



# PHẦN 1

## LẬP TRÌNH JAVASCRIPT

### BÀI 3: CẤU TRÚC ĐIỀU KHIỂN, HÀM VÀ XỬ LÝ SỰ KIỆN

- ☐ Biến và kiểu biến
- ☐ Ép kiểu
- ☐ Khai báo biến
- ☐ Toán tử
- ☐ Cấu trúc điều khiển
- ☐ Lệnh lựa chọn đơn

- ❑ Cấu trúc điều khiển (tiếp)
- ❑ Cấu trúc lựa chọn
  - ❖ Lệnh lựa chọn kép
  - ❖ Lệnh đa lựa chọn
- ❑ Cấu trúc lặp
  - ❖ Lặp không biết trước số lần lặp
  - ❖ Lặp biết trước số lần lặp
- ❑ Hàm
- ❑ Phạm vi biến
- ❑ Mảng
- ❑ Xử lý sự kiện



- ☐ Lệnh lựa chọn đơn (Bài 2)
- ☐ Lệnh lựa chọn kép
- ☐ Lệnh đa lựa chọn

□ Cú pháp

```
if (dieukien) {  
    //Thực hiện  
}  
else {  
    //Thực hiện  
}
```

□ Ví dụ:

```
var x = prompt("Hay nhap vao so be hon 100:");
```

```
if ( x > 50) {  
    alert("Ban vua nhap gia tri la: " + x + ", gia tri nay lon hon 50");  
}  
else{  
    alert("Ban vua nhap gia tri la: " + x);  
}
```

## □ Cú pháp

```
switch ( dieukien ) {  
    case giatri1:  
        //Thực hiện  
        break;  
    case giatri2:  
        //Thực hiện  
        break;  
    :  
    default:  
        //Thực hiện  
}
```

Sử dụng **break** để thoát khỏi lệnh switch, nếu không sẽ chuyển xuống thực hiện lệnh trong mệnh đề case tiếp theo

```
var diem = prompt("Hay nhap diem:")
switch ( diem ) {
    case "A":
        alert("Tuyet voi!");
        break;
    case "B":
    case "C":
        alert("Kha!");
        break;
    case "D":
    case "E":
    case "F":
        alert("Ban can co gang hon!");
        break;
    default:
        alert("Gia tri ban nhap vao khong hop le.");
}
```

```
var diem = prompt("Hay nhap diem:");  
switch ( diem ) {  
    case "A":  
        alert("Tuyet voi!");  
    case "B":  
    case "C":  
        alert("Kha!");  
    case "D":  
    case "E":  
    case "F":  
        alert("Ban can co gang hon!");  
    default:  
        alert("Gia tri ban nhap vao khong hop le.");  
}
```

*Hãy cho biết kết quả của đoạn mã trên nếu người dùng nhập vào giá trị **Tuyet voi!***



## ❑ Lệnh lặp không biết trước số lần lặp

- ❖ While

- ❖ Do ... While

## ❑ Lệnh lặp biết trước số lần lặp

- ❖ For

## □ Cú pháp

❖ Giống lệnh Do While ... Loop

```
while (dieukien) {  
    //Thực hiện  
}
```

## □ Lỗi lặp vô hạn

❖ Thay đổi giá trị điều kiện

```
var x = 0;  
while ( x < 10) {  
    //Thực hiện  
    x++;—————Thay đổi giá trị của x  
}
```

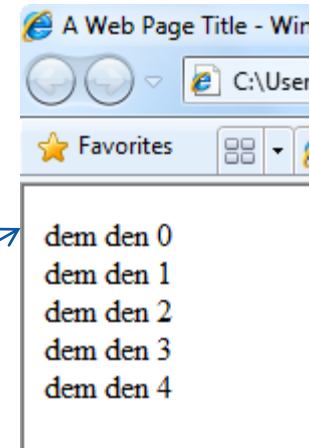
## □ Cú pháp

❖ Giống lệnh Do ... Loop While

```
do {  
    //Thực hiện  
} while ( đieukien )
```

❖ Ví dụ

```
var dem = 0;  
do {  
    document.write("dem den " + dem + "<br>");  
    dem++;  
}  
while (dem < 5);
```



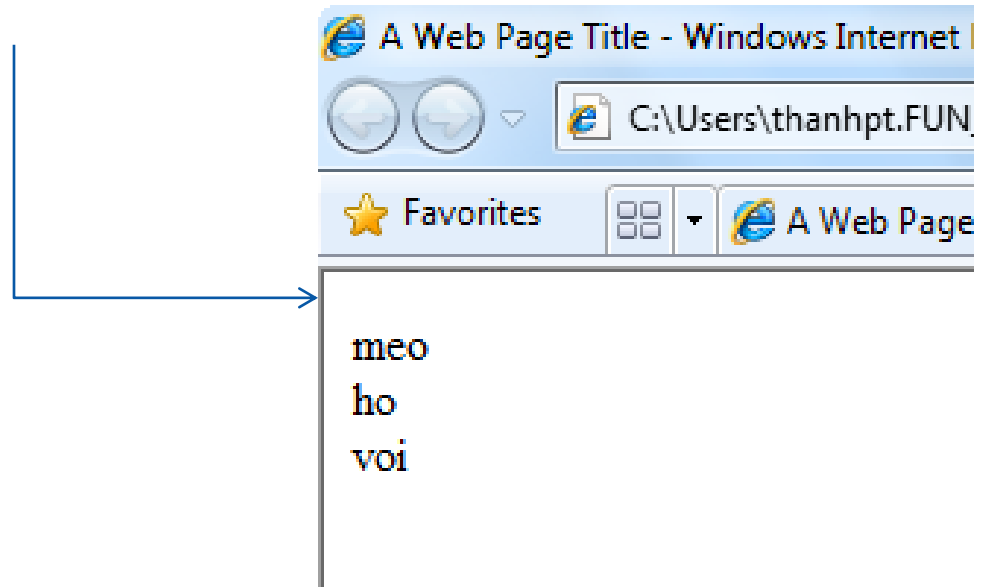
## □ Cú pháp

```
for (khoitao; dieukien; buoctang) {  
    //Thực hiện  
}
```

## □ Ví dụ

```
for (var x = 0; x < 10; x++) {  
    document.write(x + "</br>");  
}
```

```
var convat = new Array("meo", "ho", "voi");  
  
for (var i = 0; i < convat.length; i++) {  
    document.write(convat[i] + "<br>");  
}
```





## PHẦN 2

# LẬP TRÌNH JAVASCRIPT

## BÀI 3: CẤU TRÚC ĐIỀU KHIỂN, HÀM VÀ XỬ LÝ SỰ KIỆN

❑ Hàm để thực hiện một chức năng cụ thể

❑ Cú pháp

❖ Hàm không trả về giá trị (giống thủ tục Sub)

```
function tenham (thamso1, thamso2, ..., thamson) {  
    //Thực hiện  
}
```

❖ Hàm trả về giá trị (giống thủ tục Function)

```
function tenham (thamso1, thamso2, ..., thamson) {  
    //Thực hiện  
    return giatritrave;  
}
```

## □ Định nghĩa hàm

```
function myFunction (loichao1, loichao2 ) {  
    alert("Bien thu nhat la: " + loichao1);  
    alert("Bien thu hai la: " + loichao2);  
}
```

## □ Gọi hàm

```
myFunction ("Hi", "Hello");
```

- ❖ Chú ý: lời gọi hàm có thể ở một thẻ script khác với thẻ script định nghĩa hàm



## ❑ Định nghĩa hàm

```
function nhanHaiSo (x, y) {  
    return x*y;  
};
```

## ❑ Gọi hàm

```
var tich = nhanHaiSo (3, 4);  
alert (tich);
```

❑ Là hộp thoại để nhập hồi đáp từ phía người dùng

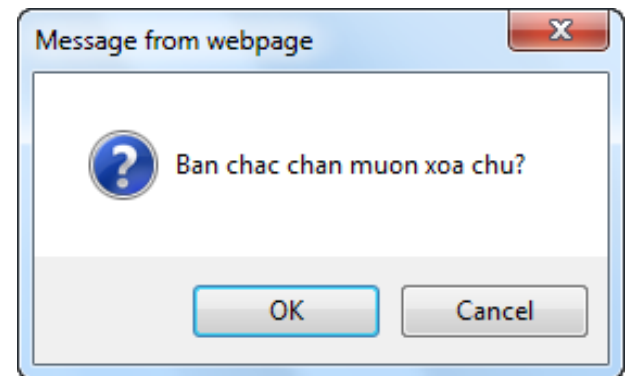
❖ Lời gọi hàm: Confirm(thông\_điệp )

❖ Hàm trả về hồi đáp của người dùng

➤ Trả về True nếu người dùng nhấn vào OK

➤ Trả về False nếu người dùng nhấn vào Cancel

```
var ok = confirm("Ban chac chan muon xoa chu?");  
if(ok == true){  
    doSometing();  
}  
else{  
    doAnythingElse();  
}
```



```
function xacNhan(traloi) {  
    var ketQua = "";  
    if (traloi) {  
        ketQua = "Tuyet voi. Chuc ban chien thang!";  
    } else {  
        ketQua = "Hen gap lai ban nhe!";  
    }  
    return ketQua;  
}
```

```
var traloi = confirm("Ban se choi game chu?");  
var thongbao = xacNhan(traloi);  
alert (thongbao);
```

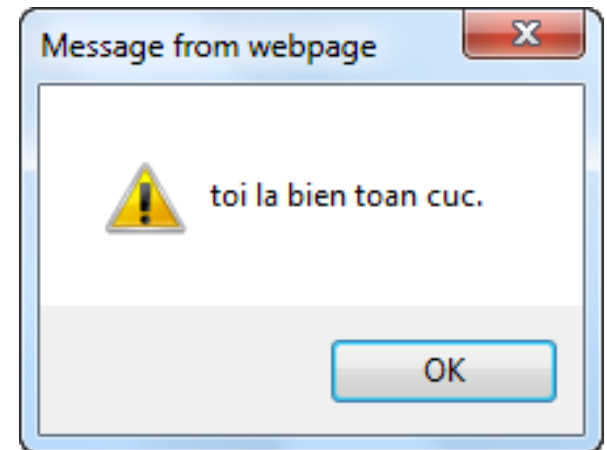
## ❑ Biến cục bộ

- ❖ Biến được khai báo trong hàm
- ❖ Chỉ được tham chiếu đến trong phạm vi khai báo

## ❑ Biến toàn cục

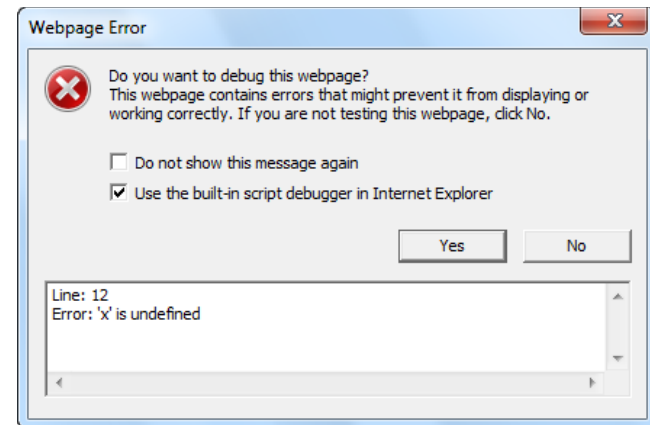
- ❖ Biến được khai báo ngoài hàm
- ❖ Có thể tham chiếu đến từ bất cứ đâu

```
<html>
<head>
  <title>Scoping Example</title>
  <script type="text/javascript">
    var x = "toi la bien toan cuc.";
  </script>
</head>
<body>
  <script type="text/javascript">
    alert(x);
  </script>
</body>
</html>
```



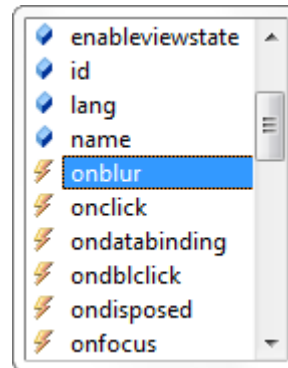
```
<html><head>  
  <title>Scoping Example</title>  
  <script type="text/javascript">  
    function() {  
      var x = "toi la bien cuc bo.";
```

```
    }  
  
  </script>  
</head>  
<body>  
  <script type="text/javascript">  
    alert(x);  
  </script>  
</body></html>
```



- ❑ Tất cả các element trên trang web đều có một tập các sự kiện tương ứng

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Trans
2 <html >
3 <head>
4 <script type="text/javascript">
5     function displayDate() {
6         alert(Date());
7     }
8 </script>
9 </head>
10 <body>
11 <button type="button" on>Display Date</button>
12 </body>
13 </html>
14
```



Chú ý: Cùng một element, các browser khác nhau hỗ trợ các tập sự kiện khác nhau

## ❑ onClick

- ❖ Được kích hoạt khi nhấn chuột vào một element

## ❑ onLoad và on Unload

- ❖ Được kích hoạt khi người dùng vào hoặc thoát khỏi trang web

## ❑ onFocus, onBlur, onChange

- ❖ Được kích hoạt khi các trường nhận được focus, mất focus hay được thay đổi giá trị

## ❑ onMouseOver

- ❖ Được kích hoạt khi người dùng di chuột lên một element HTML trên form

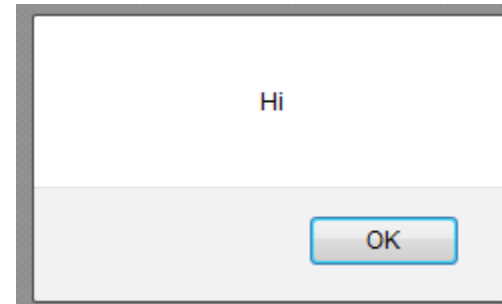


❑ Click vào bất kỳ đâu trên trang web

```
<html>  
<head>  
</head>  
<body onclick="alert('Hi')">  
</body>  
</html>
```

Sự kiện

Xử lý sự kiện



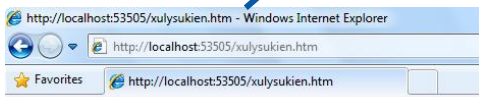
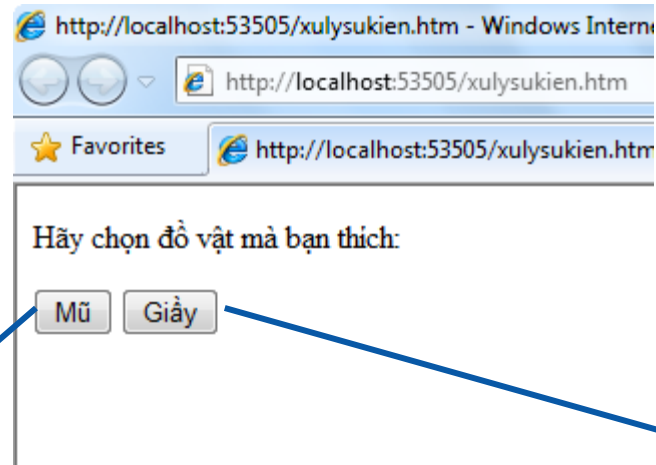
❑ Có thể thêm nhiều dòng lệnh

```
<html >  
<head>  
</head>  
<body onclick="alert('Hi'); alert('Hello');">  
</body>  
</html>
```

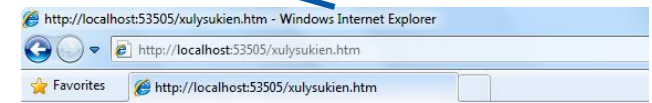
**➔ Trong trường hợp xử lý phức tạp cho sự kiện????**

## ❑ Sử dụng hàm để thực hiện các xử lý cho sự kiện

```
<html >
<head>
<script type="text/javascript">
    function hienThiLoiChao() {
        alert("Hi");
        alert("Hello");
    }
</script>
</head>
<body>
<button type="button" onclick="hienThiLoiChao();">
    Hien thi loi chao
</button>
</body>
</html>
```



**Người dùng  
nhấn vào Mũ**



**Người dùng  
nhấn vào Giày**

```
<html ><head>
  <script type="text/javascript">
    function hienThiAnh(dovat) {
      if (dovat == "mu") {
        document.write("<img src = 'mu.jpg'>");
      } else {
        document.write("<img src = 'giay.jpg'>");
      }
    }
  </script>
</head>
<body>
<p> Hãy chọn đồ vật mà bạn thích:</p>
  <input type="button" value="Mũ" onclick="hienThiAnh('mu');"/>
  <input type="button" value="Giày" onclick="hienThiAnh('giay');"/>
</body>
</html>
```

- ❑ JavaScript cung cấp các phương thức để xử lý các sự kiện thời gian
- ❑ Các phương thức này thuộc đối tượng window
- ❑ Một số phương thức quan trọng

| Phương thức   | Giải thích   |
|---------------|--|
| setTimeout    | Thực hiện công việc sau một khoảng thời gian trong tương lai |
| clearTimeout  | Hủy bỏ setTimeout trước đó                                   |
| setInterval   | Thực hiện lặp lại công việc sau một khoảng thời gian         |
| clearInterval | Hủy bỏ setInterval   |

## □ Cú pháp

```
var t=setTimeout("Lệnh_javascript",số_mili_giây );
```

- ❖ Lệnh\_javascript: Mã thực thi hoặc lời gọi hàm
- ❖ mili\_giây: Sau thời gian này mã sẽ được thực hiện
- ❖ setTimeout() trả về giá trị, giá trị được lưu trong biến t. Nếu muốn hủy bỏ setTimeout, sử dụng hàm clearTimeout và truyền vào đối số t
- ❖ Cú pháp tương tự đối với setInterval()

```
<html>
<head>
<script type="text/javascript">
    function onClickEvent() {
        var t = setTimeout("alert('Hi');", 1000);
    }
</script>
</head>
<body>
    <button onclick="onClickEvent();">
        Click here!</button>
</body>
</html>
```



```
<html>
<head>
<script type="text/javascript">
    function onClickEvent() {
        var t = setInterval("alert('Hi');", 1000);
    }
</script>
</head>
<body>
    <button onclick="onClickEvent();">
        Click here!</button>
</body>
</html>
```

Hãy so sánh sự khác nhau với ví dụ setTimeout

- ❑ JavaScript cung cấp 3 cấu trúc điều khiển: Cấu trúc tuần tự, cấu trúc lựa chọn và cấu trúc lặp
- ❑ Cấu trúc lặp gồm lệnh lựa chọn đơn (if), lệnh lựa chọn kép (if...else), lệnh đa lựa chọn (switch)
- ❑ Cấu trúc lặp gồm các lệnh lặp không biết số lần lặp (while, do...while) và lệnh lặp biết số lần lặp (for)
- ❑ Hàm thực hiện một chức năng cụ thể. Hàm có thể trả về giá trị hoặc không trả về giá trị
- ❑ Biến có phạm vi cục bộ và toàn cục

- ❑ Hàm Confirm là hàm được xây dựng sẵn dùng để lấy thông tin hồi đáp từ người dùng
- ❑ JavaScript cung cấp sự kiện cho các element của trang web. Mỗi element có một tập các sự kiện khác nhau
- ❑ JavaScript cũng cung cấp các phương thức để xử lý các sự kiện thời gian

# FPT POLYTECHNIC



KẾT THÚC