

## Лабораторна робота №3

**Тема:** Каскадні таблиці стилів CSS

**Мета:** Освоїти основні методи використання каскадних таблиць стилів CSS. Навчитися вказувати властивості для елементів гіпертекстової розмітки HTML-документа, створювати свої класи стилів.

### ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Таблиця стилів - це шаблон, який управляє форматуванням тегів HTML у Web-документі.

Таблиця стилів може міститись у самому документі або зберігатись у окремому текстовому файлі із розширенням **css**.

Існує чотири способи зв'язування документу і таблиці стилів:

- ✧ **Зв'язування** - дозволяє використовувати одну таблицю стилів для форматування багатьох сторінок;
- ✧ **Впровадження** - дозволяє задавати всі правила таблиці стилів у самому документі;
- ✧ **Імпортування** - дозволяє вбудовувати у документ таблицю стилів, розміщену на сервері;
- ✧ **Вбудовування** у теги документа - дозволяє змінити форматування конкретних елементів сторінки.

**Спосіб 1.** Зв'язування дозволяє зберігати таблицю стилів у окремому файлі і приєднувати її до документів за допомогою тега `<link>` у розділі `<head>`:

```
<link rel="stylesheet" type="text/css" href="mystyles.css">
```

Параметр `href` тега `<link>` вказує на місце знаходження файлу, що містить таблицю стилів.

**Спосіб 2.** Впровадження: використовується тег `<style>`, параметр якого `type` повинен дорівнювати `"text/css"`. Таблиця стилів обмежується коментарем. (`<!-- -->`).

Це робиться для того, щоб браузер, що не вміє працювати з і стилями, ігнорували їх.

```
<head>
  <style type="text/css">
    b {text-transform: uppercase;
      background-color: grey;
      text-align: center}
  </style>
</head>
```

прикладі визначено таблицю стилів для тегів `<b>`, `<p>`: текст, що міститься у контейнері `<b> </b>` буде записано у верхньому регістрі; текст, що міститься у контейнері `<p> </p>` буде відображатись сірим кольором та центруватись.

**Спосіб 3.** Імпортування: за допомогою тега `<style>` можна імпортувати зовнішню таблицю стилів, визначивши значення властивості `@import`, яке повинно містити URL таблиці стилів: `@import: url (mystyle.css)`

**Спосіб 4.** Вбудовування у тег: використовується для визначення стилю об'єкта, що форматується певним тегом, за допомогою параметру `style`.

```
<h1 style="color: red">Заголовок відображується
червоним кольором </h1>
```

Для зменшення розміру CSS використовується групування – різні селектори записуються через кому.

**Приклад:** `h1,h2,h3: {font-family: Arial};`

Вкладені елементи наслідують правила форматування батьківського елемента.

## Селектори в html

Селекторами називаються імена стилів, в яких вказані параметри форматування. Селектори діляться на кілька типів: селектори тегів, класи та ідентифікатори (ID).

## Селектори тегів

Правило каскадних таблиць складається з двох частин: селектора і визначення. Селектором може бути будь-який тег HTML, для якого визначення задає, яким чином необхідно тег форматувати. Саме визначення теж складається з двох частин: властивості і її значення, розділених знаком двокрапка (:).



Кожне правило починається з селектора (показчика), що вказує на ті html-елементи, до яких ми будемо застосувати css-правило. У блоці оголошень відбувається найцікавіше - ми встановлюємо правила відображення обраних нами елементів, визначаємо їх властивості - розмір, колір, грані, поля, положення на екрані і т.д.

```
h1 {color:blue; font-size: 16pt}
```

прикладі для заголовку першого рівня визначені такі властивості: колір тексту - синій, розмір тексту - 16 пунктів.

Усі розміри можна задавати у таких одиницях:

- **px** - pixels (пікселі);
- **in** - inches (дюйми);
- **em** – висота є множителем на висоту поточного шрифту для даного блоку ,
- **cm** - centimeters (сантиметри);
- **mm** - millimeters (міліметри);
- **pt** - points (пункти; 1pt=1/72 in);
- **pc** - picas (піки; 1pc= 12pt)

## Контекстні селектори

При створенні веб-сторінки часто доводиться вкладати одні теги всередину інших. Щоб стилі для цих тегів використовувалися коректно, допоможуть селектори, які працюють тільки в певному контексті. Наприклад, задати стиль для тегу **<B>** тільки коли він розташовується всередині контейнера **<P>**. Таким чином можна одночасно встановити стиль для окремого тегу, а також для тегу, який знаходиться всередині іншого.

Контекстний селектор складається з простих селектор розділених пробілом. Так, для селектора тегу синтаксис буде наступний.

```
Тег1 Тег2 { ... }
```

Наприклад:

```
P B{ font-family: Times, serif; /* Родина шрифту */  
    font-weight: bold; /* Жирний напис*/  
    color: navy; /* Синій колір тексту */ }
```

цього випадку стиль буде застосовуватися до Тегу2 коли він розміщується всередині Тегу1, як показано нижче.

```
<Тег1> <Тег2> ... </ Тег2> </Тег1>
```

Наприклад:

```
<p><b>Одночасно жирний напис тексту і виділення  
кольором</b></p>
```

## Селектори Class

Класи використовуються, коли необхідно визначити стиль для блоку тексту або задати різні стилі для одного тега. Синтаксис для класів наступний:

```
Тег.Назва класу {Параметр: Значення;}
```

Щоб вказати, що тег використовується з певним стилем, до нього додається параметр **class="Назва класу"**.

Імена класів обираються за бажанням, головне, щоб вони були зрозумілі і відповідали їх використанню. Можна, також, використовувати класи і без вказівки тегу. Синтаксис в цьому випадку буде:

Назва класу {Параметр: Значення;}

При такому записі, клас можна застосовувати до будь-якого тегу.

Наприклад:

```
<style type="text/css">
    h1.red {color: red; }
    h1.bluebgd {color: red; background-color: blue}
</style>
```

тексті документа посилання на відповідний клас задається у параметрі class:

```
<h1 class="red"> Червоний шрифт</h1>
<h1 class="bluebgd">Червоний шрифт на синьому фоні</h1> Якщо
клас повинен використовуватись для всіх елементів документа,
то конкретний тег не вказується:
<style type="text/css">
.red{color red;}
    .bluebgd {color: red; background-color: blue}
</style>
```

Тепер класи red і bluebgd можна застосовувати до будь-яких елементів документа, що мають властивості, визначені у стилі.

### Атрибутні селектори

Іноколи може виникати необхідність прописувати стилі для великої кількості одноманітних блоків, але для того, щоб прописати на всі ці блоки однаковий клас, може піти дуже багато часу. В таких випадках можна скористатися атрибутними селекторами. Синтаксис наступний:

Тег[назва атрибуту = «значення атр.»] {Параметр: Значення;}

Використання атрибутного селектора дуже зручно використовувати у випадках стилізації полів вводу у формах, тому що вони відрізняються між собою значенням атрибуту, а не тегу.

Окрім того даний селектор може приймати альтернативні варіації порівняння, окрім звичайного «=», а саме:

- «~» – одне із значені атрибуту, які перераховані через пробіл, підходить до вказаного;

- «^=» – значення атрибуту починається із вказаного;
- «\$=» – значення атрибуту закінчується вказаним;
- «\*=» – значення атрибуту містить вказане;

## Селектори ID

На відміну від класів ідентифікатори повинні бути унікальні, іншими словами, зустрічатися в коді документа лише один раз. Для цього перед ім'ям записується символ #.

Звернення до ідентифікатора відбувається аналогічно класам, але в якості ключового слова у тега використовується параметр `id`, значенням якого виступає назва ідентифікатора. Символ решітки при цьому вже не вказується.

Синтаксис ID селектора наступний.

`#ID селектор {Параметр: Значення;}`

Приклад використання ідентифікатора ID:

```
<style type="text/css">
  #help { position: absolute; /* Абсолютне позиціонування */
    left: 160px; /* Положення елемента від лівого краю */
    top: 50px; /* Положення від верхнього краю */
    width: 225px; /* Ширина блока */
    height: 180px; /* Висота блока */
    background: #f0f0f0; /* Колір фону */
  }
</style>

<div id="help">Цей елемент допомагає у разі, коли ви знаходитесь
усвідомленні того факту, що абсолютно не розумієте, хто і як вам
може допомогти. Саме в цей момент ми і підказуємо, що допомогти
вам ніхто не зможе </div>
```

Як і при використанні класів, ідентифікатори можна застосовувати до конкретного тегу. Синтаксис при цьому буде наступний.

`Тег#Назва_ідентифікатора {властивість1: значення;  
властивість2: значення; ...}`

Спочатку вказується назва тега, потім без пробілів символ решітки та назва ідентифікатора. У прикладі нижче показано використання ідентифікатора стосовно тегу <P>.

```
<style type="text/css">
  p { color: green; /* Зелений колір тексту */
      font-style: italic; /* Курсивний напис тексту */
  }
  p #sem { color: red; /* Червоний колір тексту */
          border: 1px solid #666; /* Параметри рамки */
          background: #eee; /* Колір фону */
          padding: 5px; /* Поля навколо тексту */
  }
</style>

<p>Звичайний параграф</p>
<p id="sem">Параграф незвичайний</p>
```

До одного документа можна приєднати декілька таблиць стилів, які утворюють "каскад" (тому таблиці стилів називають каскадними). Таблиці стилів мають такий пріоритет (від найменшого до найбільшого): зв'язана таблиця стилів, імпортована таблиця стилів, правило з елементом HTML у якості селектора, правило з параметром CLASS у якості селектора, правило з параметром ID у якості селектора, вбудоване у тег HTML правило.

При роботі з гіперпосиланнями використовують псевдокласи тега <a>:

```
<style type="text/css">
  a:link {color:red} /*невідвіданий зв'язок*/
  a:visited {color:blue} /*відвіданий зв'язок*/
  a:active {color:green} /*активний зв'язок*/ a:hover
  {color:beige} /*зв'язок, на якому розміщено
вказівник миші*/
</style>
```

За допомогою псевдокласу `hover` можна змінити колір гіперпосилання, на якому розміщено вказівник миші.

Деякі властивості, що використовуються у таблицях стилів:

ВЛАСТИВИСТЬ	ПРИЗНАЧЕННЯ	ПРИКЛАДИ
font-family	задає пріоритетний список шрифтів	body {font-family: "Times New Roman", serif}
font-style	стиль шрифту (normal-нормальний,	h1 {font-style: italic}

	italic-курсив, oblique-нахилений)	
font-weight	використовується для визначення товщини шрифту (normal-нормальний, bold-напівтовстий). Використовуються і числові значення з кроком 100, починаючи від 100 і закінчуючи 900.	h1 { font-weight: bold}
font-size	розмір шрифту	p {font-size :10pt}
font	одночасне встановлення властивостей шрифту	p {font: italic 12pt serif}
color	колір тексту	p {color: blue}
background-color	колір фону	body { background-color: blue}
background-image	фонове зображення	body {background-image: url(/image/image 1 .gif)}
text-decoration	тип підкреслювання (overline- верхнє підкреслення, underline- нижнє підкреслення, line-through—закреслений текст.)	h1 {text-decoration: underline}
text-align	вирівнювання тексту {left, right, center)	h2 {text-align: center}
vertical-align	вирівнювання тексту по вертикалі, (top -згори, middle - по середині, bottom - знизу)	h2 { vertical-align: middle}
text-transform	зображення усіх символів тексту у верхньому регістрі (uppercase) або у нижньому регістрі (lowercase) none-відміна використання параметра	h2 { text-transform: uppercase }
margin	значення полів (верхнє,праве,нижнє, ліве).	body { margin: 10 10 20 20}
padding	відступи від елемента	img { padding:5 5 5 7}
border-color	колір границі	img {border-color:blue}
visibility	визначає, чи є елемент видимий у даний момент (visible - видимий, hidden - прихований)	img {visibility: hidden}
list-style-type	тип маркера списку (для нумерованого списку можливі значення: lower-roman - маленькі римські цифри, upper-roman - великі римські цифри, lower-alpha маленькі латинські літери, upper-alpha -великі латинські літери. Для списку з маркерами можливі значення: disc, circle, square.)	ul {list-style-type: disc}
list-style- image	URL-зображення, яке буде використовуватися, як зображення маркер	ol {list-style-image: url (list gif)}
list-style- position	Позиція маркера (inside-у абзаці, outside-вліво від абзацу)	ol { list-style-image: url (list.gif); list-style-position: inside }



## Плавуючі елементи

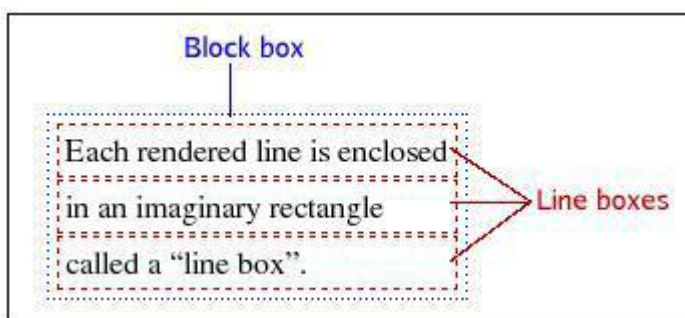
Плаваючий елемент – зміщується до однієї сторони і дозволяє тексту обтікати його з іншого боку. Очищення плаваючого елемента означає зміщення його вниз, щоб перешкодити його появі поряд з плаваючим елементом.

Властивість `float` в CSS було створено з метою реалізувати на Web-сторінках такий стиль компоновання.

CSS є бокси блочного і стрічкового рівнів. За замовчуванням елементи блочного рівня генерують бокси блокового рівня, а рядкові елементи генерують рядкові бокси. Крім боксів, що генеруються елементами, будуть також генеруватися додаткові бокси, наприклад, для текстового контенту документа. Блокові бокси зазвичай розташовуються в тому порядку, в якому елементи з'являються в розмітці, зверху вниз. Блокові бокси не можуть з'являтися поруч пліч-о-пліч, якщо не використовується деякий код CSS.

Стрічкові бокси розташовуються горизонтально. Властивість `direction` визначає, чи будуть вони розташовуватися зліва направо або справа наліво (за замовчуванням використовується зліва направо).

Це називається *поток* документа: рядкові бокси течуть горизонтально в боксах своїх блоків предків, а бокси блоків течуть



вертикально. Бокси з'являються в тому ж порядку, що і елементи в розмітці HTML.

## Як працюють плаваючі елементи?

Властивість `float` має чотири можливих значення: `left`, `right`, `none` й `inherit`. Перші два використовуються значно частіше і будуть змушувати бокс зміститися вліво або вправо.

`float: none` використовується тільки для “скасування” налаштувань в іншому правилі.

`float: inherit` використовується дуже рідко. Воно змушує елемент успадковувати значення `float` від елемента предка.

Для `float: left` бокс зміщується вліво, поки ліве поле плаваючого елементу не торкнеться лівого краю предка. Плаваючі елементи не накладаються один на одного.

### Теги Span і Div

Теги SPAN і DIV широко використовуються разом із селекторами класів. SPAN призначений для зміни стилю виділеної частини тексту, такої як окремий символ, слово або речення, а DIV дозволяє застосовувати стиль відразу до цілого блоку. Перед цим тегом, на відміну від SPAN, автоматично відбувається перехід на новий рядок, аналогічно дії тегу параграфа P.

Приклад використання тега SPAN:

```
<html>
<head>
<style type="text/css">
.capital { font-size: 150%; color: red; }
</style>
</head>

<body>
<span class=capital>Б</span>уквиця є художнім прийомом
оформлення тексту і являє собою збільшену першу літеру, базова
лінія якої нижче на один або кілька рядків базового рядка
основного тексту. Буквиця привертає увагу читача до початку
тексту, особливо, якщо сторінка позбавлена інших яскравих
деталей.
</body>
</html>
```

Тег SPAN зручний для створення виділень в тексті, буквиць, цитат та ін. За допомогою контейнера `<span>` можна перевизначити стиль фрагмента тексту всередині іншого стандартного контейнера:

```
<html>
  <head>
    <title> Приклад використання контейнера span
    </title> <style type="text/css">
      p { color:red; font-size: 24pt; font-family:
times;}
      span {color:yellow; font-size: 18pt; font-
family: courier}
    </style>
  </head>
  <body>
    <p> Текст червоного кольору
    <span>Текст жовтого кольору</span> Знову текст
червоного кольору
    </p>
  </body>
</html>
```

## ЗАВДАННЯ ДЛЯ ВИКОНАННЯ

Для створеного проекту в лабораторній роботі №2:

1. Створити файл стилів `mystyle.css`. Розмістити його в кореневій папці проекту, біля `.html` файлів.
2. Підключити файл `mystyle.scss` до всіх файлів проектів, які розроблялися на попередній лабораторній роботі.
3. Для файлу `index.html` виконати наступне:
  - 3.1. Заголовок `<h1>` форматувати таким чином:
    - 3.1.1. поставити будь-який моноширинний шрифт (наприклад Courier New);
    - 3.1.2. товщину шрифту встановити в 500;
    - 3.1.3. зробити текст зеленого кольору;
    - 3.1.4. розмістити текст по центру екрану;
  - 3.2. Для першого сегменту з монологами Гамлету:
    - 3.2.1. поставити колір заднього фону на обидва абзаца;
    - 3.2.2. монолог англійською мовою зробити жирним і курсивом, колір замінити на голубий;
    - 3.2.3. монолог українською зробити підкресленим з товщиною в 200, колір встановити на рожевий;
    - 3.2.4. розмір шрифту поставити в 20 px;
    - 3.2.5. інтервал між стрічками встановити в 150%.
  - 3.3. Для другого сегменту з монологами Гамлету:
    - 3.3.1. прибрати кольори для кожної стрічки, зробити все чорним;
    - 3.3.2. зробити всі виділені жирним кольори на розміром у 18px і червоним кольором;
    - 3.3.3. всі курсивні виділення зробити шрифтом розміром 12px, великими літерами і фіолетового кольору;
    - 3.3.4. розмістити зображення праворуч від тексту.
  - 3.4. Для елементів списку:
    - 3.4.1. для кожного тегу списку (`<ul>`, `<ol>`) поставити границі, переривчаста лінія зеленого кольору;
    - 3.4.2. для впорядкованого списку `<ol>` поставити римський стиль нумерації;

3.4.3. для кожного неупорядкованого списку `<ul>` встановити свій унікальний маркер;

3.4.4. для одного із довільно обраного неупорядкованого списку як маркер поставити емодзі.

4. Для файлу `forms.html` виконати наступне:

4.1. Додати заливку заднього фону на форму.

4.2. Для всіх тегів `<fieldset>` прописати границю чорного кольору, з подвійної лінії, шириною `6px` і заокругленням `5px`.

4.3. Для всіх тегів `<legend>` прописати заливку чорного кольору, білий шрифт і жирний шрифт.

4.4. Для всіх полів вводу прописати зелену суцільну границю, з товщиною `2px`, прибрати заокруглення, додати внутрішній відступ по `10px` і шрифт для полів встановити моноширинний шрифт (наприклад `Courier New`).

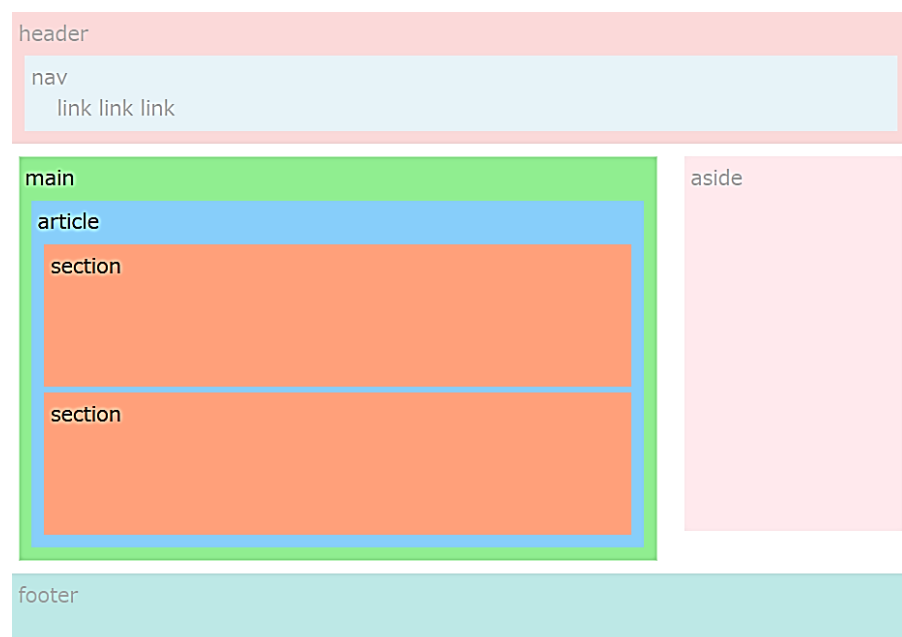
4.5. Для всіх полів вводу типу `text` поставити колір тексту в червоний.

4.6. Підписи для полів вводу зробити розміром `18px` і з курсивним виділенням.

4.7. Кнопку відправки форми стилізувати за власним смаком.

5. Для файлу `blocks.html` виконати наступне:

5.1. Використовуючи CSS властивість `float` спробувати повторити зображення:



## КОНТРОЛЬНІ ПИТАННЯ

1. Що називають таблицею стилів?
2. Чому таблиці стилів називають каскадними?
3. Як записується правило CSS?
4. Назвіть способи зв'язування CSS із документом.
5. За яким пріоритетом використовуються CSS у документі?
6. Яке призначення селектору class?
7. Наведіть приклади використання параметру id.
8. Наведіть приклади використання атрибутного селектору.
9. У яких випадках потрібно використовувати контейнер `<span>`?
10. Які ви знаєте псевдокласи тега `<a>` і для чого вони використовуються?
11. Які властивості у CSS використовуються для визначення вигляду шрифту?
12. Які властивості у CSS використовуються для роботи з кольором?
13. Які властивості у CSS використовуються для встановлення значень полів?
14. Назвіть властивості, що використовуються у CSS для форматування списків.
15. Назвіть властивості, що використовуються у CSS для форматування границь.
16. Навіщо використовувати властивість float?
17. Як розмістити плаваючий блок по центру?