

## Змістовий модуль 4. Основи веб-програмування мовою JavaScript (+jQuery)

### Тема 10: Об'єктна модель документа

Веб-документ, завантажений у вікно браузера, розпізнається цим браузером у вигляді деревоподібної структури:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Вивчаємо JavaScript</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>DOM - об'єктна модель документа</h1>
```

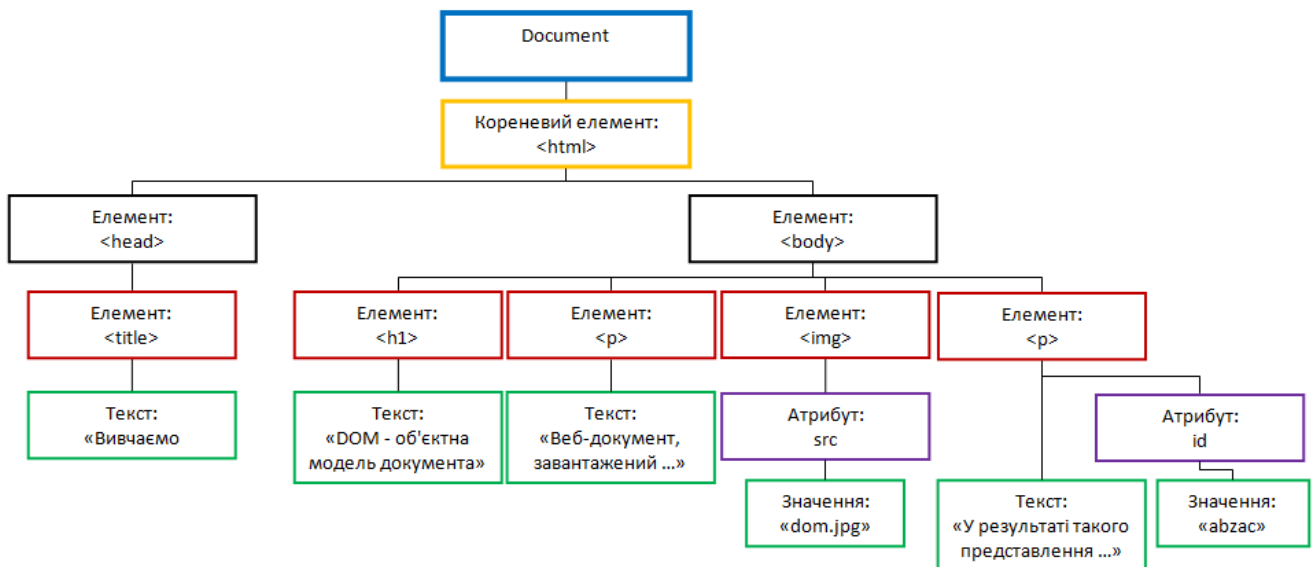
```
    <p>Веб-документ, завантажений у вікно браузера, розпізнається цим браузером у вигляді деревоподібної структури.</p>
```

```
    
```

```
    <p id="abzac">У результаті такого представлення є змога звертатись до окремих елементів сторінки, змінюючи їхні атрибути, значення, додаючи певні елементи тощо.</p>
```

```
  </body>
```

```
</html>
```



У результаті такого представлення є змога звертатись до окремих елементів сторінки, змінюючи їхні атрибути, значення, додаючи певні елементи тощо. Така обробка відбувається мовою програмування JavaScript.

## getElementsByTagName та getElementById

Для звертання до окремих об'єктів документа потрібно їх ідентифікувати у структурі, наприклад за тегом `document.getElementsByTagName("p")` чи іменем `document.getElementById("abzac")`. У першому випадку будуть вибрані усі абзаци документа, виділені тегом `p` (утворюючи масив таких об'єктів), а в другому - лише об'єкт, названий `abzac`.

## createElement та appendChild (+innerHTML та style)

Можна також створювати об'єкти, розміщуючи їх у певній ієрархічній позиції. Так, команди

```
var new_header=document.createElement("h2");
```

```
document.body.appendChild(new_header);
```

створюють новий заголовок другого рівня та додають його в кінець розділу `body` поточного документа. Зверніть увагу, що створений об'єкт зберігається у змінній `new_header`, що дозволяє змінювати його властивості. Так, `new_header.innerHTML="Текст заголовка"`; дозволяє вказати значення тексту заголовка другого рівня; також можна встановити певний стильовий атрибут у задане значення: наприклад `new_header.style.color = "green"`; (зелений колір шрифту).

Код, написаний мовою JavaScript можна розмістити як на самій сторінці (у розділі заголовка, чи в основному вмісті), або в окремому файлі, наприклад `script.js`, підключивши його до сторінки тегом `<script src="script.js" type="text/javascript" ></script>`. Потрібно враховувати, що якщо код повинен обробляти елементи, що є на сторінці, ці елементи повинні бути вже завантажені у той час, коли браузер виконуватиме скрипт JavaScript. Це означає, що код слід розмістити в самому кінці розділу основного вмісту сторінки, або прив'язати його до виконання події завантаження вмісту.

Для експериментування з кодом JavaScript рекомендуємо скористатись онлайн-сервісом <https://jsbin.com/?html.js,output> на окремих вкладках якого можна писати HTML та JavaScript коди.

## Приклад 1. Колір

Наприклад, додамо наступний код JavaScript в кінець розділу основного вмісту сторінки:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Вивчаємо JavaScript</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>DOM - об'єктна модель документа</h1>
```

```
    <p>Веб-документ, завантажений у вікно браузера, розпізнається цим браузером у вигляді деревоподібної структури.</p>
```

```
    
```

```
    <p id="abzac">У результаті такого представлення є змога звертатись до окремих елементів сторінки, змінюючи їхні атрибути, значення, додаючи певні елементи тощо.</p>
```

```
    <script type="text/javascript">
```

```
      document.getElementById('abzac').style.color = "red";
```

```
</script>
```

```
</body>
```

```
</html>
```

В результаті другий абзац документа буде оформлено червоним кольором.

Розглянемо код скрипту за допомогою ресурсу [jsfiddle.net: https://jsfiddle.net/vleonteva/npouyLzd/](https://jsfiddle.net/vleonteva/npouyLzd/) (цей код можна змінювати та експериментувати). До початкового документа додано заголовок другого рівня та зображення.

## Приклад 2. Список

Код скрипту може бути дуже різноманітний, залежно від того, що потрібно додати до сторінки. Так, наприклад, створимо список із семи елементів, розфарбовуючи їх у кольори райдуги.

Для початку додаємо до тіла документа об'єкт нумерованого списку **new\_list**. Далі у циклі, котрий виконається 7 разів, створюємо елементи **li** і додаємо їх не до загального документа, а всередину списку **new\_list**.

Створюємо масив кольорів:

```
var colors = ["red", "orange", "yellow", "green", "blue", "navy", "purple"];
```

А також масив усіх елементів, що оформлені тегом **li**:

```
var x = document.getElementsByTagName("li");
```

Для кожного з об'єктів списку **x**, встановлюємо колір та змінюємо напис, додаючи до нього назву кольору:

```
for (var i = 0; i < x.length; i++) {  
    x[i].style.color = colors[i];  
    x[i].innerHTML = x[i].innerHTML + " = " + colors[i];  
}
```

Результат можна переглянути у проекті: <https://jsfiddle.net/vleonteva/d5cefoxk/>