



# THIẾT KẾ VÀ LẬP TRÌNH WEB BẰNG NGÔN NGỮ ASP

Lý thuyết: 45 tiết  
Thực hành: 30 tiết

- GVHD: Dương Khai Phong
- Email: [khaiphong@gmail.com](mailto:khaiphong@gmail.com)
- Website: <http://sites.google.com/site/khaiphong>



# NỘI DUNG MÔN HỌC

1/ Giới thiệu tổng quan Web

2/ HTML và JavaScript

3/ Các đối tượng trong ASP.Net

4/ ADO.Net (kết nối cơ sở dữ liệu)

5/ Triển khai ứng dụng Web + Ôn tập

## PHẦN 2 (tiếp theo): (HTML và JAVASCRIPT)

# JAVASCRIPT



# NỘI DUNG: JAVASCRIPT

1. Giới thiệu
2. Kiểu dữ liệu, hằng và biến
3. Các phép toán trong javascript
4. Các đối tượng hộp thoại trong javascript
5. Các cấu trúc điều khiển cơ bản
6. Mảng và hàm trong javascript
7. Các đối tượng trong javascript
8. Xử lý sự kiện trong javascript



# 1/ Ngôn ngữ kịch bản Javascript

## a. Giới thiệu Javascript:

- Với HTML ta chỉ thiết kế được trang web để hiển thị thông tin, không tạo ra được sự tương tác từ phía người dùng.
  - JavaScript là ngôn ngữ kịch bản (do hãng Sun Microsystems và Netscape phát triển từ ngôn ngữ *LiveScripts*) dùng để tạo các client-side scripts và server-side scripts (có sự tương tác với người dùng).
  - Mặc dù có những điểm tương đồng giữa Java và JavaScript, nhưng chúng vẫn là hai ngôn ngữ riêng biệt.
- ❖ Lưu ý: trong code javascript cũng phân biệt chữ hoa và chữ thường.



# 1/ Ngôn ngữ kịch bản Javascript

## **b. Cách nhúng Javascript vào một trang HTML:**

JavaScript có thể chèn vào một tài liệu HTML theo những cách sau:

- Sử dụng thẻ SCRIPT.
- Sử dụng một file JavaScript từ bên ngoài.
- Sử dụng các biểu thức JavaScript trong các giá trị thuộc tính của thẻ
- Sử dụng JavaScript trong các trình điều khiển sự kiện



# 1/ Ngôn ngữ kịch bản Javascript

## b. Cách nhúng Javascript vào một trang HTML:

### b.1> Sử dụng thẻ **SCRIPT**:

- Khi trình duyệt gặp phải một thẻ **<SCRIPT>** nào đó, nó sẽ đọc từng dòng một cho đến khi gặp thẻ đóng **</SCRIPT>**.
- Tiếp đến nó sẽ *kiểm tra lỗi* trong các câu lệnh Javascript.
- Nếu gặp phải lỗi, nó sẽ cho *hiển thị lỗi* đó trong chuỗi các hộp cảnh báo (*alert boxes*) lên màn hình.
- Nếu không có lỗi, các *câu lệnh sẽ được biên dịch* sao cho máy tính có thể hiểu được lệnh đó.



# 1/ Ngôn ngữ kịch bản Javascript

## b. Cách nhúng Javascript vào một trang HTML:

### b.1> Cú pháp sử dụng thẻ SCRIPT:

```
<script language = "JavaScript">  
    <!--  
    Dòng ghi chú;  
    //-->  
    ....  
</script>
```

- Việc khai báo sử dụng ngôn ngữ Javascript có thể đặt bất kỳ vị trí nào trong cặp thẻ **<html>...</html>**. Tuy nhiên, nên đặt các câu lệnh script trong phần **<head>** và **</head>** để đảm bảo tất cả các câu lệnh đều được đọc và biên dịch trước khi nó được gọi từ trong phần BODY.



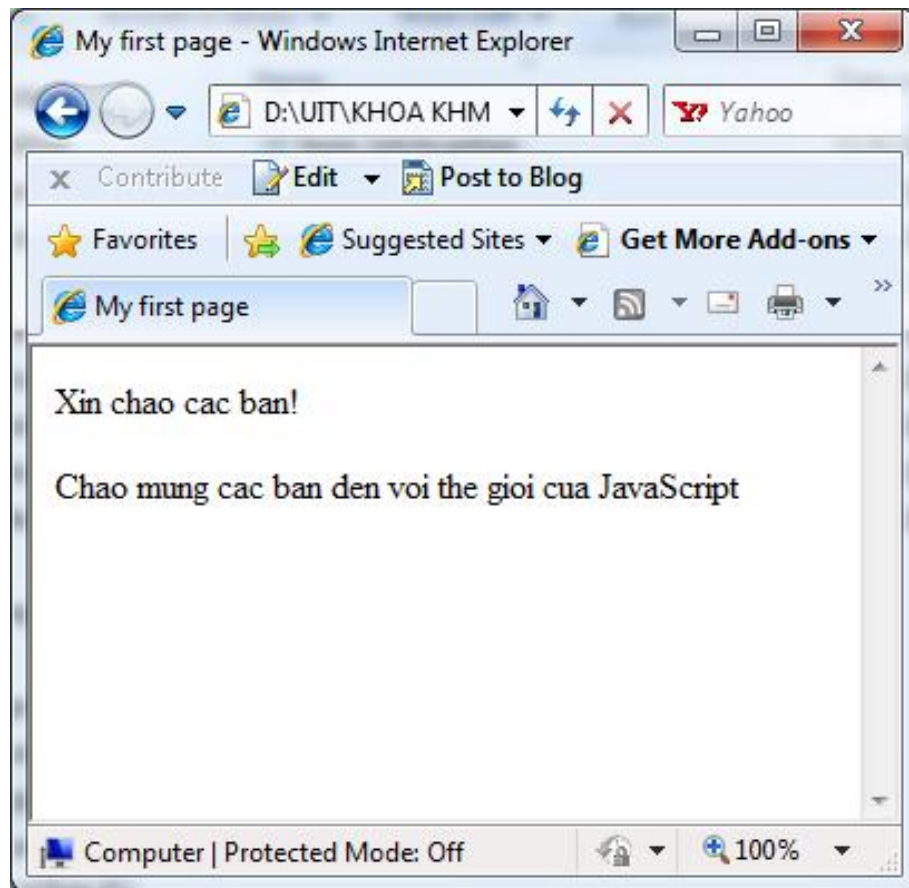


# 1/ Ngôn ngữ kịch bản Javascript

## b. Cách nhúng Javascript vào một trang HTML:

### b.1> Ví dụ sử dụng thẻ SCRIPT:

```
<html>
<head> <title>My first page</title>
      <SCRIPT LANGUAGE="JavaScript">
        document.write("Xin chao cac
        ban!");
      </SCRIPT>
</head>
<body>
  <p> Chao mung cac ban den voi the
    gioi cua JavaScript</p>
</body>
</html>
```





# 1/ Ngôn ngữ kịch bản Javascript

## b. Cách nhúng Javascript vào một trang HTML:

### b.2> Sử dụng một file JavaScript từ bên ngoài:

- **File javascript** là file văn bản chứa các mã lệnh JavaScript, file javascript có phần mở rộng là “.js”.
- Nó chỉ có thể chứa các câu lệnh và các hàm JavaScript, *không thể chứa các thẻ HTML*.
- Cú pháp khai báo:

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



# 1/ Ngôn ngữ kịch bản Javascript

## b. Cách nhúng Javascript vào một trang HTML:

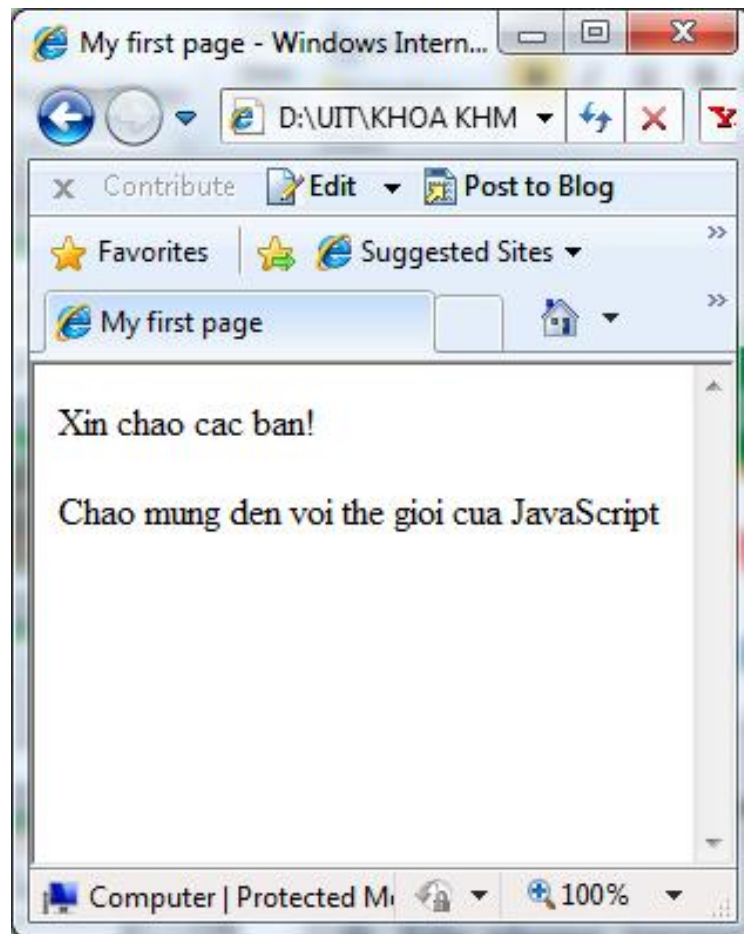
### b.2> Sử dụng một file JavaScript ở ngoài :

#### ❖ File vidu.js:

```
document.write ("Xin chao cac ban! ")
```

#### ❖ File nhung\_javascript.html:

```
<html>
<head> <title>My first page</title>
<SCRIPT LANGUAGE="JavaScript" src="vidu.js" >
</SCRIPT>
</head>
<body>
    <p> Chao mung den voi the gioi cua
    JavaScript</p>
</body>
</html>
```





## 2/ Kiểu dữ liệu, hằng và biến

### a. Kiểu dữ liệu:

- Việc xác định kiểu dữ liệu trong javascript được chuyển đổi một cách tự động trong quá trình khai báo và sử dụng các biến. Các kiểu dữ liệu thường dùng:
  - Kiểu số nguyên
  - Kiểu số thực
  - Kiểu ký tự
  - Kiểu chuỗi
  - Kiểu logic (True – False)
  - ...



## 2/ Kiểu dữ liệu, hằng và biến

### b. Khai báo hằng và biến:

- Trong javascript không cho phép định nghĩa hằng tường minh.
- Các biến trong javascript phân biệt chữ hoa và chữ thường.
- Cú pháp: **var** <tên\_biến>

```
<head>
```

```
<title>Khai báo biến</title>
```

```
<script language="JavaScript">
```

```
    var a;           // biến Glocal
```

```
    var b=2;         // biến Glocal
```

```
    var result=0;    // biến Glocal
```

```
    result=a+b;
```

```
    document.write("Ket Qua cua ham myFunction1 la : "+result+"<br>");
```

```
</script>
```

```
</head>
```

Lệnh **document.write**: dùng để xuất thông tin trên trình duyệt.  
(tương tự: **document.writeln**)



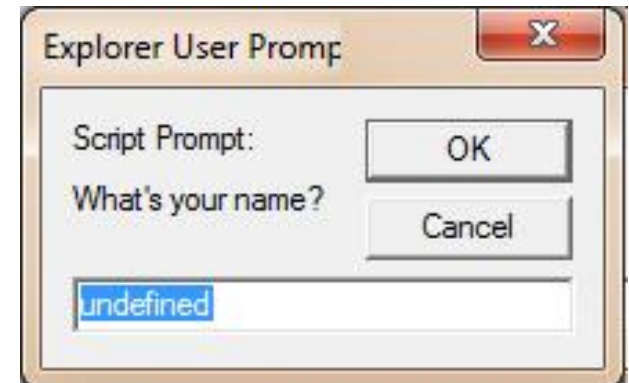
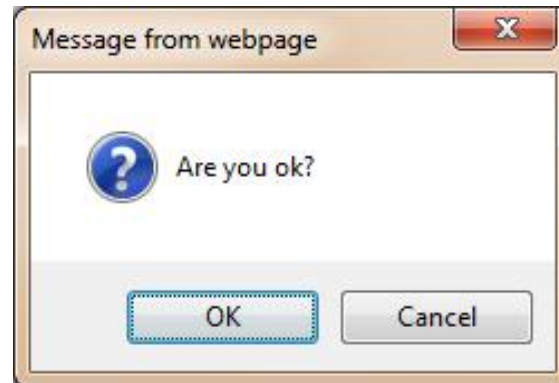
### 3/ Các phép toán trong javascript

- Trong javascript sử dụng cả hai toán tử một ngôi và hai ngôi, gồm:
  - Toán tử số học:  $+$  ,  $-$  ,  $*$  ,  $/$  ,  $++$  ,  $--$  , ...
  - Toán tử so sánh:  $>$  ,  $<$  ,  $>=$  ,  $!=$  , ...
  - Toán tử logic: AND ( $\&\&$ ), OR ( $\|\|$ ) , ...
  - Toán tử chuỗi:  $+$  (nối chuỗi)
  - Toán tử lượng giá: điều kiện ( $?$ ), `typeof`, ...
- Javascript cung cấp các thư viện hàm cho người dùng khá đầy đủ như: các hàm chuyển đổi kiểu dữ liệu, các hàm xử lý chuỗi,...



## 4/ Các đối tượng hộp thoại trong javascript

- Trong javascript cung cấp sẵn các đối tượng hộp thoại (**dialog boxes**) cho người dùng tương tác với trình duyệt trên phía client, bao gồm:
  - **Alert**: hiển thị thông báo
  - **Confirm**: xác nhận thông tin người dùng
  - **Prompt**: tương tác với người dùng bằng cách cho phép nhập giá trị mới.





## 4/ Các đối tượng hộp thoại trong javascript

### a. Alert:

- Công dụng: dùng hiển thị thông báo cho người dùng.
- Cú pháp: **Alert**("Nội dung thông báo")

### b. Confirm:

- Công dụng: dùng xác nhận lại thông tin từ phía người dùng. Hộp thoại trả về **True** nếu người dùng đồng ý.
- Cú pháp: **Confirm**("Nội dung xác nhận")

### c. Prompt:

- Công dụng: dùng nhận thông tin từ phía người dùng. Hộp thoại trả về **giá trị** người dùng đã nhập.
- Cú pháp: **Prompt**("Tiêu đề yêu cầu nội dung")





## 4/ Các đối tượng hộp thoại trong javascript

**Ví dụ:** alert, confirm, prompt

```
<html>
<head>
<title>Dialog boxes</title>
<script language="javascript">
    alert("Chào bạn");
    var namsinh=prompt("Bạn sinh năm mấy?");
    var traloi=confirm("Bạn có muốn tính tuổi của bạn không?");
    if (traloi==true)
        document.write("<h1>Bạn được "+(2011-namsinh)+"<h1>");
    else
        document.write("<h1>Chào bạn<h1>");
</script>
</head>
<body>
</body>
</html>
```



## 5/ Các cấu trúc điều khiển cơ bản trong javascript

### Cấu trúc điều kiện

If .. Else ..

Switch .. Case ..

### Cấu trúc lặp

For ..

While ..

Do .. While



## 5/ Các cấu trúc điều khiển cơ bản trong javascript

### a. Ví dụ: Cấu trúc điều kiện




1. Viết chương trình giải và biện luận phương trình bậc 1.
2. Viết chương trình giải và biện luận phương trình bậc 2.
3. Nhập 3 cạnh tam giác a , b , c: cho biết loại tam giác là tam giác đều, cân, vuông cân, vuông, thường.
4. Viết chương trình kiểm tra ngày, tháng, năm có hợp lệ?
5. Viết chương trình nhập điểm TB, cho biết xếp loại: giỏi, khá, trung bình, yếu.
6. Viết chương trình nhập năm sinh, nếu trên 18 tuổi thì in ra trên trình duyệt tuổi của người này.





## 5/ Các cấu trúc điều khiển cơ bản trong javascript

### b. Ví dụ: Cấu trúc lặp

1. Viết chương trình nhập n, in ra trên trình duyệt bảng cửu chương thứ n. 
2. Viết chương trình nhập n, in ra in ra trên trình duyệt n bảng cửu chương.
3. Viết chương trình nhập số dòng, số cột, in ra bảng biểu có số dòng và số cột đã nhập. 
4. Viết chương trình nhập tháng, năm, in ra trên trình duyệt ra lịch của tháng và năm đó.
5. Viết chương trình nhập user và password, nếu nhập đúng (user và password: “abc”) thì in ra trình duyệt câu “Xin chào”, ngược lại bắt người dùng nhập lại. 



## 6/ Mảng và hàm trong javascript

### a. Mảng:

- Trong javascript không có kiểu dữ liệu mảng *tường minh* (ví dụ: ~~int~~ ~~mang~~[10]) mà chỉ hỗ trợ thông qua đối tượng **Array** sẵn có và các thuộc tính và phương thức mà đối tượng này hỗ trợ.
- Khai báo mảng:
  - arrayObjectName = **new Array**(element0,element1,..)
  - Hoặc arrayObjectName = **new Array**(arrayLength)
- Ví dụ: tạo mảng gồm 5 phần tử  
**var Mang = new Array(5)**



## 6/ Mảng và hàm trong javascript

### a. Mảng:

- Truy cập phần tử mảng: chỉ số bắt đầu của mảng là 0
- Ví dụ:

```
<script language="javascript">  
    var arrMaVung = new Array("08","04","72","65","64");  
    var arrTenVung = new Array(5);  
    arrTenVung[0]="HCM";  
    arrTenVung[1]="Hà Nội";  
    arrTenVung[2]="Long An";  
    arrTenVung[3]="Bình Dương";  
    arrTenVung[4]="Vũng Tàu";  
    for (i=0;i<5;i++)  
        document.write(arrMaVung[i]+" "+arrTenVung[i]+"<br>");  
</script>
```



## 6/ Mảng và hàm trong javascript

### a. Mảng:

- Thuộc tính của đối tượng Array:

<b>length</b>	Trả về số phần tử của mảng
<b>index</b>	Trả về chỉ mục của phần tử.

- Phương thức của đối tượng Array:

<b>concat</b>	Nối hai mảng và trả về một mảng mới.
<b>join</b>	Kết hợp tất cả các phần tử của một mảng thành một chuỗi.
<b>pop</b>	Gỡ bỏ phần tử cuối cùng của một mảng và trả về phần tử đó.
<b>push</b>	Thêm một hoặc nhiều phần tử vào cuối một mảng và trả về độ dài mới của mảng.
<b>splice</b>	Chèn hoặc xoá một hoặc nhiều phần tử theo vị trí thứ k trong mảng
<b>reverse</b>	Hoán vị các phần tử của một mảng: Phần tử mảng đầu tiên trở thành phần tử cuối cùng và ngược lại
<b>sort</b>	Sắp xếp các phần tử của một mảng



## 6/ Mảng và hàm trong javascript

### b. Hàm:

- Javascript cung cấp sẵn cho người dùng một số hàm thông dụng:
  - Hàm **eval**: đánh giá các biểu thức hay lệnh.
  - Hàm **isFinite**: xác định xem 1 số có là hữu hạn hay không?
  - Hàm **isNaN**: kiểm tra một biến có là số?
  - Hàm **parseInt** và **parseFloat**: chuyển đổi kiểu
  - Hàm **Number** và **String**: chuyển đổi kiểu
- Ngoài ra, người dùng có thể định nghĩa các hàm tự tạo theo cú pháp sau:

```
function functionName (argument1, argument2, ...)  
{  
    statements;  
    [return value;]  
}
```





## 6/ Mảng và hàm trong javascript

### b. Hàm:

```
function functionName (argument1, argument2, ...)  
{  
    statements;  
    [return value;]  
}
```






Trong đó:

- **function**: từ khoá bắt buộc do javascript là ngôn ngữ có tính định kiểu thấp nên không cần xác định trước kiểu dữ liệu trả về.
- **functionName**: tên hàm
- **argument1, argument2, ...**: tham số đầu vào
- **[return value;]**: giá trị trả về của hàm nếu có



## 6/ Mảng và hàm trong javascript

### c. Ví dụ: Mảng và Hàm

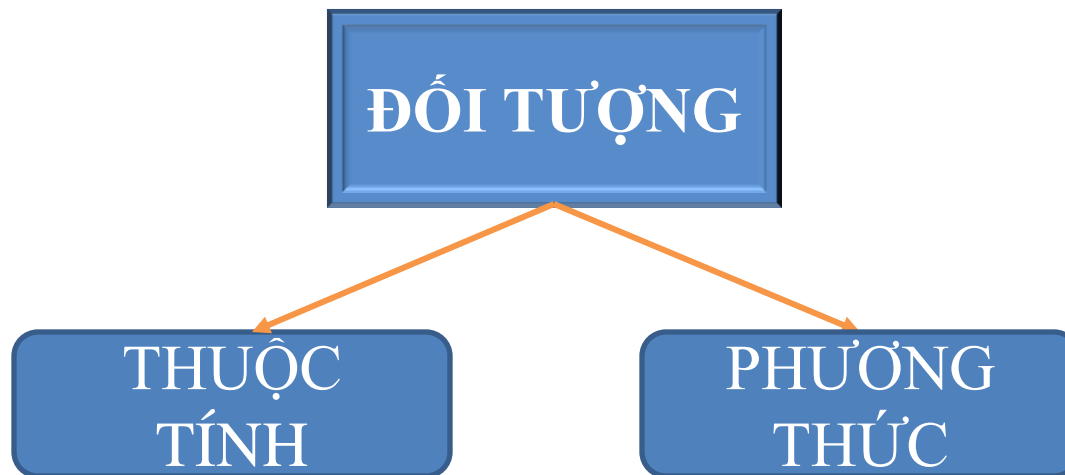
1. Viết hàm nhập/xuất mảng gồm n phần tử, xuất ra giá trị max, min của mảng. 
2. Viết chương trình nhập 2 mảng a,b gồm n phần tử, trộn 2 mảng lại và sắp xếp theo thứ tự tăng dần. 
3. Viết hàm thêm 1 phần tử x vào mảng tại vị trí thứ k . 
4. Viết hàm thêm 1 phần tử x vào mảng đã có thứ tự tăng dần sao cho sau khi thêm mảng vẫn tăng. 
5. Viết hàm xoá một phần tử x trong mảng. 



## 7/ Các đối tượng cơ bản trong javascript

### a. Giới thiệu:

- **Đối tượng** là gì?



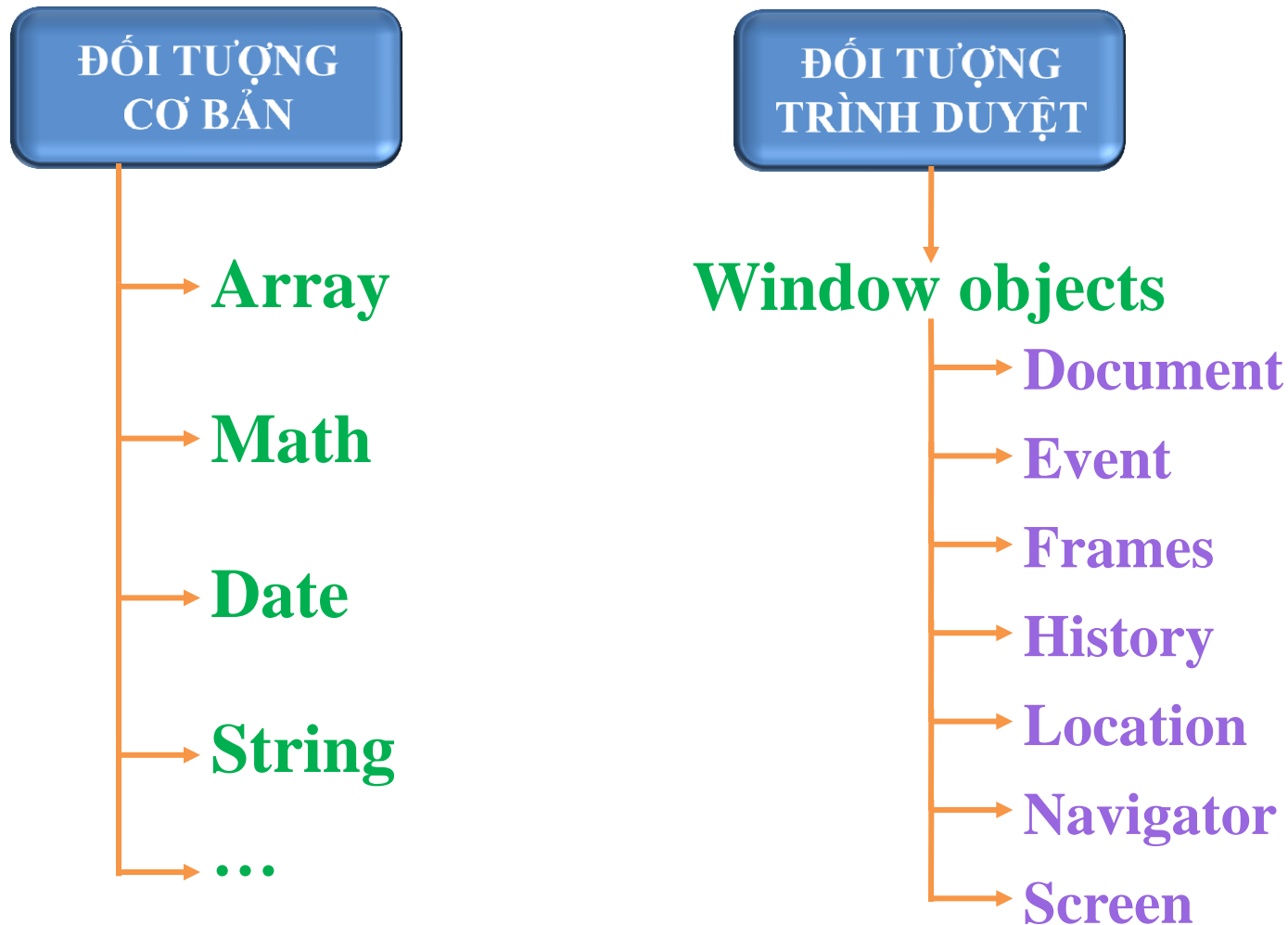
- **Ví dụ**: đối tượng SINHVIEN có các thuộc tính: mã sinh viên, họ tên, điểm, ... và các phương thức: đi học, đi thi, ...
- Truy cập thuộc tính và phương thức:
  - Tên\_đối\_tượng.**Tên\_thuộc\_tính**
  - Tên\_đối\_tượng.**Tên\_phương\_thức(...)**
- Sử dụng con trỏ **This** cho đối tượng hiện hành



## 7/ Các đối tượng trong javascript

### a. Giới thiệu:

- Các đối tượng javascript hỗ trợ:





## 7/ Các đối tượng trong javascript

### b. Đối tượng cơ bản:

- **Array**: đối tượng dùng quản lí danh sách mảng.
- **Math**: đối tượng liên quan đến các phép tính toán học.
- **String**: đối tượng dùng để thao tác với các chuỗi văn bản.
- **Date**: đối tượng liên quan đến ngày giờ.

### c. Đối tượng trình duyệt:

#### ❖ **window:**

- **Công dụng:** dùng quản lý thông tin của tất cả các đối tượng trong cửa sổ trình duyệt.
- **Các thuộc tính cơ bản:** đối tượng window được xem là đối tượng cơ bản (base class) của tất cả các đối tượng khác.
- **Các phương thức cơ bản:** alert(), blur(), close(), open(), focus(), navigate().



## 7/ Các đối tượng trong javascript

### c. Đối tượng trình duyệt:

#### ❖ document:

- **Công dụng:** dùng quản lý thông tin tài liệu HTML trong cửa sổ trình duyệt (được xem là đối tượng con của **window**)
- **Các thuộc tính và phương thức:**

Các thuộc tính:

- `alinkColor`
- `bgColor`
- `Body`
- `fgColor`
- `linkColor`
- `location`
- `Title`
- `URL`
- `vlinkColor`

Các phương thức:

- `clear()`
- `close()`
- `open()`
- `write()`
- `writeln()`



## 7/ Các đối tượng trong javascript

### c. Đối tượng trình duyệt:

#### ❖ history:

- **Công dụng:** dùng quản lý danh sách các URL đã duyệt.
- **Các thuộc tính và phương thức:**

Các thuộc tính:

- current
- length
- next
- previous

Các phương thức:

- back()
- forward()
- go(relPos hoặc string)



## 7/ Các đối tượng trong javascript

### c. Đối tượng trình duyệt:

#### ❖ location:

- **Công dụng:** dùng quản lý thông tin URL hiện tại.
- **Các thuộc tính và phương thức:**

Các thuộc tính:

- hash
- host
- hostname
- href

Các phương thức:

- reload()
- replace(URL)

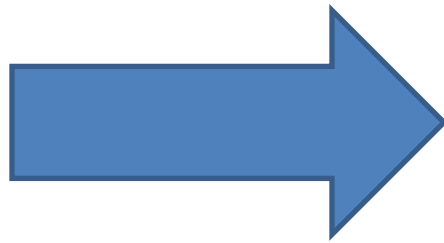




## 8/ Xử lý sự kiện trong javascript

### a. Giới thiệu:

- Javascript quản lý sự tương tác giữa người dùng và trình duyệt thông qua bộ quản lý các **sự kiện (Event)** trên Form và các đối tượng con trên Form.



**SỰ KIỆN ???**



## 8/ Xử lý sự kiện trong javascript

### b. Sự kiện: (Event)

- **Sự kiện** là kết quả thao tác của người dùng tác động lên đối tượng.
- Một sự kiện bao gồm 2 thông tin:
  - ☞ Kiểu sự kiện: click, double click, change,...
  - ☞ Vị trí của con trỏ tại thời điểm xảy ra sự kiện.
- Các sự kiện thường gặp:

- onClick
- onChange
- onFocus
- onBlur
- onSelect
- onMouseOver

- onMouseOut
- onLoad
- onUnload
- onSubmit
- onMouseDown
- onMouseUp



## 8/ Xử lý sự kiện trong javascript

### c. Các sự kiện cơ bản trên các đối tượng:

#### ❖ Đối tượng window: (onLoad - onUnload)

- **Công dụng:** sự kiện xảy ra khi người dùng mở trang hoặc đóng trang.
- **Ví dụ:** viết trang khi người dùng mở trang thì hiển thị thông báo nhập Tên → xuất “Hi, welcome my pages”, sau khi đóng trang thì hiển thị thông báo “Good bye, see you again !”

```
<html>
<head>
</head>
<body onload="hello();" onunload="goodbye();">
</body>
</html>
```

```
<script language="javascript">
    var name = "";
    function hello() {
        name = prompt('Enter Your Name:', 'Name');
        alert('Hi ' + name + ', welcome to my page!');
    }
    function goodbye() {
        alert('Goodbye ' + name + ', see you again!');
    }
</script>
```



## 8/ Xử lý sự kiện trong javascript

### c. Các sự kiện cơ bản trên các đối tượng:

#### ❖ Đối tượng form:

- Sự kiện trên form được xử lý phụ thuộc vào 2 yếu tố sau:
  - Thuộc tính sự kiện của form: **Action, Method, ...**
  - Việc xử lý các sự kiện của các đối tượng con (button, textbox,...) bên trong form: **onSubmid, onClick, onBlur, onChange, ...**
- Ví dụ về chu trình sự kiện trên form:

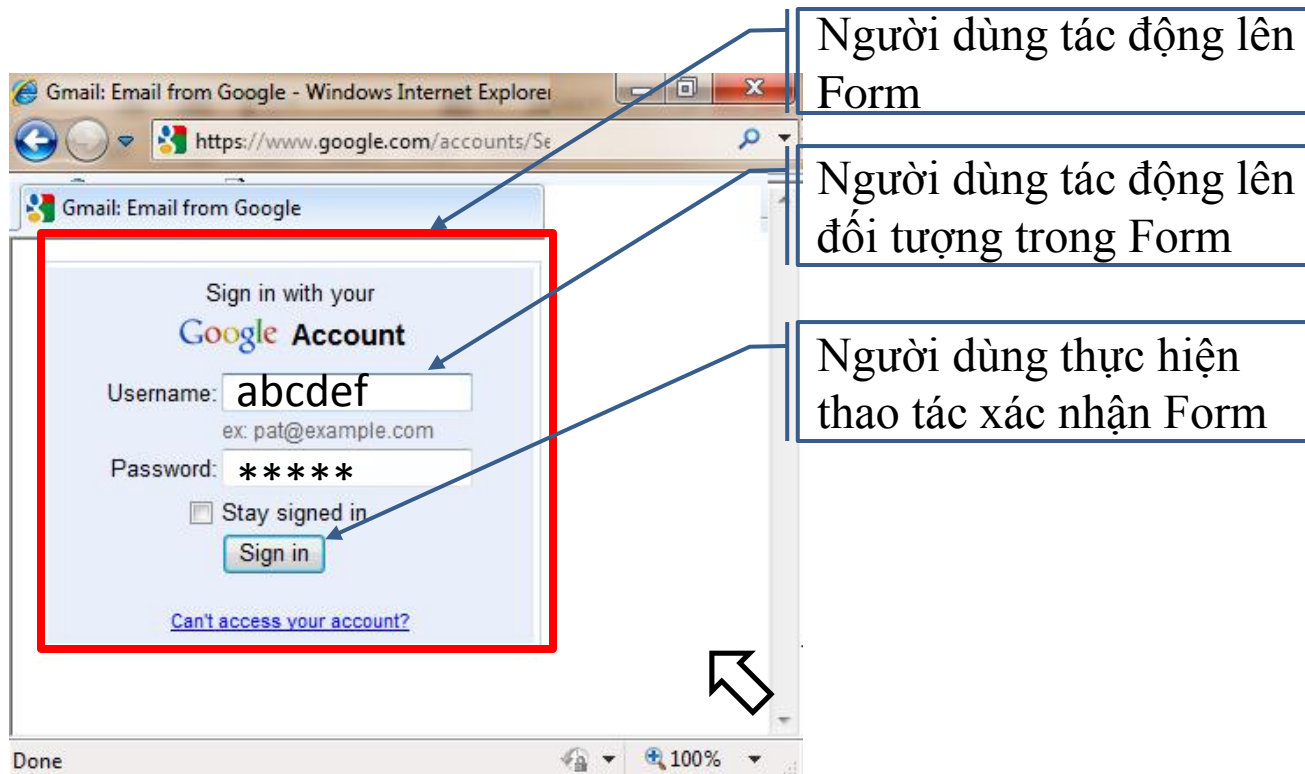


## 8/ Xử lý sự kiện trong javascript

### c. Các sự kiện cơ bản trên các đối tượng:

#### ❖ Đối tượng form:

- Ví dụ về chu trình sự kiện trên form:



Bộ lắng nghe sự kiện Javascript hoạt động.

OnMouseOver

OnChange  
(tuỳ đối tượng)

OnClick  
OnSubmit

Xử lý Form  
phụ thuộc  
vào các  
thuộc tính  
sự kiện

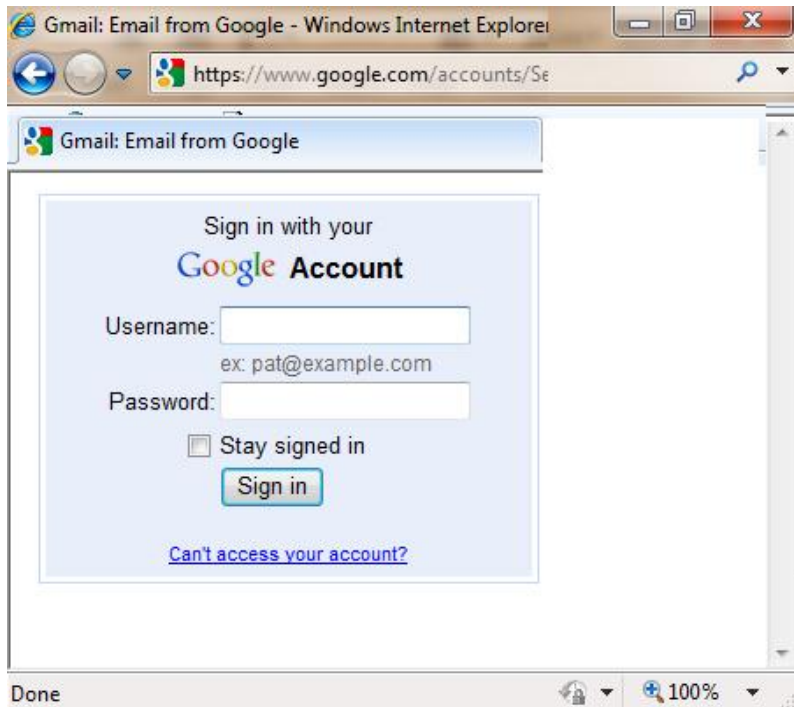


## 8/ Xử lý sự kiện trong javascript

### c. Các sự kiện cơ bản trên các đối tượng:

#### ❖ Đối tượng form:

- Ví dụ về chu trình sự kiện trên form:



#### ❖ Thuộc tính sự kiện:

☞ **Action**: thuộc tính dùng để chuyển trang hiện hành đến một trang khác theo địa chỉ URL truyền vào.

☞ **Method**: phương thức gửi nội dung đi: “get / post”

OnMouseOver

OnChange  
(tuỳ đối tượng)

OnClick  
OnSubmit

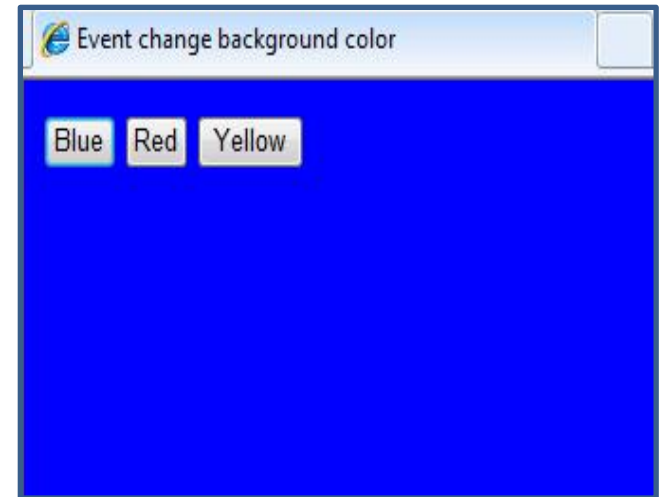
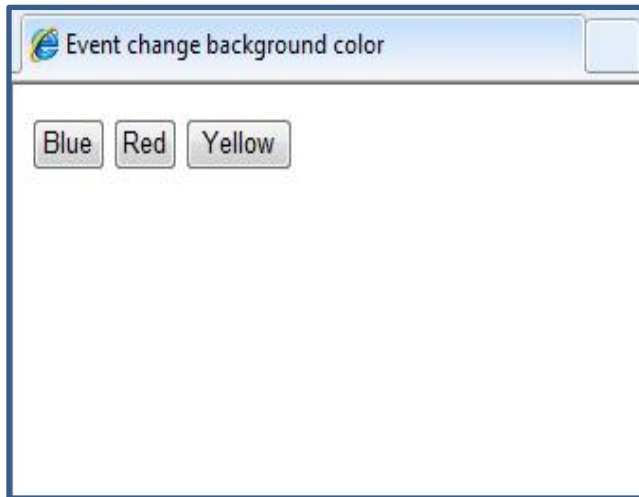
Xử lý Form  
phụ thuộc  
vào các  
**thuộc tính  
sự kiện**



## 8/ Xử lý sự kiện trong javascript

### d. Các ví dụ xử lý sự kiện:

- ❖ Ví dụ 1: thiết kế một trang gồm 3 button cho phép người dùng thay đổi màu nền của trang như sau:





## 8/ Xử lý sự kiện trong javascript

### d. Các ví dụ xử lý sự kiện:

- ❖ Ví dụ 2: thiết kế một trang cho phép nhập thông tin họ tên và năm sinh, xuất ra câu chào và cho biết tuổi người đó:

Event textfield

Họ và tên: Phong

Năm sinh: 1980

Xem tuổi



Bạn Phong được 31 tuổi



Event Textfield

Họ và tên: Phong

Năm sinh: 1980

Xem tuổi

Bạn Phong được 31 tuổi

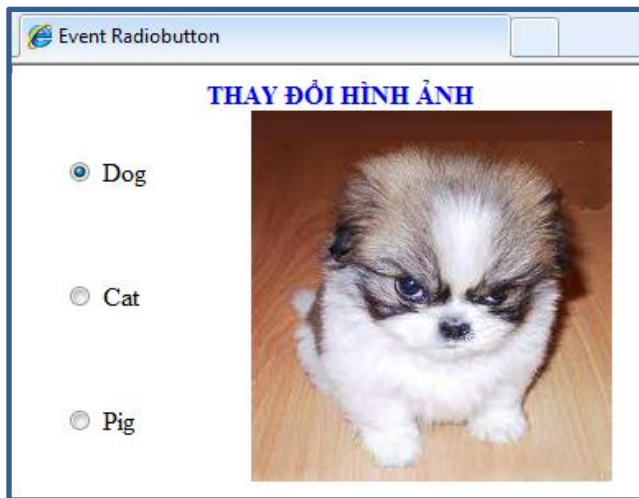




## 8/ Xử lý sự kiện trong javascript

### d. Các ví dụ xử lý sự kiện:

- ❖ Ví dụ 3: thiết kế một trang gồm 3 radio button cho phép người dùng thay đổi hình ảnh trong trang như sau:





## 8/ Xử lý sự kiện trong javascript

### d. Các ví dụ xử lý sự kiện:

- ❖ Ví dụ 4: thiết kế một trang cho phép người dùng đăng nhập, nếu thành công thì cho phép người dùng xem sản phẩm trong trang sanpham.html

Event Form

**ĐĂNG NHẬP WEBSITE**

User name:

Password:

