



# PHẦN 1

## LẬP TRÌNH JAVASCRIPT

### BÀI 6: TẠO HIỆU ỨNG VÀ VALIDATE FORM

- ❑ Giới thiệu về mô hình Document Object Model
- ❑ Giới thiệu về HTML DOM
- ❑ Cấu trúc DOM
- ❑ Thuộc tính của node
- ❑ Phương thức của node
  - ❖ Truy cập đến node
  - ❖ Thêm node
  - ❖ Xóa node
- ❑ Đối phó với các trình duyệt khác nhau

- ❑ Truy cập và thay đổi style của các element
- ❑ Viết mã tạo các hiệu ứng
  - ❖ Hiệu ứng Image Rollover
  - ❖ Hiệu ứng SlideShow
- ❑ Thao tác với các điều khiển
  - ❖ SelectBox
  - ❖ CheckBox
  - ❖ RadioButton
  - ❖ Form
- ❑ Kiểm tra hợp lệ (validate) cho Form



- ❑ CSS định nghĩa màu sắc, font, layout... cho trang web
- ❑ CSS bao gồm một tập các thuộc tính, mỗi thuộc tính có một tập giá trị nhất định  
`font-family:arial;`
- ❑ Selector cho biết thành phần nào sẽ được áp dụng  
`selector { property:value; }`
- ❑ Có 3 loại Selector
  - ❖ Thẻ `h1 { font-family:arial; }`
  - ❖ Lớp `.tenClass { font-family:arial; }`
  - ❖ ID `.tenId { font-family:arial; }`

- ❑ Có thể sử dụng JavaScript để **thay đổi style** cho trang web
- ❑ JavaScript cung cấp **đối tượng style** cho mỗi **element** của trang web để thay đổi style của trang web
- ❑ Sử dụng thuộc tính **style** của mỗi **element** để truy cập đến đối tượng style  

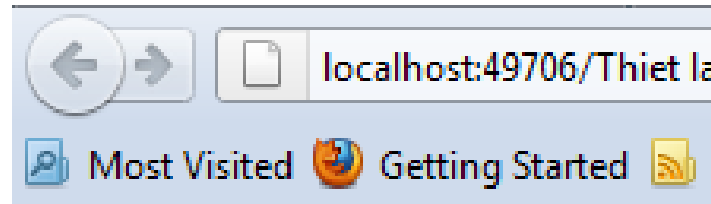
```
var hTieuDe = document.getElementById("hTieuDe");  
hTieuDe.style.fontFamily = "arial"
```
- ❑ Đối tượng style trong JavaScript có các thuộc tính tương ứng với các thuộc tính của CSS
  - ❖ Chú ý: Với các thuộc tính CSS có dấu gạch ngang thì sẽ được bỏ dấu gạch ngang và viết hoa chữ cái của từ sau dấu gạch ngang (font-family → fontFamily)

- ❑ Có thể thiết lập style cho các thành phần bằng
  - ❖ ID
  - ❖ Thẻ
  - ❖ Class

## □ Các bước thực hiện

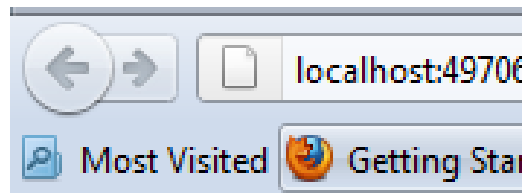
- ❖ Bước 1. Dùng ID để truy cập đến các element
- ❖ Bước 2. Sử dụng thuộc tính style để thiết lập style cho element đó

```
var hTieuDe = document.getElementById("hTieuDe");  
hTieuDe.style.fontFamily = "arial"
```



Thiet lap style bang ID

Xanh Do



Thiet lap style bang ID

Xanh Do



Thiet lap style bang ID

Xanh Do

Xem CSS



```
<html >
<head>
  <script type="text/javascript">
    function xanhButtonClick() {
      var pText = document.getElementById("pText");
      pText.style.color = "blue";
    }
    function doButtonClick() {
      var pText = document.getElementById("pText");
      pText.style.color = "red";
    }
  </script>
</head>
<body>
<p id="pText">Thiet lap style bang ID</p>
<button onclick="xanhButtonClick();">Xanh</button>
<button onclick="doButtonClick();">Do</button>
</body>
</html>
```

## ❑ Vấn đề nảy sinh:

- ❖ Muốn thay đổi style cho nhiều phần tử

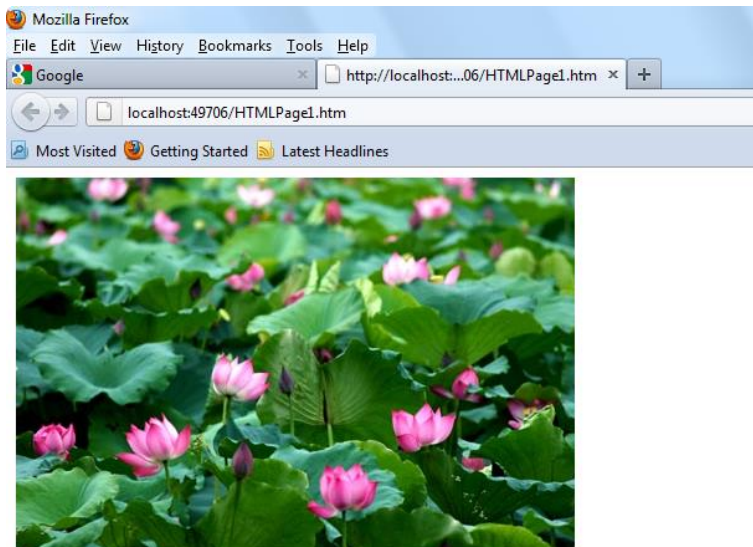
```
<p id = "pText">Thiet lap style cho nhieu thanh phan</p>  
<p id = "p1">Thanh phan 1</p>  
<p id = "p2">Thanh phan 2</p>  
<p id = "p3">Thanh phan 3</p>
```

## ❑ Giải pháp:

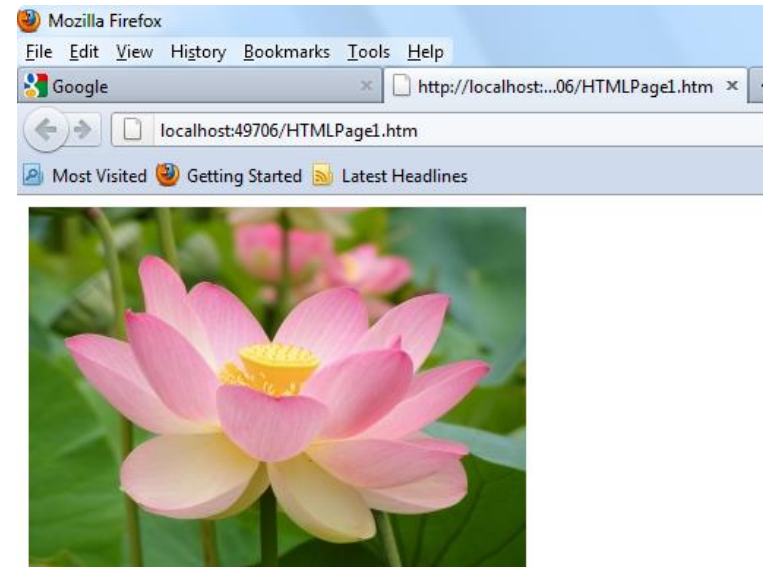
- ❖ Sử dụng phương thức `getElementsByName` hoặc `getElementsByTagName` để lấy một nhóm các phần tử
- ❖ Sử dụng vòng `For` để duyệt qua các phần tử

```
var pAr = document.getElementsByTagName("p");  
for (var i = 0; i < pAr.length; i++) {  
    pAr[i].style.color = "blue";  
}
```

❑ Image Rollover là hiệu ứng thay đổi ảnh khi di chuột lên ảnh



**Ảnh hiện lên khi  
chạy ứng dụng**



**Ảnh hiện lên khi di  
chuột lên ảnh**

- ❑ Thực hiện hiệu ứng này bằng cách xử lý sự kiện `onMouseOver` và `onMouseOut` cho thẻ `img`
  - ❖ `onMouseOver`: sự kiện được kích hoạt khi người dùng di chuột lên ảnh
  - ❖ `onMouseOut`: sự kiện được kích hoạt khi người dùng di chuột ra ngoài ảnh

```
<html>
<head>
<script type="text/javascript">
    function onMouseOverEvent() {
        document.img_hoa.src = "hoaover.jpg";
    }
    function onMouseOutEvent() {
        document.img_hoa.src = "hoaout.jpg";
    }
</script>
</head><body>
    
</body>
</html>
```

Truy cập đến phần tử  
bằng attribute name

- ❑ Vấn đề: Lần đầu tiên di chuột lên ảnh, sẽ mất một thời gian ảnh mới được load mặc dù trang web đã được mở ra từ lâu
- ❑ Nguyên nhân: Khi người dùng di chuột lên, ảnh mới được load
- ❑ Giải pháp: Load trước ảnh
  - ❖ Kỹ thuật:
    - Tạo đối tượng image cho mỗi ảnh muốn load trước
    - Gán đường dẫn của ảnh cho thuộc tính src của đối tượng ảnh đó
  - ❖ Giải thích:
    - Khi gập lệnh `imageObject.src="link_cua_anh"` thì ảnh được load ngầm bên dưới, máy tính tiếp tục thực hiện các lệnh sau lệnh này
    - ➔ Giải pháp này không làm cho việc load trang web chậm đi

```
<html><head>
<script type="text/javascript">
    var hoaover = new Image();
    var hoaout = new Image();
    function loadAnh() {
        hoaover.src = "hoaover.jpg";
        hoaout.src = "hoaout.jpg";
    }
    function onMouseOverEvent() {
        document.img_hoa.src = hoaover.src
    }
    function onMouseOutEvent() {
        document.img_hoa.src = hoaout.src
    }
</script></head>
<body onload = "loadAnh();">
    
</body></html>
```

## □ Slide Show là gì

- ❖ Là hiệu ứng ảnh được hiện ra thay thế cho ảnh trước đó
- ❖ Có thể có thanh điều khiển cho người dùng



Thanh điều khiển

<< Previous Play > Stop Next >>



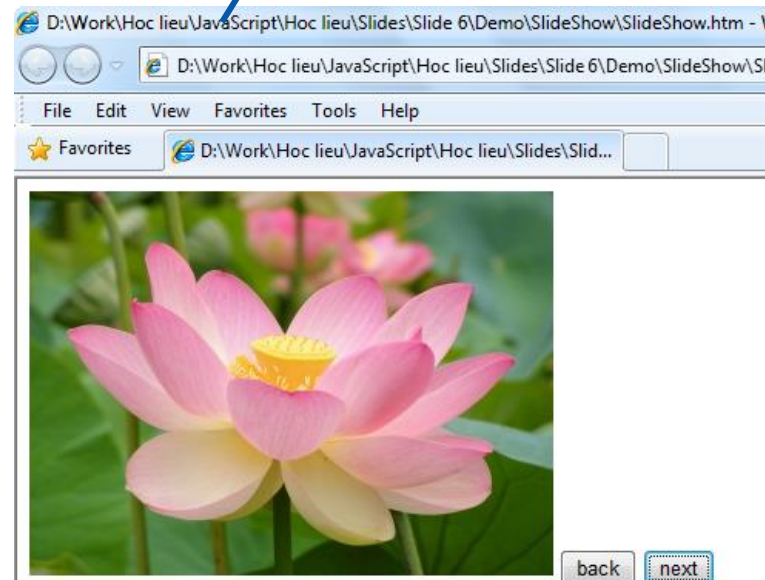
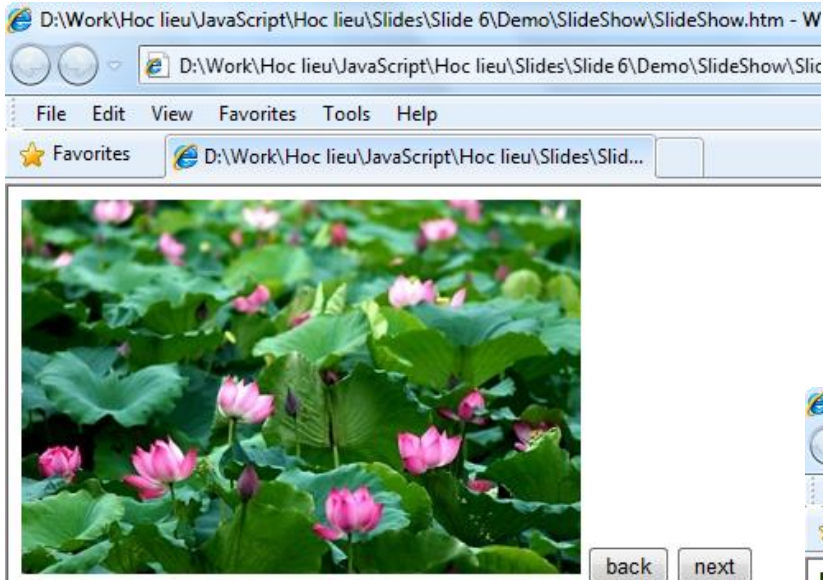
## □ Các bước để làm Slide Show

- ❖ Load trước tất cả các ảnh
- ❖ Xử lý sự kiện cho button Next
- ❖ Xử lý sự kiện cho button Back

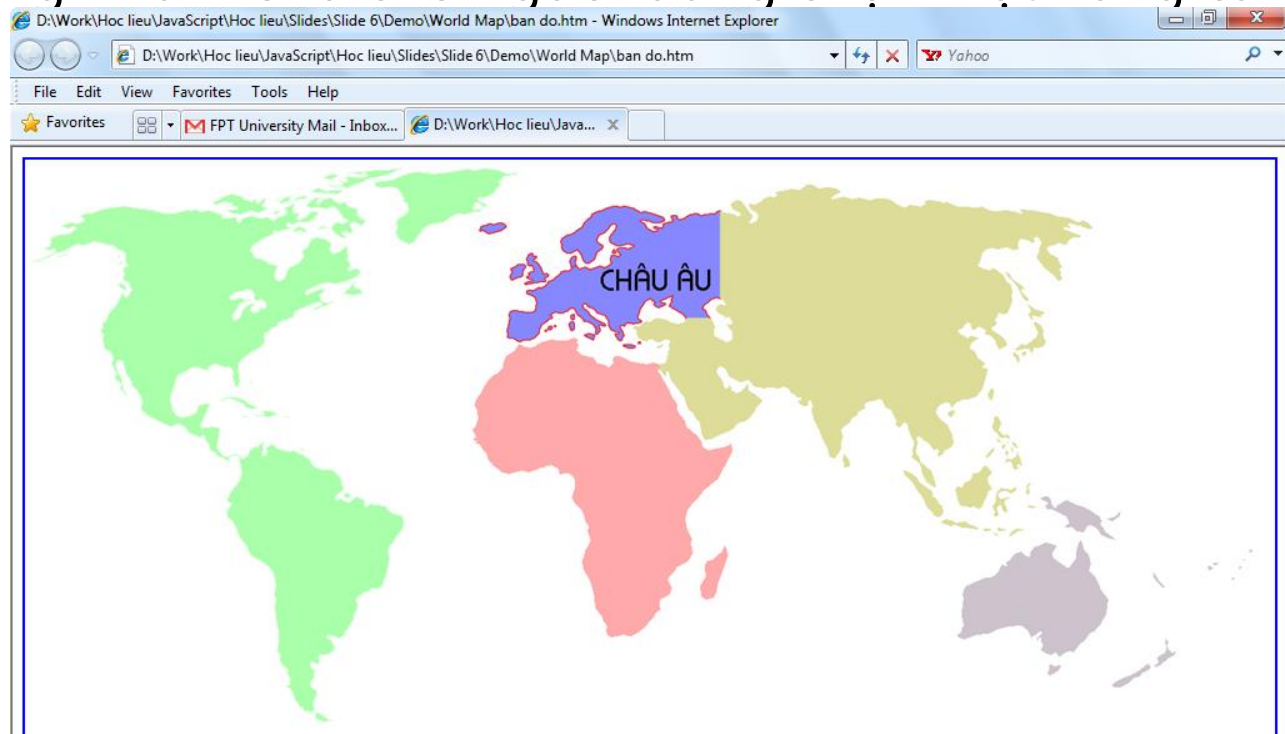
```
<body onload = "loadAnh();" >  
    
  <input type="button" value="back" onclick="back();" />  
  <input type="button" value = "next" onclick="next();"/>  
</body>
```

```
var anhAr = [];  
var currentIndex=0;  
function loadAnh() {  
    for (var i = 0; i < 6; i++) {  
        anhAr[i] = new Image();  
        anhAr[i].src = "anh"+i+".jpg";  
    }  
}  
function next() {  
    if (currentIndex < 4) {  
        currentIndex++;  
        document.getElementById("hoa").src = anhAr[currentIndex].src;  
    }  
}  
function back() {  
    if (currentIndex > 0) {  
        currentIndex--;  
        document.getElementById("hoa").src = anhAr[currentIndex].src;  
    }  
}
```

Ảnh được thay đổi  
sau khi nhấn Next



- ❑ Bản đồ ảnh là ảnh có những vùng cụ thể được định nghĩa để thực hiện một hành động nào đó khi người dùng tác động lên
  - ❖ Cho người dùng chọn khu vực (đất nước) mà người truy cập cư trú
  - ❖ Sử dụng như menu cho người dùng chọn mặt hàng cần mua



- ❑ HTML cung cấp thẻ map để tạo bản đồ ảnh
- ❑ Thẻ map đi liền sau thẻ img, thuộc tính usemap của thẻ img có giá trị bằng giá trị thuộc tính name của thẻ map tương ứng
- ❑ Thẻ area trong thẻ map để chỉ ra các vùng trên bản đồ. Vùng trên bản đồ được chỉ ra bằng thuộc tính coords

Thuộc tính usemap của ảnh có giá trị bằng  
giá trị của thuộc tính name của thẻ map

```
</p>
<map name="bando">
  <area shape="poly" coords="" href="#"
onmouseover="document.getElementById('bando').src='bando.chaumy.jpg'"/>
  <area shape="poly" coords="" href="#"
onmouseover="document.getElementById('bando').src='bando.chauphi.jpg'"/>
  <area shape="poly" coords="" href="#"
onmouseover="document.getElementById('bando').src='bando.chauau.jpg'"/>
  <area shape="poly" coords="" href="#"
onmouseover="document.getElementById('bando').src='bando.chaua.jpg';"/>
  <area shape="poly" coords="" href="#"
onmouseover="document.getElementById('bando').src='bando.chauuc.jpg';" />
</map>
```



## PHẦN 2

# LẬP TRÌNH JAVASCRIPT

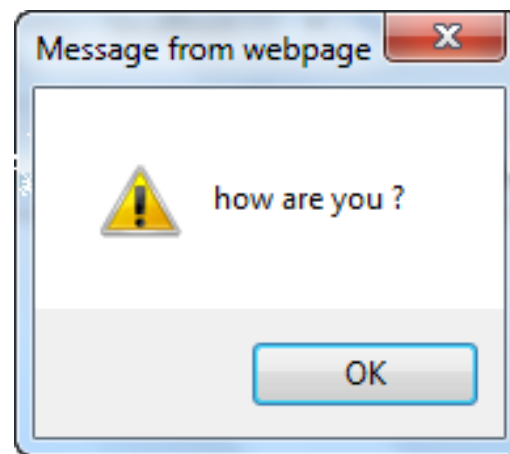
## BÀI 6: TẠO HIỆU ỨNG VÀ VALIDATE FORM

- ❑ JavaScript cung cấp các điều khiển để tương tác với người dùng
  - ❖ Select Box
  - ❖ CheckBox
  - ❖ Radio Button
  - ❖ Form



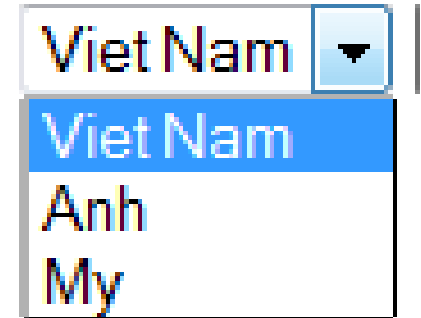
- ❑ Một số element được định nghĩa thêm thuộc tính name (ví dụ như các điều khiển checkbox, radio button...)
- ❑ Có thể sử dụng attribute name để truy cập đến một nhóm các element có cùng giá trị attribute name
- ❑ Phân biệt id và name
  - ❖ id là duy nhất, mỗi id đại diện cho 1 element
  - ❖ Nhiều phần tử có cùng giá trị của attribute name, mỗi giá trị name đại diện cho một nhóm các phần tử
- ❑ Sử dụng phương thức `getElementsByName(name)` để lấy về một mảng các element có cùng thuộc tính name

```
<html>
<body >
<input type="checkbox" name="test" value="how"/>
<input type="checkbox" name="test" value="are"/>
<input type="checkbox" name="test" value="you"/>
<input type="checkbox" name="test" value="?"/>
<script type="text/javascript">
    var ckAr = document.getElementsByName("test");
    var str = "";
    for (var i = 0; i < ckAr.length; i++) {
        str = str + " " + ckAr[i].value
    }
    alert(str);
</script>
</body>
</html>
```



## □ Select Box

```
<select id="country">  
  <option selected="selected">Viet Nam</option>  
  <option value="Anh">Anh</option>  
  <option value="My">My</option>  
</select>>
```



- Trên Firefox, truy cập đến giá trị được chọn của Select Box bằng thuộc tính value

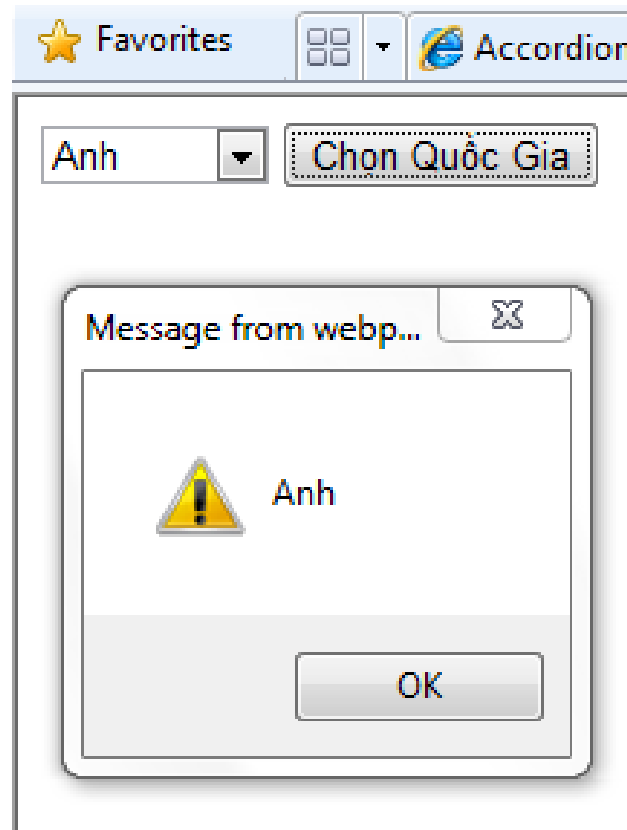
```
document.getElementById("country").value
```

- Trên Firefox cũng có thể thiết lập giá trị được chọn cho Select box bằng JavaScript

```
document.getElementById("country").value = "Mỹ"
```

## ❑ Bài toán

- ❖ Khi nhấn vào button Chọn Quốc Gia sẽ hiển thị nước được chọn



```
<html ><head>
<script type="text/javascript">
    function chonQuocGiaClick(){
        var name = document.getElementById("country").value;
        alert(name);
    }
</script>
</head>
<body>
    <select id="country">
        <option selected="selected">Viet Nam</option>
        <option value="Anh">Anh</option>
        <option value="My">My</option>
    </select>
    <input type ="button" value="Chọn Quốc Gia"
        onclick="chonQuocGiaClick()" />
</body></html>
```

## □ Bài toán

```
<body>
```

```
<p><input type="button" value="Chau A" onclick="chauAClick();" />
```

```
<input type="button" value="Chau Au" onclick="chauAuClick();" />
```

```
<input type="button" value="Chau My" onclick="chauMy();" /> </p>
```

```
<p><select id="country">
```

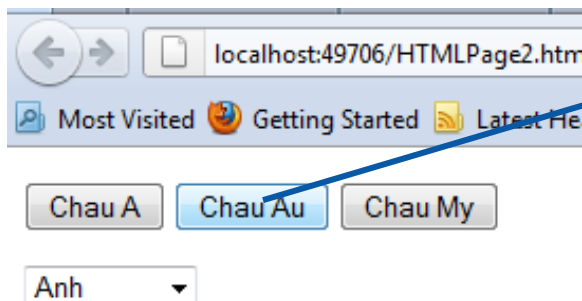
```
<option selected="selected">Viet Nam</option>
```

```
<option value="Anh">Anh</option>
```

```
<option value="My">My</option>
```

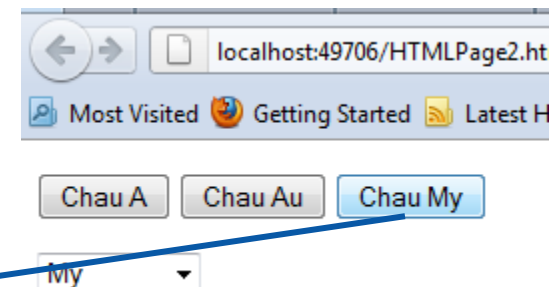
```
</select></p>
```

```
</body>
```



Click vào  
Chau Au

Click vào  
Chau My



```
<script type="text/javascript">
    function chauAClick() {
        document.getElementById("country").value = "Viet Nam";
    }
    function chauAuClick() {
        document.getElementById("country").value = "Anh";
    }
    function chauMy() {
        document.getElementById("country").value = "My";
    }
</script>
```

- ❑ Dùng để người dùng lựa chọn
- ❑ CheckBox có hai trạng thái là được chọn và không được chọn

```
<input type="checkbox" value="Kem que"  
name="ckKem" checked="checked" />
```



- ❑ Trong JavaScript, sử dụng thuộc tính checked của CheckBox để xác định và thiết lập trạng thái cho CheckBox
  - ❖ Giá trị checked = true: CheckBox được chọn
  - ❖ Giá trị checked = false: CheckBox không được chọn



## ☐ Bài tập về nhà:

- ❖ Tìm cách thiết lập và truy cập giá trị được chọn của CheckBox trên IE
- ❖ Viết lại 2 ví dụ trên để có thể chạy trên cả IE và Firefox

```
<html >
```

```
<body>
```

```
  Cac loai kem trong cua hang:<br />
```

```
  <input type="checkbox" value="Kem que" name="ckKem"/>Kem que <br />
```

```
  <input type="checkbox" value="Kem dau" name="ckKem"/>Kem dau <br />
```

```
  <input type="checkbox" value="Kem bo" name="ckKem"/>Kem bo <br />
```

```
  <input type="checkbox" value="Kem vani" name="ckKem"/>Kem vani <br />
```

```
  <input type="button" onclick="chonKem();" value="Chon kem" />
```

```
  <p id="pKemDaChon"></p>
```

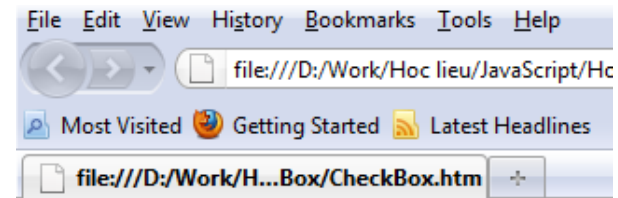
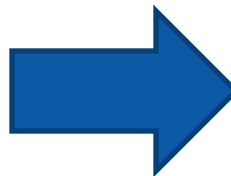
```
</body></html>
```



Cac loai kem trong cua hang:

- ☐ Kem que
- ☐ Kem dau
- ☐ Kem bo
- ☐ Kem vani
- ☐ Kem dao

Chon kem



Cac loai kem trong cua hang:

- ☒ Kem que
- ☒ Kem dau
- ☐ Kem bo
- ☐ Kem vani
- ☐ Kem dao

Chon kem

Nhung loai kem ban da chon la:

Kem que  
Kem dau

Ứng dụng liệt kê  
những loại kem  
được chọn

```
<html ><head>
  <script type="text/javascript">
    function chonKem() {
      var str = "Nhưng loại kem bạn đã chọn là:<br/>";
      var kemAr = document.getElementsByName("ckKem");
      for (var i = 0; i < kemAr.length; i++) {
        if (kemAr[i].checked) {
          str= str + kemAr[i].value + "<br/>"
        }
      }
      document.getElementById("pKemDaChon").innerHTML = str;
    }
  </script>
</head><body>
  Các loại kem trong cửa hàng:<br />
  <input type="checkbox" value="Kem que" name="ckKem"/>Kem que <br />
  <input type="checkbox" value="Kem dâu" name="ckKem"/>Kem dâu <br />
  <input type="checkbox" value="Kem bơ" name="ckKem"/>Kem bơ <br />
  <input type="checkbox" value="Kem vani" name="ckKem"/>Kem vani <br />
  <input type="button" onclick="chonKem();" value="Chọn kem" />
  <p id="pKemDaChon"></p></body></html>
```

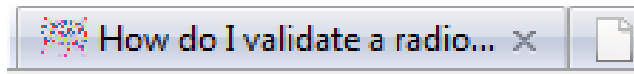
- ❑ Dùng để người dùng lựa chọn
- ❑ CheckBox có hai trạng thái là được chọn và không được chọn
- ❑ Thường được sử dụng theo nhóm, trong nhóm chỉ một Radio được chọn

```
<input type="radio" value="Lon" name="rdSize" checked/>Lon  
<input type="radio" value="Vua" name="rdSize" />Vua<br />  
<input type="radio" value="Nho" name="rdSize" />Nho<br />
```



- ❑ Các radio có cùng tên thuộc cùng một nhóm

# DEMO SỬ DỤNG RADIO BUTTON



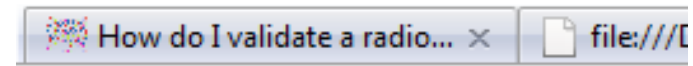
Cac loai kem trong cua hang:

- ☒ Kem que
- ☒ Kem dau
- ☒ Kem bo
- ☒ Kem vani
- ☐ Kem dao

Hay chon kích thước kem:

- ☒ Lon
- ☐ Vua
- ☐ Nho

Chon kem



Cac loai kem trong cua hang:

- ☒ Kem que
- ☒ Kem dau
- ☒ Kem bo
- ☒ Kem vani
- ☐ Kem dao

Hay chon kích thước kem:

- ☒ Lon
- ☐ Vua
- ☐ Nho

Chon kem

Nhung loai kem ban da chon la:

Kem que  
Kem dau  
Kem bo  
Kem vani

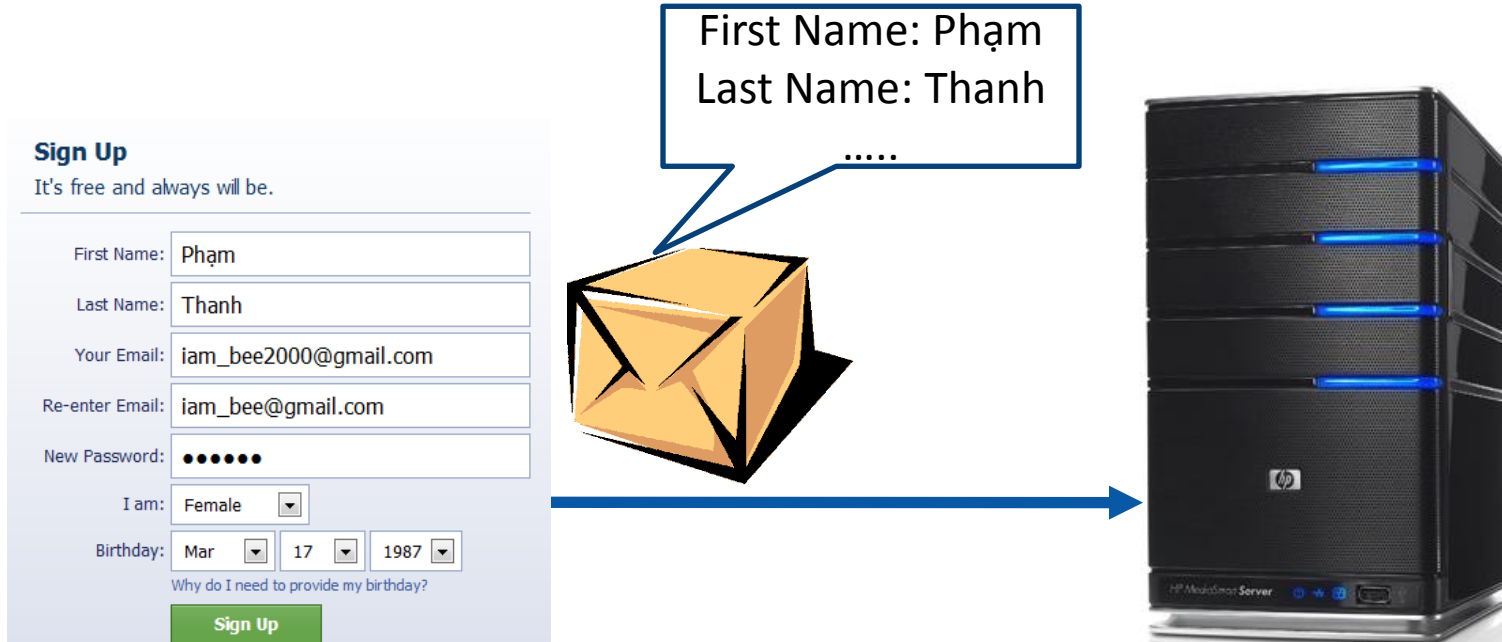
Kích thước bạn đã chọn là: Lon

**Hiển thị loại kem  
được chọn**

```
var size="";  
var sizeAr = document.getElementsByName("rdSize");  
for (var i = 0; i < sizeAr.length; i++) {  
    if (sizeAr[i].checked) {  
        size = sizeAr[i].value;  
    }  
}
```

```
str = str + "Kích thước bạn đã chọn là: " + size;
```

- ❑ Dùng để gửi dữ liệu lên server
- ❑ Khi người dùng nhấn vào Button Submit thì dữ liệu sẽ được đóng gói và gửi lên Server
- ❑ Form sử dụng phương thức GET hoặc POST để gửi dữ liệu lên Server



## ❑ Kiểm tra hợp lệ cho form:

- ❖ Kiểm tra xem thông tin cần thiết đã được điền vào form chưa?
- ❖ VÀ Thông tin điền vào có đúng yêu cầu hay không?

## ❑ Kiểm tra hợp lệ cho form rất cần thiết để đảm bảo tính đúng đắn của CSDL

The image shows the Facebook sign-up page. At the top, the Facebook logo is on the left, and the login fields (Email, Password, Log In) are on the right. Below the logo, the text "Facebook helps you connect and share with the people in your life." is displayed above a world map with user avatars. On the right, the "Sign Up" section is visible, with the text "It's free and always will be." followed by input fields for First Name (Phạm), Last Name (Thanh), Your Email (iam\_bee2000@gmail.com), Re-enter Email (iam\_bee@gmail.com), New Password (masked with dots), I am (Female), and Birthday (Mar 17 1987). A green "Sign Up" button is at the bottom of the form. A red error message box at the bottom of the page states: "Your emails do not match. Please try again." A blue arrow points from the text "Thông báo lỗi" (Error notification) to this red box.

**facebook**

Email  Password  **Log In**

☐ Keep me logged in [Forgot your password?](#)

**Facebook helps you connect and share with the people in your life.**

**Sign Up**  
It's free and always will be.

First Name:

Last Name:

Your Email:

Re-enter Email:

New Password:

I am:

Birthday:

[Why do I need to provide my birthday?](#)

**Sign Up**

Your emails do not match. Please try again.

**Thông báo lỗi**



- ❑ Trước đây việc kiểm tra hợp lệ được thực hiện trên server (server-side validation)
- ❑ Javascript ra đời, thực hiện kiểm tra hợp lệ trên browser (client-side validation) trước khi gửi dữ liệu lên server
- ❑ Với một số lượng lớn người truy cập, kiểm tra hợp lệ trên browser sẽ giảm tải cho server
- ❑ Chú ý: Vẫn phải kiểm tra hợp lệ trên server vì người dùng có thể disable javascript

- ❑ Mỗi Form có một hoặc nhiều button Submit
- ❑ Sự kiện onSubmit của form
  - ❖ Được kích hoạt khi người dùng nhấn vào button Submit
  - ❖ Nếu onSubmit có giá trị trả về là True, dữ liệu được gửi lên Server
  - ❖ Nếu onSubmit có giá trị trả về là False, dữ liệu không được gửi lên Server

- ❑ Viết hàm thực hiện việc kiểm tra hợp lệ
  - ❖ Nếu thông tin không hợp lệ, trả về giá trị False
  - ❖ Nếu thông tin hợp lệ, trả về True
- ❑ Hàm này sẽ được dùng xử lý sự kiện cho `onSubmit` cho Form

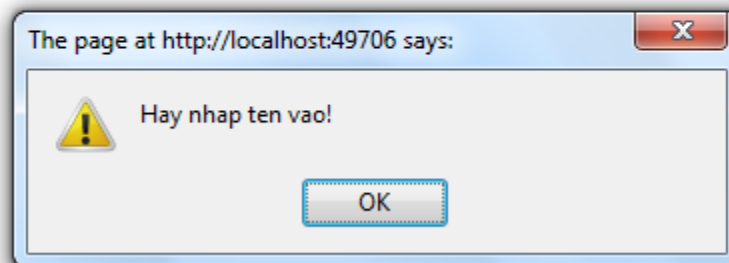
❑ Khi nhấn vào button Submit, form sẽ được kiểm tra hợp lệ

- ❖ Nếu tên chưa được nhập, sẽ hiển thị thông báo lỗi
- ❖ Nếu tên được nhập, hiển thị lời chào

```
<html><body>  
  <form action="#"      onsubmit="return validateForm()" >  
    <p>Name: <input type = "text" name="txtName" /></p>  
    <p><input type="submit" value="Submit" /></p>  
  </form>  
</body></html>
```

Name:

Submit



# DEMO KIỂM TRA HỢP LỆ CHO FORM

```
<html><head>
<script type="text/javascript">
    function validateForm() {
        var name = document.forms[0].txtName.value;
        if (name == "") {
            alert("Hay nhap ten vao!");
            return false;
        }
        else {
            alert("Hi " + name + " !" );
            return true;
        }
    }
</script></head>
<body>
    <form action="#"      onsubmit="return validateForm()" >
    <p>Name: <input type = "text" name="txtName" /></p>
    <p><input type="submit" value="Submit" /></p>
    </form></body></html>
```

Trả về False nếu dữ liệu không hợp lệ

Trả về True nếu dữ liệu hợp lệ

Xử lý sự kiện onSubmit cho Form

- ❑ Có thể truy cập đến Form bằng các cách sau
  - ❖ Chỉ số của Form: `Document.Forms[0]`
  - ❖ Tên Form: `Document.Forms["TenForm"]`
  - ❖ Chỉ số ID của Form: `Document.Forms["id"]`

- ❑ JavaScript cung cấp đối tượng style trong mỗi element để thay đổi style cho các element
- ❑ Có thể dùng JavaScript để tạo các hiệu ứng như Rollover, Slide Show
- ❑ Khi tạo hiệu ứng Rollover, khi di chuột lên ảnh, ảnh mới được load. Vì vậy, nên load trước ảnh
- ❑ JavaScript cung cấp các phương thức và thuộc tính để thao tác với các điều khiển như SelectBox, CheckBox, Radio Button và Form
- ❑ Radio Button thường được sử dụng theo nhóm. Mỗi nhóm chỉ có một Radio được chọn. Những Radio nào cùng attribute name được xem là cùng một nhóm
- ❑ Mỗi Form có một button có type submit. Khi button này được nhấn, sẽ kích hoạt sự kiện onSubmit của Form
- ❑ Nếu xử lý sự kiện onSubmit trả về True, dữ liệu được gửi đi. Nếu trả về False, dữ liệu không được gửi đi

# FPT POLYTECHNIC



KẾT THÚC