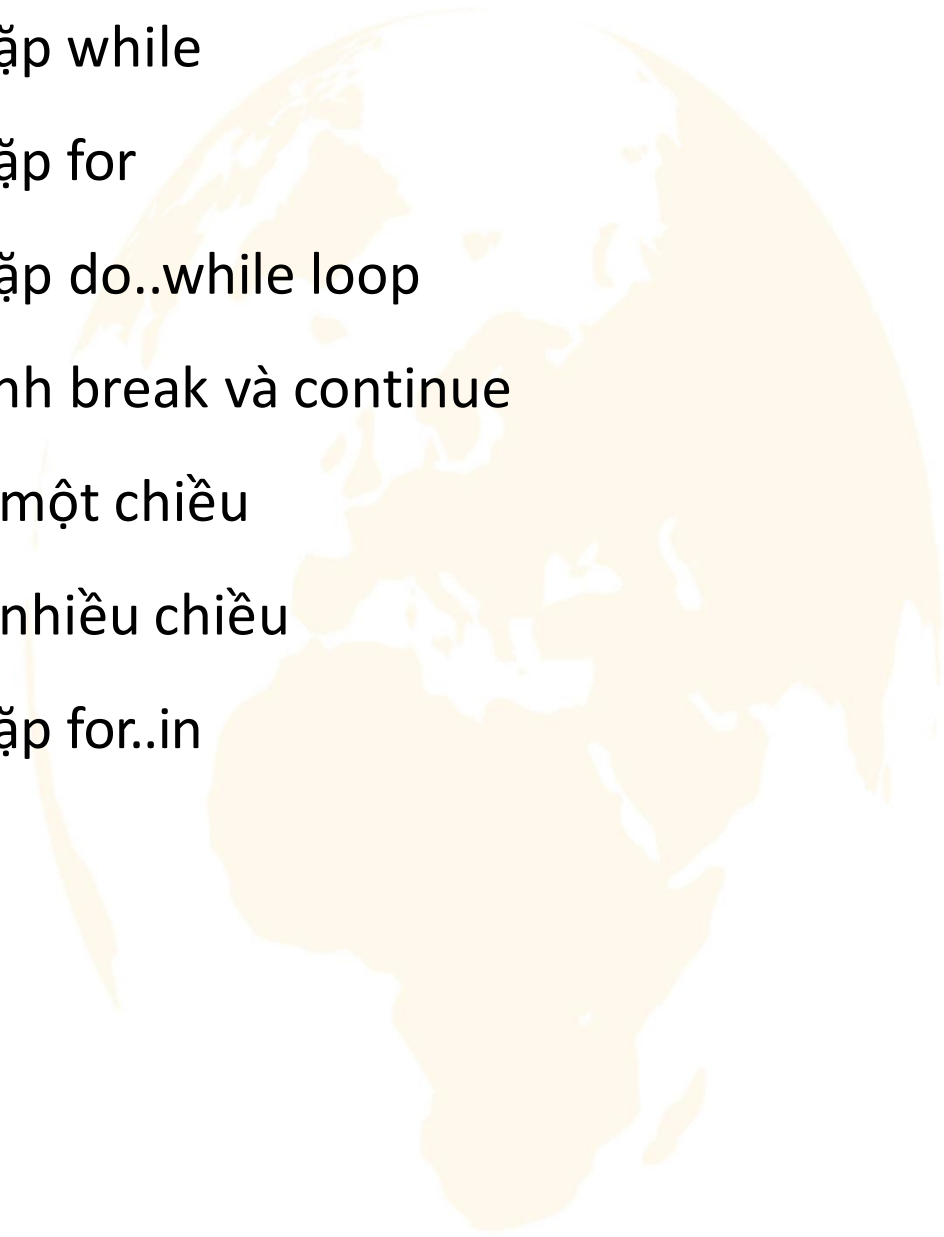


TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

BÀI 14: VÒNG LẶP VÀ MẢNG

Mục tiêu

- Giải thích vòng lặp while
- Giải thích vòng lặp for
- Giải thích vòng lặp do..while loop
- Giải thích câu lệnh break và continue
- Giải thích mảng một chiều
- Giải thích mảng nhiều chiều
- Giải thích vòng lặp for..in



Giới thiệu

Vòng cho phép bạn thực hiện một lệnh đơn hoặc một khối lệnh nhiều lần.

Chúng được sử dụng rộng rãi khi bạn muốn hiển thị một loạt các con số và chấp nhận đầu vào lặp đi lặp lại.

Một cấu trúc vòng lặp bao gồm một điều kiện là chỉ thị cho trình biên dịch số lần một khối cụ thể của mã sẽ được thực thi.

Nếu điều kiện không được chỉ định trong cấu trúc, vòng lặp tiếp tục vô hạn. Cấu trúc vòng lặp như vậy được gọi là vòng lặp vô hạn như.

- JavaScript hỗ trợ ba loại vòng được như sau:
 - `while Loop`
 - `for Loop`
 - `do-while Loop`

Vòng lặp while 1-3

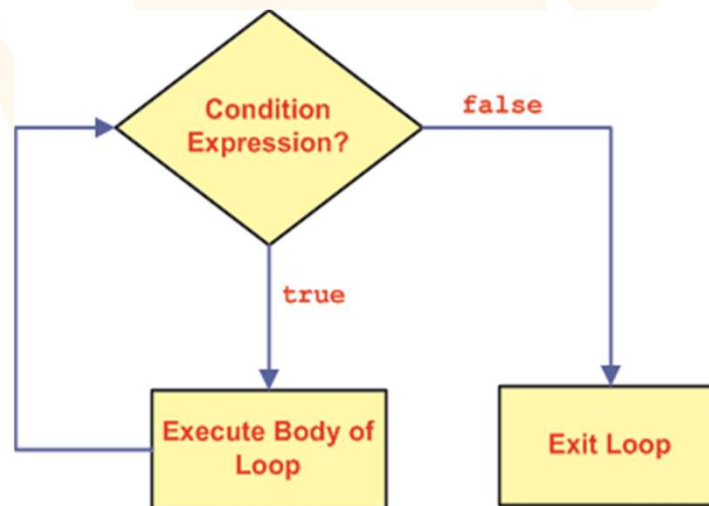
Vòng lặp while thực hiện một khối mã miễn là điều kiện vẫn còn đúng.

Vòng lặp while bắt đầu với các từ khóa while , tiếp theo là dấu ngoặc đơn có chứa một điều kiện boolean.

Nếu điều kiện này trả về true, khối lệnh trong vòng lặp while được thực hiện.

Một khi điều kiện trở thành sai, câu lệnh while dừng lại và chuyển điều khiển đến câu lệnh tiếp theo xuất hiện sau khối while.

- Hình dưới đây cho thấy các luồng thực hiện của vòng lặp while.



Vòng lặp while 2-3

Cú pháp:

```
while (condition)
{
// statements;
}
```

- Ví dụ

```
<script>
var i = 0;   var sum = 0;
while(i<=10)
{
sum = sum + i;
i = i + 1;
}
alert('Sum of first 10 numbers: ' + sum);
</script>
```

Vòng lặp while 3-3

- Kết quả



Vòng lặp for 1-4

Vòng lặp for tương tự như vòng lặp while như nó thực hiện các câu lệnh trong vòng lặp khi điều kiện là đúng.

Không giống như vòng lặp while, vòng lặp for chỉ định các câu lệnh kiểm soát vòng lặp ở đầu thay vì trong thân của vòng lặp.

Vòng lặp for bắt đầu với từ khóa for, tiếp theo là dấu ngoặc đơn có chứa ba biểu thức, mỗi trong số đó được phân cách bằng dấu chấm phẩy.

Ba biểu thức đó là **initialization expression, condition expression, increment/decrement expression.**

Cú pháp:

```
for (initialization; condition; increment/decrement)
{
// statements;
}
```

Vòng lặp for 2-4

- Các dạng for.

Simple for Loop

```
for (initialization;  
condition;  
increment/decrement)  
{  
    //statements;  
}
```

for Loop Without Expression 1

```
for ( ; condition;  
increment/decrement)  
{  
    //statements;  
}
```

for Loop Without Expression 2

```
for (initialization; ;  
increment/decrement)  
{  
    //statements;  
}
```

for Loop Without Expression 3

```
for (initialization;  
condition; )  
{  
    //statements;  
}
```

for Loop Without Expressions

```
for ( ; ; )  
{  
    //statements;  
}
```

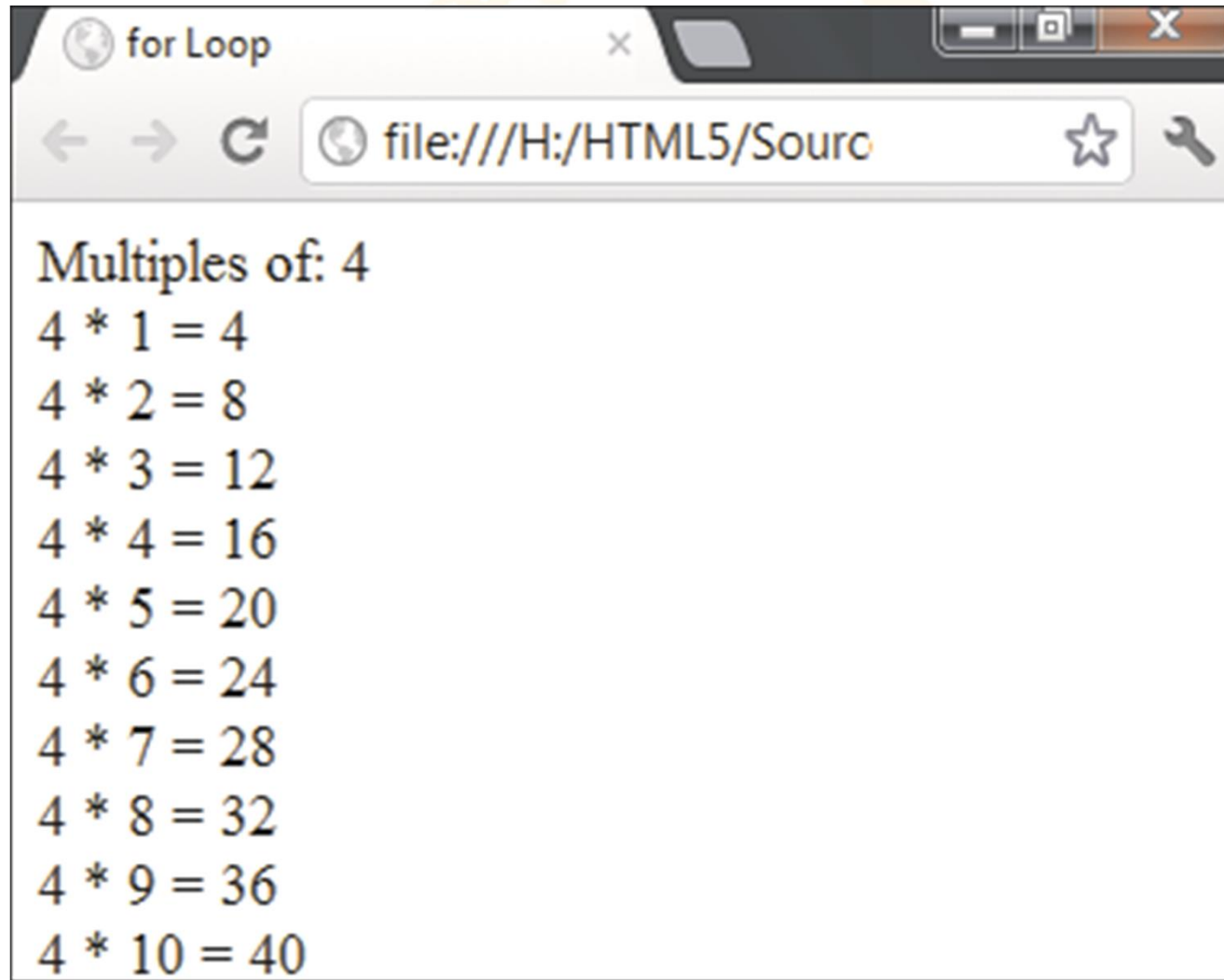

Vòng lặp for 3-4

- Ví dụ

```
<script>
var inputNum = prompt('Enter any number:');
var result = 0;
document.write ('Multiples of: ' + inputNum + '<br />');
for (var i=1; i<=10; i++)
{
    result = inputNum * i ;
    document.write (inputNum + ' * ' + i + ' = ' +
        + result + '<br />');
}
</script>
```

Vòng lặp for 4-4

- Kết quả



The screenshot shows a web browser window with the title 'for Loop'. The address bar displays 'file:///H:/HTML5/Sourc'. The main content area shows the following text:

```
Multiples of: 4
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
4 * 10 = 40
```

Vòng lặp do-while 1-4

Vòng lặp do-while cũng tương tự như vòng lặp while. Điều này là do cả hai do-while và while thực hiện việc lặp cho đến khi điều kiện trở thành sai.

Tuy nhiên, vòng lặp do-while khác bằng cách thực hiện thân của vòng lặp ít nhất một lần trước khi đánh giá điều kiện ngay cả khi điều kiện là sai.

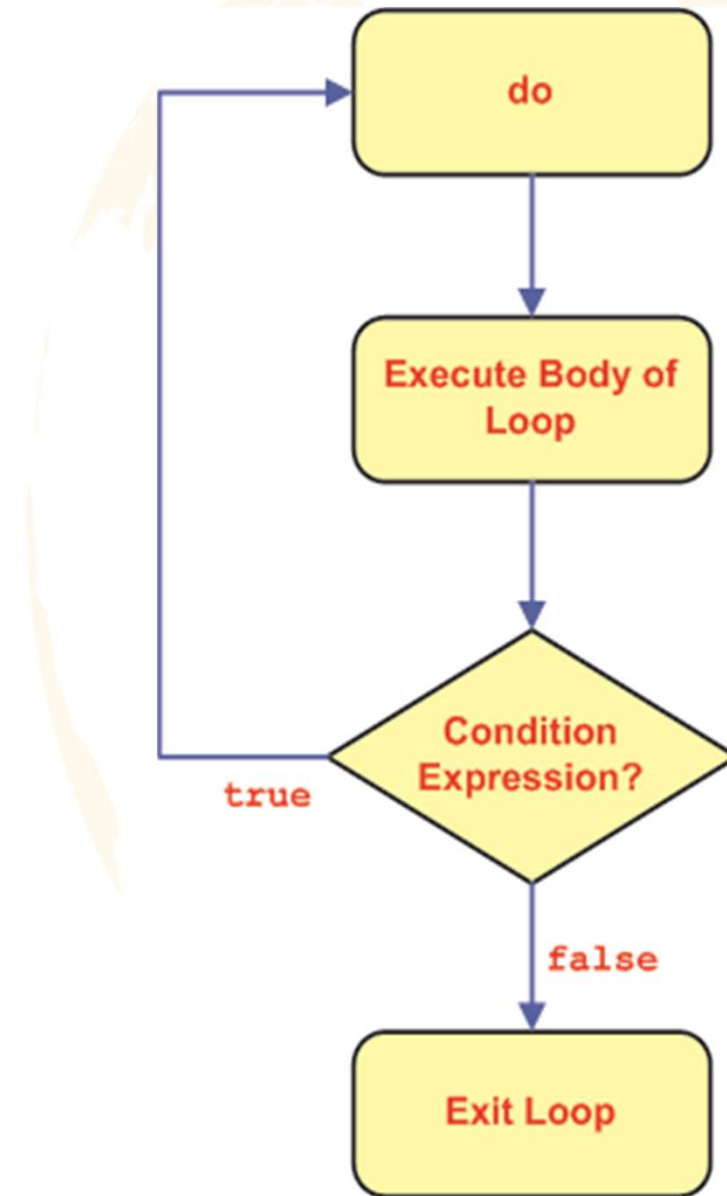
Vòng lặp do-trong khi bắt đầu với từ khóa do và được theo sau là một khối lệnh.

Vào cuối của khối, từ khóa while được xác định theo sau bởi dấu ngoặc đơn có chứa điều kiện.

Khi điều kiện trả về false, khối lệnh sau từ khóa do được bỏ qua và câu lệnh tiếp theo sau câu lệnh while được thực thi.

Vòng lặp do-while 2-4

- Sơ đồ thực thi của do-while



Vòng lặp do-while 3-4

Cú pháp:

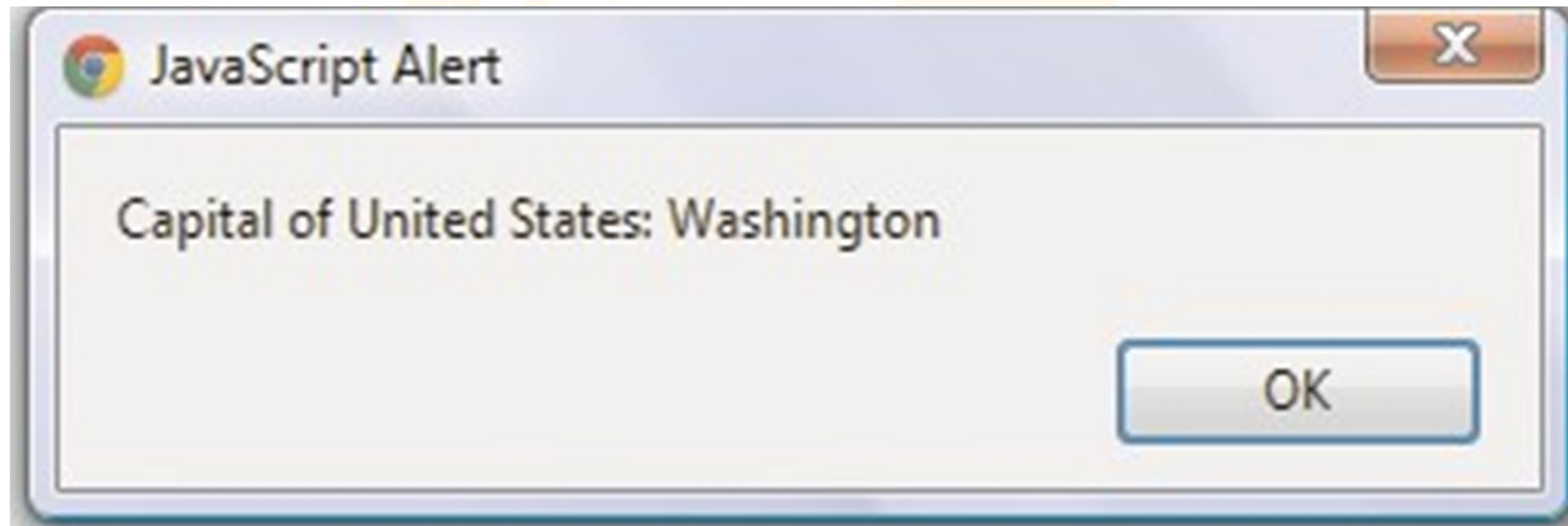
```
do
{
...
statements;
...
}while (condition);
```

• Ví dụ

```
<script>
var answer =
''; do
{
    answer = prompt('Capital of United States:', '');
}while (answer != 'Washington');
alert('Capital of United States: ' + answer);
</script>
```

Vòng lặp do-while 4-4

- Kết quả



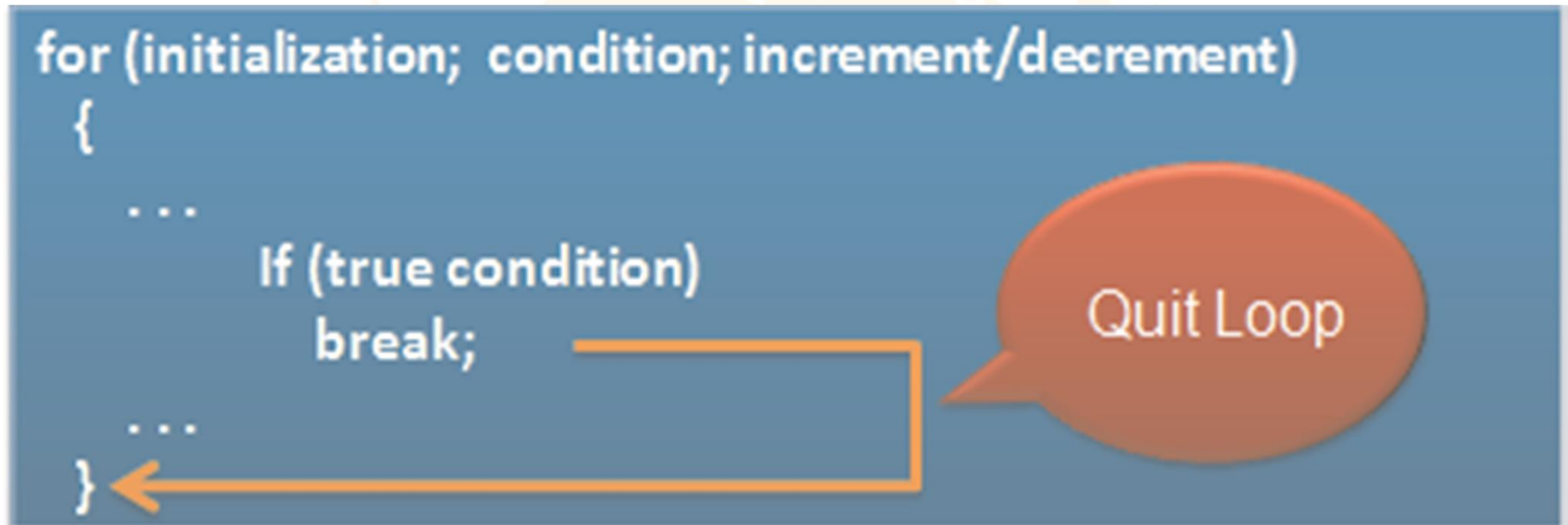
Câu lệnh break 1-3

Lệnh break có thể được sử dụng với câu lệnh ra quyết định cũng như switch và cấu trúc như vòng lặp for và while

Lệnh break được ký hiệu bằng cách sử dụng từ khoá break. Nó được sử dụng để thoát khỏi vòng lặp mà không xét các điều kiện quy định.

Điều khiển sau đó được đưa vào câu lệnh tiếp theo ngay sau vòng lặp.

- Hình sau thể hiện luồng thực thi của break



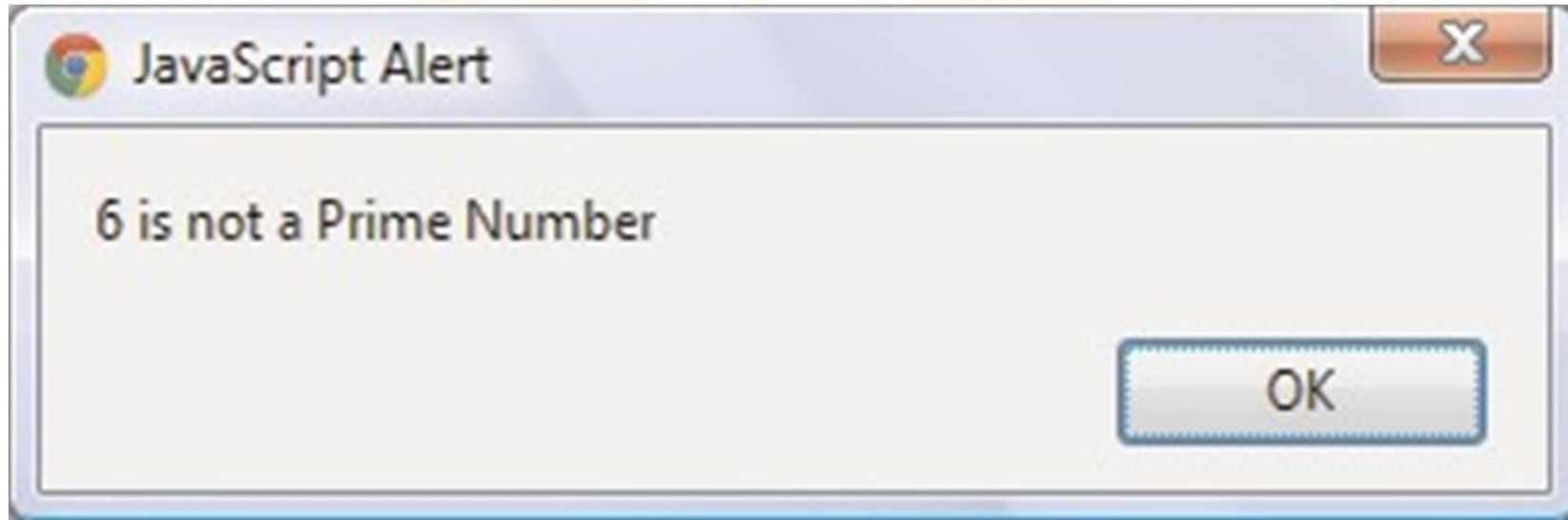
Câu lệnh break 2-3

- Ví dụ

```
<script>
var inputNum = parseInt(prompt('Enter number: ',''));
var num = 2;
while(num <= inputNum-1)
{
    if(inputNum % num == 0)
    {
        alert(inputNum + ' is not a Prime Number');
        break;
    }
    num++;
}
if(num == inputNum)
{
    alert(inputNum + ' is a Prime Number');
}
</script>
```


Câu lệnh break 3-3

- Kết quả



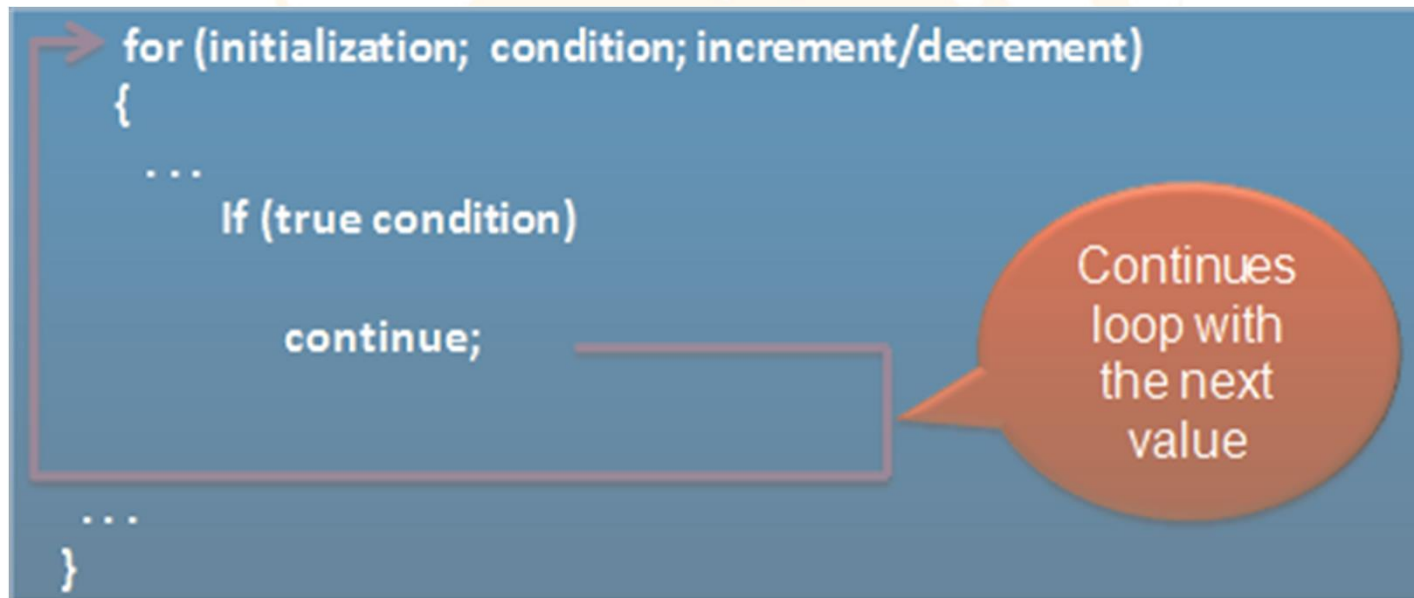
Câu lệnh continue 1-3

Câu lệnh continue chủ yếu được sử dụng trong các cấu trúc vòng lặp và được ký hiệu là từ continue.

Nó được sử dụng để chấm dứt việc thực hiện hiện tại của vòng lặp và tiếp tục với sự lặp lại tiếp theo bằng cách trả lại quyền kiểm soát cho đầu vòng lặp.

Điều này có nghĩa, câu lệnh tiếp tục sẽ không chấm dứt vòng lặp hoàn toàn, nhưng chấm dứt thực thi hiện hành.

- Hình sau thể hiện luồng thực thi của continue



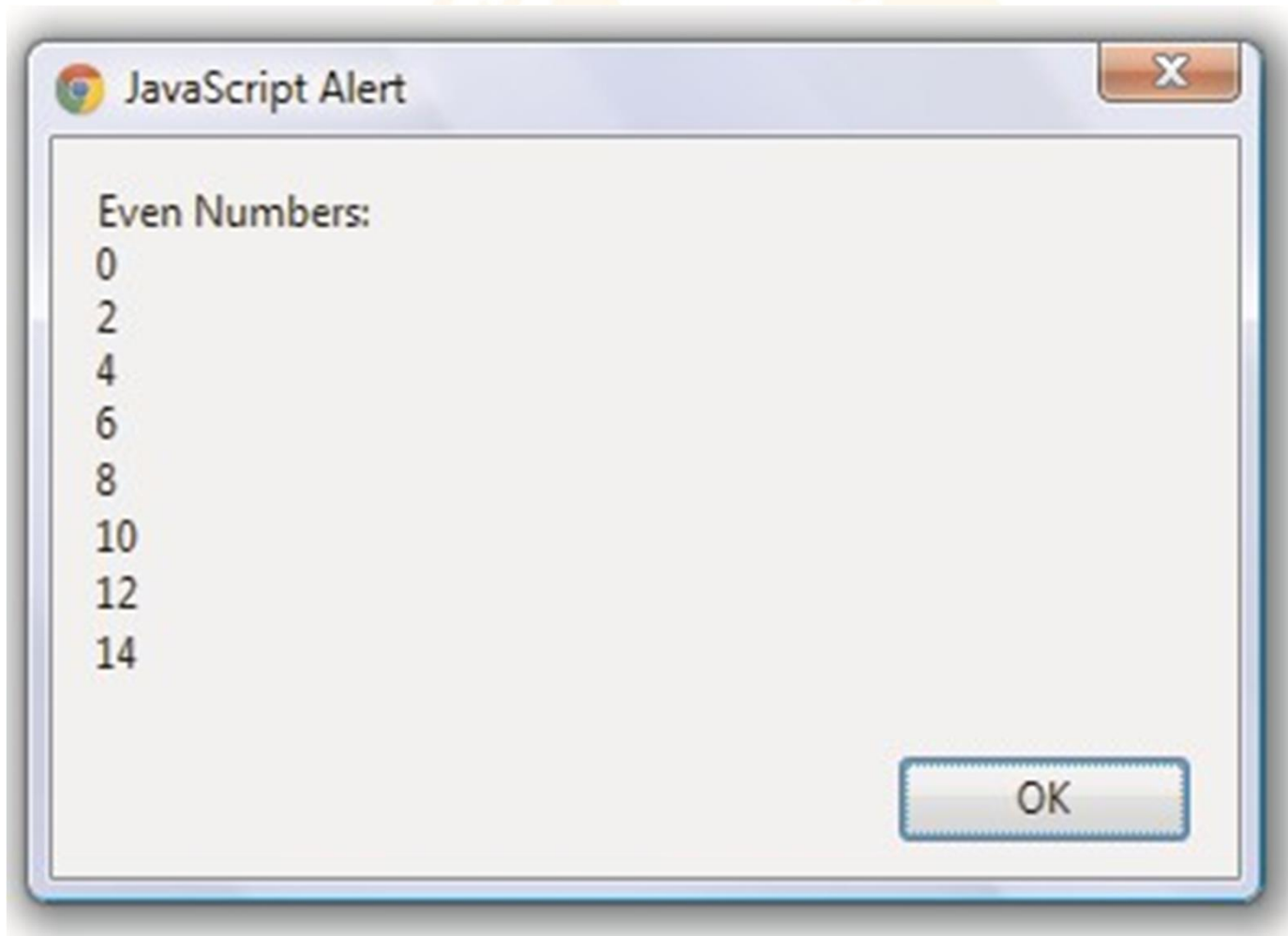
Câu lệnh continue 2-3

- Ví dụ

```
<script>
  var result = '';
  for (var i = 0; i <= 15; i++)
  {
    if((i%2) != 0)
    {
      continue;
    }
    result = result + i + '\n';
  }
  alert('Even Numbers:\n' + result);
</script>
```

Câu lệnh continue 3-3

- Kết quả



Mảng

Một mảng là một tập hợp các giá trị được lưu trữ trong bộ nhớ vị trí liên kề.

Các giá trị mảng được tham chiếu sử dụng tên mảng. Các giá trị của một biến mảng phải có cùng kiểu dữ liệu.

Những giá trị này cũng được gọi là phần tử có thể được truy cập bằng cách sử dụng chỉ số.

- Con số sau đây cho thấy việc sử dụng hiệu quả bộ nhớ đạt được bằng cách sử dụng một mảng.

Array of 100 Names

Steve	David	John	Steffen
-------	-------	------	---------	---------

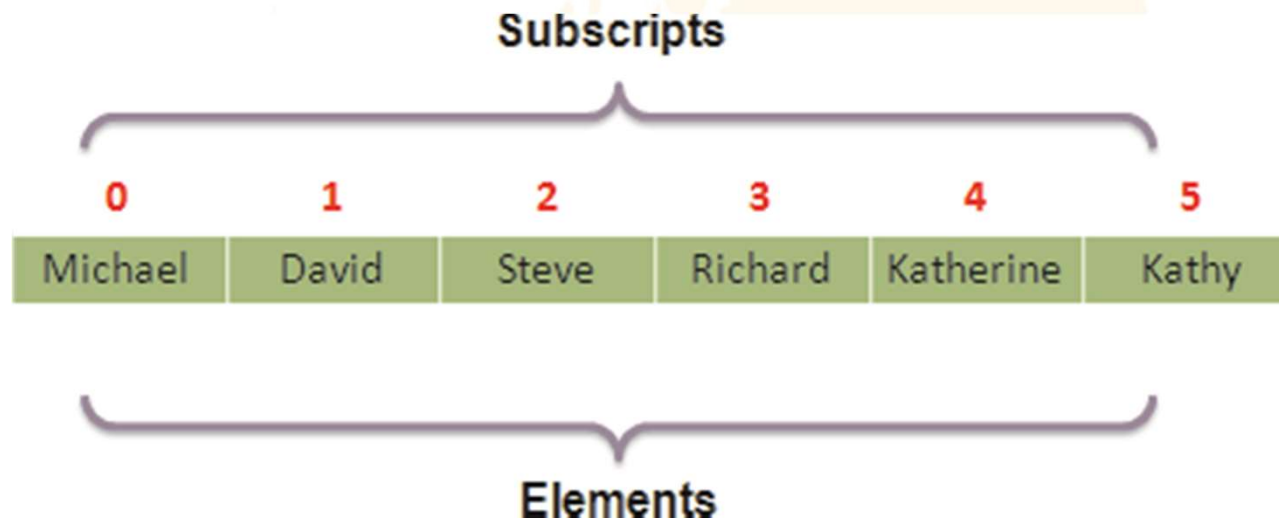
Proper Utilization of Memory

100 variables storing names of students	
<code>var studentOne</code>	Steve
<code>var studentTwo</code>	David
<code>var studentThree</code>	John
<code>var studentFour</code>	Steffen
.
... Till 100 variables	

Inefficient Memory Utilization

Mảng một chiều 1-3

- JavaScript hỗ trợ hai loại mảng đó như sau:
 - Mảng đơn chiều
- Mảng đa chiều
- Trong một mảng đơn chiều, các phần tử được lưu trữ trong một hàng duy nhất trong bộ nhớ.
- Hình dưới đây cho thấy sự phân bố của mảng đơn chiều.



- Như thể hiện trong hình, các phần tử mảng đầu tiên có số chỉ số không.
- Các phần tử mảng cuối cùng có một số chỉ số một trong ít hơn tổng số của các phần tử.
- Sự sắp xếp này giúp lưu trữ hiệu quả dữ liệu.
- Nó cũng giúp để sắp xếp dữ liệu dễ dàng và theo dõi chiều dài dữ liệu.

Mảng một chiều 2-3

- Biến mảng có thể được tạo ra bằng cách sử dụng đối tượng Array và từ khóa mới cùng với kích thước của phần tử mảng.
- Cú pháp để khai báo và khởi tạo một mảng duy nhất chiều như sau:

```
var variable_name = new Array(size); //Declaration  
variable_name[index] = 'value';
```

Mảng một chiều 3-3

- Ví dụ

```
<script>
```

```
//Declaration using Array Object and then Initialization  
var marital_status = new Array(3);  
marital_status[0] = 'Single';  
marital_status[1] = 'Married';  
marital_status[2] = 'Divorced';
```

```
//Declaration and Initialization  
var marital_status = new  
    Array('Single','Married','Divorced');
```

```
//Declaration and Initialization Without Array  
var marital_status = ['Single','Married','Divorced'];
```

```
</script>
```


Truy cập mảng một chiều 1-4

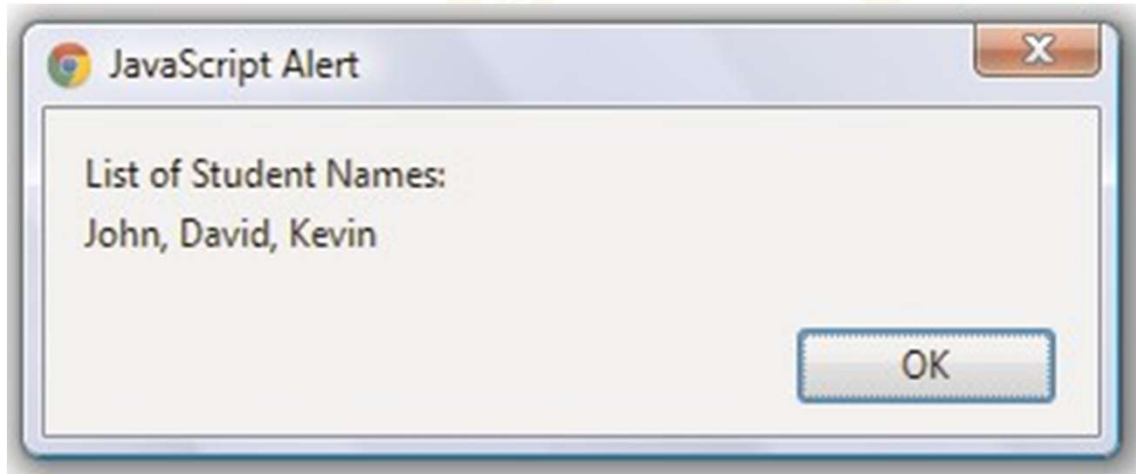
- Các phần tử mảng có thể được truy cập bằng cách sử dụng tên mảng tiếp theo là số chỉ số quy định trong dấu ngoặc vuông.

➤ Truy cập phần tử mảng không dùng vòng lặp

```
<script>
    var names = new Array("John", "David", "Kevin");
    alert('List of Student Names:\n' + names[0] + ',' + '
    ' + names[1] + ',' + ' ' + names[2]);
</script>
```

Truy cập mảng một chiều 2-4

- Kết quả



➤ Truy cập mảng sử dụng vòng lặp

- JavaScript cho phép bạn truy cập các phần tử mảng bằng cách sử dụng các vòng lặp khác nhau.
- Vì vậy, bạn có thể truy cập mỗi phần tử mảng bằng cách đặt một biến đếm của vòng lặp như chỉ số của một phần tử.
- Tuy nhiên, điều này đòi hỏi đếm các phần tử trong một mảng.
- Thuộc tính length có thể được sử dụng để xác định số phần tử trong một mảng.

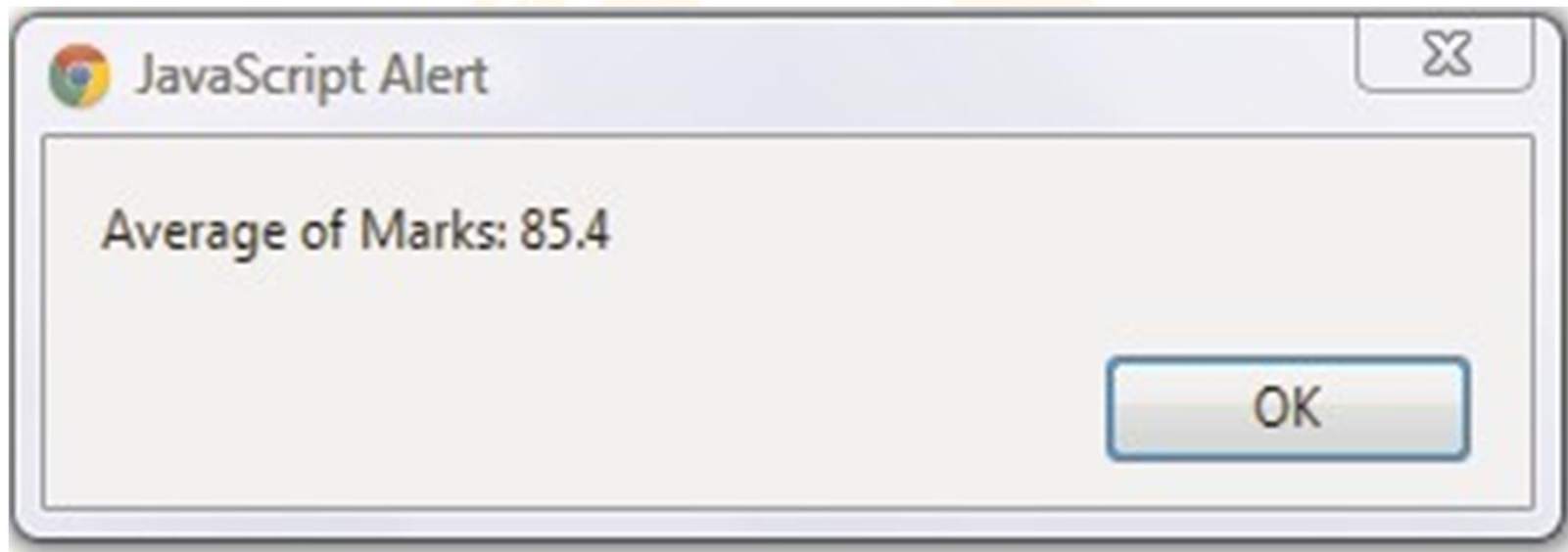
Truy cập mảng một chiều 3-4

- Ví dụ

```
<script>
  var sum = 0;
  var marks = new Array(5);
  for(var i=0; i<marks.length; i++)
  {
    marks[i] = parseInt(prompt('Enter Marks:', ''));
    sum = sum + marks[i];
  }
  alert('Average of Marks: ' + (sum/marks.length));
</script>
```

Truy cập mạng một chiều 4-4

- Kết quả



Mảng đa chiều 1-2

- Một mảng đa chiều lưu trữ sự kết hợp của các giá trị của chỉ một kiểu trong hai hoặc nhiều chiều.
- Một mảng hai chiều được biểu diễn như là các hàng và cột tương tự như của một tờ Microsoft Excel.
- Một mảng hai chiều là một ví dụ về mảng đa chiều.
- Hình dưới đây cho thấy các đại diện của một mảng đa chiều.

Employee Salaries ↓	0	1	2	3
	BASIC	HRA	ALLOWANCE	TOTAL
0	14350	10500	1500	26350
1	34350	4050	1000	39400
2	6150	4500	3250	13900
3	4920	4500	2250	11670
4	12300	9000	2000	23300

- Một mảng hai chiều là một mảng của các mảng.
- Điều này có nghĩa, một mảng hai chiều, đầu tiên là một mảng chính được khai báo và sau đó một mảng được tạo ra cho mỗi phần tử của mảng chính.

Mảng đa chiều 2-2

- Cú pháp

```
var variable_name = new Array(size); //Declaration
```

```
variable_name[index] = new Array('value1','value2'..);
```

- Ví dụ

```
var students = new Array(3);
```

← Declaration

```
students[0] = new Array ('John', '65');
```

```
students[1] = new Array('David', '70');
```

```
students[2] = new Array('Richard', '57');
```

} Initialization

Truy cập mảng hai chiều 1-3

- Mảng đa chiều có thể được truy cập bằng cách sử dụng các chỉ số của biến mảng chính cùng với chỉ số phụ mảng.

➤ Truy cập phần tử không sử dụng vòng lặp

- Code sau tạo ra một mảng hai chiều có hiển thị các chi tiết của nhân viên.

```
<script>
var employees = new Array(3);
employees[0] = new Array('John', '25', 'New Jersey');
employees[1] = new Array('David', '21', 'California');
document.write('<H3> Employee Details </H3>');
document.write('<P><B>Name: </B>' + employees[0][0] +
'</P>');
document.write('<P><B>Location: </B>' + employees[0][2]
+ '</P>');
document.write('<P><B>Name: </B>' + employees[1][0]
+ '</P>');
document.write('<P><B>Location: </B>' + employees[1][2]
+ '</P>');
</script>
```

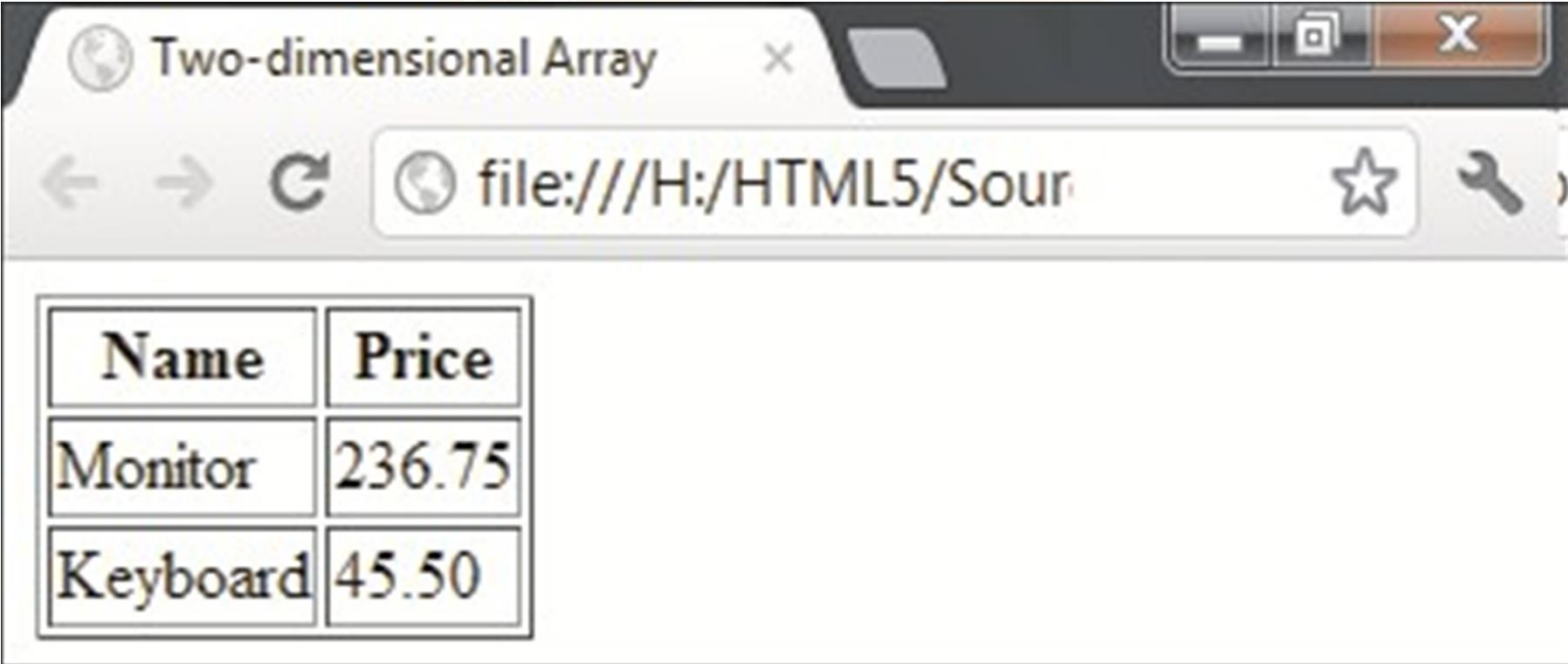
Truy cập mảng hai chiều 2-3

➤ Truy cập sử dụng vòng lặp

```
<script>
var products = new Array(2);
products[0] = new Array('Monitor', '236.75');
products[1] = new Array('Keyboard', '45.50');
document.write('<TABLE border=1><TR><TH>Name</TH>
  <TH>Price</TH></TR>');
for(var i=0; i<products.length; i++)
{
  document.write('<TR>');
  for(var j=0; j<products[i].length; j++)
  {
    document.write('<TD>' + products[i][j] + '</TD>');
  }
  document.write('</TR>');
}
document.write('</TABLE>');
</script>
```


Truy cập mảng hai chiều 4-4

- Kết quả



The screenshot shows a web browser window with the title "Two-dimensional Array". The address bar displays "file:///H:/HTML5/Sour". The main content area contains a table with the following data:

Name	Price
Monitor	236.75
Keyboard	45.50

Các phương thức của mảng 1-3

Một mảng là một tập hợp các giá trị nhóm lại với nhau và xác định bởi một tên duy nhất. Trong JavaScript, các đối tượng Array cho phép bạn tạo ra các mảng.

Nó cung cấp thuộc tính length cho phép bạn để xác định số phần tử trong một mảng.

Các phương thức khác nhau của các đối tượng Array cho phép truy cập và thao tác các phần tử mảng.

- Bảng sau liệt kê một số phương thức của mảng

Tên phương thức	Mô tả
concat	Kết hợp một hoặc nhiều biến mảng.
join	Nối tất cả các phần tử của mảng thành một chuỗi.
pop	Lấy phần tử cuối cùng của mảng.
push	Nối thêm một hoặc nhiều phần tử vào cuối của một mảng.
sort	Sắp xếp các phần tử mảng trong một thứ tự chữ cái.

Các phương thức của mảng 2-3

- Ví dụ

```
<script>
  var flowers = new Array('Rose', 'Sunflower', 'Daisy');
  document.write('Number of flowers: ' + flowers.length +
    '<br/>');
  document.write('Flowers: ' + flowers.join(', ') +
    '<br/>');
  document.write('Orchid and Lily are Added: ' +
    flowers.push("Orchid", "Lily") + '<br/>');
  document.write('Flowers (In Ascending Order): ' +
    flowers.sort() + '<br/>'); document.write('Flowers
  Removed: ' + flowers.pop()
    +'<br/>');
</script>
```

Các phương thức của mảng 3-3

- Kết quả



The screenshot shows a web browser window with the title "Manipulating Array Elements". The address bar displays the file path: `file:///H:/HTML5/Source%20Code/session%2014/cc`. The main content area of the browser displays the following text:

```
Number of flowers: 3  
Flowers: Rose, Sunflower, Daisy  
Orchid and Lily are Added: 5  
Flowers (In Ascending Order): Daisy,Lily,Orchid,Rose,Sunflower  
Flowers Removed: Sunflower
```

Vòng lặp for..in 1-3

Vòng lặp for..in là một phần mở rộng của vòng lặp for. Nó cho phép để thực hiện các hành động cụ thể về các đối tượng của mảng.

Vòng lặp đọc tất cả các phần tử trong mảng được chỉ định và thực thi một khối mã chỉ một lần cho mỗi phần tử trong mảng.

Cú pháp:

```
for (variable_name in array_name)
{
    //statements;
}
```

Vòng lặp for..in 2-3

- Ví dụ

```
<script>
  var books = new Array('Beginning CSS 3.0',
    'Introduction to HTML5', 'HTML5 in Mobile
    Development');

  document.write('<H3> List of Books </H3>');

  for(var i in books)
  {
    document.write(books[i] + '<br/>');
  }
</script>
```

Vòng lặp for..in 3-3

- Kết quả



- Một cấu trúc vòng lặp bao gồm một điều kiện là hướng dẫn các trình biên dịch số lần một khối mã sẽ được thực thi.
- JavaScript hỗ trợ ba loại vòng bao gồm: vòng lặp while, for, và do-while.
- Lệnh break được sử dụng để thoát khỏi vòng lặp mà không kiểm tra các điều kiện quy định.
- Lệnh Continue chấm dứt việc thực hiện lần lặp hiện tại của vòng lặp và tiếp tục lần lặp tiếp theo bằng cách trả lại quyền kiểm soát cho đầu vòng lặp.
- JavaScript hỗ trợ hai loại mảng cụ thể là, mảng đơn chiều và mảng đa chiều?
- Vòng lặp for...in là một phần mở rộng của vòng lặp cho phép thực hiện hành động cụ thể trên các đối tượng của mảng.

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

THANK FOR WATCH !

