
BÀI 8: MẢNG VÀ CHUỖI

Mục tiêu bài học:

Kết thúc chương này, bạn có thể:

- Làm việc với các kiểu mảng trong JavaScript
- Làm việc với đối tượng String

8.1 Mảng

Có những lúc ta muốn lưu nhiều giá trị vào trong một biến. Khi đó ta sử dụng mảng. Mảng được dùng để lưu một tập hợp các biến có cùng tên. Chỉ số của mảng dùng để phân biệt các biến này. Trong JavaScript chỉ số của mảng bắt đầu từ 0.

Tuy nhiên, JavaScript không có kiểu dữ liệu mảng. Nhưng nó có một đối tượng mảng được xây dựng sẵn. Để dùng mảng trong chương trình, ta phải sử dụng đối tượng mảng và các phương thức của nó.

➤ Tạo mảng

Câu pháp sau đây dùng để tạo một mảng:

```
arrayObjectName    =    new    Array([element0,    element1,    ...,  
elementN])
```

Trong đó **arrayObjectName** là tên của đối tượng mảng, và **elementN** là danh sách các giá trị của các phần tử mảng. Mảng được khởi tạo với các giá trị xác định như là các phần tử của nó. Thuộc tính `length` lưu số lượng các phần tử có trong mảng.

➤ Thêm các phần tử

Chúng ta có thể đưa các phần tử vào một mảng ngay khi chúng ta tạo nó. Chúng ta cũng có thể đưa dữ liệu vào một mảng bằng cách gán giá trị cho các phần tử của nó. Ví dụ,

```
emp[0] = "Ryan Dias"  
emp[1] = "Graham Browne"  
emp[2] = "David Greene"
```

Ryan Dias	Graham Browne	David Greene
-----------	---------------	--------------

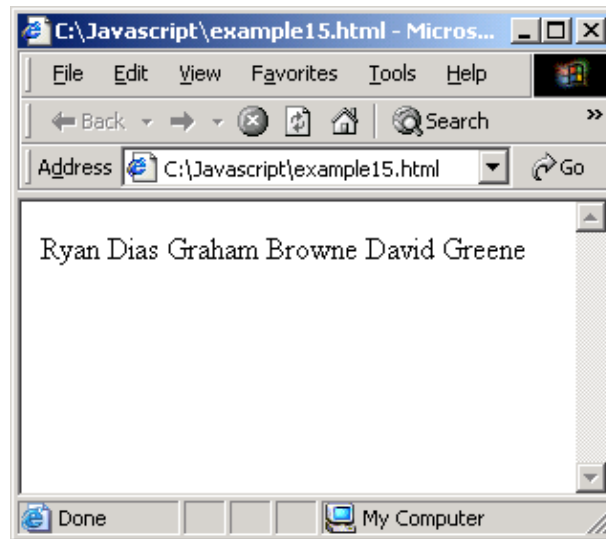
Đoạn mã trên tạo ra một mảng và gán giá trị cho từng phần tử của nó.

Ví dụ 15:

```
<html>  
  <head>  
    <script language="JavaScript">  
      <!--  
      emp = new Array(3)  
      emp[0] = "Ryan Dias"  
      emp[1] = "Graham Browne"  
      emp[2] = "David Greene";  
      document.writeln(emp[0]);  
      document.writeln(emp[1]);  
      document.writeln(emp[2]);  
      //-->  
    </script>
```

```
</head>  
</html>
```

Kết quả:



Hình 13.16: JavaScript: Mảng

➤ **Truy cập các phần tử mảng**

Có hai cách để truy cập các phần tử của mảng. Chúng ta có thể:

- Chỉ ra tên của phần tử. Ví dụ, emp ["Ryan Dias"]
- Chỉ ra chỉ số của phần tử. Ví dụ, emp [0].

➤ **Sử dụng các phương thức của mảng:**

Để thao tác trên mảng, chúng ta phải dùng các phương thức của đối tượng mảng.

Phương thức	Mô tả
-------------	-------

Join	Kết hợp các phần tử của mảng thành một chuỗi
Pop	Trả về phần tử cuối cùng của mảng, sau khi xóa nó từ mảng.
Push	Thêm một hoặc nhiều phần tử vào cuối mảng. Trả lại phần tử cuối cùng thêm vào.
Reverse	Đảo ngược các phần tử của mảng: phần tử đầu tiên của mảng trở thành phần tử cuối cùng và phần tử cuối cùng trở thành phần tử đầu tiên.
Shift	Xóa phần tử đầu tiên của mảng và trả về phần tử đó.
Sort	Sắp xếp các phần tử của mảng.

- **Phương thức sort**

Phương thức `emp.sort()` sẽ sắp xếp mảng dưới đây

Ví dụ 16:

```
<html>
<head>
  <script language="JavaScript">
    <!--
    emp = new Array(3)
    emp[0] = "Ryan Dias"
    emp[1] = "Graham Browne"
    emp[2] = "David Greene";

    emp.sort();
```

```
document.writeln(emp[0]);  
document.writeln(emp[1]);  
document.writeln(emp[2]);
```

```
//-->
```

```
</script>
```

```
</head>
```

```
</html>
```

Kết quả:

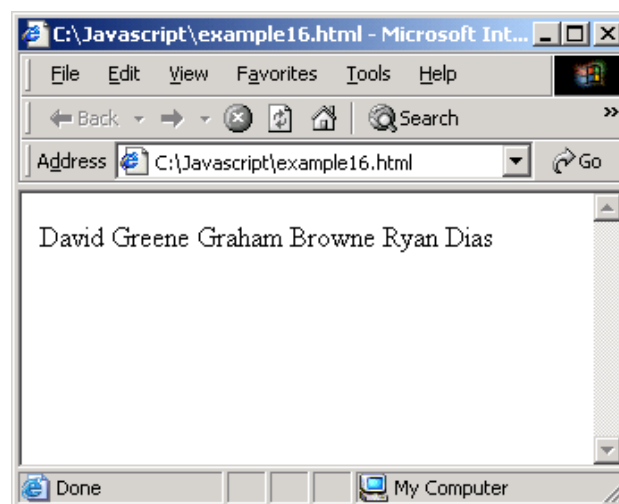


Figure 13.17: JavaScript: Sắp xếp các mảng (Sorting Arrays)

Mảng nhiều chiều

Một mảng có thể có nhiều hơn một chiều. Ví dụ, ta có thể tạo ra một mảng hai chiều để lưu trữ mã nhân viên và tên của nhân viên đó.

	Employee	Code
0	Ryan Dias	002
1	Mike Donne	003
2	Joe Dean	004
3	Robert Matey	005

Hình 13.18: Mảng hai chiều

Đoạn mã dưới đây tạo ra một mảng hai chiều và cho hiển thị giá trị của một trong những phần tử trong mảng.

Ví dụ 17:

```
<HTML>
<HEAD>
  <SCRIPT LANGUAGE = "JavaScript">
    <!--
```

```
    MyArray = new Array(5,5);
```

```
    MyArray[0, 0] = "Ryan Dias";
```

```
MyArray[0, 1] = 1;  
MyArray[1, 0] = "Mike Donne";  
MyArray[1, 1] = 2;  
MyArray[2, 0] = "Joe Dean";  
MyArray[2, 1] = 3;  
MyArray[3, 0] = "Robert Matey";  
MyArray[3, 1] = 4;  
document.write ("The name is  " + MyArray[3, 0]);  
document.write("and the code is " + MyArray[3,1]);  
  
-->  
</SCRIPT>  
</HEAD>  
</HTML>
```

Kết quả:

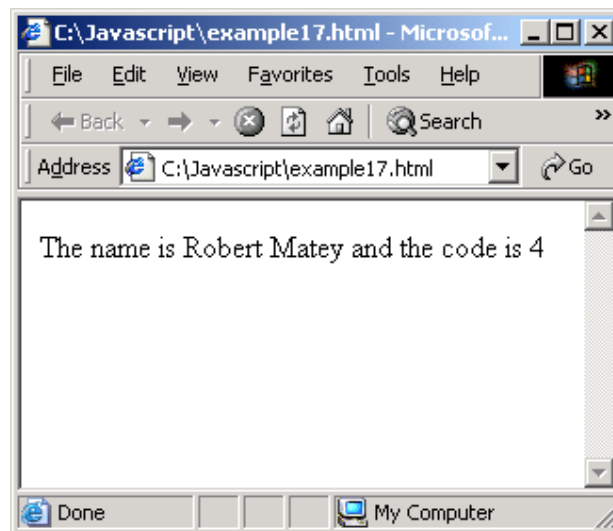


Figure 13.19: Mảng hai chiều (Two dimensional Array)

8.2 Đối tượng String

Đối tượng string được dùng để thao tác và làm việc với chuỗi văn bản. Chúng ta có thể tách nó ra thành các chuỗi con và biến đổi chuỗi đó thành các chuỗi hoa hoặc thường trong một chương trình. Cú pháp tổng quát là:

`stringName.propertyName`

Hoặc

`stringName.methodName`

Có 3 phương thức khác nhau để tạo ra chuỗi.

- **Dùng lệnh var và gán cho nó một giá trị. Ví dụ như**
`var newstr = “ This is my script”`
- **Dùng một toán tử (=) có gán với một tên biến. Ví dụ như:**
`newstr = “This is my script”`
- **Dùng hàm khởi tạo String (string). Ví dụ như:**
`var newstr = String (“This is my script”)`

Các thuộc tính:

Tên	Mô tả
length	Trả lại độ dài của chuỗi

Phương thức:

Tên	Mô tả
Big	Tăng kích thước của chuỗi văn bản
Blink	Tạo hiệu ứng nhấp nháy cho chuỗi (Internet Explorer không hỗ trợ phương thức này)
Bold	In đậm chuỗi
Fontcolor	Xác định màu của font chữ
Italics	Hiển thị chuỗi ở dạng in nghiêng
Small	Giảm kích thước của chuỗi văn bản
Strike	Hiển thị chuỗi có đường gạch ngang nằm giữa (strikethrough)
Sub	Hiển thị văn bản dưới dạng chỉ số dưới
Sup	Hiển thị văn bản dưới dạng chỉ số trên
ToLowerCase	Chuyển chuỗi thành ký tự thường
ToUpperCase	Chuyển chuỗi thành ký tự hoa

Ví dụ dưới đây hiển thị một vài phương thức và công dụng của chúng:

Ví dụ 7:

```
<HTML>
```

```
<HEAD>
```

```
  <script language = "Javascript">
```

```
    var bstr = "big";
```

```
    var sstr = "small";
```

```
    var blstr = "bold";
```

```
    var blkstr = "blink"
```

```
    var ucase = "Uppercase";
```

```
    var lcase = "Lowercase";
```

```
    document.write ("<BR>This is "+ bstr.big() + " text");
```

```
        document.write ("<BR>This is "+ sstr.small() + " text");
```

```
        document.write ("<BR>This is "+ blstr.bold() + " text");
```

```
        document.write ("<BR>This is "+ blkstr.blink() + " text");
```

```
        document.write ("<BR>This is "+ ucase.toUpperCase() + " text");
```

```
        document.write ("<BR>This is "+ lcase.toLowerCase() + " text");
```

```
  </script>
```

```
</HEAD>
```

```
</HTML>
```

