**Лабораторна робота №11**

**Масиви**

Тип Array введено в JS для можливості маніпулювання такими об’єктами, які містять списочні дані. Це - список всіх гіпертекстових посилань даної сторінки, список всіх зображень на даній сторінці, список всіх елементів форми тощо. Користувач може створити свій власний масив через конструктор Array():

new\_array = new Array()

new\_array5 = new Array(5)

colors = new Array("red", "white", "blue")

Розмірність масива може змінюватися. Можнв спершу визначити масив, а потім присвоїти одному з його елементів значення. Як тільки це трапиться, зміниться і розмірність масива:

colors = new Array()

colors[5] = "red".

В даному випадку масив буде складатися з 6 елементів, оскільки першим елементом масива вважається елемент з індексом 0.

Для масивів є чотири методи: join, reverse, sort, concat. Join об’єднує елементи масива в рядок символів, в якості аргумента задається розділювач:

colors = new Array("red", "white", "blue")

string = acolors.join(" + ")

Отримаємо: string = "red + white + blue". Другий метод, reverse, змінює порядок елементів масива на зворотній, метод sort сортує їх в лексикографічному порядку, а метод concat об’єднує два масива.

Масиви мають дві властивості: length і prototype. Length визначає кількість елементів масива. Якщо треба виконати певну рутинну операцію над всіма елементами масива, можна скористатися циклом типу:

color = new Array("red", "white", "blue")

n = 0 while(n != colors.length)

{... оператори в тілі цикла ...}

Властивість prototype дозволяє добавити властивості до об’єктів масива. Проте частіше всього в програмах на JS використовуються вбудовані масиви, в основному графічні образи (Images) і гіпертекстові посилання (Links).

Приклад 1. Створення мультиплікації з допомогою масивів.

Часто для створення мультиплікації формують масив графічних об’єктів, які потім прокручують один за другим:

img\_array = new Array()

img\_array[0] = new Image(50,100)

img\_array[1] =new Image(50,100)

...

img\_array[99] = new Image(50,100)

В об’єкта Image є 10 властивостей, із яких найважливішим є атрибут src. Так, для присвоєння конкретних зображень елементам масива img\_array слід скористатися такою послідовністю команд:

img\_array[0].src = "image1.gif"

img\_array[1].src = "image2.gif"

...

img\_array[99].src = "image100.gif"

В даному випадку можна було скористатися і циклом для присвоєння імен, оскільки вони можуть складатися з констант і значення індексної змінної.

Розширюючи приклад з масивом Image, побудуємо документ, в якому буде вбудована мультиплікація, визначена нашим масивом:

Лістинг 1. Мультиплікація.

<HTML>

<HEAD>

<SCRIPT>

function mult()

{

img\_array = new Array()

for (var i=0; i<4; i++)

img\_array[i] = new Image(50,100)

img\_array[0].src = "e1.jpg"

img\_array[1].src = "e2.jpg"

img\_array[2].src = "e3.jpg"

img\_array[3].src = "e4.jpg"

var t=new Date()

p=-1

}

function s()

{

p=p+1

document.images[0].src =img\_array[p].src

setTimeout("s()",100)

if (p==3) p=-1

}

//-->

</SCRIPT>

</HEAD>

<BODY onLoad="mult()">

<img src="e1.jpg">

<br>

<input type="button" name="But" value="Подивитись" onClick="s()">

</BODY>

</HTML>

Далі розглянемо кілька класичних задач, присвячених роботі з масивами. Наведемо функції роботи з масивами, які цінні самі по собі і можуть застосовуватися в різних програмах.

Приклад 2. Бінарний пошук з формуванням таблиці результатів

Напишемо функцію, яка реалізує алгоритм бінарного пошуку таким чином, щоб під час роботи програми формувалась таблиця значень змінних i, j, k і деяких виразів.

Лістинг 2. Пошук у впорядкованому масиві з таблицею проміжних значень

<HTML>

<HEAD>

<TITLE>Бінарний пошук з табл. проміжних значень</TITLE>

<script language="JavaScript">

var v=new Array (2, 3, 5, 6, 6, 7,10,11, 20, 25)

function testtab(obj,v,t)

{ var res="i j k v[k] t<= v[k]"+"\r\n"

var i=0

var j= v.length-1

var k

while ( i < j )

{ k=Math.round( (i+j)/2+0.5)-1

res = res + i + " "+j+" "+k+" "+"v[" + k + "]=" + v[k] + " " + t + "<=" + v[k]+"\r\n"

if (t <= v[k] )

j=k

else

i=k+l

}

res += "v[" + i + "]=" +v[i]+"\r\n"

obj.resultl.value=res

if (v[i] == t )

{ return i}

else return -1

}

function test(obj)

{ obj.datal.yalue=v}

</script>

</HEAD>

<BODY>

<H4>Бінарний пошук</H4>

<FORM name="forml">

<pre>

Масив:<INPUT type="text" size=40 name="datal" ><hr>

Елемент:<INPUT type="text" size=20 name="data2" ><hr>

Результат пошуку: <INPUT type="text" size=20 name="result" ><hr>

Таблиця проміжних значень: <BR>

<textarea cols=50 rows=7 name="result1" > </textarea><hr>

</PRE>

<input type="button" value="Визначити" onClick="test(form1); forml.result.value=testtab(form1,v,forml.data2.value)">

<input type="reset" value="Скасувати">

</FORM>

</BODY>

</HTML>

Завдання

1. Перевірити приклади із лабораторної роботи.

2. Створити мультиплікацію з використанням масива.

3. Дано одномірний масив дійсних чисел. Написати сценарій визначення кількості додатніх елементів масива.

4. Дано одномірний масив дійсних чисел. Написати сценарій пошуку максимального елемента в масиві.