3 різних продукти (наприклад, смартфони, ноутбуки, фотоапарати).

- •Назва продукту
- •Виробник
- •Ціна в магазині А
- •Ціна в магазині В
- •Ціна в магазині С

Додайте заголовки до кожного стовпця. Заповніть комірки даними про ціни на ці продукти в 3 різних магазинах. Зробіть назви продуктів жирними та виділіть їх іншим кольором. Вирівняйте стовпці з цінами по правому краю. Виділіть найнижчу ціну для кожного продукту іншим кольором. Створіть круглі межі таблиці, використовуючи властивість **border-radius**.

## 4. Створення таблиці з прогнозом погоди

Створіть HTML-код для таблиці з прогнозом погоди на тиждень.

- День тижня
- Дата
- Температура вдень (°С)
- Температура вночі (°С)
- Опис погоди (сонячно, хмарно, дощ, сніг)
- Вітер (напрямок і швидкість)

Додайте заголовки до кожного стовпця. Заповніть комірки даними про прогноз погоди на тиждень у вашому місті. Виділіть заголовки таблиці жирним шрифтом та фоновим кольором. Використовуйте іконки або зображення для опису погоди (сонце, хмари, дощ, сніжинки). Зробіть стовпці з температурами різними кольорами (для позитивних і негативних значень). Додайте умовне форматування, щоб виділити дні з найвищою та найнижчою температурами. За допомогою **border-style** властивості ви можете встановити зовнішній вигляд межі, вона повинна бути пунктирною.

## 5. Створення таблиці з рецептами страв

Створіть HTML-код для таблиці з рецептами 3 різних страв.

- Назва страви
- Інгредієнти
- Кількість порцій
- Час приготування
- Інструкція приготування

Додайте заголовки до кожного стовпця. Заповніть комірки даними про рецепти 3 страв, які вам подобаються. Виділіть назви страв жирним шрифтом та фоновим кольором. Зробіть список інгредієнтів у вигляді нумерованого списку. Додайте фотографії страв. Розбийте інструкцію приготування на абзаци з чіткими заголовками. Установіть колір рамки за власним бажання.

## 6. Створення таблиці з каталогом товарів

Створіть HTML-код для таблиці з каталогом 5 різних товарів.

- Назва товару
- Опис товару
- Ціна
- Фотографія товару
- Посилання на сторінку товару

Додайте заголовки до кожного стовпця. Заповніть комірки даними про 5 товарів з онлайн-магазину або з власного каталогу. Зробіть фотографії товарів однаковими за розміром. Вирівняйте стовпці з текстом по лівому краю. Виділіть ціни іншим кольором. Встановіть ширину четвертого стовпця на 50%, а висоту другого рядка на 100 пікселів.

## 7. Створення таблиці з рейтингом фільмів

Створіть HTML-код для таблиці з рейтингом 5 різних фільмів.

- Назва фільму
- Рік випуску
- Жанр
- Рейтинг IMDb
- Короткий опис

Додайте заголовки до кожного стовпця. Заповніть комірки даними про 5 фільмів, які вам подобаються. Виділіть назви фільмів жирним шрифтом. Додайте фоновий колір у кожному другому рядку таблиці. Зробіть стовпець з рейтингом IMDb у вигляді зірок (використовуючи іконки або зображення). Додайте посилання на трейлер або сторінку фільму в останньому стовпці.

## 8. Створення таблиці з розкладом спортивних змагань

Створіть HTML-код для таблиці з розкладом 5 спортивних змагань

- Дата змагання
- Час змагання
- Команда 1
- Команда 2
- Місце проведення
- Трансляція (канал або платформа)

Додайте заголовки до кожного стовпця. Заповніть комірки даними про 5 змагань з виду спорту, який вам подобається. Виділіть дати змагань жирним шрифтом. Зробіть стовпці з командами різними кольорами. Додайте іконки або зображення логотипів команд. Додайте вертикальні смуги зебри, на кожен другий стовпець..

## 9. Створення таблиці з розкладом занять у спортзалі

Створіть HTML-код для таблиці з розкладом занять у спортзалі на тиждень

• День тижня

- Час заняття
- Назва заняття
- Тренер
- Опис заняття

Додайте заголовки до кожного стовпця. Виділіть дні тижня жирним шрифтом та фоновим кольором. Зробіть стовпець з назвою заняття у вигляді посилань (на опис заняття). Додайте фотографії тренерів (якщо можливо). Додайте іконки для різних типів занять (силові, кардіо, йога, танці). Поєднайте вертикальні і горизонтальні смуги зебри, зробивши смуги в кожному другому рядку та кожному другому стовпці.

10. Створення таблиці з порівнянням характеристик смартфонів

Створіть HTML-код для таблиці з порівнянням характеристик 3 різних смартфонів

- Назва смартфона
- Виробник
- Екран (діагональ, роздільна здатність)
- Процесор
- Оперативна пам'ять
- Внутрішня пам'ять
- Камера (основна, фронтальна)
- Батарея (ємність)
- Ціна

Додайте заголовки до кожного стовпця. Заповніть комірки даними про характеристики 3 смартфонів, які ви хочете порівняти. Виділіть назви смартфонів жирним шрифтом та фоновим кольором. Зробіть стовпці з характеристиками однакової ширини. Використовуйте **:hover** селектор, **tr** щоб виділити рядки таблиці при наведенні миші, колір оберіть за бажанням.

## 11. Створення таблиці з розкладом рейсів літаків

Створіть HTML-код для таблиці з розкладом рейсів літаків на день

- Номер рейсу
- Авіакомпанія
- Місто вильоту
- Аеропорт вильоту
- Місто прибуття
- Аеропорт прибуття
- Час вильоту
- Час прибуття
- Статус рейсу (заплановано, затримано, скасовано)

Додайте заголовки до кожного стовпця. Заповніть комірки даними про розклад рейсів. Виділіть номери рейсів жирним шрифтом. Зробіть стовпці з містами вильоту та прибуття у вигляді посилань (на карту або інформацію про місто). Додайте іконки авіакомпаній. Використовуйте різні кольори для позначення статусу рейсу (зелений – заплановано, жовтий – затримано, червоний – скасовано). Надайте стилю першому стовпцю таблиці використовуючи елементи <colgroup> та <col>

## 12. Створення таблиці з курсами валют

Створіть HTML-код для таблиці з курсами 5 різних валют

- Валюта
- Країна
- Курс покупки
- Курс продажу
- Зміна курсу (за день)

Додайте заголовки до кожного стовпця. Заповніть комірки даними про курси валют з онлайн-курса валют. Виділіть назви валют жирним шрифтом. Зробіть стовпці з курсами різними кольорами (зелений – зростання, червоний – падіння). Додайте іконки прапорів країн. Відображайте зміну курсу у відсотках.

# 13. Створення таблиці з переліком ваших улюблених країн для подорожей.

Включіть наступні стовпці:

- Назва країни
- Головне місто
- Географічні особливості
- Відомості про культуру
- Основні туристичні визначні місця

Додайте заголовки до кожного стовпця. Заповніть комірки даними про ваші улюблені країни. Встановіть загальний стиль для таблиці (фон, шрифт, рамки). Зробіть заголовки таблиці жирними та вирівняними по центру. Виділіть назви країн курсивом. Додайте іконки прапорів країн. Основні туристичні визначні місця можуть бути представлені у вигляді зображень або ілюстрацій.

**14.** Створення таблиці з улюбленими музичними виконавцями Створіть таблицю з наступними стовпцями:

- Ім'я виконавця
- Жанр музики
- Рік початку творчої діяльності
- Країна походження
- Посилання на популярну пісню або відео виконавця.

Додайте заголовки до кожного стовпця. Заповніть комірки даними про музичних виконавців. Встановіть загальний стиль для таблиці (фон – темно-сірий, шрифт). Зробіть заголовки таблиці жирними та вирівняними по центру. Назви виконавців можна виділити білим кольором. Для кожного виконавця можна додати зображення або іконку, яка представляє їх музику. Також додайте посилання на популярні пісні.

## 15. Створення таблиці з розкладом вистав та подій

Створіть таблицю з наступними стовпцями:

- Назва події
- Дата та час проведення
- Місце проведення
- Опис або додаткові деталі
- Посилання на детальну інформацію

Додайте заголовки до кожного стовпця. Заповніть комірки даними про вистави та події. Встановіть загальний стиль для таблиці (фон – темно фіолетовий, шрифт, колір тексту – білий). Назви подій можуть бути виділені жовтим кольором. Зробіть заголовки таблиці жирними та вирівняними по центру. Додайте посилання на детальну інформацію про кожну подію.

## **Hide Columns**

You can hide specific columns with the visibility property:

MON	TUE	SAT	SUN
1	2	6	7
8	9	13	14
15	16	20	21
22	23	27	28

Note: The table columns does not collapse properly in Safari browsers.

## Контрольні питання

1. Які теги використовуються для позначення початку таблиці, рядка та комірки?

- 2. Як можна створити заголовок таблиці?
- 3. Як можна додати підпис до таблиці?
- 4. Як додати рамку до таблиці?
- 5. Як створити згорнуті межі таблиці?
- 6. Які властивості CSS можна використовувати для налаштування

стилю рамки таблиці?

- 7. Як встановити ширину таблиці?
- 8. Як встановити ширину стовпця таблиці?
- 9. Як встановити висоту рядка таблиці?
10. Як вирівняти заголовки таблиць за лівим краєм?

11. Як створити заголовок, який охоплює кілька стовпців?

12. Як додати відступ до клітинок таблиці?

13. Як використовувати colspan для охоплення клітинкою кількох стовпців?

14. Як використовувати **rowspan** для охоплення клітинкою кількох рядків?

15. Які методи можна використовувати для стилізації таблиць за допомогою CSS?

16. Як створити смуги зебри в таблиці?

17. Як додати горизонтальні роздільники до таблиці?

18. Як виділити рядок таблиці при наведенні курсору?

19. Що таке елемент **<colgroup>**?

20. Як використовувати **<colgroup>** для стилізації стовпців таблиці?

21. Які властивості CSS можна використовувати в елементі <colgroup>?

22. Як можна приховати стовпці таблиці за допомогою <colgroup>?

#### Лабораторна робота №11

**Тема роботи:** Використання списків у HTML: основні типи та їх властивості.

**Мета роботи:** Розглянути різновиди списків у HTML (невпорядковані, впорядковані, списки описів), їх структуру та особливості використання, а також навчитися стилізувати їх за допомогою CSS.

#### Завдання

\*До кожного завдання повинен йти CSS файл, в якому ви стилізуєте вашу HTML-сторінку для поліпшення зовнішнього вигляду. CSS повинен містити – фон, колір тексту, вирівнювання тексту, міжрядковий інтервал, шрифт та розмір, відступи і т.д.

#### 1. HTML Basics Quiz

- Створіть HTML-сторінку з назвою «HTML Basics Quiz».
- На сторінці повинні бути запитання з можливими відповідями.

• Додайте невпорядкований та впорядкований списки з відповідними елементами **«ul»** та **«ol»**.

#### 2. Список завдань

• Створіть HTML-сторінку з назвою «Task List».

• Створіть впорядкований список, де кожен елемент це одне завдання, яке користувач має виконати.

• Додайте короткий опис до кожного завдання.

#### 3. Послідовність кроків

• Створіть HTML-сторінку з назвою «Step-by-Step Guide».

• Створіть впорядкований список, який буде показувати послідовність кроків для виконання певного завдання.

#### 4. Список товарів

• Створіть HTML-сторінку з назвою «Product List».

• Створіть невпорядкований список для відображення різних категорій товарів вашого онлайн магазину.

## 5. Список порад

• Створіть HTML-сторінку з назвою «Tips and Tricks».

• Створіть впорядкований список, де кожен елемент буде містити коротку поради з певної тематики.

#### 6. Розклад подій

• Створіть HTML-сторінку з назвою «Event Schedule».

• Створіть впорядкований список для відображення дат та короткого опису подій, які відбудуться протягом тижня.

### 7. Список рекомендацій

• Створіть HTML-сторінку з назвою «Recommendations».

• Створіть невпорядкований список для переліку товарів, книг, фільмів або ресурсів, які ви рекомендуєте своїм користувачам.

### 8. Список ресурсів

• Створіть HTML-сторінку з назвою «Useful Resources».

• Створіть невпорядкований список для переліку корисних WEB-ресурсів для вивчення певної теми.

#### 9. Список покупок

• Створіть HTML-сторінку з назвою «Shopping List».

• Створіть невпорядкований список для переліку продуктів, які вам потрібно придбати на наступний візит до супермаркету.

# 10. Меню ресторану

• Створіть HTML-сторінку з назвою «Restaurant Menu».

• Створіть невпорядкований список, який буде містити різні категорії страв у меню вашого ресторану разом із списком страв у кожній категорії.

#### 11. Поради щодо кодування

• Створіть HTML-сторінку з назвою «Coding Tips».

• Створіть впорядкований список, де кожен елемент міститиме коротку пораду або прийом з програмування.

• Надайте короткий пояснення до кожного прийому, пояснюючи, як він може бути корисним для розробників.

### 12. Фітнес-рутина

• Створіть HTML-сторінку з назвою «Fitness Routine».

• Створіть впорядкований список, де кожен елемент буде представляти один етап або вправу у вашому фітнес-розкладі.

### 13. Список бажань для подорожей

• Створіть HTML-сторінку з назвою «Travel Bucket List».

• Створіть впорядкований список, де кожен елемент буде містити одне місце або країну, яку ви хочете відвідати.

• Для кожного місця надайте короткий опис або пояснення, чому ви хочете туди поїхати.

### 14. Музичний плейлист

• Створіть HTML-сторінку з назвою «Music Playlist».

• Створіть невпорядкований список, де кожен елемент буде містити одну пісню з вашого улюбленого плейлиста.

• Для кожної пісні додайте виконавця та короткий опис.

# 15. Порада по садівництву

• Створіть HTML-сторінку з назвою «Gardening Tips».

• Створіть невпорядкований список, де кожен елемент буде містити коротку пораду або прийом з садівництва.

• Додайте ілюстрації або фотографії для кращого зрозуміння порад.

# Контрольні питання

1. Що таке HTML-списки і яка їхня основна мета?

2. Які типи списків існують у HTML, і як вони відрізняються між собою?

3. Які теги використовуються для створення невпорядкованих списків у HTML?

4. Які атрибути можна використовувати для налаштування стилів маркерів у невпорядкованих списках?

5. Що таке вкладені списки і як їх створювати у HTML?

6. Які елементи HTML використовуються для створення впорядкованих списків?

7. Як можна налаштувати тип нумерації впорядкованих списків за допомогою атрибута type?

8. Як можна розпочати нумерацію впорядкованого списку з певного числа за допомогою атрибута start?

9. Що таке список описів і як він створюється у HTML?

10. Які елементи HTML використовуються для створення списку описів, і яка їхня роль?

#### Лабораторна робота №12

**Тема роботи:** Використання елементів рівня блоку та тегів HTML для створення WEB-сторінки.

**Мета роботи:** Навчитися використовувати елементи рівня блоку та теги HTML для створення структурованої та інформативної WEBсторінки. Оволодіти навичками додавання текстового контенту, списків, таблиць, зображень та інших елементів на WEB-сторінку.

#### Завдання

1. Створення WEB-сторінки з основними елементами рівня блоку

• Створіть HTML-документ з базовою структурою, включаючи <head> та <body> секції.

- Додайте заголовок сторінки (<h1>) та підзаголовок (<h2>).
- Створіть абзац тексту (р) з описом вашого WEB-сайту.
- Вставте зображення (img) з описом (alt).
- Створіть список (ul або ol) з важливими посиланнями.

• Додайте розділ контактів (section) з вашою адресою електронної пошти та номером телефону.

#### 2. Створення WEB-форми з різними елементами вводу

- Створіть HTML-документ з тегом <form>.
- Додайте різні елементи вводу, такі як:
  - ∘ <input type= "text"> для текстових полів.
  - ∘ <input type="password"> для полів паролів.
  - ∘ <input type="checkbox"> для прапорців.
  - ∘ <input type="radio"> для радіо-кнопок.
  - ∘ <input type="file"> для завантаження файлів.
  - 。 <input type="submit"> для кнопки надсилання.
  - ∘ <input type="reset"> для кнопки скидання.

- <textarea> для багаторядкових текстових полів.
- <select> для розкривних списків.
- Додайте підписи (label) для кожного елемента вводу.
- Використовуйте CSS для стилізації форми та її елементів.
- 3. Вбудовування відео на WEB-сторінку
- Створіть HTML-документ з тегом <video>.

• Додайте теги <source> з посиланнями на ваші відеофайли (MP4, WebM, OGG).

• Додайте текст (fallback content) між тегами <video> та </video> для браузерів, які не підтримують елемент <video>.

# 4. Створення різних типів списків

- Створіть HTML-документ з тегами та .
- Додайте елементи списку (li) з описом.

• Використовуйте різні типи маркерів для невпорядкованих списків (квадрати, круги).

- Використовуйте цифри або літери для впорядкованих списків.
- Використовуйте CSS для стилізації списків та їх елементів.
- 5. Створення галереї зображень
- Створіть HTML-документ з елементом <section> для галереї.

• Використовуйте теги <figure> та <figcaption> для кожного зображення.

• Додайте посилання на зображення (img) та описи (figcaption).

• Використовуйте CSS для стилізації галереї, щоб зображення розміщувалися сіткою, рядами або стовпцями.

# 6. Створення блогу з публікаціями

- Створіть HTML-документ з елементом <section> для блогу.
- Використовуйте теги <article> для кожної публікації.

• Додайте заголовок (h2), автор публікації, дату публікації, короткий опис (p).

• Використовуйте CSS для стилізації блогу.

# 7. Створення інтернет-магазину

• Створіть HTML-документ з елементом <section> для інтернетмагазину.

• Використовуйте теги <article> для кожного товару.

• Додайте зображення товару (img), назву товару (h3), опис товару (p), ціну товару (span) та кнопку «Купити» (button).

• Використовуйте CSS для стилізації інтернет-магазину, щоб товари відображалися в сітці, рядами або стовпцями.

# 8. Створення WEB-сайту з навігацією

• Створіть HTML-структуру для WEB-сайту з декількома сторінками.

- Використовуйте тег <nav> для створення навігаційного меню.
- Додайте посилання на кожну сторінку сайту в меню.
- Стилізуйте меню за допомогою CSS.

# 9. Створення форми для збору даних

• Створіть HTML-форму для збору даних користувача, таких як ПІБ, телефон, email, адреса.

• Використовуйте відповідні теги форми: <form>, <input>, <textarea>, <select>, <button>.

• Додайте підписи до кожного поля вводу.

# 10. Створення WEB-сайту з розділами

• Створіть HTML-структуру для WEB-сайту з чітко визначеними розділами.

• Використовуйте тег <section> для групування схожого контенту в кожному розділі.

- Додайте заголовки до кожного розділу.
- Стилізуйте розділи за допомогою CSS.

# 11. Створення форуму з темами та повідомленнями

• Створіть HTML-структуру для форуму з темами та повідомленнями.

- Використовуйте тег <dl> для списку тем.
- Використовуйте теги <dt> i <dd> для заголовка та опису теми.
- Використовуйте тег для списку повідомлень у темі.
- Стилізуйте форум за допомогою CSS.

# 12. Створення WEB-сайту з відгуками

• Створіть HTML-структуру для WEB-сайту з відгуками.

• Використовуйте тег <blockquote> для відображення кожного відгуку.

- Додайте ім'я та фото користувача до кожного відгуку.
- Стилізуйте відгуки за допомогою CSS.

# 13. Створіть НТМL-структуру для WEB-сайту з рецептами

- Використовуйте тег <article> для кожного рецепту.
- Додайте назву страви, інгредієнти, інструкції та зображення до

кожного рецепту.

• Стилізуйте WEB-сайт з рецептами за допомогою CSS.

# 14. Створення WEB-сайту з тизерами новин

• Створіть HTML-структуру для WEB-сайту з тизерами новин, що містить список новин, заголовки, короткі описи та зображення.

• Використовуйте тег <article> для кожної новини.

• Додайте заголовок, короткий опис, зображення та посилання до кожної новини.

- Відобразіть список новин у хронологічному порядку.
- Додайте кнопку «Читати далі» до кожної новини.
- Стилізуйте WEB-сайт з тизерами новин за допомогою CSS.

# 15. Створіть сторінку про вашу групу

• Використайте HTML для створення сторінки з інформацією про вашу групу.

• Використовуючи тег <h1> створіть заголовок сторінки «Група ...».

• Додайте фото групи (<img>), короткий опис вашої спеціальності на яку ви навчаєтеся (<p>) та контактну інформацію (<address>), посилання на електронну пошту чи соціальні мережі (<a>).

• Використайте CSS для стилізації сторінки та підкреслення її професійного вигляду.

#### Контрольні питання

1. Що таке елемент рівня блоку в HTML?

2. Які ключові характеристики елементів рівня блоку?

3. Які два поширених елементи рівня блоку?

4. Для чого використовується елемент ?

5. Для чого використовується елемент <div>?

6. Який тег використовується для визначення самодостатнього контенту?

7. Який тег використовується для цитування тексту з іншого джерела?

8. Який тег використовується для малювання графіки «на льоту» за допомогою скриптів?

9. Який тег використовується для групування пов'язаних елементів у формі?

10. Який тег використовується для визначення таблиці HTML?

11. Який тег використовується для визначення нижнього колонтитула для документа?

12. Який тег використовується для вбудовування відео на WEBсторінку?

13. Який тег використовується для визначення верхнього колонтитула (шапки) на WEB-сторінці?

14. Який тег використовується для визначення тематичного

розриву на сторінці HTML?

15. Який тег використовується для створення HTML-форми?

16. Який тег використовується для визначення елемента списку?

17. Який тег використовується для визначення альтернативного вмісту для користувачів, які вимкнули скрипти?

18. Який тег використовується для визначення впорядкованого списку?

19. Який тег використовується для відображення тексту на WEBсторінці точно так, як він написаний в HTML-коді?

20. Який тег використовується для визначення набору навігаційних посилань?

21. Який тег використовується для групування вмісту нижнього колонтитула в таблиці HTML?

22. Який тег використовується для визначення невпорядкованого (маркованого) списку?

#### Лабораторна робота №13

**Тема роботи:** Використання вбудованих елементів та тегів HTML для створення WEB-сторінки.

Мета роботи: Навчитися використовувати вбудовані елементи та теги HTML для створення WEB-сторінок з різним контентом та форматуванням.

#### Завдання

#### 1. Створення WEB-сторінки з посиланнями

- Створіть HTML-сторінку з назвою «index.html».
- Додайте на сторінку заголовок «Поширені HTML теги».
- Додайте список з 10 найпоширеніших HTML тегів, наприклад:
  - о <a>− посилання
  - о <b>− жирний шрифт
  - ∘ <img>- зображення
  - − абзац
  - ∘ − не нумерований список
  - о − нумерований список
  - ∘ таблиця
  - о </br>− розрив рядка
  - о <form>- форма
  - о <input> поле введення
- Для кожного тега:
  - Додайте короткий опис його призначення.
  - Наведіть приклад використання тега.
  - о Додайте посилання на документацію HTML5 для цього тега.

#### 2. Створення WEB-форми з різними елементами

- Створіть HTML-сторінку з назвою «form.html».
- Додайте на сторінку заголовок «Реєстрація користувача».

- Створіть форму з наступними елементами:
  - о Ім'я: поле введення типу text
  - о Прізвище: поле введення типу text
  - о Електронна пошта: поле введення типу email
  - о Пароль: поле введення типу password
  - о Дата народження: поле введення типу date
  - о Стать: радіокнопки чоловіча та жіноча
  - Країна: розкривний список з переліком країн
  - о Захоплення: прапорці з можливістю вибрати декілька варіантів
  - о Кнопка надсилання форми: кнопка типу submit

• Стилізуйте форму за допомогою CSS, щоб зробити її візуально привабливою.

# 3. Створення WEB-сторінки з зображеннями та відео

- Створіть HTML-сторінку з назвою «media.html».
- Додайте на сторінку заголовок «Мультимедійні елементи».
- Додайте на сторінку зображення вашого улюбленого міста.
- Додайте на сторінку відео з YouTube, яке стосується вашої теми.

• Стилізуйте зображення та відео за допомогою CSS, щоб зробити їх візуально привабливими.

# 4. Визначення абревіатур та акронімів

- Створіть HTML-сторінку з назвою «abbreviations.html».
- Додайте заголовок «Абревіатури та акроніми».

• Визначте абревіатури і акроніми за допомогою тегів <abbr> та <acronym>.

# 5. Використання тегу <br> для розділення рядків

- Створіть HTML-сторінку з назвою «line\_breaks.html».
- Додайте заголовок «Розділення рядків».
- Використовуйте тег <br> для вставки примусових переходів на

новий рядок.

- Створіть список адрес, де кожна адреса буде на новому рядку.
- 6. Вставка коду на сторінку з використанням тегу <code>
- Створіть HTML-сторінку з назвою «code\_example.html».
- Додайте заголовок «Приклад коду».
- Використовуючи тег <code>, вставте фрагмент коду.

• Застосуйте стилізацію за допомогою CSS, щоб зробити текст коду моноширинним.

# 7. Визначення термінів за допомогою тегу <dfn>

- Створіть HTML-сторінку з назвою «definitions.html».
- Додайте заголовок «Визначення термінів».

• Використовуючи тег <dfn>, визначте термін (наприклад, «HTML»). Таких термінів повинно бути не менше 5 штук.

- Додайте визначення терміну всередині тегу <dfn>.
- 8. Створення посилання на книгу та використання тегу <cite>
- Створіть HTML-сторінку з назвою «book\_citation.html».
- Додайте заголовок «Цитата з книги».

• Використовуючи тег <cite>, відобразіть назву твору (наприклад, «Кобзар»).

• Створіть посилання на саму книгу, обгорнувши текст книги тегом <a> і використовуючи атрибут href для вказання URL, який веде до інформації про книгу.

• Додайте атрибут cite, який вказує на джерело цитати, якщо це можливо.

# 9. Створення форми з полями вводу та мітками

- Створіть HTML-сторінку з назвою «form.html».
- Додайте заголовок «Форма з полями вводу».

• Додайте елементи форми за допомогою тега <input>, які відповідатимуть наступним вимогам:

- Однорядкове текстове поле для введення імені користувача з атрибутом type="text".
- о Поле для введення паролю з атрибутом type="password".
- Поле для введення електронної пошти з атрибутом type="email".
- Поле для введення числового значення з атрибутом type="number".
- о Поле для введення дати з атрибутом type="date".
- Поле для встановлення прапорця (чекбокс) з атрибутом type="checkbox".
- Кнопка для надсилання форми з атрибутом type="submit".

• Для кожного поля вводу додайте відповідну мітку за допомогою тега <label>. Пов'язуйте кожну мітку з відповідним полем вводу за допомогою атрибутів for та id.

# 10. Використання тегу <kbd>

- Створіть HTML-файл з назвою «kbd\_example.html».
- Додайте заголовок «Клавіатурні клавіші».

• Використовуючи тег <kbd>, перелічіть комбінації клавіш, які ви знайте та напишіть їх значення

# 11. Вставка відео або аудіофайлу за допомогою тегу <object>

- Створіть HTML-файл з назвою «object\_example.html».
- Додайте заголовок на вашу сторінку.

• У розділі <body> використайте тег <object>, щоб вставити відео або аудіофайл.

• Установіть атрибути <object> для вказання джерела файлу, типу медіа, розмірів та інших параметрів, які ви бажаєте використовувати.

# 12. Відображення цін продуктів з використанням тегу <span>

• Створіть новий HTML-документ з будь-якою назвою.

• Додайте заголовок для сторінки, наприклад, «Список продуктів з цінами».

• Використовуючи тег або , створіть список продуктів.

• Для кожного продукту у списку використайте тег i вкладіть у нього тег <span>.

• У тегу <span> відобразіть ціну продукту.

• Застосуйте CSS для зміни стилю ціни, наприклад, за допомогою властивостей color, background-color і т.д.

13. Виділення важливої інформації за допомогою тегу <strong>

• Створіть новий HTML-документ.

• Додайте заголовок для сторінки, наприклад, «Виділення важливої інформації».

• Вставте текст, який містить важливу інформацію.

• Використайте тег <strong> для виділення ключових слів або фраз, які необхідно підкреслити.

• Впевніться, що важлива інформація на вашій сторінці виглядає виділеною та відзначеною.

#### 14. Використання тегу <textarea> для коментарів

- Додайте новий HTML-документ.
- Створіть форму за допомогою тегу <form>.

• Додайте поле для введення тексту, використовуючи тег <textarea>.

• Визначте розмір та параметри поля за допомогою атрибутів rows ta cols.

• Забезпечте правильну назву поля та його ідентифікацію, додавши атрибут name.

• Додайте кнопку для відправлення форми, яка буде надсилати коментарі.

• Перевірте, чи правильно працює поле введення коментарів та чи можна ввести та надіслати дані.

# 15. Виведення хімічної та математичних формул за допомогою

#### тегів <sub> та <sup>

• Створіть новий HTML-документ.

• Додайте заголовок для сторінки, наприклад, «Хімічні та математичні формули за допомогою тегів <sub> та <sup>".

• Вставте текст з хімічною формулою, де будуть вказані атоми.

• Використайте тег <sub> для нижніх індексів, щоб позначити номери атомів.

• Використайте тег <sup> для верхніх індексів у математичних формулах.

• Переконайтеся, що хімічні та математичні формули на вашій сторінці виглядає коректною та всі індекси розташовані правильно.

#### Контрольні запитання

1. Який тег HTML використовується для визначення аббревіатур або акронімів?

2. Яким атрибутом <a> тегу визначається призначення посилання?

3. Як браузери зазвичай відображають невідвідані, відвідані та активні посилання за умовчанням?

4. Які теги HTML використовувалися для визначення більшого тексту та акронімів в HTML 4, але не підтримуються в HTML5?

5. Що робить тег <br> в HTML?

6. Який тег HTML використовується для визначення кнопок, які можна натиснути?

7. Які поради надаються щодо використання тега <br/>button>?

8. Що визначає тег <cite> в HTML?

9. Для чого використовується тег <code> в HTML?

10. Яким чином тег <dfn> використовується в HTML?

11. Яка семантика тегів <em> та <i>??

12. Що робить атрибут alt у тегу <img>?

13. Чому рекомендується завжди вказувати атрибути width та height y тегу <img>?

14. Яку функцію виконує тег <input> в HTML?

15. Які типи даних можна вводити за допомогою тегу <input> та атрибута type?

16. Що визначає тег <map> в HTML? Які елементи можуть міститися всередині нього?

17. Для чого використовується тег <object>? Які ресурси можуть бути вбудовані за допомогою цього тегу?

18. Як можна забезпечити зв'язок між тегами <map> та <img>?

19. Який призначення тегу <kbd> в HTML? Які його особливості?

20. Чому рекомендується використовувати атрибут id у поєднанні з тегом <label>?

21. Які призначення мають плагіни, які можуть бути вбудовані за допомогою тегу <object>?

22. Який призначення тега <output> у HTML і коли він був вперше введений?

23. Що відрізняє тег <q> від тега <blockquote>?

24. Для чого використовується тег <script> у HTML? Які можливості надає цей тег?

25. Які атрибути використовуються у тезі <select> для зазначення параметрів в розкривному списку?

26. Який призначення тега <small> і в яких ситуаціях його рекомендується використовувати?

27. Що таке тег <span> у HTML і які можливості він надає для стилізації тексту та групування елементів?

28. Для чого призначений тег <textarea> у HTML, і які атрибути використовуються для визначення розмірів текстової області?

29. Які теги використовуються для відображення тексту верхнім і нижнім індексом, і які їхні основні застосування?

30. Який призначення тегу <strong> в HTML, і яка особливість його використання порівняно з тегом <b>?

31. Яке призначення тегу <time> у HTML, і як використовується атрибут datetime для цього тегу?

32. Чому тег <tt> був вилучений у HTML5, і які теги рекомендується використовувати замість нього для відображення тексту моноширинним шрифтом?

33. Яке призначення тегу <var> у HTML, і чому рекомендується уникати використання вбудованих елементів в цьому тегу?

#### Лабораторна робота №14

**Тема роботи:** Використання вбудованих елементів та тегів HTML для створення WEB-сторінки.

Мета роботи: Навчитися використовувати вбудовані елементи та теги HTML для створення WEB-сторінок з різним контентом та форматуванням.

#### Завдання

- 1. Створення WEB-сторінки з посиланнями
- Створіть HTML-сторінку з назвою «index.html».
- Додайте на сторінку заголовок «Поширені HTML теги».
- Додайте список з 10 найпоширеніших HTML тегів, наприклад:
  - о <a>− посилання
  - <b>- жирний шрифт
  - ∘ <img>− зображення
  - − абзац
  - ∘ − не нумерований список
  - о нумерований список
  - 。 таблиця
  - о </br>− розрив рядка
  - о <form>- форма
  - о <input> поле введення
- Для кожного тега:
  - Додайте короткий опис його призначення.
  - Наведіть приклад використання тега.
  - о Додайте посилання на документацію HTML5 для цього тега.

#### 2. Створення WEB-форми з різними елементами

- Створіть HTML-сторінку з назвою «form.html».
- Додайте на сторінку заголовок «Реєстрація користувача».
- Створіть форму з наступними елементами:

- о Ім'я: поле введення типу text
- о Прізвище: поле введення типу text
- о Електронна пошта: поле введення типу email
- о Пароль: поле введення типу password
- о Дата народження: поле введення типу date
- о Стать: радіокнопки чоловіча та жіноча
- Країна: розкривний список з переліком країн
- о Захоплення: прапорці з можливістю вибрати декілька варіантів
- о Кнопка надсилання форми: кнопка типу submit

• Стилізуйте форму за допомогою CSS, щоб зробити її візуально привабливою.

# 3. Створення WEB-сторінки з зображеннями та відео

- Створіть HTML-сторінку з назвою «media.html».
- Додайте на сторінку заголовок «Мультимедійні елементи».
- Додайте на сторінку зображення вашого улюбленого міста.
- Додайте на сторінку відео з YouTube, яке стосується вашої теми.

• Стилізуйте зображення та відео за допомогою CSS, щоб зробити їх візуально привабливими.

# 4. Визначення абревіатур та акронімів

- Створіть HTML-сторінку з назвою «abbreviations.html».
- Додайте заголовок «Абревіатури та акроніми».

• Визначте абревіатури і акроніми за допомогою тегів <abbr> та <acronym>.

# 5. Використання тегу <br> для розділення рядків

- Створіть HTML-сторінку з назвою «line\_breaks.html».
- Додайте заголовок «Розділення рядків».

• Використовуйте тег <br> для вставки примусових переходів на новий рядок.

- Створіть список адрес, де кожна адреса буде на новому рядку.
- 6. Вставка коду на сторінку з використанням тегу <code>
- Створіть HTML-сторінку з назвою «code\_example.html».
- Додайте заголовок «Приклад коду».
- Використовуючи тег <code>, вставте фрагмент коду.

• Застосуйте стилізацію за допомогою CSS, щоб зробити текст коду моноширинним.

# 7. Визначення термінів за допомогою тегу <dfn>

- Створіть HTML-сторінку з назвою «definitions.html».
- Додайте заголовок «Визначення термінів».

• Використовуючи тег <dfn>, визначте термін (наприклад, «HTML»). Таких термінів повинно бути не менше 5 штук.

• Додайте визначення терміну всередині тегу <dfn>.

# 8. Створення посилання на книгу та використання тегу <cite>

- Створіть HTML-сторінку з назвою «book\_citation.html».
- Додайте заголовок «Цитата з книги».

• Використовуючи тег <cite>, відобразіть назву твору (наприклад, «Кобзар»).

• Створіть посилання на саму книгу, обгорнувши текст книги тегом <a> і використовуючи атрибут href для вказання URL, який веде до інформації про книгу.

• Додайте атрибут cite, який вказує на джерело цитати, якщо це можливо.

# 9. Створення форми з полями вводу та мітками

• Створіть HTML-сторінку з назвою «form.html».

• Додайте заголовок «Форма з полями вводу».

• Додайте елементи форми за допомогою тега <input>, які відповідатимуть наступним вимогам:

о Однорядкове текстове поле для введення імені користувача з

атрибутом type="text".

- Поле для введення паролю з атрибутом type="password".
- Поле для введення електронної пошти з атрибутом type="email".
- Поле для введення числового значення з атрибутом type="number".
- о Поле для введення дати з атрибутом type="date".
- Поле для встановлення прапорця (чекбокс) з атрибутом type="checkbox".
- Кнопка для надсилання форми з атрибутом type="submit".

• Для кожного поля вводу додайте відповідну мітку за допомогою тега <label>. Пов'язуйте кожну мітку з відповідним полем вводу за допомогою атрибутів for та id.

### 10. Використання тегу <kbd>

- Створіть HTML-файл з назвою «kbd\_example.html».
- Додайте заголовок «Клавіатурні клавіші».

• Використовуючи тег <kbd>, перелічіть комбінації клавіш, які ви знайте та напишіть їх значення

# 11. Вставка відео або аудіофайлу за допомогою тегу <object>

- Створіть HTML-файл з назвою «object\_example.html».
- Додайте заголовок на вашу сторінку.

• У розділі <body> використайте тег <object>, щоб вставити відео або аудіофайл.

• Установіть атрибути <object> для вказання джерела файлу, типу медіа, розмірів та інших параметрів, які ви бажаєте використовувати.

#### 12. Відображення цін продуктів з використанням тегу <span>

• Створіть новий HTML-документ з будь-якою назвою.

• Додайте заголовок для сторінки, наприклад, «Список продуктів з цінами».

• Використовуючи  $\operatorname{ter} a fo$ , створіть список продуктів.

• Для кожного продукту у списку використайте тег i вкладіть у нього тег <span>.

• У тегу <span> відобразіть ціну продукту.

• Застосуйте CSS для зміни стилю ціни, наприклад, за допомогою властивостей color, background-color і т.д.

# 13. Виділення важливої інформації за допомогою тегу <strong>

• Створіть новий НТМL-документ.

• Додайте заголовок для сторінки, наприклад, «Виділення важливої інформації».

• Вставте текст, який містить важливу інформацію.

• Використайте тег <strong> для виділення ключових слів або фраз, які необхідно підкреслити.

• Впевніться, що важлива інформація на вашій сторінці виглядає виділеною та відзначеною.

# 14. Використання тегу <textarea> для коментарів

- Додайте новий HTML-документ.
- Створіть форму за допомогою тегу <form>.

• Додайте поле для введення тексту, використовуючи тег <textarea>.

• Визначте розмір та параметри поля за допомогою атрибутів rows ta cols.

• Забезпечте правильну назву поля та його ідентифікацію, додавши атрибут name.

• Додайте кнопку для відправлення форми, яка буде надсилати коментарі.

• Перевірте, чи правильно працює поле введення коментарів та чи можна ввести та надіслати дані.

15. Виведення хімічної та математичних формул за допомогою тегів <sub> та <sup>

• Створіть новий HTML-документ.

• Додайте заголовок для сторінки, наприклад, «Хімічні та математичні формули за допомогою тегів <sub> та <sup>".

• Вставте текст з хімічною формулою, де будуть вказані атоми.

• Використайте тег <sub> для нижніх індексів, щоб позначити номери атомів.

• Використайте тег <sup> для верхніх індексів у математичних формулах.

• Переконайтеся, що хімічні та математичні формули на вашій сторінці виглядає коректною та всі індекси розташовані правильно.

#### Контрольні запитання

1. Який тег HTML використовується для визначення абревіатур або акронімів?

2. Яким атрибутом <a> тегу визначається призначення посилання?

3. Як браузери зазвичай відображають невідвідані, відвідані та активні посилання за умовчанням?

4. Які теги HTML використовувалися для визначення більшого тексту та акронімів в HTML 4, але не підтримуються в HTML5?

5. Що робить тег <br> в HTML?

6. Який тег HTML використовується для визначення кнопок, які можна натиснути?

7. Які поради надаються щодо використання тега <br/>button>?

8. Що визначає тег <cite> в HTML?

9. Для чого використовується тег <code> в HTML?

10. Яким чином тег <dfn> використовується в HTML?

11. Яка семантика тегів <em> та <i>??

12. Що робить атрибут alt y тегу <img>?

13. Чому рекомендується завжди вказувати атрибути width та

height y tery <img>?

14. Яку функцію виконує тег <input> в HTML?

15. Які типи даних можна вводити за допомогою тегу <input> та атрибута type?

16. Що визначає тег <map> в HTML? Які елементи можуть міститися всередині нього?

17. Для чого використовується тег <object>? Які ресурси можуть бути вбудовані за допомогою цього тегу?

18. Як можна забезпечити зв'язок між тегами <map> та <img>?

19. Який призначення тегу <kbd> в HTML? Які його особливості?

20. Чому рекомендується використовувати атрибут іd у поєднанні з тегом <label>?

21. Які призначення мають плагіни, які можуть бути вбудовані за допомогою тегу <object>?

22. Який призначення тега <output> у HTML і коли він був вперше введений?

23. Що відрізняє тег <q> від тега <blockquote>?

24. Для чого використовується тег <script> у HTML? Які можливості надає цей тег?

25. Які атрибути використовуються у тезі <select> для зазначення параметрів в розкривному списку?

26. Який призначення тега <small> і в яких ситуаціях його рекомендується використовувати?

27. Що таке тег <span> у HTML і які можливості він надає для стилізації тексту та групування елементів?

28. Для чого призначений тег <textarea> у HTML, і які атрибути використовуються для визначення розмірів текстової області?

29. Які теги використовуються для відображення тексту верхнім і нижнім індексом, і які їхні основні застосування?

30. Який призначення тегу <strong> в HTML, і яка особливість

його використання порівняно з тегом <b>?

31. Яке призначення тегу <time> у HTML, і як використовується атрибут datetime для цього тегу?

32. Чому тег <tt> був вилучений у HTML5, і які теги рекомендується використовувати замість нього для відображення тексту моноширинним шрифтом?

33. Яке призначення тегу <var> у HTML, і чому рекомендується уникати використання вбудованих елементів в цьому тегу?

#### Лабораторна робота №15

Тема роботи: Використання атрибута класу в HTML.

**Мета роботи:** Ознайомитися із синтаксисом та застосуванням атрибуту класу в HTML для створення структурованого та стильового контенту на WEB-сторінках.

#### Завдання

#### 1. Стилізація елементів за допомогою класів

• Створіть HTML-сторінку з наступною структурою:

о Заголовок рівня 1 з текстом «Моя сторінка».

о Два абзаци тексту з довільним вмістом.

о Список, що складається з трьох пунктів.

 Зображення з довільним посиланням та альтернативним текстом «Моя картинка».

• Визначте три CSS-класи .title, .paragraph, та .image.

• За допомогою цих класів стилізуйте кожен розділ наступним чином:

о . title: зелений колір тексту, розмір шрифту 24 пікселі, курсив.

 .paragraph: вирівнювання тексту за лівим краєм, мінімальна ширина 200 пікселів, відступи з усіх сторін по 10 пікселів.

 .image: обрамлення рамкою товщиною 2 пікселі, колір рамки синій, вирівнювання по центру.

#### 2. Стилізація елементів за допомогою класів

• Створіть HTML-сторінку з наступною структурою:

о Заголовок рівня 1 з текстом «Мої улюблені ігри».

о П'ять абзаців тексту з коротким описом вашої улюбленої гри.

о П'ять зображень обкладинок або скріншотів цих ігор.

• Визначити два CSS-класи: .title та .game-description.

о .title: вирівняти текст по центру та змінити колір на темно-

сірий.

о .game-description: зробити текст курсивним та змінити колір на світло-сірий.

• Впорядкувати зображення обкладинок або скріншотів у ряди.

3. Додавання кількох класів до одного елемента

• Створіть HTML-сторінку з абзацом тексту, який містить цитату та ім'я автора.

• Визначте два CSS-класи: .quote та .author.

• За допомогою цих класів стилізуйте цитату та ім'я автора наступним чином:

о .quote: курсив, відступ зліва 40 пікселів.

о .author: вирівнювання по правому краю, курсив, будь-який шрифт на ваш вибір.

• Додайте обидва класи до абзацу тексту, щоб стилізувати як цитату, так і ім'я автора.

4. Додавання кількох класів до одного елемента

• Створіть HTML-сторінку з абзацом тексту, який містить опис місця події та дату.

• Визначте два CSS-класи: .event-description та .date.

• Стилізуйте опис місця події та дату відповідно до вказаних умов:

о .event-description: колір тексту синій, жирний шрифт, вирівнювання за лівим краєм.

о .date: розмір шрифту 14 пікселів, курсив, колір тексту сірий.

• Додайте обидва класи до абзацу тексту, щоб стилізувати як опис місця події, так і дату.

5. Використання класів для динамічного додавання стилів за допомогою JavaScript

• Створіть HTML-сторінку з кнопкою та трьома елементами div.

• Кожному елементу div присвойте клас box.

• Додайте до кнопки подію onclick, яка буде додавати/видаляти клас active до першого елемента div при кожному кліку.

- Визначте два CSS-класи: .box та .active.
- Стилізуйте елементи з класом .box наступним чином:

 .box: ширина 100 пікселів, висота 100 пікселів, фоновий колір #ccc, відступ 10 пікселів.

- Стилізуйте елементи з класом .active наступним чином:
  - о .active: фоновий колір #000, колір тексту #fff.

# 6. Використання класів для динамічного додавання стилів за допомогою JavaScript

• Створіть HTML-сторінку зі списком елементів .

• При кліку на будь-який елемент списку, його текст повинен змінюватися.

• Визначте два CSS-класи: .list-item та .active.

• Стилізуйте елементи з класом .list-item та .active відповідно до вказаних умов:

 .list-item: курсор повинен змінювати форму при наведенні на елемент.

о .active: колір тексту змінюється на червоний та робимо текст жирним.

# 7. Використання класів для динамічного додавання стилів за допомогою JavaScript

• Створіть HTML-сторінку зі списком зображень.

• При наведенні курсора миші на будь-яке зображення, його розмір має збільшуватися на 20%.

• Визначте два CSS-класи: .image та .enlarged.

• Стилізуйте елементи з класом .image та .enlarged відповідно до вказаних умов:

 .image: встановлюємо початкові розміри зображень та застосовуємо плавну анімацію для зміни розміру.  .enlarged: застосовуємо збільшення розміру на 20% до зображень, які мають клас .enlarged.

#### 8. Використання JavaScript для вибору елементів за класом

• Створіть HTML-сторінку з декількома елементами, які мають клас іtem. Це можуть бути абзаци, списки або будь-які інші елементи HTML.

• Додайте CSS-код для стилізації цих елементів. Ви можете встановити колір фону, шрифт, розміри та інші властивості.

• Додайте кнопку до HTML-коду. Ви можете використовувати будь-який дизайн та текст для кнопки.

• При натисканні на кнопку використовуйте JavaScript, щоб:

о Знайти всі елементи на сторінці з класом іtem.

о Змінити колір фону кожного знайденого елемента на червоний.

#### 9. Створення картки продукту

• Створіть HTML-структуру для картки продуктів, яка містить зображення, назву, опис, ціну та кнопку «Додати до кошика».

• Визначте CSS-класи для стилізації картки продуктів, зображення, тексту та кнопки:

 product-card: Цей клас буде застосовано до основного контейнера картки продукту.

о product-image: Цей клас буде застосовано до зображення продукту.

о product-name: Цей клас буде застосовано до назви продукту.

о product-description: Цей клас буде застосовано до опису продукту.

о product-price: Цей клас буде застосовано до ціни продукту.

о add-to-cart-button: Цей клас буде застосовано до кнопки «Додати до кошика».

• Додайте JavaScript, який буде:

о Додавати клас active до кнопки «Додати до кошика» при

наведенні курсору миші.

• Видаляти клас active з кнопки «Додати до кошика», коли курсор миші виходить з неї.

#### 10. Створення галереї зображень з фільтрами

• Створіть HTML-сторінку з галереєю зображень, де кожне зображення має категорію.

• Додайте кнопки фільтрів для відображення лише зображень певних категорій.

• Визначте CSS-класи для стилізації галереї, зображень, кнопок фільтрів та активної кнопки фільтра:

о gallery: Цей клас буде застосовано до основного контейнера галереї.

о gallery-image: Цей клас буде застосовано до кожного зображення в галереї.

о filter-button: Цей клас буде застосовано до кнопок фільтра.

о active-filter: Цей клас буде додано до активної кнопки фільтра.

• Додайте JavaScript, який буде:

о Фільтрувати зображення на основі вибраної категорії.

о Додавати клас active до активної кнопки фільтра.

о Видалити клас active з інших кнопок фільтра.

#### 11. Створення інтерфейсу чату

• Створіть HTML-структуру для інтерфейсу чату, який містить поле для введення повідомлення, список повідомлень та кнопку надсилання.

• Визначте CSS-класи для стилізації інтерфейсу чату, поля введення, списку повідомлень та окремих повідомлень:

о chat-container: Цей клас буде застосовано до основного контейнера чату.

о chat-input: Цей клас буде застосовано до поля введення повідомлення.

о chat-messages: Цей клас буде застосовано до списку повідомлень.

о chat-message: Цей клас буде застосовано до кожного окремого повідомлення.

о incoming-message: Цей клас буде додано до вхідних повідомлень.

о outgoing-message: Цей клас буде додано до вихідних повідомлень.

• Додайте JavaScript, який буде:

о Додавати нові повідомлення до списку повідомлень.

 Эмінювати стиль повідомлень залежно від їхнього типу (вхідні, вихідні).

о Автоматичне прокручування списку повідомлень донизу.

#### 12. Створення інтерактивної таблиці

• Створіть HTML-сторінку з таблицею, яка містить дані про продукти.

• Додайте кнопки сортування для сортування таблиці за різними стовпцями.

• Визначте CSS-класи для стилізації таблиці, заголовків стовпців, рядків даних та активного стовпчика сортування:

 .table: Клас для основної таблиці. Це допоможе вам застосувати загальні стилі до всієї таблиці.

 .table-header: Клас для заголовка таблиці. Використовуйте його для застосування стилів до заголовків стовпців.

 .table-row: Клас для рядків таблиці. Це допоможе вам застосувати стилі до кожного рядка даних.

 .table-cell: Клас для комірок таблиці. Використовуйте його для застосування стилів до комірок даних.

о .active: Клас для активного стовпчика сортування. Цей клас можна додати до заголовка стовпця, щоб підкреслити, який стовпчик

сортується.

 .max-value: Клас для виділення максимальних значень.
Використовуйте його для стилізації рядків, які містять максимальні значення в кожному стовпці.

• Додайте JavaScript, який буде:

о Сортувати таблицю за вибраним стовпчиком.

о Додавати клас active до активного стовпчика сортування.

 Эмінювати стиль рядків даних залежно від їхнього значення (наприклад, виділяти максимальне значення).

### 13. Динамічні зміни класів

- Створити HTML-структуру:
  - о Створити елементи HTML, які будуть змінювати свої стилі.
- Визначити CSS стилі:
  - о Визначити стандартні стилі елементів за допомогою CSS.

о Визначити додаткові стилі для спеціальних станів, таких як наведення миші.

• Використати класи CSS:

о Визначити два класи CSS: .default для початкового стану та .special для змінених станів.

• Написати JavaScript код:

о Використовувати addEventListener для додавання обробників подій, таких як mouseover або click.

 В функціях обробників подій додати або видалити клас .special для зміни стилів елементів.

# 14. JavaScript із використанням атрибуту class

• Створити HTML сторінку:

○ Створити декілька елементів зі спільним класом, наприклад, <div> або , і додати їм клас, який хочете приховати при натисканні на кнопку.

• Написати JavaScript-функцію:

о Створити JavaScript-функцію, яка приховує всі елементи з певним класом.

о Використовуйте метод getElementsByClassName для отримання всіх елементів з певним класом.

о Змініть стиль або встановіть атрибут style.display на "none" для кожного отриманого елемента.

• Додати кнопку на сторінку:

○ Додайте кнопку <button> на сторінку HTML.

 Пов'яжіть цю кнопку з написаною JavaScript-функцією за допомогою події onclick.

• Використати класи:

о В HTML створіть декілька елементів зі спільним класом, наприклад, class="hideable".

о У JavaScript використовуйте getElementsByClassName("hideable"), щоб отримати всі елементи з цим класом.

• У JavaScript, встановіть стиль або додайте клас, який приховує елементи при натисканні на кнопку.

#### 15. Форма з використанням класів

• Створити HTML форму:

 Створити HTML форму з декількома полями введення, такими як текстові поля, чекбокси, радіокнопки тощо.

 Додати класи до цих елементів форми для подальшої стилізації та ідентифікації в JavaScript.

• Використати класи для стилізації:

 Додати CSS класи для елементів форми, щоб задати їм вигляд, розташування і стилізацію.

• Взаємодія з елементами форми в JavaScript:

 Використати JavaScript для взаємодії з елементами форми за допомогою їх класів.

• Наприклад, можна використовувати document.querySelector для вибору елементів за їх класами та встановлення подій на ці елементи для обробки введених даних або взаємодії з користувачем.

#### Контрольні запитання

- 1. Що визначає атрибут HTML class?
- 2. Які цілі використовуються для атрибуту class у HTML?
- 3. Як можна використовувати атрибут class y JavaScript?

4. Яким чином JavaScript отримує доступ до елементів за їхнім класом?

5. Яка особливість назв класів у відношенні до регістру?

6. Як визначаються кілька класів для одного елемента HTML?

7. Які методи можна використовувати для отримання елементів з певним класом у JavaScript?
# Лабораторна робота №16

#### Тема роботи: HTML Iframes

Мета роботи: Навчитися використовувати HTML Iframes для створення інтерактивної інформаційної платформи

#### Завдання

#### 1. Вбудовування WEB-сторінки YouTube

- Створіть HTML-сторінку з тегом <iframe>.
- Визначте атрибут src для вбудовування відео з YouTube.
- Додайте атрибут title з описом відео.

• Налаштуйте розмір iframe за допомогою атрибутів height та width.

#### 2. Створення динамічної галереї зображень

• Створіть HTML-сторінку з контейнером для галереї.

• Використовуйте JavaScript для динамічного завантаження зображень з масиву URL-адрес.

• Для кожного зображення створіть тег <iframe> з відповідним URL-адресою.

- Додайте кнопки навігації для переходу між зображеннями.
- 3. Вбудовування карти Google Maps
- Створіть HTML-сторінку з тегом <iframe>.

• Визначте атрибут src для вбудовування карти Google Maps з певною локацією.

• Додайте атрибути width та height для налаштування розміру карти.

• Використовуйте параметри URL-адреси для налаштування додаткових функцій карти, таких як маркери, масштабування та елементи керування.

4. Розробка системи голосування

• Створіть HTML-сторінку з трьома <iframe>: один для відображення опитування, один для відображення результатів голосування та один для відображення форми голосування.

• Використовуйте HTML та CSS для створення форми голосування та відображення результатів.

• Дозвольте користувачам голосувати за варіанти опитування в третьому <iframe>.

#### 5. Вбудовування потокової трансляції

• Створіть HTML-сторінку з іframe для відображення потокової трансляції з WEB-камери або іншого джерела.

• Використовуйте технології HTML5 Media Streams або сторонні плеєри для відтворення потоку.

• Додайте елементи керування, такі як пауза, відтворення та регулювання гучності.

#### 6. Портфоліо

• Створіть WEB-сайт з портфоліо, де будете використовувати iframe для вбудовування зразків своїх робіт з різних WEB-сайтів. Це може включати WEB-дизайни, код, графіку, відео тощо.

• Використовуйте різні атрибути iframe, такі як height та width, щоб контролювати розмір та макет вбудованих елементів.

• Додайте заголовки та описи до кожного iframe, щоб користувачі могли зрозуміти, що вони дивляться.

# 7. Новинний сайт

• Розробіть WEB-сайт новин, який використовує iframe для вбудовування контенту з різних джерел, таких як BBC, CNN, Reuters тощо.

• Створіть різні розділи для різних категорій новин, наприклад, політика, спорт, бізнес, розваги.

• Використовуйте iframe для відображення заголовків, коротких описів та посилань на повні статті з кожного джерела.

нтернет-магазин:

# 8. Інтернет-магазин

• Створіть інтернет-магазин, який використовує iframe для вбудовування продуктових відео з YouTube.

• На сторінці кожного продукту відображайте основне зображення, опис, ціну та кнопку «Додати в кошик».

• Додайте iframe під описом продукту, щоб вбудувати відповідне відео з YouTube.

# 9. Освітній WEB-сайт

• Розробіть WEB-сайт для освіти, який використовує iframe для вбудовування інтерактивних ігор та вікторин.

• Створіть різні розділи для різних предметів, наприклад, математика, наука, історія, мова.

• Використовуйте iframe для вбудовування iгор та вікторин з таких сайтів, як Education.com, Funbrain або Khan Academy.

# 10. Вбудувати календар у WEB-сайт

• Виберіть онлайн-календар, який пропонує код для вбудовування (наприклад, Google Calendar або FullCalendar).

• Створіть базовий HTML-файл з заголовком та описом.

• Використовуйте тег <iframe> для вбудовування коду календаря у ваш HTML-файл.

• Налаштуйте розмір та стиль iframe за допомогою атрибутів height, width ta style.

• Додайте заголовок до iframe за допомогою атрибуту title.

# 11. Вбудувати форму для збору даних

• Створіть HTML-файл з заголовком та описом форми.

• Використовуйте тег <iframe> для вбудовування форми Google Forms або іншої онлайн-форми.

- Налаштуйте розмір та стиль iframe.
- Додайте заголовок до iframe за допомогою атрибуту title.

# 12. Вбудувати онлайн-гру

• Знайдіть онлайн-гру, яка пропонує код для вбудовування (наприклад, HTML5 game aбo Unity web game).

• Створіть HTML-файл з контейнером для онлайн-гри.

• Використовуйте тег <iframe> для вбудовування коду онлайнгри у ваш HTML-файл.

- Налаштуйте розмір та стиль iframe.
- Додайте заголовок до iframe за допомогою атрибуту title.

# 13. Вбудувати інструмент для перекладу

• Використовуйте онлайн-сервіс перекладу, який пропонує код для вбудовування.

• Створіть HTML-файл з контейнером для інструменту перекладу.

• Використовуйте тег <iframe> для вбудовування коду вбудовування інструменту перекладу у ваш HTML-файл.

• Налаштуйте розмір та стиль iframe.

• Додайте елементи керування інструментом перекладу, такі як кнопки вибору мови.

# 14. Вбудувати онлайн-калькулятор

• Знайдіть онлайн-калькулятор, який пропонує код для вбудовування (наприклад, Desmos calculator або Mathway calculator).

• Створіть HTML-файл з контейнером для онлайн-калькулятора.

• Використовуйте тег <iframe> для вбудовування коду вбудовування онлайн-калькулятора у ваш HTML-файл.

- Налаштуйте розмір та стиль iframe.
- Додайте заголовок до iframe за допомогою атрибуту title.
- 15. Вбудовування аудіоплеєра

• Знайдіть онлайн-сервіс для зберігання аудіо (наприклад, SoundCloud або Bandcamp).

- Завантажте аудіофайл, який ви хочете вбудувати.
- Скопіюйте код для вбудовування з онлайн-сервісу.
- Створіть HTML-файл з контейнером для аудіоплеєра.
- Вставте код для вбудовування у ваш HTML-файл.

# Контрольні запитання

- 1. Що таке HTML Iframes і яку функцію вони виконують?
- 2. Який синтаксис використовується для визначення iframe в

# HTML?

- 3. Що атрибут src робить в iframe?
- 4. Чому рекомендується завжди включати атрибут title для <iframe>?
  - 5. Як встановити висоту і ширину iframe?
  - 6. Як видалити межу (рамку) навколо iframe?
  - 7. Як iframe може бути використаний як ціль для посилання?

#### Лабораторна робота №17

**Тема роботи:** Роль JavaScript у підвищенні інтерактивності та динамічності WEB-сторінок.

**Мета роботи:** Аналіз методів вставки JavaScript коду в HTML документ, вивчення основних можливостей JavaScript, розгляд використання тегу HTML <noscript> для надання альтернативного вмісту користувачам.

#### Завдання

\*До кожного завдання повинен йти CSS файл, в якому ви стилізуєте вашу HTML-сторінку для поліпшення зовнішнього вигляду. CSS повинен містити – фон, колір тексту, вирівнювання тексту, міжрядковий інтервал, шрифт та розмір, відступи і т.д.

#### 1. Створення інтерактивної форми з валідацією

- Створіть HTML-форму з полями для введення даних:
  - о Ім'я
  - о Електронна пошта
  - о Телефонний номер
  - о Пароль (з підтвердженням)
  - о Коментар (необов'язкове)

• Використовуйте відповідні теги HTML-форм (<input>, <textarea>, <select>, <label> тощо) для кожного поля.

• Використовуйте JavaScript для перевірки даних, введених користувачем, на:

 Обов'язкові поля: Переконайтеся, що всі обов'язкові поля заповнені.

о Формат: Перевірте правильність формату даних, таких як електронна пошта (наприклад, email-regex) та телефонний номер.

о Пароль: Переконайтеся, що пароль та його підтвердження

збігаються.

• Використовуйте методи JavaScript (addEventListener, preventDefault, querySelector, validity тощо) для:

о Перехоплення подій вводу користувача.

о Перевірки значень введених даних.

о Відображення повідомлень про помилки або підтвердження успішного введення.

# 2. Динамічний генератор порад

• Створити HTML-сторінку з заголовком, кнопкою та контейнером для відображення порад.

• Написати JavaScript-код, який:

 Эберігає масив або список з різними короткими порадами (текстовими рядками).

- При натисканні на кнопку:
  - о Випадково обирає пораду з масиву.

 ⊙ Динамічно створює елемент HTML (наприклад, ) для відображення обраної поради.

# 3. Робота з атрибутами

- Створіть HTML-сторінку з елементом, таким як:
  - о <img> з атрибутом src, який вказує на URL зображення.
  - $\circ$  <a> з атрибутом href, який вказує на URL посилання.
- Використовуйте методи JavaScript getAttribute() та setAttribute()

для:

- о Отримання поточного значення атрибута.
- о Зміни значення атрибута на нове значення.

# 4. Використання тегу <noscript> для альтернативного вмісту

• Додайте на сторінку деякий вміст, який буде відображатися тільки якщо JavaScript вимкнений у браузері:

- о Інтерактивні елементи (форми, кнопки, анімації)
- о Динамічно завантажувані зображення або текст

• В тегу <noscript> вставте альтернативний контент, який буде відображатися, якщо JavaScript вимкнено.

• Зробіть цей вміст корисним для користувача, який не має можливості використовувати JavaScript, наприклад, надайте пояснення або альтернативний спосіб взаємодії.

#### 5. Сховати/показати елемент

- Створіть HTML-сторінку з:
  - о Кнопка з текстом «Сховати/показати».
  - о Елемент, який ви хочете приховати/показати (наприклад, ).
- За замовчуванням елемент має бути видимим.

• Використовуйте JavaScript, щоб приховати елемент при натисканні на кнопку.

• При повторному натисканні кнопки елемент має знову з'являтися.

#### 6. Динамічний список завдань (To-Do List)

- Створіть форму з:
  - о Полем вводу для тексту завдання.
  - о Кнопкою «Додати».

• Створіть список завдань (наприклад, ), де будуть відображатися завдання.

- Кожне завдання має містити:
  - о Текст завдання.
  - о Прапорець для позначення виконання.
  - о Кнопку видалення.

• Використайте JavaScript для додавання нових завдань у список, видалення завдань та відмітки їх як виконані.

• Створіть поле введення з можливістю автозаповнення.

• Використайте JavaScript для зчитування даних з сервера або локального набору даних та автоматичного заповнення поля вводу зі списком можливих варіантів під час введення користувачем.

7. Односторінковий WEB-сайт з динамічним переходом між розділами

• Створіть HTML-файл для вашого односторінкового WEBсайту.

• Розділіть контент на окремі секції, кожна з яких буде представляти окремий розділ.

• Додайте меню навігації з кнопками, які відповідають кожному розділу.

• Напишіть JavaScript-код, який буде обробляти натискання кнопок меню навігації.

• При натисканні кнопки:

о Знайдіть відповідну секцію за її HTML-ідентифікатором.

о Використовуйте JavaScript-метод scrollIntoView(), щоб плавно прокрутити цю секцію до верху вікна.

#### 8. Анімація елементів сторінки

• Створіть базову HTML-структуру з елементом head та body.

• Додайте HTML-елемент, який ви хочете анімувати, наприклад, зображення (<img>) або блок тексту (<div>).

• Використайте JavaScript для створення анімацій, таких як переміщення, зміна розміру або зміна кольору цього елемента при взаємодії користувача зі сторінкою.

# 9. Зміна теми сторінки

• Створіть сторінку з двома або більше темами оформлення (наприклад, light-theme та dark-theme).

• Додайте елементи керування, такі як кнопки, щоб користувачі могли перемикатися між темами.

• Використайте JavaScript для зміни оформлення сторінки при виборі користувачем певної теми.

# 10. Зміна кольору тексту

• Створіть HTML-сторінку з абзацом тексту.

• Використовуйте JavaScript, щоб змінити колір тексту на будьякий інший колір при наведенні курсора миші на абзац.

• Коли курсор миші виходить з абзацу, колір тексту має повертатися до вихідного.

#### 11. Зміна розмірів та масштабування зображень

• Створіть сторінку з зображеннями, які можна масштабувати.

• Використовуйте JavaScript для зміни розмірів зображень за допомогою ползунка або кнопок.

• Додайте можливість перегляду зображення в повноекранному режимі.

#### 12. Зміна розмірів та масштабування зображень

• Створіть сторінку з зображеннями, які можна масштабувати.

• Використовуйте JavaScript для зміни розмірів зображень за допомогою ползунка або кнопок.

• Додайте можливість перегляду зображення в повноекранному режимі.

# 13. Попередження про вимкнення JavaScript

• Створіть HTML-сторінку з банером, який з'являється, якщо JavaScript вимкнено.

• Банер має повідомляти користувача про те, що для повноцінного використання сайту рекомендується ввімкнути JavaScript.

• В тег <noscript> можна додати посилання на інструкції щодо ввімкнення JavaScript в різних браузерах.

# 14. Менеджер паролів

• Створіть сторінку для зберігання та управління паролями користувача.

- Додайте форму для введення та редагування паролів.
- Додайте список для відображення збережених паролів.

• Використовуйте JavaScript для додавання, видалення та зміни

паролів у списку.

• Зашифруйте паролі, щоб забезпечити безпеку користувача.

#### 15. Анімація інтерактивних кнопок

• Створіть набір інтерактивних кнопок, наприклад, кнопок з іконками ваших соціальних мереж та посиленнями на них.

• Використовуйте JavaScript для додавання анімаційних ефектів, наприклад, зміни кольору чи розміру кнопки при наведенні миші.

• Додайте анімаційний перехід між станами кнопки (наприклад, зміна кольору фону при кліку).

# Контрольні запитання

1. Що робить JavaScript у WEB-розробці?

2. Як використовується тег *<script>* в HTML документі?

3. Як внутрішній JavaScript відрізняється від зовнішнього JavaScript у контексті тегу <script>?

4. Як можна вибрати елемент HTML за допомогою JavaScript?

5. Які можливості має JavaScript щодо маніпулювання вмістом, стилями та атрибутами HTML-елементів?

6. Що робить тег **<noscript>** і коли він використовується?

#### Лабораторна робота №18

**Тема роботи:** Створення WEB-форми з використанням HTML та JavaScript.

Мета роботи: Ознайомитися з основними тегами та атрибутами HTML для створення WEB-форм. Навчитися використовувати JavaScript для перевірки та валідації даних, введених користувачами у WEB-формі.

#### Завдання

#### 1. Інтерактивна форма ресстрації

• Створіть форму реєстрації для нового користувача на WEBсайті.

• Форма повинна містити поля для введення імені, адреси електронної пошти, пароля, підтвердження пароля та інших релевантних даних (наприклад, дата народження, стать).

• Використовуйте атрибути HTML5 для перевірки введення (наприклад, required, pattern, min, max, email).

• Запропонуйте користувачеві можливість вибрати бажаний спосіб отримання новин (наприклад, email, SMS).

 Додайте поле для завантаження фотографії профілю (з обмеженням розміру та типу файлу).

• Реалізуйте логіку JavaScript для перевірки підтвердження пароля та відображення повідомлення про помилку, якщо воно не співпадає з паролем.

• Додайте кнопку «Зареєструватися», яка надсилатиме дані форми на сервер для обробки.

- Оформити візуальне представлення форми:
  - о Кольори, шрифти, розміри елементів.
  - о Розташування полів.
  - о Ефекти при наведенні курсору, натисканні.

#### 2. Форма для опитування

• Створіть форму для опитування, щоб отримати відгуки від користувачів.

• Форма повинна містити заголовок опитування, вступний текст, опис мети опитування, різноманітні запитання (з вибором однієї/кількох відповідей, текстові) та кнопку «Відправити».

• Використовуйте різні типи запитань, такі як:

о Запитання з вибором однієї відповіді (radio buttons).

о Запитання з вибором кількох відповідей (checkboxes).

о Текстові поля для відкритих відповідей.

• Додайте кнопку «Відправити», яка надсилатиме дані опитування на сервер для зберігання та аналізу.

• Запропонуйте користувачеві можливість пройти опитування анонімно.

• Зробіть форму візуально привабливою та зручною для користувачів.

#### 3. Форма для завантаження файлів

• Створіть форму, яка дозволяє користувачам завантажувати файли на сервер.

• Дозвольте користувачеві вибрати один або кілька файлів для завантаження.

• Використовуйте елемент input type="file" для вибору файлів.

• Обмежте розмір та тип завантажуваних файлів.

• Відобразіть хід завантаження файлу та повідомлення про успішне або невдале завантаження.

- Додайте можливість перегляду.
- Використовувати CSS для візуального оформлення форми:
  - о Зовнішній вигляд елементів input, button, div.

о Шрифти, кольори, вирівнювання.

#### 4. Форма замовлення

• Створіть HTML-форму для замовлення товарів або послуг.

• Форма повинна містити поля для введення інформації про клієнта (ім'я, адреса, контактні дані), інформації про замовлення (товари, кількість, ціна).

• Кнопка «Замовити» для надсилання форми.

• Відображення повідомлень про підтвердження або помилку.

• Використовуйте атрибути HTML5 для перевірки введення (наприклад, required, email).

• Додайте повідомлення про помилки для недійсного введення.

• Запропонуйте користувачеві вибрати спосіб доставки та оплати.

- Використовуйте CSS для:
  - о Стилізації форми (розмір, колір, відступи).
  - о Стилізації елементів керування (поля введення, списки вибору,

кнопки).

о Забезпечення чіткої та зручної структури для користувачів.

# 5. Форма з валідацією на стороні клієнта

• Створіть HTML-форму, яка використовує атрибути HTML5 та JavaScript для перевірки даних, введених користувачем.

- Перевіряйте наявність помилок, таких як:
  - о Незаповнені поля.
  - о Неправильний формат електронної пошти.
  - о Неправильний формат дати.
  - о Неправильний формат номера телефону.

• Відображайте повідомлення про помилки користувачеві, щоб він міг їх виправити:

о span або div елементи з класом error-message біля кожного поля.

о Початкове значення display: none для повідомлень про

помилки.

- **Атрибути HTML5**:
  - о required для позначення обов'язкових полів.
  - о pattern для визначення формату даних (email, дата, телефон).
  - о min та max для обмеження діапазону значень (числові поля).
- Кнопка «Відправити»:
  - о button type="submit" для надсилання форми.
- Використовуйте CSS для:
  - о Стилізації форми (розмір, колір, відступи).
- Стилізації елементів керування (поля введення, списки вибору, кнопки).
  - о Забезпечення чіткої та зручної структури для користувачів.
  - 6. Форма з динамічною генерацією полів

• Створіть HTML-форму, яка динамічно генерує поля введення залежно від вибраних користувачем параметрів.

• Використовуйте JavaScript, щоб додати або видалити поля з форми на основі вибраних значень.

• Функція для додавання поля:

о Створити елементи div, label, input (з відповідним типом) та кнопку «Видалити».

о Додати динамічно створені елементи до div#dynamicFields.

 Налаштувати події для кнопки «Видалити» (видалення елемента).

• Це може бути корисно для форм, які потребують різної кількості інформації залежно від ситуації.

- Використовуйте CSS для:
  - о Стилізації форми (розмір, колір, відступи).

 Стилізації елементів керування (поля введення, списки вибору, кнопки).

о Забезпечення чіткої та зручної структури для користувачів.

# 7. Форма вибору кольору

• Створіть HTML-форму, яка дозволяє користувачеві вибрати колір за допомогою поля введення типу «color».

- Відобразіть вибраний колір у форматі НЕХ або RGB.
- Підтримка браузерів:

о Переконайтеся, що поле input type="color" правильно підтримується в браузерах, які використовує ваша цільова аудиторія.

- Використовуйте CSS для:
  - о Стилізації форми (розмір, колір, відступи).
- Стилізації елементів керування (поля введення, списки вибору, кнопки).
  - о Забезпечення чіткої та зручної структури для користувачів.

# 8. Форма календаря

• Створіть HTML-форму, яка дозволяє користувачеві вибирати дату за допомогою поля введення типу "date".

- Дозвольте користувачеві вибрати діапазон дат.
- Відобразіть вибрані дати в зручному для користувача форматі.

• Інтегруйте календар з іншими полями форми, наприклад, з полем для введення ідентифікатора замовлення.

- Використовуйте CSS для:
  - о Стилізації форми (розмір, колір, відступи).

Стилізації елементів керування (поля введення, списки вибору, кнопки).

о Забезпечення чіткої та зручної структури для користувачів.

# 9. Форма з рейтингом

• Створіть HTML-форму, яка дозволяє користувачам оцінювати щось (продукт, послугу, статтю) за допомогою зірок або інших елементів рейтингу.

- Відобразіть середню оцінку та кількість оцінок.
- Дозвольте користувачам залишати відгуки разом зі своїми

оцінками.

• Використовуйте CSS для:

о Стилізації форми (розмір, колір, відступи).

Стилізації елементів керування (поля введення, списки вибору, кнопки).

о Забезпечення чіткої та зручної структури для користувачів.

# 10. Форма з кроками

• Створіть HTML-форму, яка розділена на кілька кроків.

• Переміщуйте користувача між кроками форми за допомогою кнопок «Наступний» та «Попередній».

• Перевіряйте введені дані на кожному кроці перед переходом до наступного.

• Зберіть усі дані форми після завершення користувачем останнього кроку.

- Використовуйте CSS для:
  - о Стилізації форми (розмір, колір, відступи).

Стилізації елементів керування (поля введення, списки вибору, кнопки).

о Забезпечення чіткої та зручної структури для користувачів.

# 11. Форма пошуку

• Створіть сторінку яка має бути присвячена певній темі та містити інформативний текст, що стосується цієї теми.

• Текст має бути добре структурований, розділений на заголовки, абзаци та списки, якщо це необхідно.

• Створити вкінці сторінки HTML-форму для пошуку, яка дозволяє користувачам вводити ключові слова.

• Використовуйте поле text для введення ключових слів.

• Додайте кнопку «Пошук», яка надсилає форму та запускає пошук.

• Використовуйте атрибут placeholder для підказки користувачам

про те, що шукати.

• Після надсилання запиту на пошук сторінка має динамічно генерувати та відображати результати, відповідні введеним ключовим словам.

• Використовуйте CSS для:

о Стилізації форми (розмір, колір, відступи).

о Стилізації елементів керування (поля введення, списки вибору, кнопки).

о Забезпечення чіткої та зручної структури для користувачів.

#### 12. Форма бронювання

• Створіть HTML-форму для бронювання готелю, яка дозволяє користувачам:

- о Вводити дати заїзду та виїзду.
- о Вибирати тип номера.
- о Вказувати кількість гостей.
- о Вибирати додаткові послуги.
- о Вводити спеціальні запити.
- о Надсилати форму для підтвердження бронювання.

• Використовуйте елементи select та date для вибору дат заїзду та виїзду, типу номера та кількості гостей.

- Додайте елементи checkbox для вибору додаткових послуг.
- Додайте поле textarea для спеціальних запитів.

• Додайте кнопку «Бронювати», яка надсилає форму та підтверджує бронювання.

• Використовуйте атрибут min та max для обмеження дати заїзду та виїзду.

- Використовуйте CSS для:
  - о Стилізації форми (розмір, колір, відступи).

 Стилізації елементів керування (поля введення, списки вибору, кнопки). о Забезпечення чіткої та зручної структури для користувачів.

# 13. Інтерактивна форма для бронювання столиків у ресторані

• Створіть форму, яка дозволяє користувачам бронювати столики в ресторані.

• Включіть поля для введення дати, часу, кількості людей, а також імені та електронної пошти користувача.

• Перевірте, чи доступні столики на вказану дату та час.

• Кнопка «Бронювати»:

о button елемент з класом book-table-button та текстом «Бронювати».

• Надішліть користувачеві підтвердження бронювання.

• Використовуйте CSS для:

о Стилізації форми (розмір, колір, відступи).

о Стилізації елементів керування (поля введення, списки вибору, кнопки).

о Забезпечення чіткої та зручної структури для користувачів.

14. Форма для збору пожертв

• Створіть HTML-форму для збору пожертв для вашої некомерційної організації, яка дозволяє:

о Користувачам вводити суму пожертви.

о Користувачам вводити свою контактну інформацію (ім'я, email, телефон).

 Надавати користувачам можливість вибрати спосіб оплати (PayPal, кредитна картка, чек).

• Кнопка «Пожертвувати»:

о button елемент з класом donate-button та текстом «Пожертвувати».

• Використовуйте CSS для:

о Стилізації форми (розмір, колір, відступи).

о Стилізації елементів керування (поля введення, списки вибору,

кнопки).

о Забезпечення чіткої та зручної структури для користувачів.

# 15. Форма для подачі заявок на вакансії

• Створіть HTML-форму, яка дозволяє користувачам:

о Подавати заявки на вакансії у вашій компанії.

о Вводити свою контактну інформацію (ПІБ, email, телефон).

о Надавати інформацію про свою освіту (навчальні заклади, дати навчання, дипломи).

о Завантажувати резюме (у форматі PDF або Word).

• Кнопка «Відправити заявку»:

о button елемент з класом submit-application-button та текстом «Відправити заявку».

• Використовуйте CSS для:

о Стилізації форми (розмір, колір, відступи).

Стилізації елементів керування (поля введення, списки вибору, кнопки).

о Забезпечення чіткої та зручної структури для користувачів.

# Контрольні запитання

1. Що таке HTML-форма?

2. Який тег використовується для створення HTML-форми?

3. Які різні типи елементів можна використовувати в HTMLформі?

4. Який атрибут використовується для визначення дії, яка буде виконана під час надсилання форми?

5. Який атрибут використовується для визначення того, де буде відображатися відповідь, отримана після надсилання форми?

6. Який атрибут використовується для визначення методу НТТР, який буде використовуватися під час надсилання даних форми?

7. Який атрибут використовується для визначення, чи буде

автозаповнення увімкнено або вимкнено у формі?

8. Який атрибут використовується для визначення, чи повинні дані форми перевірятися під час надсилання?

9. Який елемент використовується для створення текстового поля?

10. Який елемент використовується для створення прапорця?

11. Який елемент використовується для створення розкривного списку?

12. Який елемент використовується для створення багаторядкового текстового поля?

13. Який елемент використовується для створення кнопки?

14. Який тип введення використовується для створення поля для введення тексту?

15. Який тип введення використовується для створення поля пароля?

16. Який тип введення використовується для створення кнопки надсилання?

17. Який тип введення використовується для створення кнопки скидання?

18. Який тип введення використовується для створення радіокнопки?

19. Який атрибут використовується для визначення початкового значення поля введення?

20. Який атрибут використовується для визначення того, чи поле введення доступне лише для читання?

21. Який атрибут використовується для визначення того, чи поле введення має бути вимкнено?

22. Який атрибут використовується для визначення видимої ширини поля введення?

23. Який атрибут використовується для визначення максимальної

кількості символів, дозволених у полі введення?

24. Який атрибут використовується для визначення мінімального та максимального значень для поля введення?

25. Який атрибут використовується для визначення того, чи користувачеві дозволено вводити більше одного значення в поле введення?

26. Який атрибут використовується для визначення регулярного виразу, за яким перевіряється значення поля введення?

27. Який атрибут використовується для визначення короткої підказки, яка описує очікуване значення поля введення?

28. Який атрибут використовується для визначення того, чи поле введення має бути заповнене перед надсиланням форми?

29. Який атрибут використовується для визначення допустимих інтервалів чисел для поля введення?

30. Який атрибут використовується для визначення того, чи поле введення має автоматично отримувати фокус під час завантаження сторінки?

31. Який атрибут використовується для визначення висоти та ширини зображення?

32. Який атрибут використовується для зв'язування поля введення з елементом <datalist>?

33. Який атрибут використовується для визначення, чи буде автозаповнення увімкнено або вимкнено для поля введення?

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1. https://www.w3.org/html/
- 2. https://html.spec.whatwg.org/
- 3. https://www.w3schools.com/html/