**Лабораторна робота №4**

**Методи**

Під час інтерпретації HTML-документа броузером створюються об’єкти JS. Властивості об’єктів визначаються параметрами тегів HTML. Структура документа відображається в ієрархічній структурі об’єктів, що відповідають HTML-тегам. Предком є об’єкт windows, розташований на найвищому рівні ієрархії, він працює з вікном броузера і створюється при запуску броузера. Щоб відкрити нове вікно в сценарії JS і показати в ньому новий документ, застосовується метод open, для закриття вікна є метод close. Метод alert об’єкта windows показує діалогове вікно з текстом в якості параметра. Властивості об’єкта windows стосуються всього вікна, в якому показується документ.

Підлеглими об’єктами (або об’єктами нижчого рівня) є об’єкти document, history, location, frame. Властивостями об’єкта history є адреси раніше завантажених HTML-сторінок. Властивості об’єкта location пов’язані з URL-адресою документа, об’єкта frame - із спеціальним способом подання даних.

Властивості об’єкта document визначаються вмістом самого документа: шрифт, колір фону, форми, зображення тощо. Об’єкт document залежно від свого вмісту може мати підлеглі (дочірні) об’єкти. Зокрема підлеглими для об’єкта document є об’єкти form, image, link, area та ін.

Для кожної сторінки створюється один об’єкт document. Окремі його властивості відповідають атрибутам тега <BODY>: bgColor, fgcolor, linkcolor, alinkcolor, vlinkcolor. Методи write і writeln записують в документ текст, заданий параметром.

Якщо документ містить зображення, то доступ до об’єкта, що визначає зображення, можна отримати з допомогою змінної, вказаної в атрибуті name тега <Img>. Об’єкт image має властивість images, яка містить посилання на всі зображення, розташовані в документі. Посилання нумеровані з нуля. Доступ до першого зображення можна отримати з допомогою складної конструкції document.images[0], до другого - document.images[1] і т.д.

Розглянемо приклади, в яких використовуються різні властивості об’єктов.

Для вбудовування зображень в HTML-документ є тег <IMG> із обов’язковим атрибутом src для задання URL-адреси файла із зображенням. Можна задавати розміри показуваного зображення. Значення атрибутів width та height задають ширину та висоту зображення. Ці значення можуть не співпадати з дійсними розмірами зображень, тоді при завантаженні зображення автоматично виконується масштабування.

Зображення може отримати рамку – з допомогою атрибута border. Значенням атрибута має бути число, що задає товщину рамки в пікселях. Стандартно рамка відсутня, якщо зображення не є посиланням.

Атрибут alt задає альтернативний текст - при наведенні курсора на зображення появляється коментар із даного атрибута.

Приклад 1. Переставляння зображень

Напишемо сценарій обміну картинок в документі. Нехай в документі є три зображення, пронумеровані від 1 до 3. В текстових полях вказуються номери зображень, які необхідно поміняти місцями. Натискання кнопки «Переставити» зображення перемістились на потрібні місця.

Спершу перевіримо, чи правильно задані номери зображень, і якщо це не так, то покажемо повідомлення. Змінна z - для запам’ятовування адреси першого зображення. Доступ до зображення з номером r1 відбувається з допомогою конструкції document.images[r1-1]. Щоб на місце зображення з номером r1 помістити зображення з номером r2, требая виконати оператор присвоєння:

document.images[r1-1].src=document.images[r2-1].src

І, нарешті, на місце зображення з номером r2 ставимо зображення, яке раніше було на місці з номером r1, і адреса якого запам’ятали в змінній z:

document.images[r2-1].src=z

Повний сценарій в лістингу 1.

Лістинг 1. Переставляння зображень

<HTML>

<HEAD>

<TITLE>Переставляння зображень</TITLE>

<script>

function chan(obj)

{ var r1=Number(obj.a1.value)

var r2=Number(obj.a2.value)

if ((r1<1)||(r1>3)||(r2<1)||(r2>3))

alert ("Неправильно задані номери зображень!")

else

{ var z=document.images[r1-1].src

document.images[r1-1].src=document.images[r2-1].src;

document.images[r2-1].src=z

} }

</script>

</HEAD>

<BODY>

<CENTER>

<Р>Галерея рисунков</Р><br>

<IMG src="p1.gif" width="90" name="pic1">

<IMG src="p2.gif" width="90" name="pic2">

<IMG src="p3.gif" width="90" name="pic3"> <br>

<FORM name=form1>

Зображення з номерами<br>

<input type="text" name="a1" size=1> и

<input type="text" name="a2" size=1><P>

<input type="button" value=" Переставити" onClick="chan(form1)">

</FORM>

</CENTER>

</BODY>

</HTML>

Приклад 2. Просте вертикальне меню

Напишемо сценарій вертикального графічного меню. При наведенні курсора на пункт меню змінюється палітра виділеного пункта меню.

Кожному пункту меню відповідає два зображення: перше зображення, коли пункт меню не обраний, друге - при обраному пункті меню, колірна палітра малюнка змінена. Графічні зображення, відповідні ситуації, коли пункти меню не вибрані, зберігаються в файлах з іменами pch1.gif, pch2.gif. Відповідні їм графічні зображення із зміненою палітрою зберігаються в файлах з іменами wpch1.gif, wpch2.gif.

Функція img має два параметра. Перший параметр задає вибір пункта меню, другий параметр - n - визначає номер пункта меню. Від цього параметра залежить, яке зображення в документі треба змінити document.images[n-1].src (вставити на цьому місці зображення "wpch"+n+".gif" або pch"+n+".gif). Назва файла формується динамічно конкатенацією рядків, одна із складових якої - значення другого параметра. Якщо імена файлів не підлягають загальному правилу, то в функції буде потрібний додатковий аналіз, який файл підтягнути. Це зробити неважко, якщо знати місце в документі, з якого прийшов виклик функції. Документ із сценарієм вертикального графічного меню в лістингу 2.

Лістинг 2. Просте вертикальне меню

<HTML>

<HEAD>

<TITLE>Просте вертикальне меню</TITLE>

<script language="JavaScript">

function img(n, action)

{if (action)

{document.images[n-1].src="wpch"+n+".gif"}

else

{document.images[n-1].src="pch"+n+".gif"}

}

</script>

</HEAD>

<BODY background="fon1.jpg">

<H2><FONT соlоr="#0000ff">Зміст</FONT></H2>

<A href="pg1.htm" onmouseover="img(1,1)" onmouseout="img(1,0)">

<IMG src="pch1.gif" аlt="пункт 1" border="0" width="200" height="50"></A><br>

<A href="pg2.htm" onmouseover="img(2,1)" onmouseout="img(2,0)">

<IMG src="pch2.gif" аlt=" пункт 1" border="0" width="200" height="50"></A><br>

<A href="pg3.htm" onmouseover="img(3,1)" onmouseout="img(3,0)">

<IMG src="pch3.gif" аlt=" пункт 1" border="0" width="200" height="50"></A><br>

</BODY>

</HTML>

При попаданні курсора в область зображення виникає подія mouseover, атрибут обробки події onMouseOver отримує значення img(p1,true).

Завдання

1. Перевірити приклади лабораторної роботи.

2. Написати сценарій вибору із трьох зображень одного, яке вставляється нижче цих трьох.

3. Написати сценарій картинки з "ефектом наближення", тобто збільшення розмірів як реакція на попадання курсора в поле зображення (використовувати властивості width і height).

4. Написати сценарій графічного горизонтального меню з появою стрілки над пунктом, на який наведено курсор.