**Методичка до практичного заняття №3: "Основи CSS для стилізації веб-сайтів"**

**І. Основні правила СSS.**

CSS (Cascading Style Sheets) - це мова стилізації, яка використовується для визначення вигляду та форматування документів, написаних на мовах розмітки, таких як HTML. CSS дозволяє контролювати колір, розмір, шрифт, відстань між елементами, структуру сторінки та інші візуальні характеристики веб-сайтів.

Ось основні концепції, що стосуються CSS:

1. **Селектори та властивості**: В основі CSS лежить поняття селекторів та властивостей. Селектор вказує, до якого елементу HTML слід застосовувати стилі, а властивості визначають, як саме ці стилі повинні виглядати.

p { color: red; font-size: 16px; }

У цьому прикладі, **p** - це селектор, який вказує на усі абзаци в HTML-документі. **color** та **font-size** - це властивості, які задають колір та розмір шрифту відповідно.

1. **Каскадність**: Це означає, що стилі, задані на вищому рівні (наприклад, для всього документа), можуть бути перевизначені стилями на нижчому рівні (наприклад, для конкретного елемента).
2. **Наслідування**: Багато стилів, визначених для одного елемента, наслідуються його дочірніми елементами. Наприклад, якщо ви встановите колір тексту для елемента **<body>**, всі текстові елементи всередині нього за замовчуванням будуть цього ж кольору, якщо не вказано інше.
3. **Медіа-запити**: Це інструменти CSS, які дозволяють застосовувати стилі на основі характеристик пристрою, наприклад, ширини екрану. Це корисно для створення адаптивних веб-сайтів, які виглядають добре на різних пристроях.

@media (max-width: 600px) { body { font-size: 14px; } }

1. **Зовнішні, внутрішні та вбудовані стилі**: CSS може бути включений прямо в HTML-документ (вбудовані стилі), розміщений в заголовку документа (внутрішні стилі) або розміщений в окремому файлі (зовнішні стилі).

Щоб використовувати CSS, вам слід знати основи мови розмітки (зазвичай HTML), а також зрозуміти, як стилі застосовуються до елементів цієї розмітки. З часом ви зможете додавати складніші стилі та анімації для поліпшення вигляду та взаємодії вашого веб-сайту.

CSS, як і більшість мов програмування та стилізації, має строгі правила синтаксису. Ось деякі ключові аспекти пунктуації, граматики та логіки у CSS:

1. **Селектори і блоки визначень**:
	* Селектор вказує на елемент(и) HTML, до яких застосовується певний стиль.
	* Після селектора слідує блок визначень, який обмежується фігурними дужками **{}**.

селектор { властивість: значення; }

1. **Властивості та значення**:
	* Всередині блоку визначень розташовуються пари "властивість: значення".
	* Кожна пара "властивість: значення" закінчується крапкою з комою **;**.

p { color: red; font-size: 16px; }

1. **Коментарі**:
	* Коментарі у CSS починаються з **/\*** і закінчуються на **\*/**.

/\* Це коментар \*/ p { color: blue; /\* Це теж коментар \*/ }

1. **Чутливість до регістру**:
	* CSS нечутливий до регістру, але рекомендується використовувати один і той же регістр у всьому коді для послідовності.
2. **Комбіновані селектори**:
	* Селектори можуть бути комбіновані для визначення стилів для конкретних елементів або груп елементів.

h1, h2, h3 { font-family: Arial, sans-serif; }

1. **Псевдокласи та псевдоелементи**:
	* Використовуються для стилізації конкретних станів елементів або частин елементів.

a:hover { color: green; } p::first-line { font-weight: bold; }

1. **Медіа-запити**:
	* Використовуються для застосування стилів на основі характеристик пристрою.

@media (max-width: 600px) { body { font-size: 14px; } }

1. **Імпорт**:
	* Дозволяє імпортувати один CSS-файл у інший.

@import url('path/to/stylesheet.css');

1. **Значення та їх одиниці**:
	* Значення можуть бути вказані в різних одиницях, таких як **px**, **em**, **%**, **rem**, **vh**, **vw** тощо.
2. **Вкладеність та специфічність**:
	* Селектори можуть бути вкладені, і це впливає на специфічність, яка визначає, який стиль матиме перевагу, якщо декілька правил конфліктують між собою.

Використання CSS у текстовому файлі (наприклад, у форматі блокнота **.txt**) вимагає певних кроків. Якщо у вас є файл **.txt** з HTML-кодом, ви можете додати до нього стилі CSS різними способами:

1. **Вбудовані стилі**: Ви можете додати стилі безпосередньо до конкретних елементів HTML за допомогою атрибута **style**.

<p style="color: red;">Це червоний текст</p>

1. **Внутрішні стилі**: Ви можете додати блок стилів у заголовок (**<head>**) вашого HTML-документа.

<head> <style> p { color: blue; } </style> </head> <body> <p>Це синій текст</p> </body>

1. **Зовнішні стилі**: Якщо у вас є окремий **.txt** файл з CSS-кодом, ви можете посилатися на нього з вашого HTML-документа. Проте це буде працювати, лише якщо обидва файли будуть перейменовані відповідно (**.html** для HTML-коду та **.css** для CSS-коду) та розміщені в одній директорії.

<head> <link rel="stylesheet" type="text/css" href="styles.txt"> </head>

Примітка: Для оптимальної роботи ці файли слід зберігати з розширенням **.html** та **.css** замість **.txt**.

1. **Зберігання файлу**: Після того як ви додали CSS до вашого **.txt** файлу, збережіть його з розширенням **.html** (наприклад, **mywebsite.html**). Це дозволить вам відкривати файл у веб-браузері і бачити стилізований контент.

ІІ. Основні CSS-селектори та властивості

1. **Селектори**
	* Тегові селектори: **h1**, **p**, **a**.
	* Класові селектори: **.className**.
	* Ідентифікатори: **#idName**.
2. **Властивості тексту**
	* **font-family**: задає шрифт тексту.
	* **font-size**: розмір шрифта.
	* **color**: колір тексту.
3. **Властивості блоку**
	* **margin**: відступи навколо елемента.
	* **padding**: відступи всередині елемента.
	* **border**: границя навколо елемента.
	* **background-color**: колір фону елемента.

**ІІІ. Використання шрифтів
1. Використання Google Fonts:**

1. **Вибір шрифту**:
	* Перейдіть на [Google Fonts](https://fonts.google.com/).
	* Виберіть потрібний шрифт, натиснувши на нього.
2. **Вибір стилів**:
	* Виберіть стилі шрифту, які вам потрібні (наприклад, Regular 400, Bold 700 тощо).
3. **Підключення шрифту**:
	* Натисніть на кнопку "Select this style".
	* Скопіюйте посилання на шрифт з розділу "Embed" і вставте його в **<head>** вашого HTML-документа.

Наприклад:

<link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap">

1. **Використання шрифту**:
	* Скопіюйте CSS-правило, яке надається Google Fonts, і використайте його у своєму CSS.

Наприклад:

font-family: 'Roboto', sans-serif;

вбудував шрифт "Fuggles" у ваш HTML-файл:

**Підключення шрифту**

Для початку, слід додати код у розділ **<head>** вашого HTML-документа, наприклад:

<link rel="preconnect" href="https://fonts.googleapis.com"> <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin> <link href="https://fonts.googleapis.com/css2?family=Fuggles&display=swap" rel="stylesheet">

Цей код дозволяє вашому браузеру завантажити шрифт "Fuggles" з Google Fonts.

**2. Застосування шрифту в CSS**

Після того, як шрифт було підключено, слід додати наступний CSS-код, щоб застосувати шрифт "Fuggles" до заголовків **h1** та **h2**:

h1, h2 { font-family: 'Fuggles', cursive; }

Цей CSS-код говорить браузеру використовувати шрифт "Fuggles" для всіх елементів **h1** та **h2** на сторінці.

**2. Використання локальних шрифтів:**

1. **Завантаження шрифту**:
	* Завантажте файл шрифту (зазвичай у форматі .ttf або .woff) з джерела, якому ви довіряєте.
2. **Підключення шрифту**:
	* Розмістіть файл шрифту у каталозі вашого веб-проекту.
	* Використовуйте **@font-face** у CSS, щоб підключити файл.

Наприклад:

@font-face { font-family: 'MyCustomFont'; src: url('path\_to\_font\_file.woff2') format('woff2'), url('path\_to\_font\_file.woff') format('woff'); }

1. **Використання шрифту**:
	* Після підключення шрифту ви можете використовувати його у своєму CSS.

Наприклад:

font-family: 'MyCustomFont', sans-serif;

**Завдання**

**Мета:** Стилізувати веб-сторінку, створену під час практичного заняття №2.

**Кроки виконання:**

1. Виберіть і підключіть зовнішній шрифт для вашої сторінки (наприклад, з [Google Fonts](https://fonts.google.com/)).
2. Використовуючи CSS, задайте шрифт, колір тексту та фон для вашої сторінки.
3. Задайте відступи та границі для окремих елементів.
4. Стилізуйте посилання на вашій сторінці, використовуючи різні стани (**hover**, **active**).

**Приклад стилізації:**

body { font-family: 'Arial', sans-serif; background-color: #f4f4f4; color: #333; margin: 20px; } h1 { color: #ff9800; } a { color: #3f51b5; text-decoration: none; } a:hover { text-decoration: underline; } .header, .nav, .main, .footer { margin-bottom: 20px; padding: 10px; border: 1px solid #e0e0e0; background-color: #ffffff; }

Ось додатковий список з 25 CSS-властивостей та функцій, які можна використовувати для стилізації веб-сайтів:

1. **font-weight** - визначає товщину шрифта (наприклад, **normal**, **bold**, **bolder**, **lighter**).
2. **font-style** - стилізація шрифта (наприклад, **normal**, **italic**, **oblique**).
3. **text-align** - вирівнювання тексту (наприклад, **left**, **right**, **center**, **justify**).
4. **text-transform** - зміна регістру тексту (**uppercase**, **lowercase**, **capitalize**).
5. **text-decoration** - декорування тексту (**none**, **underline**, **overline**, **line-through**).
6. **line-height** - визначає відстань між рядками тексту.
7. **letter-spacing** - відстань між буквами тексту.
8. **border-radius** - створює закруглені границі для елемента.
9. **box-shadow** - додає тінь до елемента.
10. **background-image** - встановлює зображення як фон для елемента.
11. **background-repeat** - визначає, як фонове зображення буде повторюватися.
12. **background-size** - визначає розмір фонового зображення.
13. **background-position** - встановлює положення фонового зображення.
14. **opacity** - встановлює прозорість елемента.
15. **transition** - дозволяє встановити плавний перехід між двома станами елемента.
16. **transform** - застосовує 2D або 3D перетворення до елемента (наприклад, **scale**, **rotate**, **translate**).
17. **display** - визначає, як елемент буде відображатися на сторінці (**block**, **inline**, **none**, **flex** тощо).
18. **position** - визначає, як елемент розміщується в межах його батьківського контейнера (**static**, **relative**, **absolute**, **fixed**).
19. **top, right, bottom, left** - встановлює положення елемента при використанні певних значень властивості **position**.
20. **z-index** - встановлює порядок наявних елементів по осі Z.
21. **float** - розміщує елемент зліва або справа від його батьківського контейнера.
22. **clear** - визначає, як елемент має взаємодіяти з плаваючими елементами (**left**, **right**, **both**).
23. **list-style** - встановлює стиль маркера для елементів списку.
24. **cursor** - встановлює тип курсору, який відображається, коли вказівник миші знаходиться над елементом (**pointer**, **text**, **wait** тощо).
25. **@media** - медіа-запит, який дозволяє вам вказувати стилі, що застосовуються в залежності від характеристик пристрою (наприклад, розмір екрану).