# ЛАБОРАТОРНА РОБОТА 10

**Тема.** Створення інтерактивних Web-документів

**Мета:** освоєння практики створення інтерактивних Webдокументів засобами JavaScript та CSS.

### Хід роботи

1. У папці HTML створити файл anim.html та записати у ньому такий код («кістяк» HTML-сторінки):

<html>

<head>

<meta http-equiv=«Content-Type» content=«text/html; charset=windows-1251»>

<title>Створення інтерактивних Web-документів</title>

<script>

</script>

</head>

<body>

</body>

</html>

1. Створимо скрипт для того, щоб ця Web-сторінка завжди відкривалась у повновіконному режимі. Для цього:
   1. Визначаємо вертикальний та горизонтальний розмір екрана:

...

h=window.screen.height; w=window.screen.width;

</script>

* 1. Визначаємо функцію, яка буде переміщати вікно з нашою Web-сторінкою у лівий верхній кут екрана та встановить розміри вікна, рівні розмірам екрана:

...

function wr() { window.moveTo(0,0); window.resizeTo(w,h);

}

</script>

* 1. Модифікуємо тег <body> для звернення до функції wr() при завантаженні документа у вікно браузера:

<body onload=«wr()»>

1. Створимо на нашій Web-сторінці вертикальне меню із двох елементів (кнопок). Вибір користувачем певного елемента меню повинен приводити до показу на сторінці відповідної інформації. Для цього:
   1. Додамо на нашу Web-сторінку три таблиці. У першій таблиці розмістимо два управляючі елементи меню (кнопки), у другій таблиці запишемо зміст першого розділу меню, а у третій – зміст другого розділу. Другій таблиці присвоїмо *id=«r1»*, а третій – *id=«r2»*:

...

<table align=«left»>

<tr><td><input type=«button» value=«Перший пункт меню» style=«width: 200px;»></td></tr>

0<tr><td><input type=«button» value=«Другий пункт меню» style=«width: 200px;»></td></tr>

</table>

<table id=«r1»>

<tr><td>Зміст першого розділу меню (Таблиця 2)</td></tr>

</table>

<table id=«r2» style=«display: none;»>

<tr><td>Зміст другого розділу меню (Таблиця 3)</td></tr>

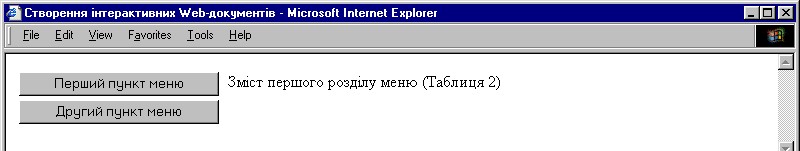
</table>

</body>

3.2. Визначимо, що при завантаженні Web-сторінки відображатись буде тільки зміст першого розділу меню, тобто таблиця 3 буде невидимою. Для цього модифікуємо код третьої таблиці:

*<table id=«r2» style=«display: none;»>*

3.3. При перегляді нашої Web-сторінки зміст другого пункту меню (таблиця 3) повинен бути невидимим (рис. 10.1).



**Рис. 10.1. Відображення першого розділу меню при завантаженні Web-сторінки**

3.4. Додамо функцію, яка буде відображати на екрані тільки той елемент HTML-сторінки, id якого передається цій функції як параметр:

...

function disp(myid) {

document.getElementById(‘r1’).style.display=«none»; document.getElementById(‘r2’).style.display=«none»; document.getElementById(myid).style.display=«block»;

}

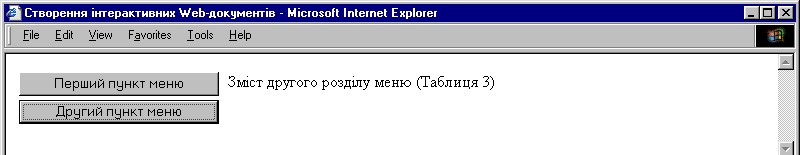
</script>

3.5. Модифікуємо код кнопок так, щоб вибір користувачем певної кнопки приводив до відображення тільки відповідного розділу меню:

<input type=«button» value=«Перший пункт меню» style=«width: 200px;» onClick=«disp(‘r1’)»>

<input type=«button» value=«Другий пункт меню» style=«width: 200px;» onClick=«disp(‘r2’)»>

3.6. Перевіримо функціонування меню. При виборі кнопки меню повинен відображатись тільки відповідний розділ (рис. 10.2).



**Рис. 10.2. Відображення другого розділу меню**

4. Створимо скрипт для прокрутки тексту «Привіт всім!!!» у статусному рядку браузера. Для цього після визначення функції *disp* запишемо наведений нижче програмний код:

//визначаємо змінну pos var pos=0;

//визначаємо функцію для прокрутки тексту

function status() {

//визначаємо текст, призначений для відображення str=«Привіт всім!!!»;

//визначаємо фрагмент тексту, що буде

//показаний у рядку статуса

//визначення ралізується за рахунок копіювання рядка str //з символу з номером pos по символ з номером pos+17 у рядок str1

str1=str.substring (pos, pos+27); //показуємо змінну str1 у рядку статуса window.defaultStatus=str1;

//збільшуємо значення змінної pos на 1 pos++

//перевіряємо значення змінної pos if (pos == 27) pos=0;

//рекурсивний виклик функції status з частотою 1 раз у 0,03 сек. setTimeout(«status()»,30);

}

</script>

Зазначимо, що у даному випадку коментарі призначені для кращого розуміння принципів функціонування скрипта. З цієї причини записувати їх не обов’язково.

1. Модифікуємо функцію *wr* (викликається внаслідок реалізації події onload) для того, щоб прокрутка починалась після завантаження HTML-сторінки у вікно браузера:

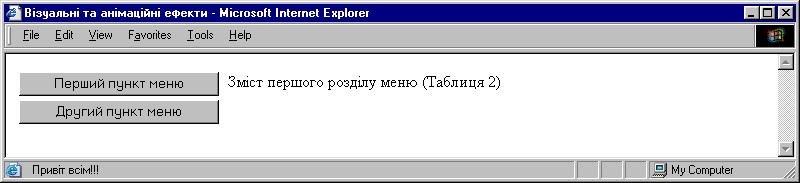
function wr() { window.moveTo(0,0);

window.resizeTo(w,h); status();

myopen();

}

1. Перегляд HTML-сторінки (рис. 10.3) повинен підтвердити прокрутку тексту у статусному рядку.



**Рис. 10.3. Прокрутка тексту у статусному рядку**

1. Створимо функцію *myopen*, що реалізує завантаження нового HTML-документа у нове вікно браузера. Подібні функції досить часто використовуються для показу користувачеві короткої та особливо актуальної інформації сайта. Таку інформацію можливо показувати у вікні браузера без панелі інструментів, рядка меню та смуг прокрутки. Для цього використаємо метод *open* стандартного об’єкта *window*. Методу *open* необхідно передати три параметри: ім’я файла, який буде відображено у новому вікні, ім’я та параметра вікна (параметри задаються одним рядком). У даному випадку параметри такі: панель інструментів, рядок меню та смуги прокрутки відсутні, ширина вікна 300, а висота 140 пікселів, лівий верхній кут вікна знаходиться нижче на 100 і лівіше на 200 пікселів від верхнього лівого кута екрана. Запишемо код функції *myopen*:

***...***

function myopen() {

/\* **Увага**! Значення змінної str необхідно записати в одному рядку без переносута пробілів. \*/

str=«toolbar=0,menubar=0,Scrollbars=0,width=300,height=140, top=100,left=200»;

window.open(«info.html»,»newa»,str);

}

</script>

1. У папці HTML створимо файл info.html та визначимо у ньому такий HTML-код:

<html>

<head>

<title>Увага!!! Актуальна інформація!!!</title>

</head>

<body>

<h1 align=«center»>Увага!!! <br>Актуальна інформація!!!</h1>

</body>

</html>

1. Зазначимо, що показувати користувачеві актуальну інформацію доцільно при перегляді ним певної сторінки сайта. Тому виклик функції *myopen* реалізуємо у функції *wr*, яка, у свою чергу, викликається при завантаженні нашої HTML-сторінки (*anim.html*). Для цього модифікуємо код функції *wr* так:

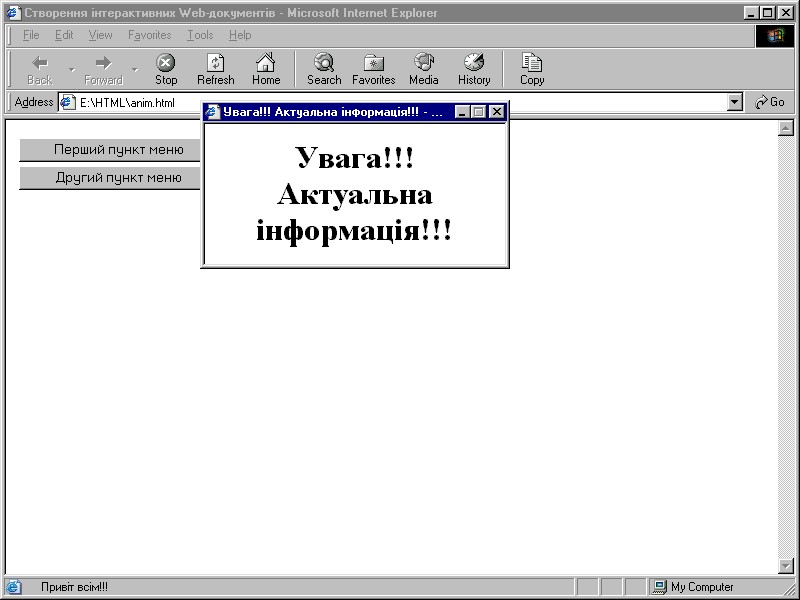
function wr() { window.moveTo(0,0);

window.resizeTo(w,h); status();

myopen();

}

1. Під час відкриття HTML-документа *anim.html* повинно відкритись нове вікно браузера із завантаженим у ньому файлом *info.html* (рис. 10.4).



**Рис. 10.4. Відкриття нового вікна браузера**

1. Створити меню із 5 пунктів. Вибір пункту меню повинен привести до відображення на Web-сторінці певного рисунка.
2. Модифікувати функцію *wr* так, щоб вікно браузера займало тільки половину екрана.
3. Оформити звіт.

### Запитання для самоперевірки

1. Яке призначення методу open об’єкта window?
2. Які параметри необхідно передати методу open об’єкта window?
3. Яка подія відповідає завантаженню HTML-документа у вікно браузера?
4. Яка подія відповідає вибору користувачем певного елемента Web-сторінки?
5. Яким чином можливо реалізувати рекурсивний виклик функції через певний проміжок часу?
6. Яке призначення методу moveTo об’єкта window?
7. Яке призначення методу resizeTo об’єкта window?
8. Як визначити власну функцію у коді JavaScript?
9. Як скопіювати частину символів із однієї змінної в іншу?
10. Як показати інформацію у рядку статусу вікна браузера?