

Методичні вказівки

«Розробка web-сайту. Робота з каскадними таблицями стилів (CSS)»

Мета - вивчення можливостей каскадних таблиць для форматування і оформлення зовнішнього вигляду html-сторінки.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Стиль - це набір параметрів, що задає зовнішнє уявлення деякого об'єкта в тому чи іншому середовищі.

CSS (від Cascading Style Sheets) значно полегшують створення веб-сайтів. CSS працює зі шрифтами, полями, таблицями, відступами, картинками та ін. І представляє значно ширші можливості, ніж простий html.

Синтаксис і принцип роботи CSS

CSS має строго певний синтаксис, тобто правила за якими створюються таблиці стилів. Основною структурною одиницею тут є правило, яке визначає, як буде виглядати той чи інший елемент в документі.

Розглянемо структуру правила:



Рисунок 1 – Визначення стилю

Як видно з рисунка вище, спочатку записується так званий селектор, який показує, до якого html-тегу застосовується та чи інша властивість.

Далі, безпосередньо за селектором, пишеться блок оголошення стилів, який обов'язково полягає в фігурні дужки.

Кожне оголошення в свою чергу складається з *властивості* і його *значення*. Після властивості ставиться двокрапка. Правило може містити в собі кілька оголошень. В такому випадку вони повинні бути відокремлені один від одного крапкою з комою, причому після останнього оголошення крапку з комою можна не ставити.

Показане вище правило вказує на те, що всі заголовки першого рівня в документі будуть блакитного кольору з розміром шрифту 14 пікселів.

Селектор служить для того, щоб в HTML-документі однозначно визначити той елемент, до якого слід застосувати дане CSS-правило.

Селектор по елементу

Як селектор може використовуватися ім'я елемента HTML. Наприклад, для відображення тексту всіх абзаців в HTML-документі було написано шрифтом *Arial*, звичайним шрифтом і розміром 14 пунктів, то слід застосувати таке правило:

```
P {font: normal 14pt Arial;}
```

Як селектор тут обраний елемент P, що позначає в мові HTML окремий абзац.

Селектор по класу

Якщо ж буде потрібно виділити абзаци двох типів: перший, аналогічний описаному вище, а другий з текстом, колір якого вже повинен бути не чорним, а синім. Можна вийти з положення декількома шляхами. Перший - це вказати колір тексту в атрибуті *style* елемента *p*.

```
<p style="color:blue">Колір цього абзаца буде синім</p>
```

Однак мова йде про ситуацію, коли на сторінці повинен існувати не один абзац такого виду, а цілий клас, тому описаний вище спосіб не дуже підходить. По-перше, для того щоб все абзаци класу стали синіми, потрібно в кожному прописати атрибут *style* = "color: blue". По-друге, якщо пізніше прийдеться поміняти колір виділених абзаців, доведеться шукати кожне входження в код записи *style* = "color: blue" і виправляти в ній колір. А якщо, окрім кольору, теба збільшити ще і шрифт, то будуть потрібні ще більш серйозні зміни в коді.

Другий варіант вирішення цього завдання - використання селектора по класу. Для того, щоб визначити клас, в HTML-коді використовується атрибут *class*, який можна застосовувати в будь-якому елементі. У HTML-коді ми повинні написати наступне:

```
<p class="blue"> Колір всіх абзаців буде синім </p>
```

У таблиці стилів при цьому слід написати таке правило, де в якості селектора буде використовуватися назва елемента *p* і через точку (.) від нього ім'я класу:

```
<style type="text/css">
p.blue {
    color: blue;
}
</style>
```

Такий синтаксис нагадує синтаксис об'єктно-орієнтованих мов програмування, де назви властивостей і методів об'єкта відокремлюються від імені об'єкта точкою. Якщо знадобиться виділити ще один клас абзаців, наприклад червоного кольору, можна створити ще один клас *red*. Тоді потрібно написати такий код:

```
<style type="text/css">
p.red {
    color: red;
}
</style>
...
<p class="red">Червоний колір</p>
```

Можна створити **уніфікований клас**, який можна буде використовувати в будь-яких елементах HTML. Наприклад, створимо клас червоних об'єктів. Для цього в таблиці стилів потрібно написати таке правило:

```
<style type="text/css">
.red {
    color: red;
}
</style>
```

Перед ім'ям класу можна не вказувати назву елемента. Далі можна написати такий код.

```
<h1 class="red">Червоний  
заголовок</ h1>  
<p class="red">Червоний текст</p>
```

Селектор по ідентифікатору

Можлива інша ситуація, коли в HTML-документі повинен бути присутнім унікальний елемент, стиль оформлення якого відрізняється від всіх інших на цій сторінці. Тоді для цього елемента за допомогою атрибута *id* потрібно визначити його власний унікальний ідентифікатор, який також може застосовуватися в будь-якому елементі HTML. Існує вимога, за яким в HTML-документі може бути тільки один елемент із заданим *id*. Тому, якщо на кожній сторінці вашого сайту найперший заголовок повинен відрізнятися від всіх інших, то його слід оформити з використанням ідентифікатора:

```
<h1 id="firstheader">первый заголовок на странице</  
h1>
```

Щоб записати CSS-правило для визначення стилю цього заголовка, використовується такий же прийом, як і для селектор по класу, але як роздільник використовується не крапка, а символ решітки (#).

```
h1#firstheader {  
color: red;  
font-weight: bold;  
text-align: center  
}
```

За аналогією з уніфікованим селектором по класу можна створити **уніфікований селектор** за ідентифікатором. Наведемо приклад:

```
#name {  
background: wheat;  
padding: 5px;  
}
```

Псевдокласи

Крім псевдоелементів, в CSS існують ще й псевдокласи, які ґрунтуються на інформації, яка не є частиною HTML-документа. Є псевдокласи, що позначають відвідані і не відвідані, а також активні посилання. Однак в коді документа ви не зможете визначити, яка з посилань є відвідується, яка в даний момент активна і т.д. Цією інформацією володіє безпосередньо браузер.

В CSS існує чотири псевдокласи, що дозволяють працювати з посиланнями:

link - позначає просте посилання;

active - позначає активне посилання;

visited - позначає відвідану посилання;

hover - позначає посилання, на яку наведений курсор миші.

Таким чином, щоб за допомогою CSS задати зовнішній вигляд посилань, можна написати такі правила:

```
a:link {стиль оформления }
```

```
a:active {стиль оформлення }  
a:visited {стиль оформлення }  
a:hover {стиль оформлення }
```

Псевдокласи можна використовувати в комбінації зі звичайними класами і ідентифікаторами.

```
<a class="pagelink" href="page.html">текст посилання</a>  
a.pagelink:hover {  
    ... стиль ...  
}  
  
або:  
  
<a id="mylink" href="page.html">текст посилання</a>  
a#mylink:visited {  
    ... стиль ...  
}
```

Групування селекторів

Якщо блоки описів для декількох селекторів повністю збігаються, то ці селектори можна згрупувати. Угрупування - це об'єднання селектор з однаковими оголошеннями з метою зменшення загального обсягу коду. При угрупуванні селектори повинні розділятися комою. Наприклад, якщо ви хочете для всіх заголовків (з першого по шостий рівні) в документі змінити колір, то можете написати таке правило CSS:

```
h1,h2,h3,h4, h5, h6 {  
    color: blue  
}
```

Підключення CSS-таблиць до HTML-документа

CSS розроблений з метою маніпулювання зовнішнім виглядом елементів web-сторінки. Каскадні таблиці стилів включаються в HTML-документ чотирма способами:

1. Вбудовуються в теги.
2. Розміщуються в секторі <head> </head>.
3. У секторі <head> </head> прописується посилання на зовнішній .css файл.
4. Імпорт опису стилю в документі

Пряме включення CSS в HTML теги за допомогою style:

Приклад:

```
<p style="color:red; font-weight:bold">Текст червоного кольору,  
напівжирний</p>
```

Розміщення CSS в секторі <head> </head>

Приклад:

```
<html>
```

```
<head>
  <title> CSS приклад </title>
  <style type="text/css">
    p {color:#006633}
    .forexample {color:Yellow}
    #ident {color:#ffffff; font-weight:bold}
body {background-color:#66cc66}
  </style>
</head>
<body>
  <p>Текст параграфов документа темно-зеленого кольору</p>
  <p>Текст</p>
  <p class="forexample">Текст жовтий</p>
  <p id="ident">Текст білого кольору, напівжирний</p>
  <h2 class="forexample">Заголовок</h2>
  <p>Текст</p>
</body>
</html>
```

Посилання на зовнішній файл з розширенням .css

Приклад, **file.css** знаходиться в поточній папці.

```
@charset "windows-1251";
/* CSS Document */
body {background-color:#0000cc}
p {color:#222255}
.forexample {color:orange}
#ident {color:#ffffff; font-weight:bold}
```

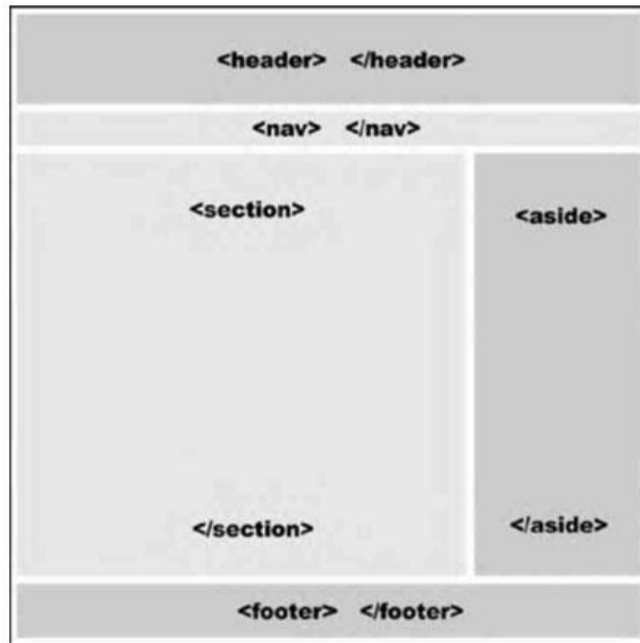
Файл **file.html**

```
<html>
<head>
<title> </title>
<link rel="stylesheet" type="text/css" href="file.css" />
</head>
<body>
<p>Текст параграфов документа темно-синього кольору</p>
<p>Текст</p>
<p class="forexample">Текст помаранчевий</p>
<p id="ident">Текст білого кольору, напівжирний</p>
<h2 class="forexample">Заголовок</h2>
<p>Текст</p>
</body>
</html>
```

ЗАВДАННЯ ДО ВИКОНАННЯ РОБОТИ

1. Для роботи використовувати html-сторінки, які були створені вами в попередніх трьох лабораторних роботах. Попередньо видаліть з них теги і атрибути для форматування зовнішнього вигляду тексту. Використовуйте тільки каскадні таблиці стилів.

Змініть структуру html-документів на блокову з використанням елементів наступним чином



2. Для сторінки **text.html**
 - a. Задайте наступні параметри:
 - колір фону
 - розмір шрифту
 - колір шрифту
 - сімейство шрифту (наприклад, Arial)
 - b. Використовуйте пряме включення CSS в HTML теги за допомогою властивості style
 - c. Використовуйте тип селектора по елементу
3. Для сторінки **list.html**
 - a. Задайте параметри:
 - b. для нумерованого списку - тип маркера маленькі латинські літери
 - c. для маркованого списку - тип маркера диск
 - d. використовуючи блоковий елемент div, для кожного виду списку задати свій колір фону
 - e. Використовуючи елементи для роботи з таблицями CSS, задайте наступні параметри:
 - рамку навколо таблиці і комірок
 - для заголовків жовтий фон комірок, для інших - сірий
 - f. Використовуйте розміщення CSS в секторі <head> </head>
 - g. Використовуйте тип селектора по ідентифікатору
4. Для сторінки **image.html**
 - a. . Задайте власні стилі
 - b. Використовуйте розміщення CSS в секторі <head> </ head>
 - c. Використовуйте тип селектора по класу і ідентифікатору