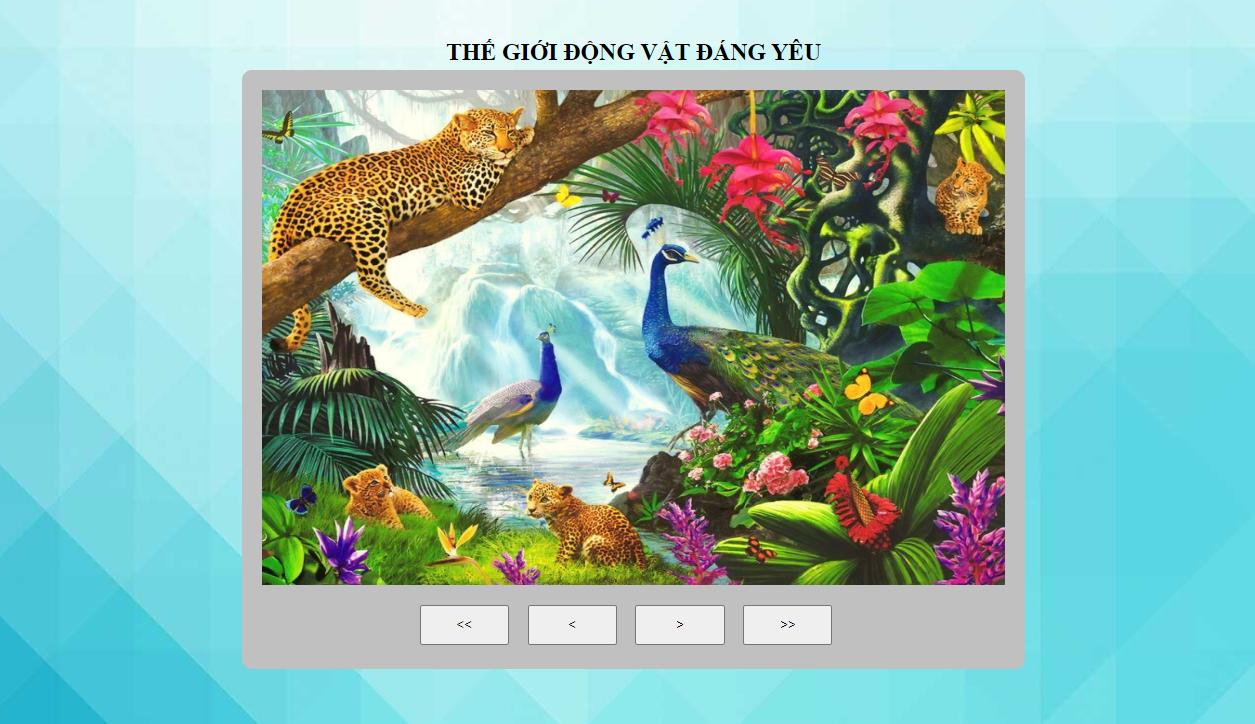
**LAB 6 MÔN LẬP TRÌNH JAVASCRIPT CƠ SỞ**

### Bài 1 : Slideshow

Xây dựng trang web cho phép hiển thị các hình ảnh dạng slideshow có giao diện như hình sau



Mô tả hoạt động: khi user nhắp các nút:

*  : thay bằng hình tiếp sau. Nếu là hình cuối sẽ trở lại hình đầu tiên
*  : thay bằng hình kế trước. Nếu là hình đầu tiên thì hiển thị hình cuối
*  : thay bằng hình đầu tiên
*  : thay bằng hình cuối cùng

Hướng dẫn thực hiện:

**Chuẩn bị:**

* Tạo folder <MaSV\_Ten>\_Lab6 để làm lab trong đó
* Trong folder vừa tạo, tạo thêm folder con có tên **images**

#### Tìm hình

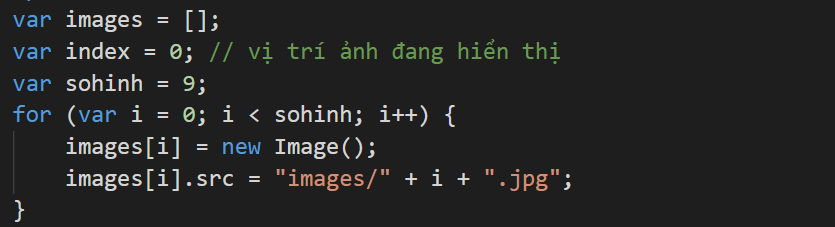
Tìm ít nhất 8 hình có kích thước tương đương, theo 1 chủ đề nào đó (động vật, phong cảnh , hoa, công nghệ, sản phẩm…) và lưu vào folder **images** vừa tạo ở trên. Đặt tên file từ 0 trở đi (Ví dụ : 0.jpg, 1.jpg, 2.jpg …)

#### Tạo file

Tạo file **bai1.html** và code html:

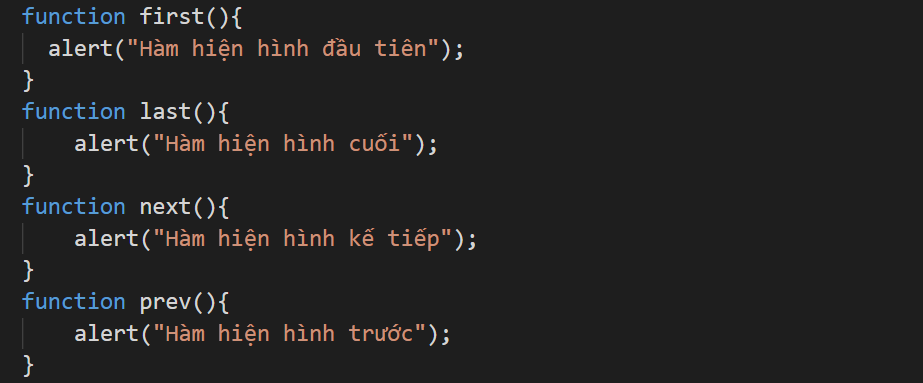


#### Viết mã javascript nạp sẵn các hình vào bộ nhớ



*(Thay số giá trị 9 trong code trên bằng số hình mà bạn có)*

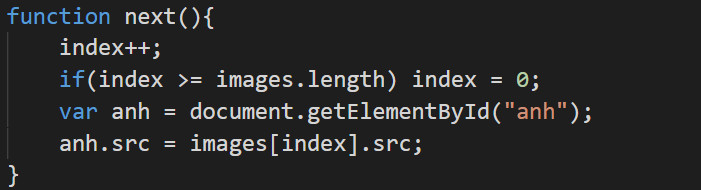
#### Khai báo 4 hàm



Test : Nhắp các button trong trang, phải thấy các thông báo tương ứng khi nhắp. (Nếu không được: mua máy mới ☺ )

#### Hiện hình kế:

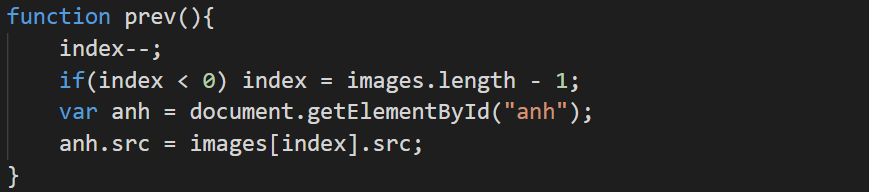
* Code lại bên trong hàm next: tăng biến index và gán địa chỉ hình kế vào ảnh:



* Test: nhắp nút  phải thấy hình kế.

#### Hiện hình trước:

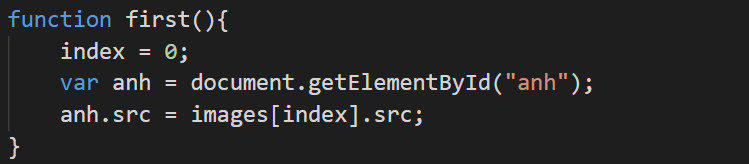
* Code lại bên trong hàm prev: giảm biến index và gán địa chỉ hình trước vào ảnh:



* Test: nhắp nút  phải thấy hình kế

#### Hiện hình đầu tiên:

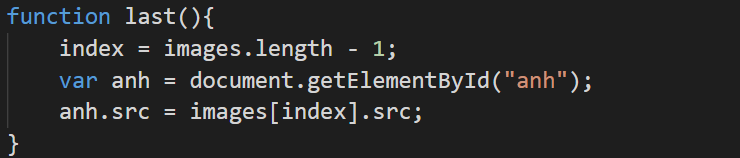
* Code lại bên trong hàm first: gán biến index về 0 và gán địa chỉ hình đầu tiên vào ảnh:



* Test: nhắp nút  phải thấy hình đầu tiên

#### Hiện hình cuối cùng:

* Code lại bên trong hàm last: gán biến index về chỉ số của hình cuối và gán địa chỉ hình cuối vào ảnh:



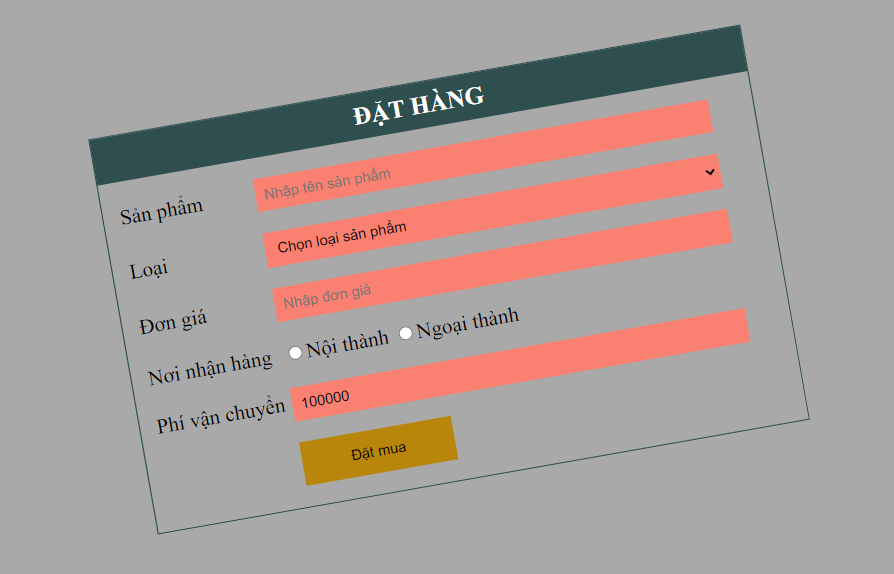
* Test: nhắp nút  phải thấy hình cuối

#### Định dạng css

Bổ sung tiêu đề phù hợp với các hình của bạn (như hình mẫu là Thế giới động vật…)

Định dạng css cho trang: border, canh giữa, width height các nút , màù nền trang , màu chữ…

### Bài 2 : Kiểm lỗi trong form



1. Tạo file **bai2.html** và code html:

Mô tả hoạt động

* Nếu chọn nơi nhận hàng là **Ngoại thành** thì hiển thị dòng **Phí vận chuyển** (xem hình trên), nếu chọn **Nội thành** thì ẩn dòng Phí vận chuyển
* Khi nhắp nút đặt mua thì kiểm lỗi như sau:

+ Sản phẩm không được để trống

+ Loại : phải được chọn

+ Đơn giá không được để trống và phải là số dương

+ Nơi nhận hàng phải được chọn

#### Tạo file

* Tạo file **bai3.html** và code html:

<form>

    <p> <label>Sản phẩm</label> <input id="tensp" placeholder="Nhập tên sản phẩm"/> </p>   <p> <label>Loại</label>

        <select id="loai">

            <option value="">Chọn loại sản phẩm</option>

            <option value="DT">Điện thoại</option>

            <option value="MT">Máy tính</option>

        </select>

    </p>

    <p> <label>Đơn giá</label> <input id="gia" placeholder="Nhập đơn giá" /> </p>

    <p> <label>Nơi nhận hàng</label>

        <input type="radio" name="diachi">Nội thành

        <input type="radio" name="diachi">Ngoại thành

    </p>

    <p id="vc"><label>Phí vận chuyển</label> <input id="phi\_vc" value="100000"></p>

<p> <label></label> <button type="button">Đặt mua</button></p>

  <hr>

 <div id="kq"></div>

</form>

* Code css định dạng

form {

    width: 600px;

    margin:100px auto;

    border: 1px solid darkslategrey;

box-shadow: 2px 2px 1px darkcyan;

    border-radius: 5px;

    font-size: 1.2em;

    transform: rotate(-10deg);

}

label { width: 120px; display: inline-block;}

p, h2 { margin-left: 15px;}

input[id], select {

    padding: 8px;

    width: 420px;

    background-color: salmon;

    border: none;

}

button {

    width: 140px;

    height: 40px;

    border: none;

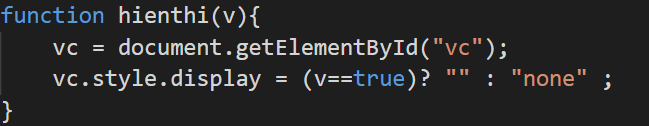
    background-color: darkgoldenrod;

}

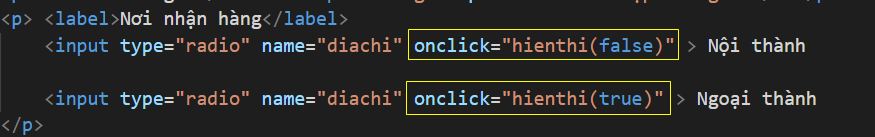
Xem thử trang web và tìm hiểu code css

#### Ẩn hiện phần Phí vận chuyển

* Định nghĩa hàm hienthi() để ẩn hiện phí vận chuyển

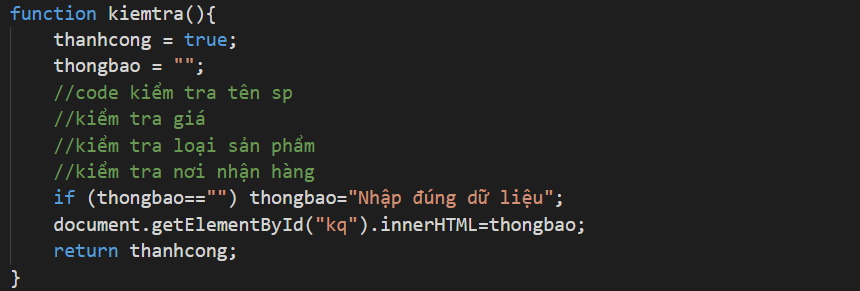


* Gọi hàm trong sự kiện click của 2 radio



#### Định nghĩa hàm kiểm tra dữ liệu trong form

* Code hàm kiemtra()



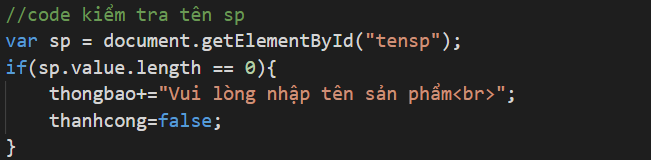
* Gọi hàm trong sự kiện click của button:



* Test: nhắp nút Đặt mua , phải thấy kết quá : Nhập đúng dữ liệu

#### Kiểm tra tên sản phẩm

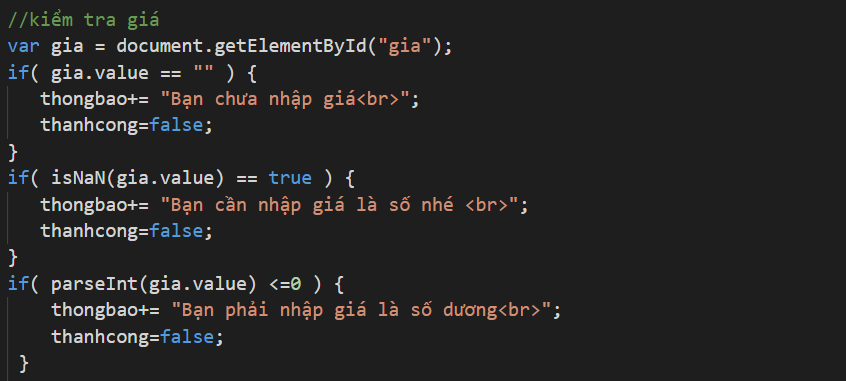
* Tại vị trí kiểm tra giá trong hàm kiemtra(), code:



* Test: Nhắp nút **Đặt mua** khi chưa nhập tên sp, sẽ thấy thông báo lỗi. Có nhập sẽ báo thành công.

#### Kiểm tra giá sản phẩm

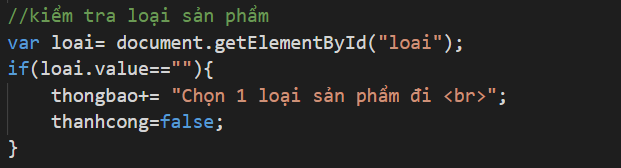
* Tại vị trí kiểm tra giá trong hàm kiemtra(), code để kiểm tra:



* Test: Nhắp nút **Đặt mua** khi chưa nhập giá hoặc nhập chữ hoặc nhập số âm sẽ thấy báo lỗi.

#### Kiểm tra loại sản phẩm

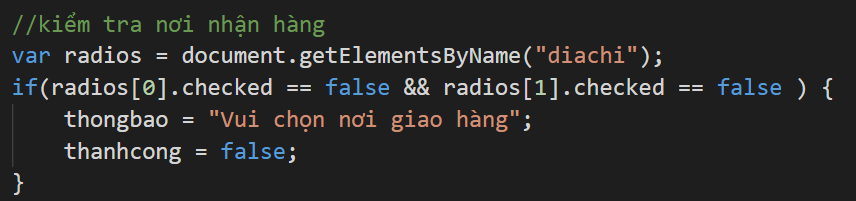
* Tại vị trí kiểm tra loại sp trong hàm kiemtra(), code:



* Test: Nhắp nút **Đặt mua** khi chưa chọn loại sp, sẽ thấy thông báo lỗi.

#### Kiểm tra nơi nhận hàng

* Tại vị trí kiểm tra tên nhận hàng trong hàm kiemtra(), code:



* Test: Nhắp nút **Đặt mua** khi chưa chọn nơi giao hàng, sẽ thấy thông báo lỗi.

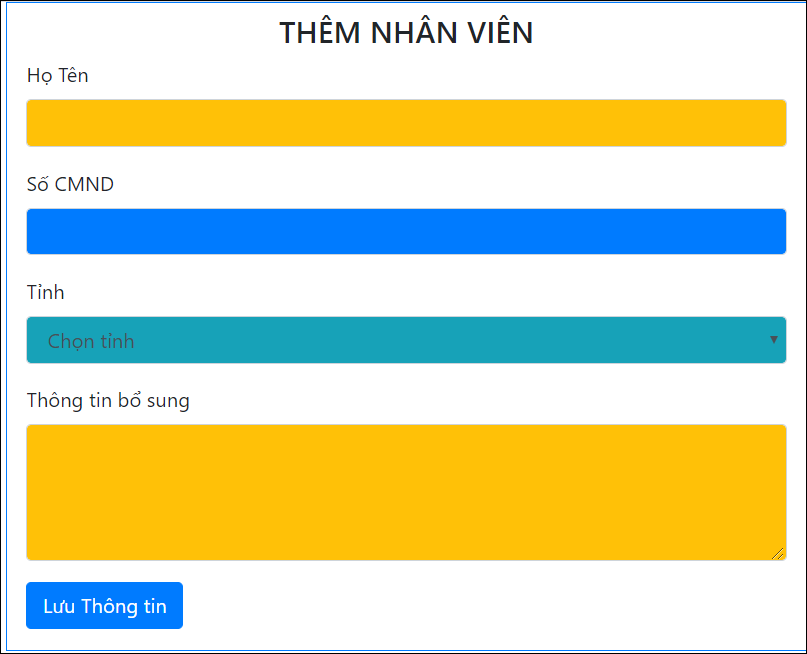
#### Định dạng CSS

* Định dạng cho đẹp : cho trang, div kq, ẩn phí vận chuyển khi mới hiện trang …

### Bài 3: Kiểm lỗi trong form

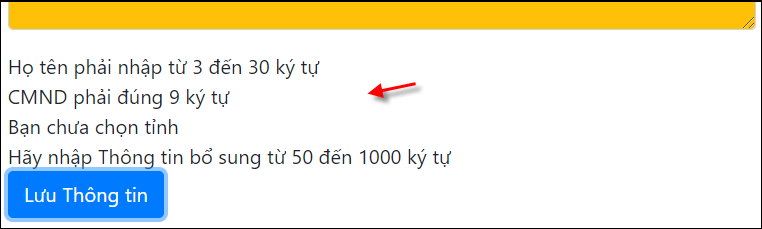
Tạo form như hình dưới, sau đó viết hàm kiểm tra dữ liệu nhập như sau:

* Họ tên: phải nhập, từ 3 đến 30 ký tự
* CMND: phải đúng 9 ký tự
* Tỉnh phải có chọn (value>0)
* Thông tin bổ sung: từ 50 đến 1000 ký tự



**Thông báo lỗi:**

Kiểm tra tất cả các điều kiện và hiện 1 lần trong div#loi. Ví dụ hiện lỗi như sau:



1. Tạo trang **bai4.html** và code html

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<form name="frm1" class="col-6 mx-auto border border-primary">

<h4 class="m-0 p-2 text-center">THÊM NHÂN VIÊN</h4>

<div class="form-group">

    <label> Họ Tên</label>  <input type="text" name="ht" class="form-control bg-warning">

</div>

<div class="form-group">

    <label>Số CMND</label>  <input type="text" name="cmnd" class="form-control bg-primary">

</div>

<div class="form-group">

    <label> Tỉnh</label>

    <select name="tinh" class="form-control bg-info">

        <option value="0">Chọn tỉnh</option>

        <option value="1">TPHCM</option>

        <option value="2">Hà nội</option>

        <option value="3">Cần Thơ</option>

    </select>

</div>

<div class="form-group">

    <label> Thông tin bổ sung</label>

    <textarea  name="bosung" class="form-control bg-warning" rows="4"></textarea>

</div>

<div class="form-group">

    <div id="loi"></div>  <button class="btn btn-primary">Lưu Thông tin</button>

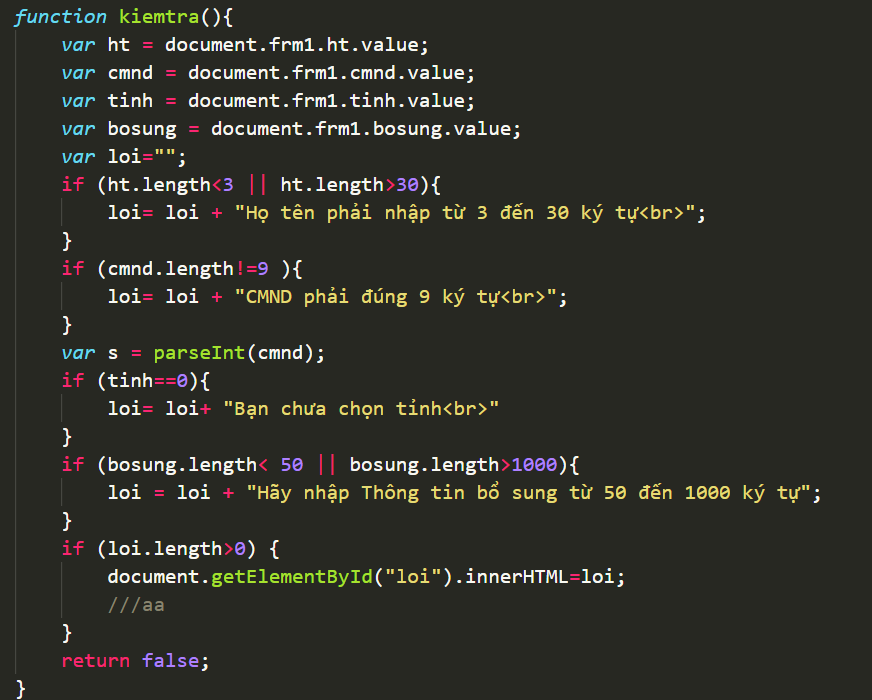
</div>

</form>

1. Gọi hàm kiểm tra trong sự kiện submit của form



1. Viết hàm kiểm tra



1. Định dạng thông báo lỗi: chữ nghiêng + màu chữ, cỡ chữ,
2. Thêm họ tên sinh viên + mã sinh viên vào tag title
3. Khi hiện lỗi ra, 10 giây sau thì cho lỗi biến mất

Gợi ý: gọi hàm setTimeout(“…”, thờigian); Trong đó … là lệnh gán chuỗi rỗng vào innerHTML của div#loi

### Bài 4: Kiểm tra form search, login

#### Kiểm tra form search

* Tạo file search.html
* Code html để tạo form như sau:

<form name=formtim action="kqtim.php" method="get">

<input name="tukhoa" id="tukhoa" type="text" value="Tìm kiếm"/>

<input name="btntim" id="btntim" type="submit" value=" TÌM "/>

</form>

* Chọn textfield tukhoa 🡺 vào code và thêm code như sau:

<input onclick="if(this.value=='Tìm kiếm') this.value=''" onblur="if(this.value=='') this.value='Tìm kiếm'" name="tukhoa" id="tukhoa" type="text" value="Tìm kiếm" />

* Tạo file checksearch.js và viết hàm kiểm tra form search

function checksearch (){

if(document.formtim.tukhoa.value=='Tìm kiếm') {

alert("Bạn chưa nhập từ khóa tìm kiếm…");

document.formtim.tukhoa.focus();

return false; // Chặn form không cho submit

}

return true; // Cho submit form khi đã kiểm tra xong

}

* Chèn file checksearch.js vào trong tag head của file search.html và gọi hàm kiểm tra như sau:

<head>

<script src="checksearch.js"></script>

</head>

<body>

<form onsubmit="return checksearch();" … >

#### Kiểm tra form flogin

* Tạo trang login.html và code html để tạo form như sau: .

<form name="flogin" method="post" action="xulylogin.php">

Username:<br/> <input name="u" id="u" type="text" /> <br/>

Password:<br/> <input name="p" id="p" type="password"/> <br/>

<input type="submit" name="sub" id="sub" value="Login" />

</form>

* Tạo file checklogin.js và code

function checklogin(){

if(document.flogin.u.value==''){

alert("Bạn chưa nhập username");

document.flogin.u.focus(); //Chọn đối tượng u

return false; // Chặn form không cho submit

}

if(document.flogin.p.value=='') {

alert("Bạn chưa nhập password");

document.flogin.p.focus(); //Chọn đối tượng p

return false;

}

return true; // Cho submit form khi đã kiểm tra xong

}

* Chèn file checklogin.js vào file login.html và gọi hàm kiểm tra form như sau:

<head>

<script src="checklogin.js"></script>

</head>

<body>

<form onsubmit="return checklogin();"name="flogin" method="post" action="xulylogin.php">

### Bài 5: Bé làm toán (1đ)

* Tạo file belamtoan.html
* Tạo 3 textfield Số A, Số B, Đáp Án và 1 button Trả lời.
* Hiện ngẫu nhiên 2 số trong 2 input.
* Bé nhập đáp án rồi nhắp nút Trả lời sẽ gọi 1 hàm
* Trong hàm xử lý: Nếu đúng thì khen GIỎI QUÁ BÉ ƠI, nếu sai thì hiện SAI RỒI BÉ IU. Hiện đáp án trong 1 thẻ div
* Định dạng thẻ div cho đẹp: màu chữ , màu nền, border, font-size…