Array

1 大綱

2 基本

Array object (陣列物件)允許我們在程式中讓一個變數存放多個值。

【例】

```
<script>
a = new Array(); // 利用new 建立陣列，可以知道陣列是一個物件。
a[0] = "This is a pen"; //自動添加一個元素（長度為1）然後指派字串
a[1] = 3.14; //
a[2] = [1,2,3];  // 元素a[2]是一個陣列
document.write("a[0] = " + a[0] + "<br>");
document.write("a[1] = " + a[1] + "<br>");
document.write("a[2] = " + a[2] + "<br>");
</script>
```

> 存取每一個元素的值，可以利用 a[0], a[1] , a[2] 。

而每一個值的存取，都可以藉由索引號碼來操作，最大號碼是$2^{32}-1$。

上述程式先宣告一個空陣列(a) ，然後在程式碼中，分別指派元素值。

也可以宣告的時候指派。例如，

```
var fruits = new Array( "apple", "orange", "mango" );
```

除了使用關鍵字new 以外，也可以

```
var fruits = [ "apple", "orange", "mango" ];
```

陣列的宣告方式，下面4種都可以：

```
var a=[]; //空陣列，
var a= new Array; //建立物件的方式 new
var a=Array(); //直接使用constructor
var a= new Array();  //利用constructor 又利用new ，有點多餘。
```

在javascript中，Array也可以拿來作為資料結構stack的使用，例如方法：

push(),pop()。

【例】

如果未經宣告為陣列(例如var a;)，直接使用a[0]=3會發生錯誤

a[0]=3

Uncaught TypeError: Cannot set property '0' of undefined

at <anonymous>

正確

```
var a=[];
a[5]=9;
```

此時a的內容

(6) [empty × 5, 9]

長度為6。

但是，利用push,pop 可以避免宣告。

```
a.push(3);
//此時a的長度為1
a.pop()
//此時a的長度為0
```

2.1 合併

水平

var a=[1,2,3];

var b=[4,5,6];

方法1

c=[];

c.concat(a);

c.concat(b);

方法2

c=a.concat(b); //c=[1,2,3,4,5,6],a=[1,2,3]

垂直

c=[];
c.push(a);
c.push(b);

2.2 array內容的列印

var a=[1,2,3];
document.write(a);
console.table(a);

自行定義
function displayArray(a)
{
}

3 內建屬性和方法

3.1 Array Properties

陣列物件的方法

Property	Description
constructor	Returns a reference to the array function that created the object.
index	The property represents the zero-based index of the match in the string
input	This property is only present in arrays created by regular expression matches.
length	Reflects the number of elements in an array.
prototype	The prototype property allows you to add properties and methods to an object.

3.2 Array Methods

Here is a list of the methods of the Array object along with their description.

Method	Description

concat()	合併陣列。
	`<script type="text/javascript">`
	`var alpha = ["a", "b", "c"];`
	`var numeric = [1, 2, 3];`
	`var alphaNumeric = alpha.concat(numeric);`
	`document.write("alphaNumeric : " + alphaNumeric );`
	`</script>`

every()	Returns true if every element in this array satisfies the provided testing function.
filter()	Creates a new array with all of the elements of this array for which the

	provided filtering function returns true.
forEach()	Calls a function for each element in the array.
indexOf()	Returns the first (least) index of an element within the array equal to the specified value, or -1 if none is found.
join()	每個元素利用分隔字元隔開的字串表示方式。
lastIndexOf()	Returns the last (greatest) index of an element within the array equal to the specified value, or -1 if none is found.
map()	Creates a new array with the results of calling a provided function on every element in this array.
pop()	從尾端去掉第一個。
push()	從尾端加入。
reduce()	Apply a function simultaneously against two values of the array (from left-to-right) as to reduce it to a single value.
reduceRight()	Apply a function simultaneously against two values of the array (from right-to-left) as to reduce it to a single value.
reverse()	傳回一個元素位置反轉的陣列.
shift()	從開頭去掉第1個。
slice()	傳回陣列的一個區段
some()	Returns true if at least one element in this array satisfies the provided testing function.
toSource()	Represents the source code of an object
sort()	Sorts the elements of an array
splice(start, howmany)	移除陣列中的元素，然後傳回被刪除的元素，而且可以在原位插入新元素。第2個參數可以省略，則為最後。例如，若x為10個元素的陣列，則x.splice(0,3) 傳回索引0到2 的陣列,且x的大小便為7（參考這裡）.
toString()	將陣列利用字串表示。
unshift()	從陣列開頭加入。

```
function getRandomArray(num){
  var x=[];
  for(var i=0;i<num;i++){
    x.push(Math.floor(Math.random()*100) );
  }
  return x;
}
```

```
x=getRandomArray(10)
```

(10) [86, 11, 55, 99, 22, 98, 67, 8, 34, 56]

```
x.splice(0,3) //0:start , end 3 (但不包含)
```

(3) [86, 11, 55]

```
x //顯示 x 的內容
```

(7) [99, 22, 98, 67, 8, 34, 56]

```
x.slice(0)  ////傳回索引 0 到最後一個元素的陣列
```

(7) [99, 22, 98, 67, 8, 34, 56]

```
x.slice(0,3)  //傳回索引 0 到 2 的陣列
```

(3) [99, 22, 98]

```
x  //利用 slice ,x 本身這個陣列不會被改變
```

(7) [99, 22, 98, 67, 8, 34, 56]

```
x.splice(5)
```

(2) [34, 56]

4 其他

4.1 判斷array是否是空的

```
if(array && array.length){
   // not empty
} else {
   // empty
}
```

或者也可以：

```
Objects.keys(__array__).length
```

typeof 測試

```
function ck_typeof(r,c,v) {
  if (typeof v===undefined) console.log("by typeof");
  if ( v===undefined) console.log("no typeof");

}
```

```
ck_typeof(1,2);
```

>> no typeof

利用開發人員工具分析為什麼第1個if沒作用

在console上測試為什麼：

```
typeof v
```

>> "undefined"

可以知道typeof v的結果為字串

```
typeof v === undefined
```

>>false

因此使用=== 的比較結果是false。

- 利用typeof ，不管變數有沒有宣告，如果沒有給定初始值，都是字串 "undefined"
- 如果已經宣告，則預設被指定的內容為 undefined（不是字串）

typeof v===undefined	false
typeof v==undefined	false
v===undefined	true
v==undefined	true

 問題：利用 == 的結果是什麼？

結果仍然是false 。但是我預期是true，因為== 不是說可以型態轉換？利用這個測試

```
function ck_typeof(r,c,v) {
  if (typeof v===undefined) console.log("=== by typeof");
  if (typeof v==undefined) console.log("== by typeof");
  if ( v===undefined) console.log("=== no typeof");
  if ( v==undefined) console.log("==no typeof");

}
```

判斷元素是否有意義

JavaScript 的陣列索引從0開始，一直到length-1。這些索引是連續的，不存在間斷。要判斷某個位置是否有值，可以

```
if (index < array.length) {
  // do stuff
}
```

但是因為陣列的元素值可以是null, undefined, NaN, Infinity, 0, 甚至是其他非數值。因此，要判斷是否數值有意義，或已經定義可以利用

```
if (typeof array[index] !== 'undefined') {
```

或

```
if (typeof array[index] !== 'undefined' && array[index] !== null) {
```

利用==和!= 可以簡化上面第二種判斷方式，因為這兩個比較符號，認為null 和 undefined 相等：

```
if (array[index] != null) {
```

5 範例

5.1 矩陣相乘

```
<html>
<head>
<script>
function multiplyMatrices(m1, m2) {
    var result = [];
    for (var i = 0; i < m1.length; i++) {
        result[i] = [];
        for (var j = 0; j < m2[0].length; j++) {
            var sum = 0;
            for (var k = 0; k < m1[0].length; k++) {
                sum += m1[i][k] * m2[k][j];
            }
            result[i][j] = sum;
        }
    }
    return result;
}

var m1 = [[1,2],[3,4]];
var m2 = [[5,6],[7,8]];

var mResult = multiplyMatrices(m1, m2);

/*In Google Chrome and Firefox you can do:*/

console.table(mResult) /* it shows the matrix in a table */
</script>
</head>
<body>
```

```
</body>
</html>
```

5.2 output to html table

```
function createTable(tableData) {
  var table = document.createElement('table');
  var row = {};
  var cell = {};

  tableData.forEach(function(rowData) {
    row = table.insertRow(-1); // [-1] for last position in Safari
    rowData.forEach(function(cellData) {
      cell = row.insertCell();
      cell.textContent = cellData;
    });
  });
  document.body.appendChild(table);
}
```

5.3 read csv file to array

<参考 [JS] 20 reading CSV file.docx >

TNote: 需要瞭解string 物件。

利用HTML5 API 李的物件FileReader。

1.Create a input field

```
<input type="file" id="file" name="file">
```

2.Trigger a response on change of this input

```
var file = document.getElementById('file');
file.addEventListener('change', function() {
    var reader = new FileReader();
    var f = file.files[0];
    reader.onload = function(e) {
        var CSVARRAY = parseResult(e.target.result); //this is where the csv array will
  be
    };
    reader.readAsText(f);
});
```

3.Parse the result to an array by using split/push. This uses \n as row delimiter and , as cell delimiter.

```
function parseResult(result) {
    var resultArray = [];
    result.split("\n").forEach(function(row) {
        var rowArray = [];
        row.split(",").forEach(function(cell) {
            rowArray.push(cell);
        });
        resultArray.push(rowArray);
    });
    return resultArray;
}
```