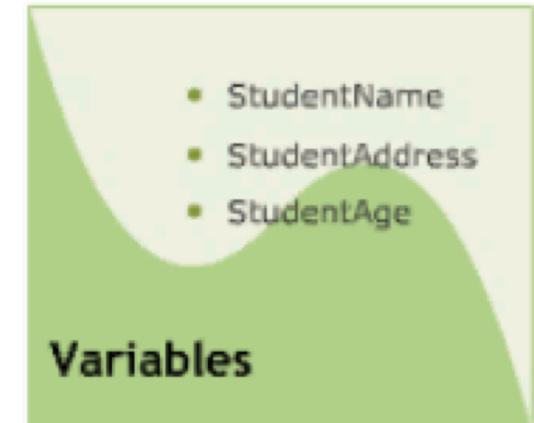
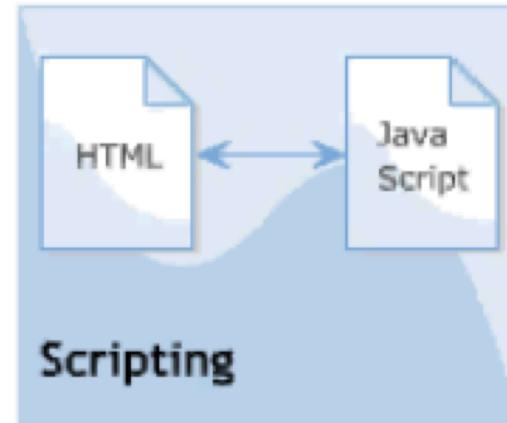


Chương 8

Mục tiêu bài học

- Kịch bản (Scripting)
- Giới thiệu về JavaScript
- Các vấn đề căn bản trong JavaScript



The diagram shows a snippet of JavaScript code within a yellow rounded rectangle:
<SCRIPT>
//Using
//Methods
</SCRIPT>

The word 'JavaScript Basics' is written below the code.

- Kịch bản được xem như là một dãy các câu lệnh được thông dịch và thực thi tuần tự ngay lập tức khi có một sự kiện xảy ra.
- Sự kiện là một hành động được phát sinh do người dùng lúc tương tác với trang Web. Ví dụ như click một nút, di chuột qua một đối tượng, chọn một mục trên menu...vv
- Ngôn ngữ kịch bản thường được nhúng vào trong trang HTML nhằm thay đổi cách ứng xử (behavior) của trang Web theo yêu cầu của người dùng.
- Có hai loại ngôn ngữ kịch bản:
 - Kịch bản phía client
 - Kịch bản phía server.

JavaScript là gì?

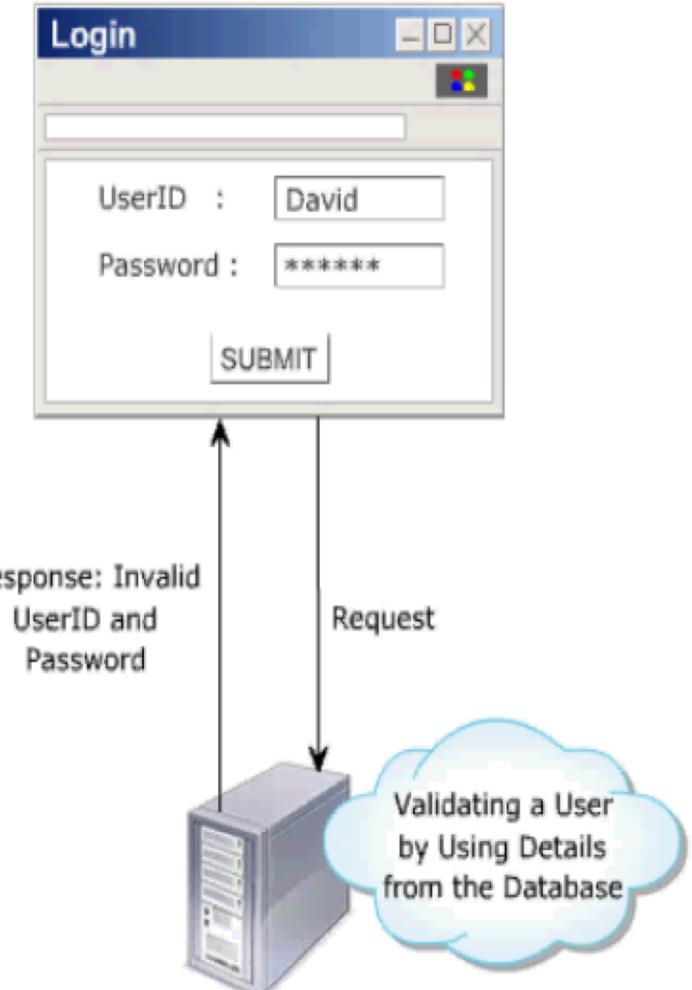
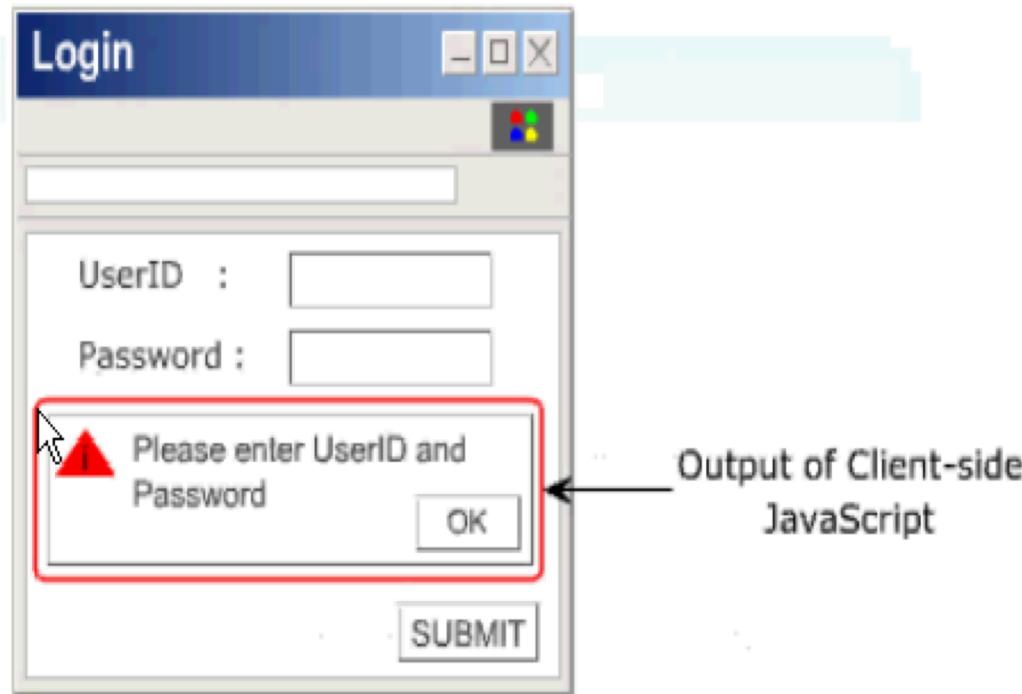
- JavaScript là ngôn ngữ kịch bản dùng để tạo các kịch bản phía client (client-side) và phía server (server-side).
- JavaScript làm cho việc tạo các trang Web động và tương tác trên Internet dễ dàng hơn.
- JavaScript là một ngôn ngữ kịch bản được hãng Sun Microsystems và Netscape phát triển.
- JavaScript được phát triển từ Livescript. Của Netscape

- JavaScript cung cấp tính tính động và tính tương tác trong các trang web qua các chức năng như sau:
 - Đáp lại (responding) nhanh chóng các yêu cầu (request) của người dùng.
 - Sinh ra các trang HTML bằng việc viết nội dung HTML.
 - Kiểm tra tính hợp lệ của dữ liệu do người dùng nhập.
 - Thực hiện các tính toán đơn giản phía client.

- Tương tự các ngôn ngữ khác, JavaScript cũng có các quy tắc cú pháp như:
 - Phân biệt chữ hoa, chữ thường.
 - Luôn có cặp kí hiệu mở và đóng như { }, ()
 - Sử dụng thêm các kí tự trắng, hoặc các tab giúp cho ta dễ dàng đọc hay sửa file script.
 - Sử dụng các dòng chú thích tạo các ghi chú về chức năng của đoạn script và thời gian tạo.

Môi trường thực thi

- Môi trường thực thi
 - Các Scripting ở phía Client
 - JavaScript trên Web Server



JavaScript ở phía Client

JavaScript ở phía Server

- Các công cụ sinh mã JavaScript và phần mềm có giao diện phát triển ứng dụng IDE được sử dụng giúp tự động tạo ra các đoạn mã JavaScript code. Một vài chức năng được sinh mã:
 - Dialog Box
 - Pop – up Menu Builder
 - Remotes

- JavaScript có thể chèn vào một tài liệu HTML theo những cách sau :
 - Sử dụng thẻ SCRIPT:

```
<script language="JavaScript">
  <!--
    JavaScript statements;
  //-->
</script>
```

- Sử dụng một file JavaScript ở ngoài

```
<script language="JavaScript" src="filename.js">
</script>
```

- Sử dụng JavaScript trong các trình điều khiển sự kiện

```
<INPUT type="button" onClick="alert('Hello !');">
```

- Mỗi biến là một vị trí duy nhất trong bộ nhớ máy tính để lưu trữ giá trị.
- Mỗi biến có một tên duy nhất và giá trị của nó có thể thay đổi trong khi kịch bản thực thi.
- Sử dụng từ khoá ‘var’ để khai báo biến.

ví dụ: var A = 10;

- Các biến có một phạm vi được xác định, khi chúng khai báo trong script.
 - Biến toàn cục.
 - Biến cục bộ.
- Nguyên dạng (literal) là các giá trị không đổi được dùng trong script.

- Dùng các kí tự chữ, số và dấu nối (underscore) để đặt tên.
- Phải bắt đầu bằng kí tự chữ hoặc dấu nối.
- Không chứa các kí tự đặc biệt như +, -, *, /, %, ...vv
- Không chứa kí tự trắng.
- Không trùng với các từ khóa.

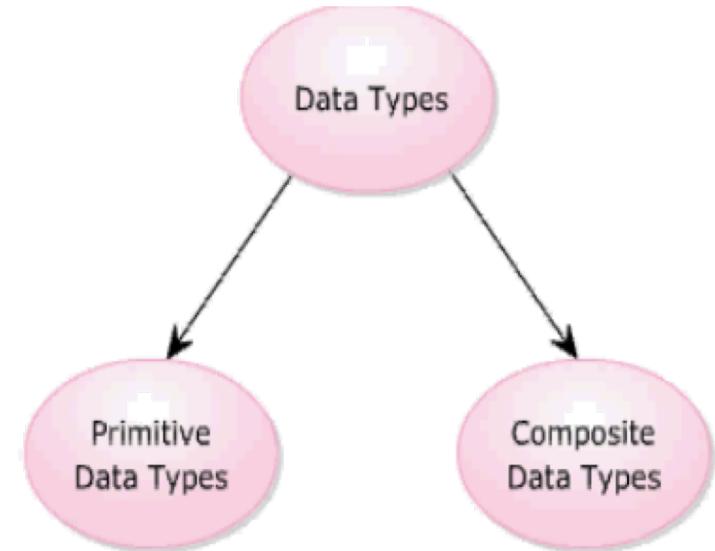
- Có hai loại kiểu dữ liệu

- Kiểu nguyên thủy

- number: kiểu số
 - boolean: giá trị logic.
 - string: chuỗi
 - null: giá trị rỗng

- Kiểu phức hợp

- Objects: Tham chiếu đến các đối tượng trong JavaScript
 - Functions: Hàm, là một tập các câu lệnh
 - Arrays:



- Hiển thị thông tin
- Chú thích (comment)
- Các kí tự escape sequense
- Các hàm dựng sẵn (built-in functions)

- Sử dụng phương thức write và writeln của đối tượng document

- Cú pháp write

```
document.write("<dữ_liệu>"+biến)
```

- <dữ_liệu>: là một chuỗi được đặt trong cặp nháy kép.
 - biến: tên của biến, giá trị của nó sẽ được hiển thị.

- Cú pháp writeln

```
document.writeln("<dữ_liệu>"+biến)
```

- Cũng giống như write, phương thức writeln chèn thêm kí tự xuống dòng

- Chú thích một dòng

```
// nội dung dòng chú thích đặt ở đây
```

- Chú thích trên nhiều dòng

```
/*
    nội dung các dòng chú thích đặt ở đây
*/
```

Escape Sequences

Escape Sequence	Non-Printing Character
\b	Backspace.
\f	Form feed.
\n	New line.
\r	Carriage return.
\t	Horizontal tab.
\'	Single quote.
\"	Double quote.
\\\	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'.
\uaaaaa	Represent the Unicode encoding character, where aaaaa are four hexadecimal numbers. For example, the character \u0020 represents a space.

Các hàm dựng sẵn (built-in functions)

- JavaScript cung cấp nhiều hàm dựng sẵn, là các hàm được định nghĩa trước đáp ứng một vài bài toán. Xem bảng sau

Hàm	Mô tả	Ví dụ
alert()	Hiển thị hộp thoại với một chuỗi thông tin và một nút OK .	alert('Các trường không được để trống')
confirm()	Hiển thị một thoại với nút OK và Cancel . Thường dùng để xác minh lại một hành động do người dùng thực hiện.	confirm('Are you sure you want to delete it ?')
parseInt()	Chuyển một giá trị chuỗi sang một giá trị số nguyên.	parseInt('25 years') Trả về số 25
parseFloat()	Chuyển một chuỗi số sang giá trị số thực.	parseFloat('10.33') Trả về số 10.33
eval()	Định giá trị một biểu thức và trả về kết quả.	eval("2+2") Trả về 4
isNaN()	Kiểm tra một giá trị xem nó đúng là không phải là một số không ?	isNaN("Hello") Trả về true
prompt()	Hiển thị một thoại cho phép nhập vào một giá trị	prompt("Enter your name", "Name")