

План уроку виробничого навчання

Майстра в/н Цируль Г.В.

_____2023р.

Модуль ООІ та ПЗ-2.6. Підготовка та розміщення матеріалів у мережі Інтернет

ООІ та ПЗ-2.6.3 Програмування веб-сторінок та веб-сайту мовою розмітки

Тема уроку: «Створення та підключення стилів CSS. »

Мета уроку:

а) навчальна – ознайомити, навчити створювати та підключати стилі CSS в HTML;

б) розвивальна – виробити в учнів професійні навички оператора при роботі на ПК з використанням усіх правил техніки безпеки та охорони праці.

в) виховна – виховувати професійні риси, почуття відповідальності, вміння об'єктивно оцінювати себе та інших, культуру користувача ПК.

Тип уроку: урок вдосконалення вмінь та формування навичок.

Дидактичне забезпечення (інтернет, конспект майстра, роздатковий матеріал)

Міжпредметні зв'язки: «охорона праці»,

Хід уроку:

1. Організаційна частина – 5 хв.

- 1.1 Привітання.
- 1.2 Перевірка наявності учнів.

2. Вступний інструктаж – 50 хв.

- 2.1 Повідомлення учням теми та мети уроку.
- 2.2 Актуалізація опорних знань
- 2.3 Вивчення нової теми за планом уроку.
- 2.4 Закріплення нової теми.
- 2.5 Інструктаж з охорони праці перед початком роботи.
- 2.6 Доведення денного завдання та критеріїв оцінювання.

3. Поточний інструктаж – 205 хв.

- 3.1 Перевірка організації робочих місць.
- 3.2 Робота з вправами.
- 3.3 Пояснення і демонстрування завдання невстигаючим учням.
- 3.4 Перевірка правильності виконання вправ.
- 3.5 Надання консультацій учням, у яких виявленні помилки.

4. Заключний інструктаж – 10 хв.

- 4.1 Підведення підсумків за урок.
- 4.2 Аналіз допущених помилок та недопущення їх в подальшій роботі.
- 4.3 Відзначення успішних робіт учнів.
- 4.4 Оголошення оцінок за виконане завдання.
- 4.5 Домашнє завдання: вивчити з конспекту нові записи і терміни даної теми.

Каскадні таблиці стилів (Cascading Style Sheets, CSS) — спеціальна мова, яку використовують для запису оформлення сторінок, написаних мовами розмітки даних.

Таблиці стилів дають змогу спростити процес створення сторінок і поліпшити їх зовнішній вигляд. Концепція стилів подібна до ідеї стилів, яку реалізовано в сучасних текстових редакторах — текст спочатку вводять, а потім форматують, користуючись стилями. Застосування стилів дає змогу вводити на сторінку потрібні тексти та інші елементи, не задумуючись над їхнім зовнішнім виглядом і розташуванням.

Таблиці стилів зазвичай створюють окремо від html-файлу. Під час створення html-файлу увагу концентрують на змісті сторінки, а не на її зовнішньому вигляді, а під час створення таблиці стилів — навпаки. Отже, стилі дають змогу розділити етапи створення html-файлу й удосконалення зовнішнього вигляду сторінки.

Стилізація HTML за допомогою CSS

CSS описує, як елементи HTML повинні відображатися на екрані, папері або в інших носіях .

CSS економить багато роботи. Він може керувати розташуванням декількох веб-сторінок одночасно.

CSS можна додати до елементів HTML у три способи:

- **лінійний** - за допомогою атрибута **style** у HTML-елементах
- **Внутрішній** - за допомогою **<style>** елемента в **<head>** розділі
- **Зовнішній** - за допомогою зовнішнього файлу CSS

Найпоширенішим способом додавання CSS є збереження стилів у окремих файлах CSS.

Вбудований CSS

Вбудований CSS використовується для застосування унікального стилю до одного елемента HTML.

Вбудований CSS використовує атрибут **style** елемента HTML.

Цей приклад встановлює синій колір тексту **<h1>** елемента:

Приклад **<h1 style="color:blue;">This is a Blue Heading</h1>**

Внутрішній CSS

Внутрішній CSS використовується для визначення стилю для однієї сторінки HTML.

Внутрішній CSS визначається в **<head>** розділі HTML-сторінки в межах **<style>** елемента:

Приклад

```
<html>
<head>
  <style>
    body {background-color: powderblue;}
    h1 {color: blue;}
    p {color: red;}
  </style>
</head>
<body>
```

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Зовнішній CSS

Для визначення стилю для багатьох сторінок HTML використовується зовнішня таблиця стилів.

За допомогою зовнішньої таблиці стилів ви можете змінити вигляд всього веб-сайту, змінивши один файл!

Щоб використовувати зовнішню таблицю стилів, додайте посилання на неї в **<head>** розділі HTML-сторінки:

Приклад

```
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

В будь-якому текстовому редакторі можна записати зовнішню таблицю стилів. Файл **не повинен містити жодного HTML-коду** і повинен бути збережений з **розширенням .css.**

Ось як виглядає "styles.css":

```
body {
background-color: powderblue;
}
h1 {
color: blue;
}
p {
color: red;
}
```

Шрифти CSS

Color - Властивість CSS визначає колір тексту, який буде використовуватися.

font-family - Властивість CSS визначає використовуваний шрифт.

font-size - Властивість CSS визначає розмір тексту, який буде використовуватися.

Приклад

```
<html>
<head>
  <style>
    h1 {
      color: blue;
      font-family: verdana;
      font-size: 300%;
    }
    p {
      color: red;
      font-family: courier;
      font-size: 160%;
    }
  </style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Можна створити окремий документ
наприклад

styles.css

в якому указати стилі

```
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
```

Тоді даний документ буде записаний

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="st
vles.css">
```

Межі рамок CSS

Border - Властивість CSS визначає межу навколо елемента HTML:

Приклад

```
p {
  border: 1px solid powderblue;
}
```

Padding CSS

Padding - Властивість CSS визначає **внутрішній відступ** (пробіл) між текстом і межею:

Приклад

```
p {
  border: 1px solid powderblue;
  padding: 30px;
}
```

margin CSS

margin - властивість CSS визначає **зовнішній відступ** (пробіл) елемента від іншого елемента:

Приклад

```
p {
  border: 1px solid powderblue;
  margin: 50px;
}
```

Атрибут ідентифікатора

Щоб визначити певний стиль для одного спеціального елемента, додайте до нього

id - атрибут:

```
<p id="p01">I am different</p>
```

потім визначити стиль для елемента з певним ідентифікатором:

Приклад

```
#p01 {  
color: blue;  
}
```

Примітка: Ідентифікатор елемента повинен бути унікальним у межах сторінки, тому селектор ідентифікаторів використовується для вибору одного унікального елемента!

Клас Атрибут

Щоб визначити стиль для спеціальних типів елементів, додайте **class** - до елемента атрибут:

```
<p class="error">I am different</p>
```

потім визначити стиль для елементів з певним класом:

Приклад

```
p.error {  
color: red;  
}
```

```
.error {  
color: red;  
}
```

Зовнішні посилання

На зовнішні таблиці стилів можна посилатися з повним **URL-адресою або шляхом**, пов'язаним з поточною веб-сторінкою.

Цей приклад використовує повну **URL-адресу** для посилання на таблицю стилів:

Приклад

```
<link rel="stylesheet" href="https://www.w3schools.com/html/styles.css">
```

Цей приклад посилається на таблицю стилів, розташовану **в папці html** на поточному веб-сайті:

Приклад

```
<link rel="stylesheet" href="/html/styles.css">
```

Цей приклад посилається на таблицю стилів, розташовану **в тій же папці**, що й поточна сторінка:

Приклад

```
<link rel="stylesheet" href="styles.css">
```

Практичне завдання №1

Оформіть веб-сторінку за допомогою зовнішньої таблиці стилів.

1. Відкрийте **блокнот** створіть новий документ і збережіть його у папку зі своїм прізвищем з назвою **mystyle.css** і введіть таке визначення тегу <P>:

```
p{
color:red;
font-size:40px;
font-family:"Comic Sans MS";
}
```

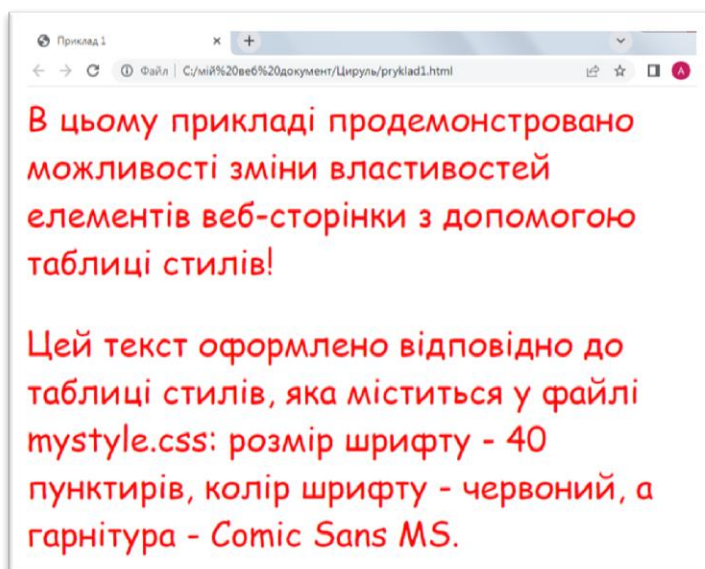
2. Збережіть файл

```
<html>
<head>
<title> Приклад 1</title>
<link rel="stylesheet"href="mystyle.css">
</head>
<body>
<p>В цьому прикладі продемонстровано можливості зміни властивостей
елементів веб-сторінки з допомогою таблиці стилів!</p>
<p>Цей текст оформлено відповідно до таблиці стилів,
яка міститься у файлі mystyle.css:
розмір шрифту - 40 пунктирів, колір шрифту - червоний,
а гарнітура - Comic Sans MS.</p>
</body>
</html>
```

3. В **блокноті** створіть новий документ і збережіть його у папку зі своїм прізвищем з назвою **pryklad1.html**:

Увага! Збережіть цей файл з іменем **pryklad1.html** у тій самій папці, що і **mystyle.css**.

5. Відкрийте файл **pryklad1.html** у вікні браузера (рис. 5.1). Як бачите, хоча в цьому документі не задано жодних параметрів для тегу <P>...</P>, текст абзацу набув нових властивостей, які описані в зовнішній таблиці стилів **mystyle.css**.

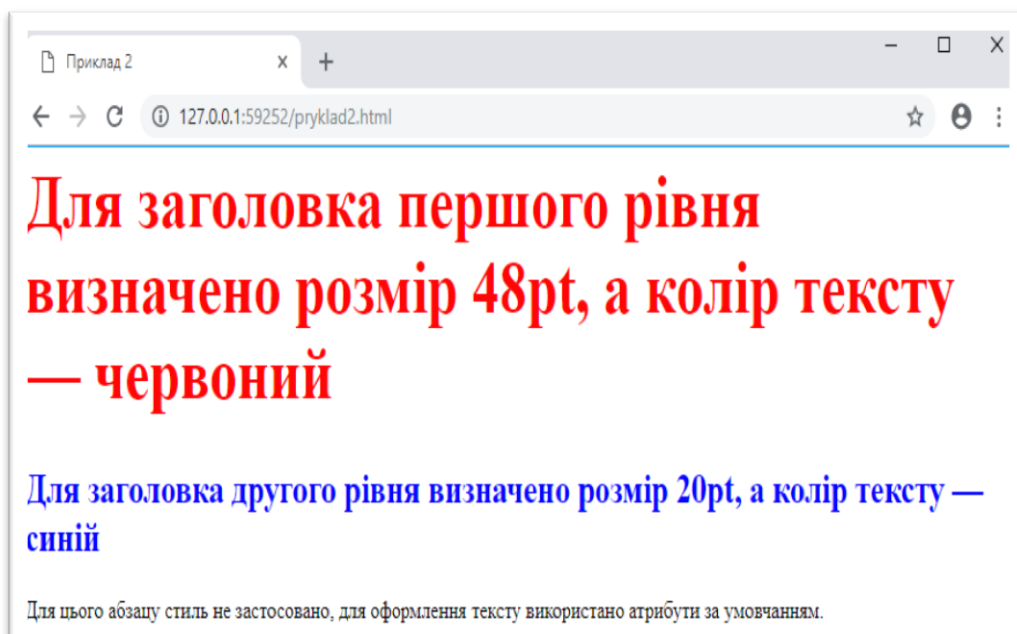


Практичне завдання №2

Визначимо стилі заголовків першого та другого рівнів із використанням тегу <STYLE>.

```
<html>
<head>
<title> Приклад 2</title>
<style>
h1 {color:red;font-size:48px;}
h2 {color:blue;font-size:20px;}
</style>
</head>
<body>
<h1>Для заголовка першого рівня визначено розмір 48px, а колір тексту -
червоний</h1>
<h2>Для заголовка другого рівня визначено розмір 20px, а колір тексту -
синій</h2>
<p> Для цього абзацу стиль не застосовано, для оформлення тексту використано
атрибут за замовчуванням.</p>
</body>
</html>
```

У цьому описі стилю змінено оформлення заголовків H1 та H2. Зокрема, для першого призначено розмір 48 пунктів і червоний колір, а для другого — розмір 20 пунктів і синій колір. У вікні браузера цей документ матиме вигляд, як на рис. Нижче

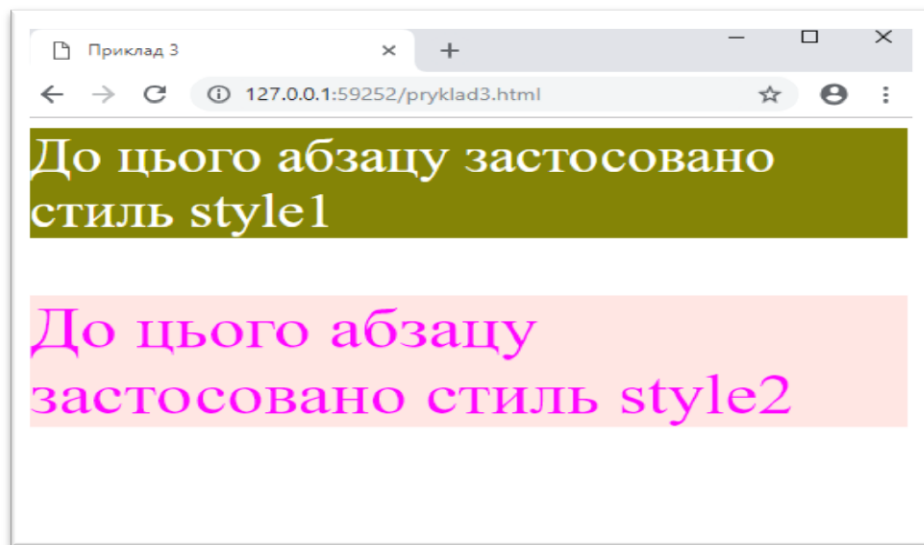


Практичне завдання №3

Застосовуючи атрибут **CLASS** тегу **</P>**, задайте для двох абзаців різні стилі: першому надайте розмір шрифту 38 пунктів, білий колір символів і оливковий колір тла, другому — розмір шрифту 46 пунктів, фіолетовий колір символів і рожевий колір тла.

```
<html>
<head>
<title> Приклад 3</title>
<style>
.style1{ font-size:38px;color:white;background-color:olive;}
.style2{ font-size:46px;color:magenta;background-color:mistyrose;}
</style>
</head>
<body>
<p class="style1">Для цього абзацу застосовано стиль style1 </p>
<p class="style2">Для цього абзацу застосовано стиль style2 </p>
</body>
</html>
```

Який вигляд має цей документ у вікні браузера, показано на рис. 5.3.



Стилі дають змогу сформувати сторінку з цікавими ефектами, які інакше можна створити лише за допомогою графіки. Це, наприклад, ефект об'ємного тексту або накладання (часткового перекриття) фрагментів сторінки.

Стилі дають змогу сформувати сторінку з цікавими ефектами, які інакше можна створити лише за допомогою графіки. Це, наприклад, ефект об'ємного тексту або накладання (часткового перекриття) фрагментів сторінки.

Практичне завдання №4

Створення текстових ефектів за допомогою стилів

Визначимо для тегу <BODY> такий стиль: шрифт **Arial** чорного кольору розміром 16 пунктів. Внаслідок цього весь текст, що міститься між тегами <BODY> та </BODY>, буде автоматично відформатований відповідно до цих властивостей. Тепер визначимо стилі з іменами **тинь**, **основа**, **шар1** і **шар2**. Текст запишемо в шарах, які накладаються. Спочатку відображатиметься шар **тинь**, а на нього накладатимуться шари **основа**, **шар1** та **шар2**. Порядок накладання задано розташуванням фрагментів тексту в HTML-документі.

Буде застосовано такі властивості:

- **margin-top** — відступ згори (за від'ємних значень можна забезпечити накладання фрагментів);
- **color** — колір;
- **font-size** — розмір шрифту;
- **font-family** — сімейство шрифтів (гарнітура);
- **line-height** — висота рядка.

Визначені в документі стилі мають такі параметри:

- **.тин** — колір зі значенням #DBDBDB (світло-сірий), вирівнювання по центру, відступ згори — 30 пікселів, розмір шрифту — 80 пікселів, висота рядка — 270 пікселів, гарнітура — Times;
- **.основа** — червоний колір шрифту, відступ згори — -230 пікселів, розмір — 70 пікселів, висота рядка — 250 пікселів, гарнітура — Times;
- **.шар1** — чорний колір шрифту, відступ згори — 100 пікселів, розмір шрифту — 50 пікселів, висота рядка 65 пікселів, гарнітура — Arial;
- **.шар2** — зелений колір шрифту, відступ згори — 30 пікселів, розмір шрифту — 35 пікселів, висота рядка — 45 пікселів, гарнітура — Arial.

Для виділення частини HTML-документа використовують тег <DIV>. Він нічого не форматує, а лише відзначає фрагмент тексту, який виступає як окремий об'єкт. Атрибут **CLASS** цього тегу дає змогу посилатися на стилі внутрішньої таблиці й тим самим задавати стиль подання тексту, розташованого між тегами <DIV CLASS...> та </DIV>.

```
<html>
<head>
<title> Приклад 4</title>
<style>
body{color:black;font-size:16px;font-family:'Arial';}
.тин{color:#DBDBDB;text-align:center;margin-top:30px;font-size:80px;line-
height:270px;font-family:'Times';}
.основа{color:red;margin-top:-230px;font-size:70px;line-height:250px;font-
family:'Times';}
.шар1{color:black;margin-top:100px;font-size:50px;line-height:65px;font-
family:'Arial';}
.шар2{color:green;margin-top:30px;font-size:35px;line-height:45px;font-
family:'Arial';}
</style>
```

```

</head>
<body>
Приклад використання каскадних стилів
<center>
<table width=500 cellpadding=0 cellspacing=0 border=0>
<tr>
<td align=center valign=top>
<div class='tin'>Текст з тінню</div>
<div class='osnova'>Текст з тінню</div>
<div class='shar1'>Приклад тексту з тінню</div>
<div class='shar2'>Це приклад використання каскадних стилів</div></td></tr>
</table>
</center>
</body>
</html>

```

Вигляд цього документа після відкриття його у вікні браузера, показано на рис.

нижче

