



*NexTGen*



*Web*



## Bài 12

# *Giới thiệu về JavaScript*

# Mục tiêu

- Giải thích về scripting
- Giải thích về ngôn ngữ JavaScript
- Giải thích về client-side và server-side JavaScript
- Biến và kiểu dữ liệu trong JavaScript
- Một số phương thức hiển thị thông tin trong JavaScript
- Các ký tự không in ra và hàm xây dựng sẵn trong JavaScript
- Sự kiện và trình điều khiển sự kiện
- Giải thích về jQuery
- Sử dụng jQuery Mobile

# Scripting 1-3

Scripting đề cập đến một loạt các lệnh được diễn giải và thực hiện tuần tự và ngay lập tức khi xảy ra một sự kiện.

Sự kiện này là một hành động được tạo ra bởi một người dùng khi tương tác với một trang web.

Ví dụ về các sự kiện bao gồm nút bấm, lựa chọn một sản phẩm từ một menu...

Ngôn ngữ scripting dùng để chỉ một bộ các hướng dẫn cung cấp một số chức năng khi người dùng tương tác với một trang web.

Các ngôn ngữ Scripting ngôn ngữ được thường được nhúng trong các trang HTML để thay đổi hành vi của các trang web theo yêu cầu của người dùng.

# Scripting 2-3

- Hình dưới đây hiển thị cần thiết cho kịch bản.



# Scripting 3-3

- Có hai loại ngôn ngữ kịch bản. Họ là như sau:

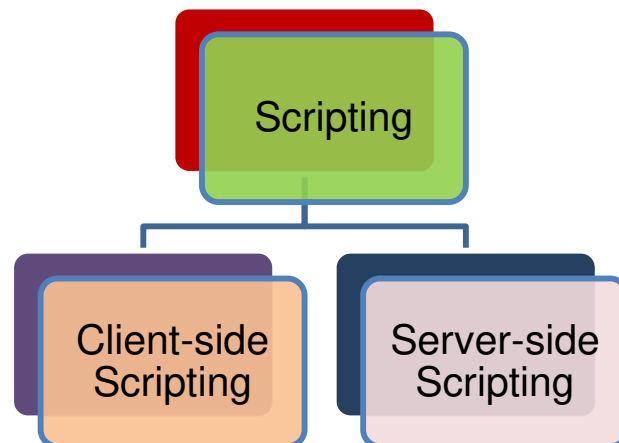
## Client-side Scripting:

- Dùng để chỉ một kịch bản được thực hiện trên máy tính của khách hàng bằng các trình duyệt.

## Server-side Scripting:

- Dùng để chỉ một kịch bản được thực hiện trên một máy chủ Web để tạo ra các trang HTML động.

- Hình dưới đây hiển thị các loại kịch bản.



# JavaScript 1-2

JavaScript là một ngôn ngữ kịch bản cho phép xây dựng các trang web động bằng cách đảm bảo tương tác người dùng tối đa.

JavaScript là một ngôn ngữ dựa trên đối tượng, có nghĩa là nó cung cấp các đối tượng để xác định chức năng.

Trong cuộc sống thực, một đối tượng là một thực thể có thể nhìn thấy như một chiếc xe hoặc cái bảng có một số thuộc tính và hành động nhất định.

Tương tự như vậy, trong một ngôn ngữ kịch bản, một đối tượng có một nhận dạng duy nhất, trạng thái và hành vi.

Danh tính của các đối tượng phân biệt nó với các đối tượng khác cùng loại.

Trạng thái của các đối tượng liên quan đến đặc điểm của nó, trong khi hành vi của các đối tượng bao gồm các hành động có thể của nó.

Các đối tượng lưu danh tính và trạng thái của nó trong các trường (còn gọi là biến) và phơi bày hành vi của mình thông qua chức năng (hành động).

# JavaScript 2-2

- Hình dưới đây sẽ hiển thị một số đối tượng thế giới thực.



Identity: AXA 43 S

State: Color-Red, Wheels-Four

Behavior: Running



Identity: T002

State: Color-Brown

Behavior: Stable

## Objects

# Các phiên bản của JavaScript

- Phiên bản đầu tiên của JavaScript được phát triển bởi Brendan Eich tại Netscape vào năm 1995 và được đặt tên là JavaScript 1.0.
- Bảng dưới đây liệt kê các phiên bản khác nhau của ngôn ngữ JavaScript.

Phiên bản	Mô tả
1.1	Được hỗ trợ từ phiên bản 3.0 của Netscape Navigator và Internet Explorer.
1.2	Được hỗ trợ bởi Internet Explorer từ phiên bản 4.0.
1.3	Được hỗ trợ bởi Internet Explorer từ phiên bản 5.0, Netscape Navigator từ phiên bản 4.0, và Opera từ phiên bản 5.0.
1.4	Được hỗ trợ bởi các máy chủ của Netscape và Opera 6.
1.5	Được hỗ trợ bởi Internet Explorer từ phiên bản 6.0, Netscape Navigator từ phiên bản 6.0, và Mozilla Firefox từ phiên bản 1.0.
1.6	Được hỗ trợ trong phiên bản mới nhất của trình duyệt Internet Explorer và Netscape Navigator. Nó cũng được hỗ trợ bởi trình duyệt Mozilla Firefox từ phiên bản 1.5.
1.7	Được hỗ trợ trong phiên bản mới nhất của trình duyệt Internet Explorer và Netscape Navigator. Nó cũng được hỗ trợ bởi trình duyệt Mozilla Firefox từ phiên bản 2.0.



# Client-side JavaScript 1-2

A Client-side JavaScript (CSJS) được thực hiện bởi trình duyệt trên máy trạm của người dùng.

A client-side script có thể chứa các lệnh cho trình duyệt để xử lý tương tác người dùng.

Những lệnh này có thể là để thay đổi giao diện hoặc nội dung của trang web dựa trên các yếu tố đầu vào người sử dụng.

Ví dụ như hiển thị một trang chào mừng với tên người dùng, ngày và thời gian hiển thị, xác nhận rằng các chi tiết người sử dụng cần được điền vào...

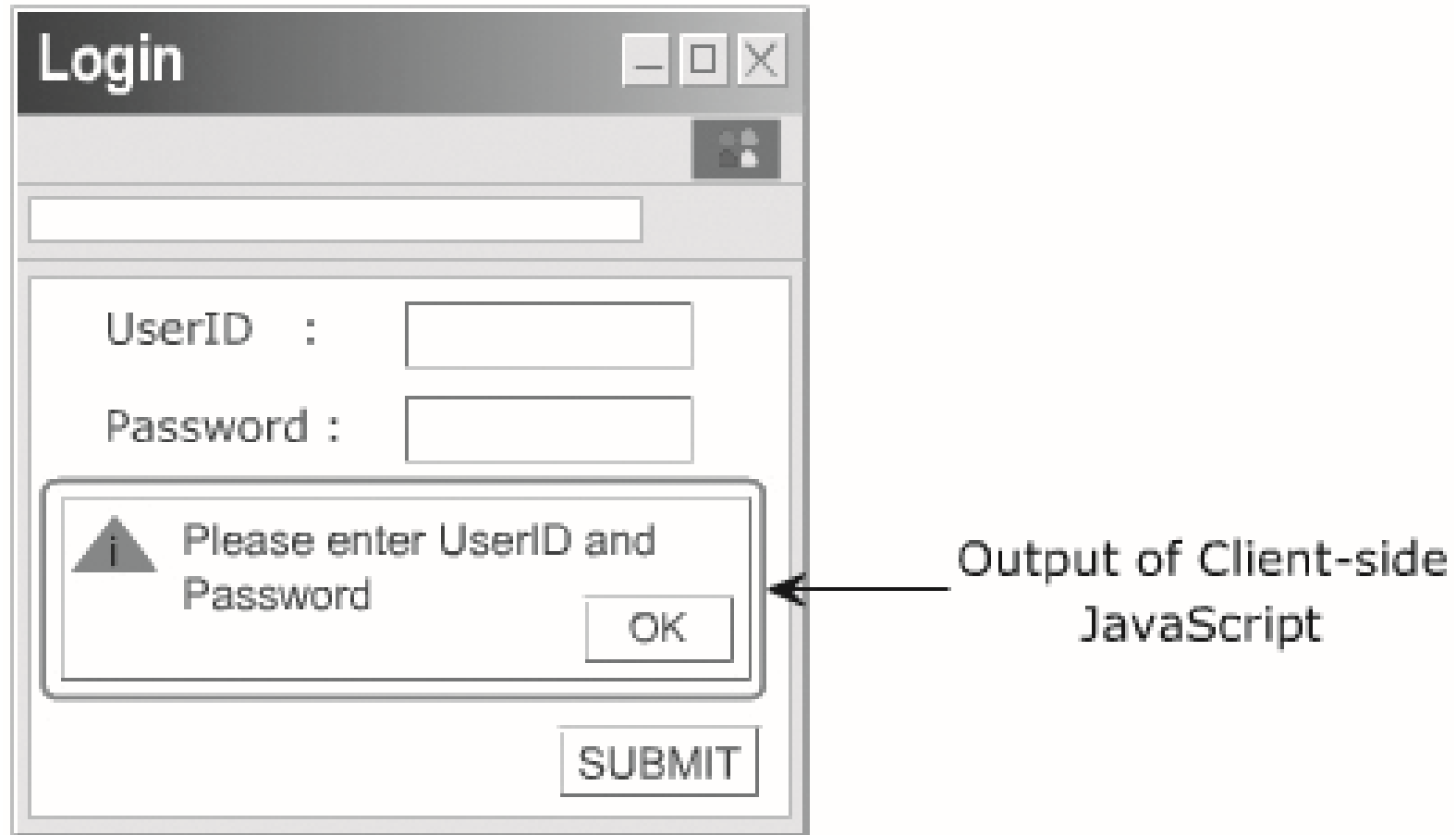
A JavaScript có thể nhúng trong một trang HTML hoặc tạo riêng tại một tập tin, mà sẽ được lưu với phần mở rộng .js

Trong client-side script, khi HTML được yêu cầu, máy chủ Web sẽ gửi tất cả các tập tin cần thiết vào máy tính của người dùng.

Trình duyệt web thực thi kịch bản và hiển thị các trang HTML cho người dùng cùng với bất kỳ kết quả hữu hình của kịch bản.

# Client-side JavaScript 2-2

- Hình dưới đây sẽ hiển thị đầu ra của một JavaScript phía máy khách.



# Server-side JavaScript 1-2

A Server-side JavaScript được thực hiện bởi các máy chủ khi một trang HTML được yêu cầu của người dùng và đầu ra được hiển thị bởi trình duyệt.

A server-side JavaScript có thể tương tác với cơ sở dữ liệu, lấy thông tin cần thiết và hiển thị nó cho người dùng.

Server-side scripting đáp ứng mục tiêu cung cấp nội dung động trong các trang web.

Không giống client-side JavaScript, Trang HTML sử dụng server-side JavaScript được biên dịch vào các tập tin bytecode trên máy chủ.

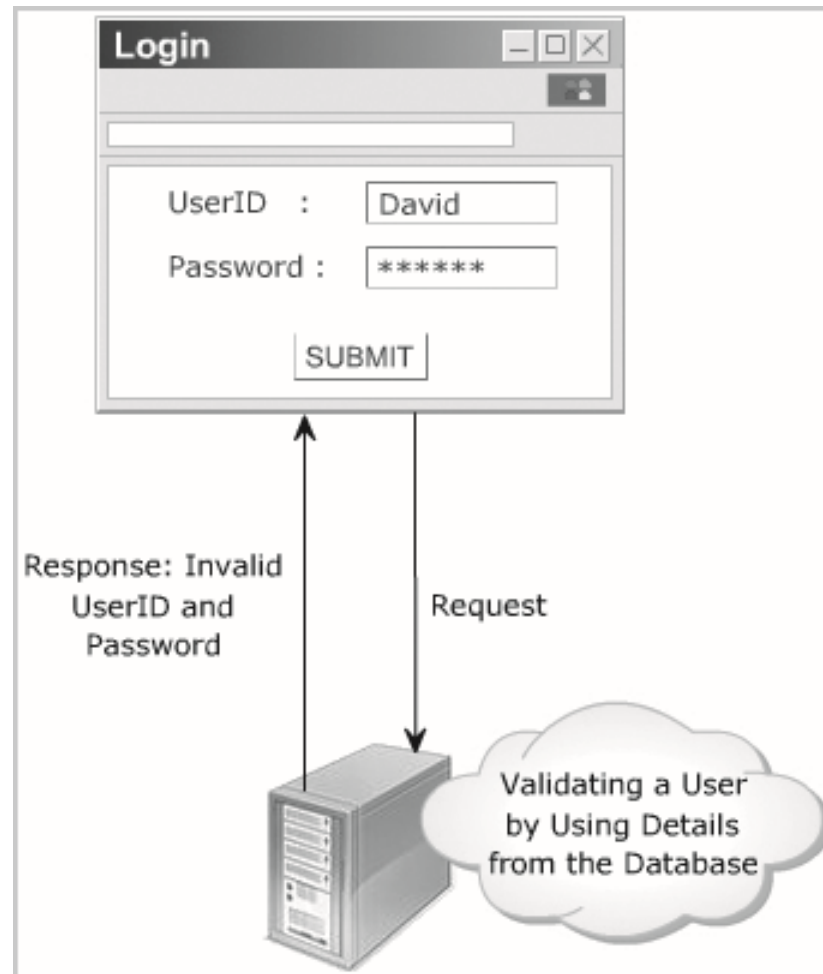
A JavaScript có thể nhúng trong một trang HTML hoặc viết riêng tại một tập tin, mà sẽ được lưu với phần mở rộng .js

Biên dịch là một quá trình chuyển đổi mã vào mã máy độc lập.

Mã máy độc lập này được gọi là bytecode, mà là một tập tin thực thi mà máy chủ Web chạy để tạo ra các đầu ra mong muốn.

# Server-side JavaScript 2-2

- Hình dưới đây sẽ hiển thị đầu ra của một JavaScript phía máy khách.



# Thẻ <Script> 1-2

Thẻ <script> xác định một kịch bản cho một trang HTML để làm cho chúng tương tác.

Trình duyệt có hỗ trợ thông dịch và thực thi các kịch bản ở trong thẻ <script> khi tải các trang trong trình duyệt.

Bạn có thể trực tiếp chèn mã JavaScript dưới thẻ <script>.

Bạn có thể xác định nhiều từ khóa script trong các thẻ <head> hoặc trong phần tử <body> của một trang HTML.

Trong HTML5, thuộc tính type xác định ngôn ngữ kịch bản không còn cần thiết vì nó là tùy chọn.

# Thẻ <Script> 2-2

- Ví dụ

```
<!doctype html>
<html>
  <head>
    <script>
      document.write("Welcome to the Digital World");
    </script>
  </head>
  <body>
    ....
  </body>
</html>
```

Có hai mục đích chính của thẻ <script> định như sau:

Xác định một phân khúc nhất định kịch bản trong trang HTML.

Tải một tập tin kịch bản bên ngoài.

# Các biến trong JavaScript

Một biến dùng để chỉ một tên biểu tượng chứa một giá trị mà có thể sẽ thay đổi.

Ví dụ, age của một học sinh và salary của một nhân viên có thể được coi là các biến.

Trong JavaScript, một biến là một vị trí duy nhất trong bộ nhớ máy tính của để lưu trữ một giá trị và có một tên duy nhất.

Tên của biến được sử dụng để truy cập và đọc các giá trị được lưu trữ trong đó.

Một biến có thể lưu trữ các loại dữ liệu khác nhau chẳng hạn như một nhân vật, một số, hoặc một chuỗi.

# Khai báo các biến 1-4

- Khai báo một biến đề cập đến việc tạo ra một biến bằng cách xác định tên biến.
- Ví dụ, người ta có thể tạo ra một biến có tên để lưu trữ tên của một học sinh.

## Trong JavaScript

từ khóa var được sử dụng để tạo ra một biến bằng cách cấp phát bộ nhớ cho nó.

một từ khóa là một từ dành riêng chứa một ý nghĩa đặc biệt.

biến có thể được khởi tạo tại thời điểm tạo ra các biến hoặc sau đó.

khởi tạo là gán giá trị cho một biến.

một khi biến được khởi tạo, bạn có thể thay đổi giá trị của một biến theo yêu cầu.

biến cho phép theo dõi các dữ liệu trong quá trình thực của kịch bản.

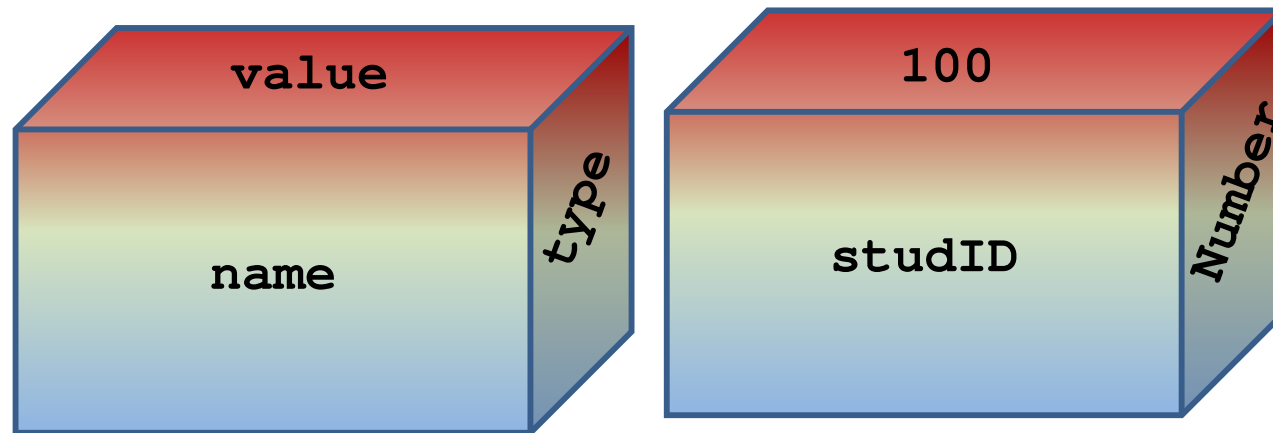
trong khi đề cập đến một biến, bạn đang đề cập đến giá trị của biến đó.

người ta có thể khai báo và khởi tạo nhiều biến trong một câu lệnh đơn.



# Khai báo các biến 2-4

- Hình dưới đây sẽ hiển thị như thế nào để khai báo các biến.



- Sau cú pháp chứng minh làm thế nào để khai báo các biến trong JavaScript.

**Cú pháp:**

```
var <variableName>;
```

# Khai báo các biến 3-4

- Sau cú pháp chứng minh làm thế nào để khởi tạo các biến trong JavaScript.

## Cú pháp:

```
<variableName> = <value>;
```

- Cú pháp trình bày cách khai báo và khởi tạo nhiều biến trong một câu lệnh duy nhất, được phân cách bởi dấu phẩy.

## Cú pháp:

```
var <variableName1> = <value1>, <variableName2> = <value2>;
```

# Khai báo các biến 4-4

- Mã số câu liên quan khai báo hai biến cụ thể là, studID và studName và gán giá trị cho họ.

```
var studID;  
var studName;  
studID = 50;  
  
studName = "David Fernando";
```

- Đoạn mã gán giá trị cho biến studID và studName bằng cách sử dụng toán tử gán (=).
- Giá trị tên là David Fernando được quy định trong dấu ngoặc kép.
- Mã số câu liên quan chứng minh làm thế nào để khai báo và khởi tạo nhiều biến trong một tuyên bố duy nhất trong JavaScript.

```
var studName = David, studAge = 15;
```

# Quy tắc đặt tên biến

- JavaScript là một ngôn ngữ phân biệt chữ hoa và chữ thường.
- Các biến X và x được coi như là hai biến khác nhau.
- JavaScript bao gồm quy tắc nhất định để đặt tên một biến như sau:

Trong JavaScript, một tên biến

có thể bao gồm chữ số, dấu gạch dưới, và các ký tự chữ cái.

phải bắt đầu bằng một chữ cái hoặc ký tự gạch dưới.

không thể bắt đầu với một số và không thể chứa bất kỳ dấu chấm câu.

không thể chứa bất kỳ loại ký tự đặc biệt như +, \*, %, vv.

không thể chứa khoảng trắng.

không thể là một từ khóa JavaScript.

# Các kiểu dữ liệu trong JavaScript 1-3

- Để xác định loại dữ liệu có thể được lưu trữ trong một biến, JavaScript cung cấp các loại dữ liệu khác nhau.
- Các kiểu dữ liệu trong JavaScript được phân thành hai loại lớn cụ thể là, các kiểu dữ liệu nguyên thủy và kiểu tổng hợp.
- Kiểu dữ liệu chỉ chứa một giá trị duy nhất, trong khi các kiểu dữ liệu tổng hợp có chứa một nhóm các giá trị.

## ➤ PRIMITIVE DATA TYPES

- Một kiểu dữ liệu nguyên thủy có chứa một giá trị cụ thể duy nhất như một số hoặc một chuỗi.
- Một chữ là một giá trị tĩnh mà bạn có thể gán cho biến.

# Các kiểu dữ liệu trong JavaScript 2-3

- Bảng sau liệt kê các kiểu dữ liệu nguyên thủy.

Kiểu dữ liệu cơ bản	Mô tả
Boolean	Chỉ chứa hai giá trị cụ thể là, đúng hay sai
null	Chỉ chứa một giá trị cụ thể là null, null là một từ khóa.
number	Bao gồm số âm, dương, và thập phân. Một số ví dụ hợp lệ bao gồm 6, 7,5, -8, 7.5e-3, vv
string	Bao gồm giá trị chuỗi đặt trong cặp nháy kép.

# Các kiểu dữ liệu trong JavaScript 3-3

## ➤ COMPOSITE DATA TYPES

- Một kiểu dữ liệu tổng hợp lưu trữ một bộ sưu tập của nhiều giá trị có liên quan, không giống như các kiểu dữ liệu nguyên thủy.
- Trong JavaScript, tất cả các loại dữ liệu tổng hợp đang được coi là đối tượng.
- Một kiểu dữ liệu tổng hợp có thể là được xác định trước hoặc người dùng định nghĩa trong JavaScript.
- Bảng sau liệt kê các kiểu dữ liệu composite.

Kiểu tổng hợp	Mô tả
Objects	Kiểu đối tượng trong JavaScript
Functions	Kiểu hàm
Arrays	Kiểu mảng

# Phương thức 1-3

- JavaScript cho phép bạn hiển thị thông tin bằng cách sử dụng phương thức của đối tượng document.
- Đối tượng document là một đối tượng được xác định trước trong JavaScript, đại diện cho trang HTML và cho phép quản lý trang động.
- Mỗi đối tượng trong JavaScript bao gồm các phương thức, mà thực hiện một nhiệm vụ cụ thể.
- Có hai phương thức của đối tượng document hiển thị bất kỳ loại dữ liệu trong trình duyệt.
- Những phương thức này như sau:
  - `write()`: Hiển thị bất kỳ dữ liệu gì.
  - `writeln()`: Hiển thị bất kỳ dữ liệu gì sau đó xuống dòng mới.

## Cú pháp:

```
document.write("<data>" + variables);
```



# Phương thức 2-3

- Cú pháp của `document.writeln()`

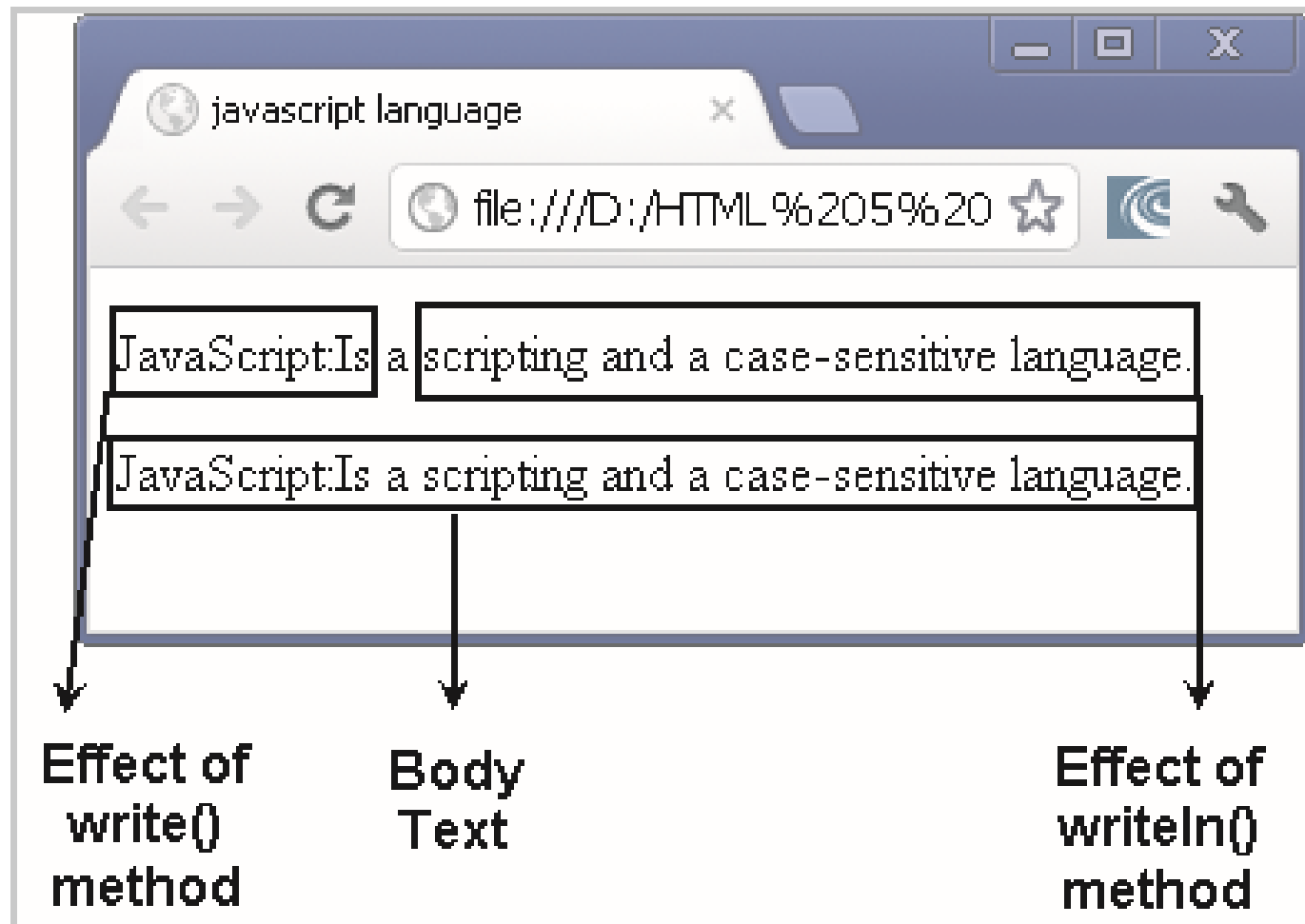
## Cú pháp:

```
document.writeln("<data>" + variables);
```

- Ví dụ của phương thức `write()`.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> JavaScript language </title>
    <script>
      document.write("<p> JavaScript:");
      document.writeln("is a scripting");
      document.write("and a case-sensitive language.");
    </script>
  </head>
  <p>
    JavaScript: is a scripting and a case-sensitive language.
  </p>
</html>
```

# Phương thức 3-3



# Sử dụng chú thích

- Chú thích cung cấp thông tin về một đoạn mã trong kịch bản.
- Khi kịch bản được thực thi, trình duyệt xác định chú thích vì chúng được đánh dấu bằng ký tự đặc biệt và không hiển thị chúng.
- JavaScript hỗ trợ hai loại chú thích . Đây là như sau:

## ➤ Chú thích một dòng

```
// This statement declares a variable named num.  
var num;
```

## ➤ Chú thích nhiều dòng

```
/* This line of code  
   declares a variable */  
var num;
```

# Các ký tự không in ra 1-2

- Một ký tự tự thoát ra là một ký tự đặc biệt là trước bởi một dấu gạch chéo ngược (\).
- Các ký tự thoát được sử dụng để hiển thị các ký tự không in ra như ký tự tab, ký tự space hay backspace.
- Trong JavaScript, các ký tự thoát phải được đặt trong dấu ngoặc kép.
- Bảng sau liệt kê các nhân vật dãy thoát.

Ký tự thoát	Mô tả
\b	Back space
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\'	Single quote
\"	Double quote

# Các ký tự không in ra 2-2

Ký tự thoát	Mô tả
\\	Backslash
\\aaa	Matches a Latin-1 encoding character using octal representation, where aaa are three octal numbers. For example, \251 represents the copyright symbol
\\xaa	Matches a Latin-1 encoding character using hexadecimal representation, where aa are two hexadecimal numbers. For example, \x61 represents the character 'a'
\\uaaaa	Represent the Unicode encoding character, where aaaa are four hexadecimal numbers. For example, the character \u0020 represents a space

- Ví dụ.

```
<script>
  document.write("You need to have a \u0022credit card\u0022, if
                 you want to shop on the \'Internet\'.");
</script>
```

# Các hàm có sẵn 1-4

- Một hàm là một đoạn mã mà thực hiện một số hoạt động trên các biến để hoàn thành một nhiệm vụ cụ thể.
- Nó cần một hoặc nhiều giá trị đầu vào, xử lý chúng, và trả về một giá trị đầu ra.
- Bảng sau liệt kê các hàm được xây dựng trong JavaScript

Hàm	Mô tả	Ví dụ
<code>alert()</code>	Hiển thị một hộp thoại với một số thông tin và nút OK	<code>alert("Please fill all the fields of the form");</code>
<code>confirm()</code>	Hiển thị một hộp thoại với nút OK và Cancel. Đây xác thực một hành động, mà người dùng muốn thực hiện	<code>confirm("Are you sure you want to close the page?");</code>
<code>parseInt()</code>	Chuyển đổi một giá trị chuỗi thành một giá trị số	<code>parseInt("25 years");</code>
<code>parseFloat()</code>	Chuyển đổi một chuỗi thành một số với dấu thập phân	<code>parseFloat("10.33");</code> Returns 10.33

# Các hàm có sẵn 2-4

Hàm	Mô tả	Ví dụ
<code>eval()</code>	Tính toán một biểu thức và trả về kết quả.	<code>eval("2+2");</code> Returns 4
<code>isNaN()</code>	Kiểm tra xem một giá trị không phải là số	<code>isNaN("Hello");</code> Returns true
<code>prompt()</code>	Hiển thị một hộp thoại mà chấp nhận một giá trị đầu vào thông qua một hộp văn bản. Nó cũng chấp nhận giá trị mặc định cho hộp văn bản	<code>prompt("Enter your name", "Name");</code>

- Ví dụ.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> JavaScript language </title>
    <script>
      var value = "";
```

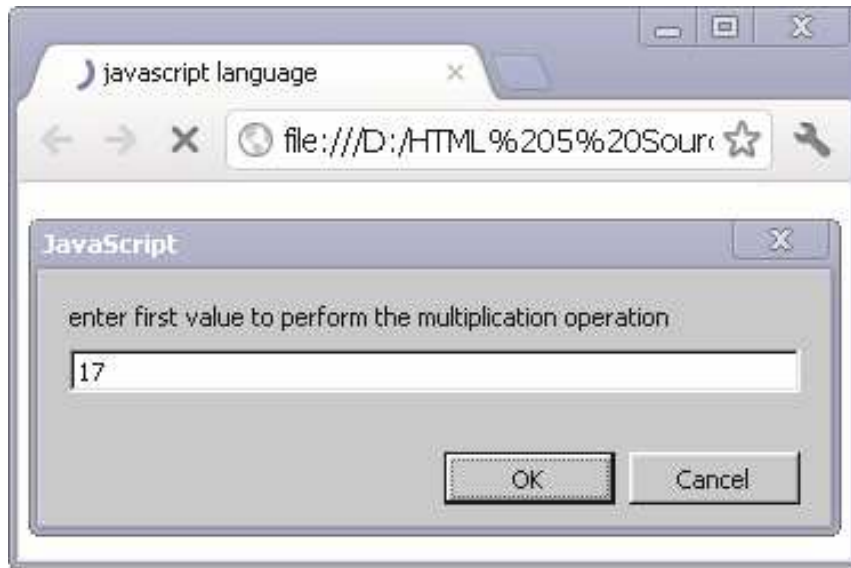
# Các hàm có sẵn 3-4

```
var numone = prompt("enter first value to perform the  
multiplication operation", value);  
  
var numtwo = prompt("enter second value to perform the  
multiplication operation", value);  
var result = eval(numone * numtwo);  
document.write("The result of multiplying: " + numone +  
"and " + numtwo + " is: " + result + "." );  
  
</script>  
</head>  
</html>
```

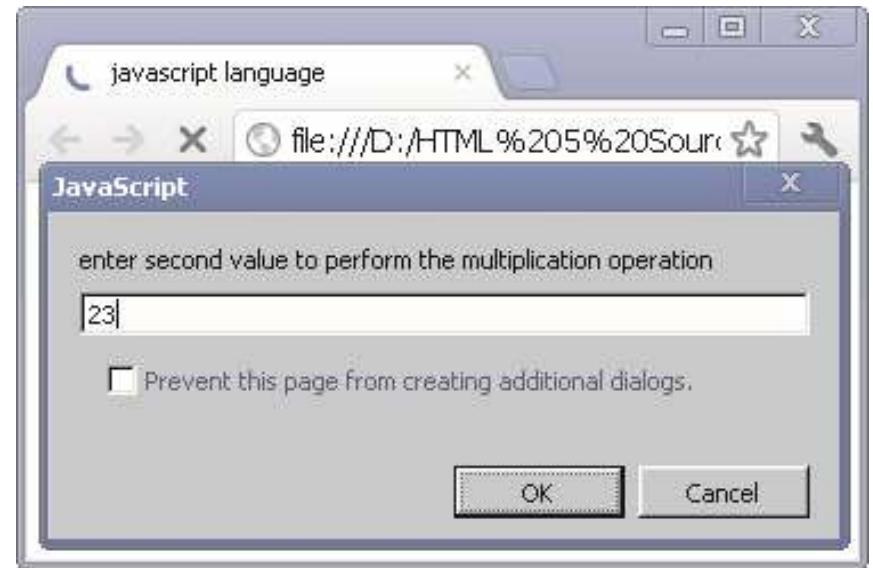


# Các hàm có sẵn 4-4

- Hình dưới đây sẽ hiển thị số đầu tiên đầu vào.



- Hình dưới đây sẽ hiển thị số thứ hai đầu vào.



- Hình dưới đây sẽ hiển thị kết quả.



# Các sự kiện

- Một sự kiện xảy ra khi người dùng tương tác với các trang web.
- Một số sự kiện thường được tạo ra là cú click chuột, gõ phím,..
- Quá trình xử lý các sự kiện này được gọi là xử lý sự kiện.
- Hình dưới đây sẽ hiển thị các sự kiện.



# Điều khiển sự kiện 1-2

Xử lý sự kiện là một quá trình xác định các hành động được thực hiện khi sự kiện xảy ra. Đây được thực hiện bằng cách sử dụng một trình xử lý sự kiện.

Xử lý sự kiện là một mã kịch bản hay một chức năng xác định các hành động được thực hiện khi sự kiện này được kích hoạt.

Khi một sự kiện xảy ra, một hàm xử lý sự kiện được liên kết với các sự kiện cụ thể được gọi.

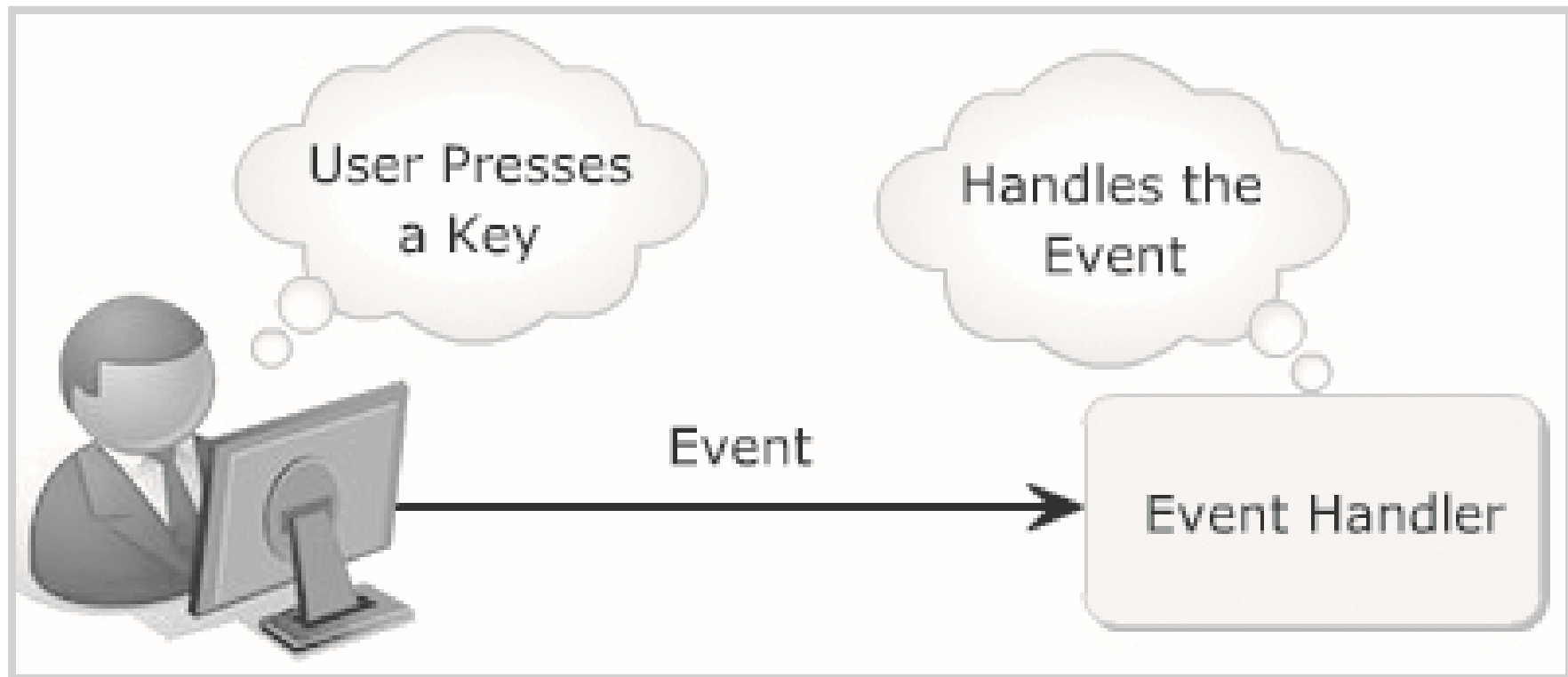
Các thông tin về sự kiện này được tạo cập nhật trên các đối tượng sự kiện.

Các đối tượng sự kiện là một đối tượng được xây dựng sẵn, có thể được truy cập thông qua các đối tượng cửa sổ.

Nó xác định trạng thái sự kiện, bao gồm các thông tin như vị trí của con trỏ chuột, phần tử mà một sự kiện xảy ra trên nó, và trạng thái của các phím trong bàn phím.

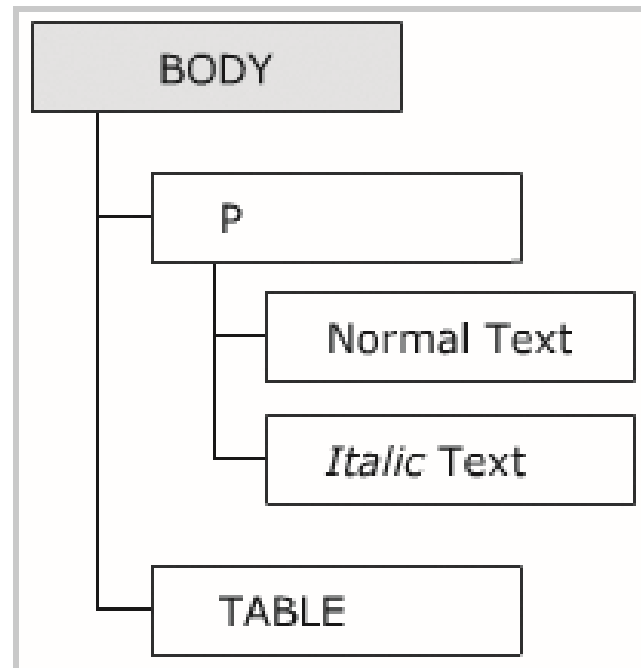
# Điều khiển sự kiện 2-2

- Ví dụ



# Sự kiện nổi lên

- Sự kiện nổi lên là một cơ chế cho phép người dùng để xác định một xử lý sự kiện chung cho tất cả các phần tử con.
- Điều này có nghĩa rằng các phần tử cha mẹ xử lý tất cả các sự kiện được tạo ra bởi các phần tử con.
- Ví dụ, hãy xem xét một trang web bao gồm một đoạn văn và một bảng. Đoạn bao gồm nhiều lần xuất hiện của văn bản in nghiêng.
- Để thay đổi màu sắc của mỗi văn bản nghiêng của một đoạn khi người dùng nhấn một nút đặc biệt, thay vì tuyên bố một xử lý sự kiện cho mỗi văn bản in nghiêng, có thể khai báo trong phần tử <P>.
- Hình dưới đây hiển thị các sự kiện nổi bọt.



# Vòng đời của một sự kiện

- Vòng đời của một sự kiện bắt đầu khi người dùng thực hiện một hành động tương tác với các trang web.
- Nó cuối cùng đã kết thúc khi xử lý sự kiện cung cấp một phản ứng với hành động của người dùng.
- Các bước liên quan trong vòng đời của một sự kiện như sau:

1. Người dùng thực hiện một hành động để phát một sự kiện.

2. Các đối tượng sự kiện được cập nhật để xác định trạng thái sự kiện.

3. Sự kiện lóe lên.

4. Bọt sự kiện xảy ra như sự kiện bong bóng thông qua các yếu tố của hệ thống phân cấp.

5. Xử lý sự kiện được gọi là thực hiện các hành động cụ thể.

# Các sự kiện về bàn phím 1-2

- Sự kiện bàn phím là những sự kiện xảy ra khi một phím hoặc tổ hợp phím được nhấn hoặc phát hành từ bàn phím.
- Những sự kiện này xảy ra cho tất cả các phím của bàn phím.
- Các sự kiện bàn phím khác nhau như sau:

## Onkeydown

- Xảy ra khi một phím được nhấn xuống.

## Onkeyup

- Xảy ra khi phím được nhả ra.

## Onkeypress

- Xảy ra khi một phím được nhấn và nhả.

# Các sự kiện về bàn phím 2-2

- Ví dụ

```
function numericonly()
{
    if(!event.keyCode >=48 && event.keyCode<=57))
        event.returnValue=false;
}

function countWords()
{
    var message = document.getElementById('txtMessage').value;
    message= message.replace(/\s+/g, ' ');
    var numberOfWords = message.split(' ').length;
    document.getElementById('txtTrack').value = words
    Remaining:  ` + eval(50 - numberOfWords);
    if(numberOfWords > 50)
        alert("too many words.");
}
```



# Các sự kiện về chuột 1-4

- Các sự kiện chuột xảy ra khi người dùng nhấp vào nút chuột.
- Bảng sau liệt kê các sự kiện chuột.

Sự kiện	Mô tả
<code>onmousedown</code>	Xảy ra khi nút chuột được nhấn
<code>onmouseup</code>	Xảy ra khi chuột được thả
<code>onclick</code>	Xảy ra khi nút chuột được nhấn và thả
<code>ondblclick</code>	Xảy ra khi nút chuột là kích đúp
<code>onmousemove</code>	Xảy ra khi con trỏ chuột được di chuyển từ một địa điểm khác
<code>onmouseover</code>	Xảy ra khi con trỏ chuột di chuyển trên phần tử
<code>onmouseout</code>	Xảy ra khi con trỏ chuột được di chuyển ra khỏi các phần tử

# Các sự kiện về chuột 2-4

- Ví dụ

```
<tr>
  <td> Arrival Date: </td>
  <td> <input id="txtArrival" type="text" /></td>
</tr>
<!DOCTYPE HTML>
<html>
  <tr>
    <head>
      <td> Departure Date: </td>
      <title> Reservation </title>
      <td> <input id="txtDeparture" type="text" /></td>
      <script src="form.js">
    </tr>
    </script>
  </head>
  <body>
    <td> Number of Person: </td>
    <td> <input id="txtPerson" type="text" maxlength="3"
      <td> Hotel Reservation Form</td>
      <form id="frmreservation">
    </tr>
    <table>
      <tr>
        <td>
          <td>
            <img id="imgSubmit" width="120px" height="30px"
              <label for="txtName">Name:</label>
            src="submit.jpg" alt="Submit",
          <td>
            onmousedown="showImage(this,
              <input id="txtName" type="text" />submitdown.jpg');"
            <td>
              onmouseup="showImage(this, 'submit.jpg');",
            </tr>
            onclick="frmReservation.submit();" />
          </td>
        </td>
      </tr>
    </table>
  </body>
</html>
```

# Các sự kiện về chuột 3-4

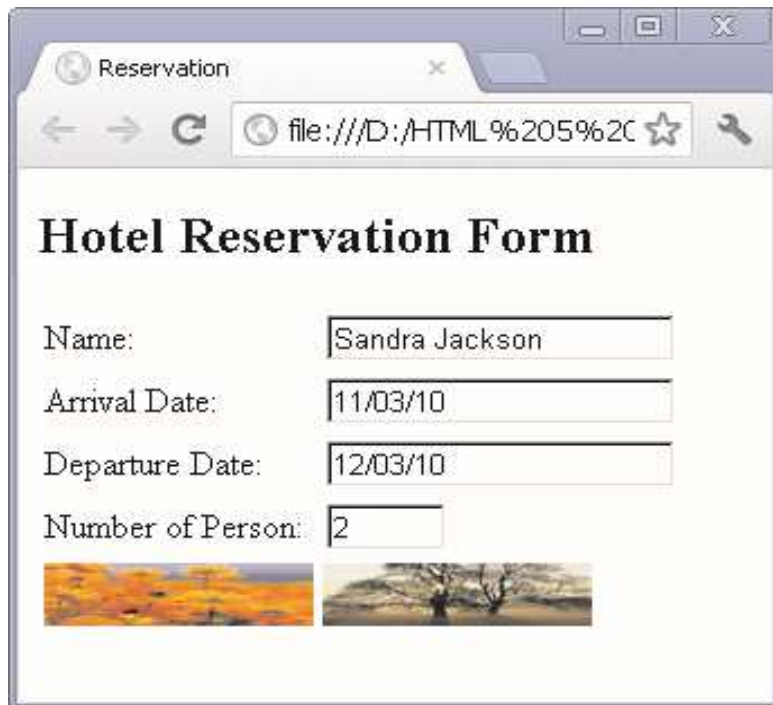
```
<td>
  
</td>
</tr>
</table>
</form>
</body>
</html>
```

# Các sự kiện về chuột 4-4

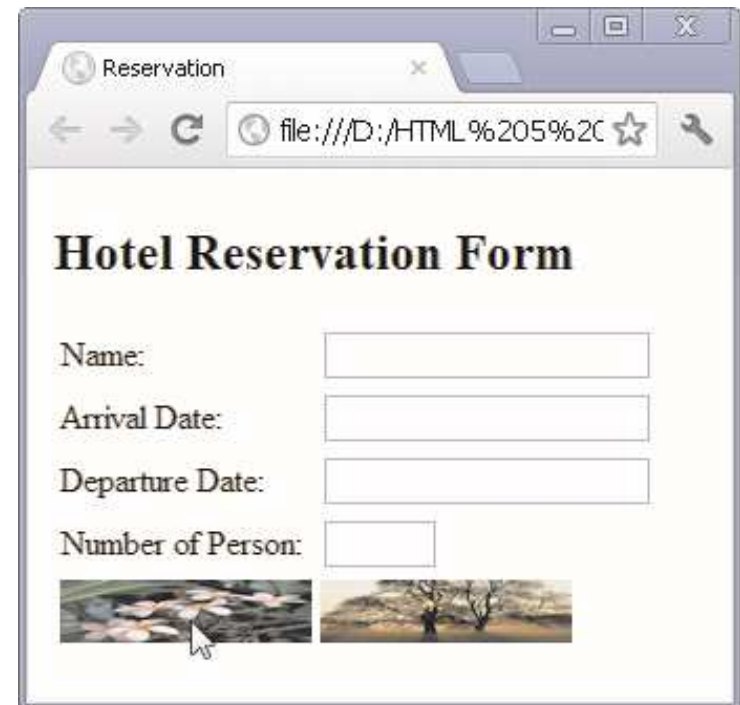
- Mã số câu liên quan chứng minh việc tải các hình ảnh trong một tập tin JavaScript.

```
function showImage(object,url)
{
    object.src=url;
}
```

- Hình dưới đây sẽ hiển thị đầu ra của mouseup.



- Hình dưới đây sẽ hiển thị kết quả trên mousedown.



# Các sự kiện Focus và Selection 1-4

- Các sự kiện focus xác định khi các phần tử được kích hoạt hoặc nhận con trỏ.
- Các sự kiện selection xảy ra khi lựa chọn một thành phần hoặc một phần của một phần tử trong một trang Web được chọn.
- Bảng sau liệt kê các sự focus và selection.

Sự kiện	Mô tả
<code>onfocus</code>	Xảy ra khi một phần tử nhận được focus
<code>onblur</code>	Xảy ra khi một phần tử mất focus
<code>onselectstart</code>	Xảy ra khi lựa chọn của một phần tử bắt đầu
<code>onselect</code>	Xảy ra khi lựa chọn hiện tại thay đổi
<code>ondragstart</code>	Xảy ra khi các phần được chọn được di chuyển

# Các sự kiện Focus và Selection 2-4

- Ví dụ

```
</head>
<body>
  <!DOCTYPE HTML>
  <html>Feedback Form</html>
  <form id="frmreservation">
    <table>
      <thead>
        <tr>
          <th>Reservation </th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>
            <label for="txtName">Name:</label></td>
            <td>
              <input id="txtName" type="text"
                onfocus="showStyle(this);"
                onblur="hideStyle(this);"
                onselect=setFontStyle(this); />
            </td>
          </tr>
          <tr>
            <td colspan="2">field.style.backgroundColor = `#FFFFFF`;</td>
          </tr>
          <tr>
            <td>
              <label for="txtEmail">E-mail:</label></td>
            <td>
              <input id="txtEmail" type="text"
                onfocus="showStyle(this);"
                onblur="hideStyle(this);"
                onselect=setFontStyle(this); />
            </td>
          </tr>
        </tbody>
      </table>
    </form>
  </body>
</html>
```

# Các sự kiện Focus và Selection 3-4

```
<tr>
  <td> <label for="txtComment">Comment:</label></td>
  <td> <textarea id="txtComment" cols="15" rows="3"
    onfocus="showStyle(this);"
    onblur="hideStyle(this);"
    onselect=setFontStyle(this);> </textarea>
  </td>
</tr>
<tr>
  <td> <input id="btnSubmit" type="button" type="button"
    value="Submit" /></td>
  <td> <input id="btnReset" type="reset" /></td>
</tr>
</table>
</form>
</body>
</html>
```

# Các sự kiện Focus và Selection 4-4

- Kết quả



The screenshot shows a web browser window titled "Reservation". The address bar displays the file path: `file:///D:/HTML%205%20Source%20Co`. The main content area features a "Feedback Form" with the following elements:

- Name:** A text input field containing the text "David".
- E-mail:** A text input field containing the text "david@abc.com".
- Comment:** A text area containing the text "Please provide your feedback to".
- Buttons:** Two buttons labeled "Submit" and "Reset" are positioned at the bottom of the form.



# jQuery 1-2

- jQuery là một thư viện JavaScript ngắn và nhanh chóng được phát triển bởi John Resig vào năm 2006 với một khẩu hiệu tuyệt vời: Viết ít hơn và làm nhiều hơn nữa.
- jQuery đơn giản hoá client-side script, tạo hoạt ảnh cho tập tin HTML, xử lý sự kiện, đi lại, và phát triển các ứng dụng web dựa trên AJAX.
- Nó giúp trong việc phát triển ứng dụng Web nhanh chóng bằng cách viết mã ít hơn.
- Sau đây là các tính năng chính được hỗ trợ bởi jQuery.

## Event Handling

- jQuery có một cách thông minh để nắm bắt một loạt các sự kiện, chẳng hạn như người dùng nhấp vào một liên kết, mà không làm cho phức tạp HTML code với xử lý sự kiện.

## Animations

- jQuery có khá nhiều các hiệu ứng hoạt hình mà người dùng có thể sử dụng trong khi xây dựng các trang web của họ.

## DOM Manipulation

- jQuery dễ dàng lựa chọn, đi qua, và sửa đổi DOM bằng cách sử dụng các trình duyệt bộ máy mã nguồn mở có tên sizzle.

# jQuery 2-2

## Cross Browser Support

- jQuery có hỗ trợ cho qua trình duyệt và hoạt động tốt với các trình duyệt sau:
- Internet Explorer 6 trở lên
- Firefox 2.0 trở lên
- Safari 3.0 trở lên
- Cơ rô-m
- Opera 9.0 trở lên

## Lightweight

- jQuery có một thư viện nhẹ 19 KB.

## AJAX Support

- jQuery giúp bạn phát triển các trang web tính năng phong phú và đáp ứng bằng cách sử dụng công nghệ AJAX.

## Latest Technology

- jQuery hỗ trợ cú pháp XPath và CSS3 selectors cơ bản.

# Sử dụng thư viện Query

- Để làm việc với jQuery thực hiện các bước sau:

1. Tải về thư viện jQuery từ trang web <http://jquery.com/>.

2. Đặt tập tin jquery-1.7.2.min.js trong thư mục hiện tại của trang web.

- Người sử dụng có thể bao gồm thư viện jQuery trong tập tin của họ.
- Ví dụ.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>The jQuery Example</title>
    // Using jQuery library
    <script src="jquery-1.7.2.min.js">
      // The user can add our JavaScript code here
    </script>
  </head>
  <body>
  </body>
</html>
```

# Gọi các hàm thư viện jQuery 1-2

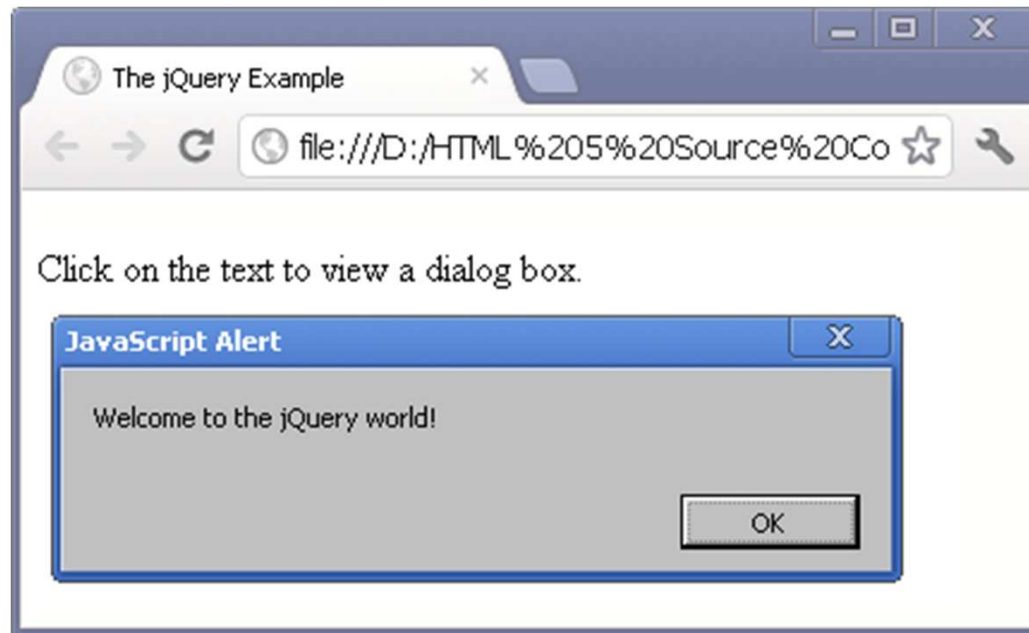
- Trong khi jQuery được đọc hoặc thao tác đối tượng Document Object Model (DOM), người dùng có thể thêm các sự kiện khi các đối tượng DOM đã sẵn sàng.
- Nếu người dùng muốn các sự kiện trên trang của họ sau đó người dùng để gọi sự kiện này trong hàm `$(document).ready()`.
- Người sử dụng cũng đăng ký các sự kiện sẵn sàng cho các tài liệu.
- Đặt tập tin jquery-1.7.2.min.js trong thư mục hiện tại và xác định vị trí của tập tin này trong thuộc tính src.
- Mã số câu liên quan cho thấy cách gọi hàm thư viện jQuery và sự kiện đã sẵn sàng trong DOM.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>The jQuery Example</title>
    <script src=" jquery-1.7.2.min.js">
    </script>
    <script>
      $(document).ready(function() {
        $("div").click(function() {
          alert("Welcome to the jQuery world!");
        });
      });
    </script>
```

# Gọi các hàm thư viện jQuery 2-2

```
</head>
<body>
  <div id="firstdiv">
    Click on the text to view a dialog box.
  </div>
</body>
</html>
```

- Kết quả



# jQuery Mobile 1-6

- jQuery mobile là một framework phát triển Web giao diện người dùng (UI) cho phép người sử dụng để xây dựng các ứng dụng web di động mà làm việc trên máy tính bảng và điện thoại thông minh.
- Khuôn khổ jQuery mobile cung cấp nhiều tiện nghi bao gồm XML DOM và thao tác và duyệt qua trong HTML, thực hiện giao tiếp máy chủ, xử lý các sự kiện, hiệu ứng hình ảnh, và hình ảnh động cho trang web.
- Các tính năng cơ bản của jQuery mobile như sau:

## Simplicity

- Framework này dễ sử dụng và cho phép phát triển các trang web bằng cách sử dụng đánh dấu hướng với tối thiểu hoặc không có JavaScript.

## Accessibility

- Framework hỗ trợ Accessible Rich Internet Applications (ARIA) giúp phát triển các trang web truy cập cho du khách khuyết tật.

## Enhancements and Degradation

- Mobile jQuery chịu ảnh hưởng của HTML5 mới nhất, JavaScript, và CSS3.

# jQuery Mobile 2-6

## Themes

- Framework này cung cấp chủ đề cho phép người dùng đưa ra phong cách riêng của họ.

## Smaller Size

- Kích thước của framework jQuery mobile thì nhỏ hơn 6KB cho CSS và cho thư viện JavaScript là 12KB.

### • Ví dụ.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="jquery.mobile-1.0a3.min.css" />
    <script src="jquery-1.5.min.js"></script>
    <script src="jquery.mobile-1.0a3.min.js"></script>
  </head>
  <body>
    <div data-role="page">
      <div data-role="header">
        <h1>Car Rental</h1>
      </div>
```

# jQuery Mobile 3-6

```
<div data-role="content">
  <p>Choose from the listed car models</p>
  <ul data-role="listview" data-inset="true">
    <li><a href="#">Ford</a></li>
    <li><a href="#">Ferrari</a></li>
    <li><a href="#">BMW</a></li>
    <li><a href="#">Toyota</a></li>
    <li><a href="#">Mercedes-Benz</a></li>
  </ul>
</div>
<div data-role="footer">
  <h4>&copy; DriveCars 2012.</h4>
</div>
</div>
</body>
</html>
```



# jQuery Mobile 4-6

- Trong đoạn mã này, ba tập tin được bao gồm, CSS (jquery.mobile-1.0a3.min.css), jQuery thư viện (jquery-1.5.min.js), và thư viện di động jQuery (jquery.mobile-1.0a3.min.js).
- Một người sử dụng cũng có thể tải các thư viện jQuery từ trang web <http://code.jquery.com/>.

jQuery Mobile có thể HTML và làm cho họ trên các thiết bị di động. Để làm việc này, HTML có sử dụng các thuộc tính dữ liệu.

jQuery sử dụng các thuộc tính như các chỉ số làm cho nó trên các trang web.

jQuery cũng giống như div sử dụng một giá trị dữ liệu vai trò đặc biệt như trang, nội dung, tiêu đề, và footer.

Có nhiều khối div thêm vào mã cho trang, nội dung, tiêu đề, và footer.

Để hiển thị các mẫu xe khác nhau một listview dữ liệu được thêm vào để tăng cường giao diện của trang web điện thoại di động.

# jQuery Mobile 5-6

- Một người sử dụng cần phải cài đặt Opera Mobile Emulator từ trang web của Opera.
- Sau khi cài đặt Opera Mobile Emulator, thực hiện các bước sau đây để áp dụng cài đặt vào trình giả lập:

1. Chọn **All Programs** → **Opera Mobile Emulator** → **Opera Mobile Emulator**. Hộp thoại **Opera Mobile Emulator** sẽ hiển thị.

2. Trong **Profile**, chọn **Samsung Galaxy Tab**.

Trong **Resolutio**, chọn **WVGA Portrait(480x800)**.

4. Chọn **Update**.

5. Chọn **Launch**. **Samsung Galaxy** được hiển thị.

# jQuery Mobile 6-6

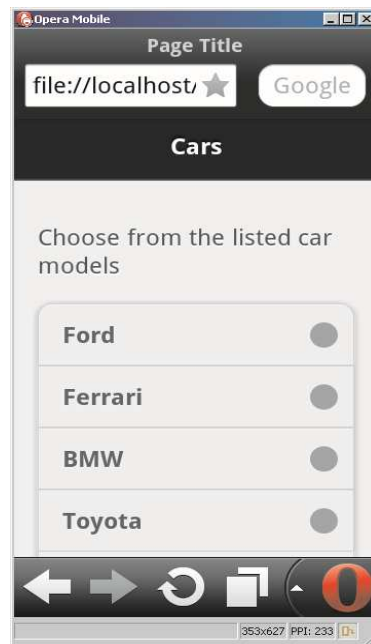
- Một người sử dụng cần cài đặt Opera Mobile Emulator từ trang web của Opera và thực hiện các bước sau đây để áp dụng cài đặt vào trình giả lập: :

1. Thêm Opera Mobile Emulator trong trình soạn thảo CoffeeCup bằng cách nhấn **Tools → Additional Browsers → Test with Additional Browser 1** và cung cấp cho các vị trí của Opera Mobile Emulator được cài đặt trên hệ thống của bạn.

2. Mở file jQuery trong trình soạn thảo CoffeeCup và lưu lại.

3. Chọn **Tools → Additional Browser → Test with Additional Browser 1**.

- Kết quả.



# Tổng kết

- Scripting đề cập đến một loạt các lệnh được thông dịch và thực hiện tuần tự và ngay lập tức một sự kiện xuất hiện.
- JavaScript là một ngôn ngữ kịch bản, có thể được thực hiện trên phía máy khách và phía máy chủ.
- Một biến dùng để chỉ một tên biểu tượng chứa một giá trị, trong đó có thể thay đổi giá trị.
- Một kiểu dữ liệu nguyên thủy có chứa một giá trị đơn như một số hoặc một chuỗi.
- Một hàm là một đoạn mã mà thực hiện một số hành động trên các biến để hoàn thành một nhiệm vụ cụ thể.
- Xử lý sự kiện là một quá trình xác định các hành động được thực hiện khi sự kiện xảy ra.
- Sự kiện bọt là một cơ chế cho phép bạn chỉ định một xử lý sự kiện chung cho tất cả các phần tử con.
- jQuery mobile là một framework để phát triển giao diện Web, cho phép người sử dụng để xây dựng các ứng dụng web di động hoạt động trên máy tính bảng và điện thoại thông minh.