

## ПРАКТИЧНА РОБОТА №2

**Тема:** Вивчення й практичне застосування стилів CSS.

**Мета:** Познайомитися із стилями CSS, синтаксисом, основними елементами.

**Знати:** Структуру HTML документа, основні підходи стилів CSS.

**Вміти:** Компонувати текстову інформацію при створенні гіпертекстових документів з використанням простих текстових редакторів.

### Теоретичні відомості

Керовані WEB-дизайнером стилі з'явилися в HTML відносно недавно й відразу наблизили HTML до сучасних засобів підготовки публікацій.

Стиль - це можливість, один раз описавши правила форматування ділянки документа, використовувати потім ці правила простим посиланням на назву стилю.

Стилі дозволяють легко вирішити проблему "фірмового" оформлення сайтів. Є можливість не формувати тисячі об'єктів вручну, а просто описати стиль і посилатися на нього там, де потрібно. Тоді Ви зможете б, змінивши форматування стилю в одному єдиному місці, домогтися зміни форматування відповідних об'єктів у всьому сайті.

Насправді стилі як такі були присутні в HTML із самого початку. Наприклад, теги <H1>, <H2>, <BLOCKQUOTE>, і т.д. є ні що інше, як стилі. Однак веб-дизайнер не міг управляти ними. Наприклад, не можливо було вказати, що всі заголовки рівня <H1> повинні бути в 20 пунктів висотою й синім кольором. З CSS така можливість з'явилася.

Крім того, стилі дають цілий ряд нових, раніше не доступних, засобів форматування документа. І, нарешті, стилі - річ динамічна! Це означає, що зовнішній вигляд документа можна змінювати із програм-сценаріїв прямо на льоту (наприклад, у відповідь на те, що користувач завів покажчик миші на картинку або на посилання).

На жаль, існує деякий розбід у реалізації всіх цих можливостей у найбільш популярних браузерях. Фірма Netscape Communications, наприклад, до останнього часу впливала своєму унікальному стандарту JSSS (JavaScript Style Sheets), що не підтримувався ніякими іншими клієнтами. Компанія Microsoft просувала стандарт CSS (Cascading Style Sheets), який підтримувався WWW консорціумом. Нарешті точка зору консорціуму взяла гору й компанію Netscape Communications оголосила про підтримку CSS, починаючи з Netscape Navigator

4.04. Проте реалізація CSS у різних браузерях трохи різна. Якоюсь мірою в цьому може допомогти таблиця сумісності, складена людьми, які просто взяли, і все зрівняли. Однак, кращий спосіб не помилитися, це перевіряти, як виглядають сторінки в різних браузерах перед тим, як "випустити їх у світло".

Отже, перелічимо по пунктах, що ми можемо робити, використовуючи стилі:

- використовувати поля, кеглі й гарнітури шрифтів, індивідуальні кольори тексту й фону для окремих абзаців, слів і навіть букв. Оформляти нові рядки, буквиці та ін.;
- змінювати форматування сторінок і всього сайту в цілому, не доторкаючись до HTML файлів;
- зменшити кількість тегів в HTML тексті й зробити його кращим для сприйняття;
- визначати варіації оформлення за рахунок спадкування класів. Наприклад, визначивши дизайн для тега <P>, можна успадкувати від нього теги <P.exclamation>, <P.question> і т.д., які будуть виглядати як <P>, за винятком тих атрибутів, які явно перевизначені в похідних стилях;
- гнучко управляти розміщенням елементів документа, включаючи накладення їх один на одного, і інші ефекти;
- управляти форматуванням сторінок під час друку HTML документа;
- створювати різні ефекти для тексту й зображень (тіні, димку, і т.д.).

### **Вставка в документ**

Існує три способи вставити стиль в HTML документ.

- посиланням з HTML документ на спеціальний файл, що містить визначення стилів. При такому підході Ви можете посилатися на той самий файл стилів з будь-якої кількості документів. Тоді, при зміні стилів у цьому файлі всі документи поміняють свій вид;
- визначити всі стилі, що використовуються на початку документа, а потім використовувати. При такому підході Ви можете змінювати стилі на початку документа (один раз) і Ваші зміни відіб'ються у всіх місцях документа, де ці стилі використовуються;
- вставляти вказівки на правила форматування безпосередньо в тег. Звичайно це використовується для унікального форматування, що більше (крім як у цьому місці документа) ніде не потрібно.

Звичайно при розробці сайту доводиться використовувати всі три підходи. При цьому, у випадку конфлікту, найвищий пріоритет у властивостей,

вставлених безпосередньо в теги, потім - у блоків стилів, вбудованих у сторінку, і найнижчий - у блоків, на які є посилання.

Як резюме: якщо потрібно однаково відформатувати дві й більше ділянки тексту, то Вам належить визначити стиль. Двічі писати однакове форматування - *дуже дурний тон*.

## **Синтаксис**

Синтаксис визначення стилю дуже простий. Пишеться ім'я стилю, потім у фігурних дужках перераховуються властивості, розділені символом ";". Перед закриваючою фігурною дужкою ";" не ставиться. Властивостей може бути дуже багато, пізніше ми розглянемо деякі з них.

Кілька описів стилів утворюють блок стилів. Наприклад, визначимо блок, що задає червоний колір і кегль в 36 пунктів для заголовків <H1> і голубий колір і кегль в 18 пунктів для заголовків <H2>:

```
H1 {
    font-size: 36pt;
    color: red
}
H2 {
    font-size: 18pt;
    color: blue
}
```

## **Вставка в документ**

Стили можна використовувати в документі трьома способами:

- визначаючи посилання на них,
- вставляючи в початок документа
- вставляючи властивості стилю прямо в теги.

## **Вставка за допомогою посилання**

Для того, щоб зробити в документі посилання на окремий файл стилів, варто записати:

```
<LINK REL=STYLESHEET TYPE="text/css" HREF="URL ">
```

Де URL - адреса файлу з визначенням стилів (стандартне розширення - CSS).

Наприклад,

```
<LINK REL=STYLESHEET TYPE="text/css" HREF="style2.css">
```

у файлі style2.css утримується наведене вище визначення <H1> і <H2>.

### **Вставка безпосередньо в документ**

Для того, щоб вставити блок стилів прямо в документ, потрібно укласти його в

тег <STYLE> ... </STYLE>. Наприклад, це може виглядати так:

```
<STYLE>
  H1 {
    font-size: 36pt;
    color: red
  }
  H2 {
    font-size: 18pt;
    color: blue
  }
</STYLE>
<H1>Заголовок рівня 1</H1>
Звичайний текст<BR>
Звичайний текст<BR>
<H2>Заголовок рівня 2</H2>
Звичайний текст<BR>
Звичайний текст<BR>
<H2>Ще заголовок рівня 2</H2>
Звичайний текст<BR>
Звичайний текст<BR>
<H1>Ще заголовок рівня 1</H1>
Звичайний текст<BR>
Звичайний текст<BR>
<H2>Знов заголовок рівня 2</H2>
Звичайний текст<BR>
Звичайний текст<BR>
```

Результат повинен бути таким же, як і минулого разу.

### **Важливе зауваження:**

Хоча, тут цього не зроблено, існують дуже поважні причини завжди вставляти <STYLE> ... </STYLE> у заголовну частину документа. Тобто між тегами <HEAD> і </HEAD>. Намагайтеся завжди робити саме так.

### **Вставка властивостей у теги**

Властивості стилів можна вставляти прямо у відповідні теги. Розглянемо такий приклад:

```
<P>Цей абзац відформатований за замовчуванням. Цей абзац
відформатований за замовчуванням. Цей абзац відформатований</P>
<P STYLE=" margin-left: 2cm; margin-right: 2cm">
Цей абзац має відступи праворуч і ліворуч по 2 сантиметри.
Цей абзац має відступи праворуч і ліворуч по 2 сантиметри.</P>
<P STYLE=" font-size: 1in; color: red; font-style: italic">
Цей абзац написаний червоними похилими буквами висотою в один
дюйм.</P>
```

Елементи оформлення можна вставляти практично в будь-який тег. Наприклад, у тег `<DIV STYLE="...">` вони будуть ставитися до описуваної секції, у тег `<UL STYLE="...">` - до списку, у тег `<LI STYLE="...">` - до елемента списку і т.д.

Існує новий спеціальний тег `<SPAN>`, він у всім аналогічний `<DIV>` з тої лише різницею, що ніяк не впливає на розбивку документа на рядки. У такий спосіб його можна використовувати для завдання стильових особливостей окремих слів і навіть букв.

Отже, стилі описують форматування та властивості тегів, суттєво розширюючи можливості їх представлення.

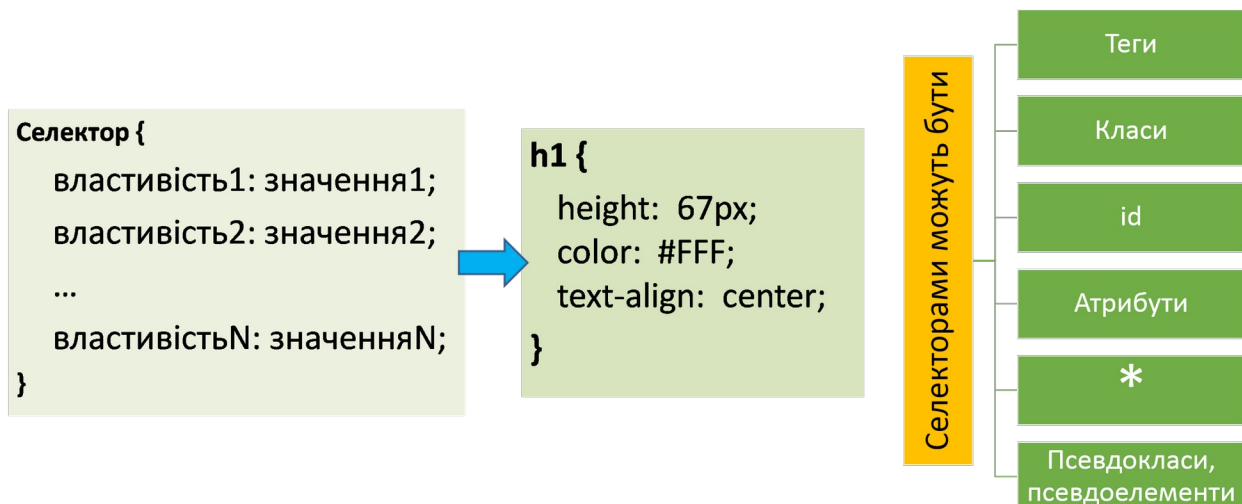


Рисунок 2.1 - Стилi та селектори

Пошук місця застосування визначеного стилю (якщо це не застосування у середині конкретної теги) відбувається за допомогою селектора, який шукає необхідний елемент в потоці документа.

Загальний синтаксис стилю визначають так



Рисунок 2.2 - Синтаксис стилів

Записувати стиль зі всіма властивостями можна в один рядок або в кілька — кожен властивість в окремому рядку для зручності.

## Класи

Приємною властивістю стилів є те, що вони можуть успадковувати властивості один одного. У програмуванні об'єкти, що успадковують властивості "батьків", називаються **класами**.

Давайте визначимо, що всі теги <P> у документі будуть задавати абзаци, написані шрифтом розміром в 20 пунктів блакитного кольору. Потім визначимо, що в нас ще будуть спеціальні абзаци, такі ж, але зрушені ліворуч на пів дюйма. І, нарешті, у нас будуть абзаци ще й третього виду - такі ж, як перші, але написані похилими буквами. Рішення цього завдання може виглядати, наприклад, так:

```
<STYLE>
  P { font-size: 20pt; color: blue}
  P.m { margin-left: 0.5in}
  P.i { font-style: italic}
</STYLE>
<P>Це звичайний абзац.</P>
<P CLASS=m>Це зрушений абзац.</P>
<P CLASS=i>Це похилий абзац.</P>
```

Подивіться, як це виглядає.

*Це звичайний абзац.*

*Це зрушений абзац.*

*Це похилий абзац.*

Рисунок 2.3 - Результат виконання

Зрозуміло, так можна оформляти не тільки спеціальні абзаци, але й спеціальні заголовки, розділи документа й т.д.

Особливий випадок представляють посилання. Взагалі можна визначити стиль для тега <A>, і посилання будуть виглядати так, як ми хочемо. Однак, існують ще й посилання, на яких ми вже побували, і активні посилання. Для них спеціально визначені класи:

A:link - посилання,

A:visited - посилання, на якій уже побували й

A:active - посилання активні в цей момент.

Наприклад, зробимо так, щоб посилання на нашій сторінці були висотою в 20 пунктів, перекреслені. Ті посилання на яких ми не були - зелені. Ті, на яких були - червоні.

Рішення:

```
<STYLE>
  A:link { font-size: 20pt; color: green; text-decoration: line-
through }
  A:visited { font-size: 20pt; color: red; text-decoration:
line-through }
</STYLE>
<A HREF="fictive">Фіктивне посилання, на ньому ми ще не були</A>
<P>
<A HREF="http://elearn.nubip.edu.ua">На цьому посиланні ми вже
були</A>
```

### **Успадкування**

HTML документ має ієрархічну структуру. Тіло документа укладене в теги <BODY>...</BODY>. Усередині документа можуть бути розділи, укладені в теги <DIV>...</DIV>. Усередині розділів можуть бути ще розділи і т.д.

Ідея спадкування полягає в тому, що всі формати, які Ви визначите для <BODY>, будуть успадковуватися всіма розділами. Ви можете перевизначити в розділі деякі формати, а всі інші він успадкує від <BODY>.

Таким же чином, розділ, вкладений в інший розділ, успадковує формати розділу що його охоплює, але може щось перевизначити для себе.

Розглянемо приклад:

```
<DIV STYLE=" font-size: 16pt; color=black;">
```

Це перший розділ. Чорні букви розміром в 16 пунктів.

```
<DIV STYLE=" text-decoration: underline; margin-left: 1cm">
```

Це другий розділ усередині першого. Усе успадковано від першого, але букви підкреслені й всі зсунуто на 1 сантиметр ліворуч від першого розділу.

```
<DIV STYLE="color: blue; margin-left: 1cm">
```

Це третій розділ усередині другого. Усе успадковано від другого, але букви блакитні й всі зсунуто на 1 сантиметр ліворуч від другого розділу.

```
</DIV>
```

Це триває другий розділ (третій закінчився)

```
</DIV>
```

Це триває перший розділ (другий закінчився)

```
</DIV>
```

Результат показано на рис.2.2.

Це перший розділ. Чорні букви розміром в 16 пунктів.

Це другий розділ усередині першого. Усе успадковано від першого, але букви підкреслені й всі сдвинуто на 1 сантиметр ліворуч від першого розділу.

Це третій розділ усередині другого. Усе успадковано від другого, але букви блакитні й всі сдвинуто на 1 сантиметр ліворуч від другого розділу.

Це триває другий розділ (третій закінчився)

Це триває перший розділ (другий закінчився)

Рисунок 2.4 - Результат виконання

Дивіться, вкладені розділи успадковують форматування батьків і можуть додавати своє (а якщо треба, то й змінювати батьківське). Однак, як тільки вкладений розділ завершився й триває батьківський, всі його формати відновлюються.



У цьому й полягає зміст спадкування стильового оформлення в CSS. Наприклад, використання нового тега <SPAN>.

Багато з людей люблять виділяти товстим фломастером місця, що сподобалися їм, у газетах і книгах. При цьому текст на зафарбованій ділянці продовжує проступати, але фон змінює колір. Спробуємо використовувати такий прийом виділення головних думок. Оскільки ми виділяємо не в газеті, скористаємося своєю перевагою й, ще трохи зробимо товстішим шрифт у виділених ділянках. Рішення може бути таким:

```
<STYLE>
  SPAN.NB {background: yellow;font-weight: 700}
</STYLE>
<BR>
За деякими оцінками, через WWW може бути поставлено під загрозу
існування урядів. <SPAN CLASS="NB">З винаходом технологій, які
забезпечать істинну демократію на відміну від наявної
представницької демократії, весь процес управління може суттєво,
якщо і не докорінно, зміниться.</SPAN>
<P ALIGN=RIGHT>Том Армстронг</P>
```

## Властивості

Дотепер ми користувалися властивостями стилів такими як font-size, text-decoration, color і ін. ніяк не визначаючи які взагалі властивості бувають. Насправді їх дуже багато.

Далеко не все працює скрізь, а дещо не працює просто ніде. Проте це - стандарт, і те, що не працює зараз, заробить у нових версіях браузерів.

Тепер, коли Ви розумієте, що таке стилі, як і навіщо їх використовувати, Ви можете просто брати з таблиці властивості й уписувати їх у свої визначення стилів. Особливих проблем (крім несумісності браузерів) виникати не повинно.

Володіння цією технікою дозволить Вам заощадити масу часу й сил при розробці більших сайтів.

Наприклад, використовуючи перераховані властивості можна досягти такого виду як на рис. 2.5

background — колір тла  
font-family — стиль шрифту (гарнітура)  
font-size — розмір шрифту  
font-weight — жирність шрифту  
color — колір шрифту  
text-decoration — оздоблення тексту

text-align — вирівнювання тексту

margin — відступ від рамки об'єкту

padding — відступ від тексту до рамки об'єкту

line-height — висота рядка

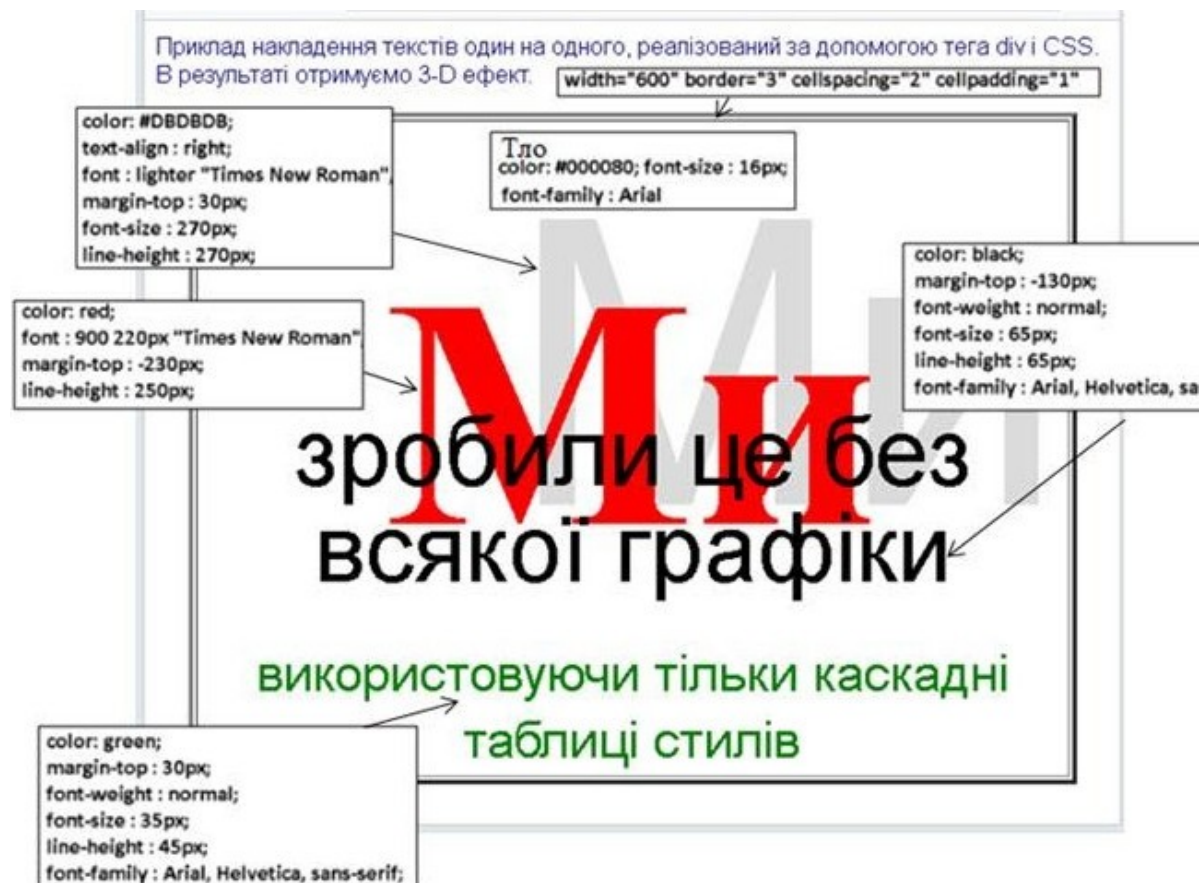


Рисунок 2.5 - Приклад використання стилів

## CSS 3

Все, про що ми тут говорили, належить до базового стандарту CSS. На сьогодні вже існує новий стандарт CSS3. Там додане керування друком, фільтри для різних ефектів, і багато чого іншого, що ми будемо розглядати в наступних темах.

## Завдання

На основі HTML сторінок практичної роботи №1, виконати форматування з врахуванням виконання наступного: додати колір тла сторінки, визначити форматування шрифтів і заголовків сторінки, вказувати розміри елементів на сторінці через стилі, до заголовку сторінки додати тінь (`text-shadow`), вибрати відповідно тематики сторінок використовуючи CSS.

Всі формати стилів робити за допомогою CSS. Всі стилі розмістити в окремому файлі.

**Контрольні питання:**

- 1 Що представляє собою CSS?
- 2 Як використовується CSS?
- 3 У яких випадках який спосіб застосування CSS кращий?
- 4 Який синтаксис визначення стилю?
- 5 В чому полягає ідея спадкування форматів у документі?
- 6 Що означає клас і як його визначити?
7. Для чого використовується id в тегах та стилях?
8. Що таке селектор? Які селектори ви знаєте?
9. Що означає селектор \* (зірочка)?
10. Що таке зовнішня таблиця стилів?
11. З яких елементів складається стиль?
12. Як записують коментар в CSS?