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

### Bài 1 : Accordion

1. Tạo file bai1.hhtml và code html

<div class="listBH">

    <h3 class="tieude">Tên bài hát 1</h3>

    <div class="noidung">Nội dung bài hát 1</div>

    <h3 class="tieude">Tên bài hát 1</h3>

    <div class="noidung">Nội dung bài hát 1</div>

    <h3 class="tieude">Tên bài hát 1</h3>

    <div class="noidung">Nội dung bài hát 1</div>

</div>

1. Định dạng css

.listBH { width: 1000px; margin: auto;}

.tieude { background: darkslategrey; color: white;

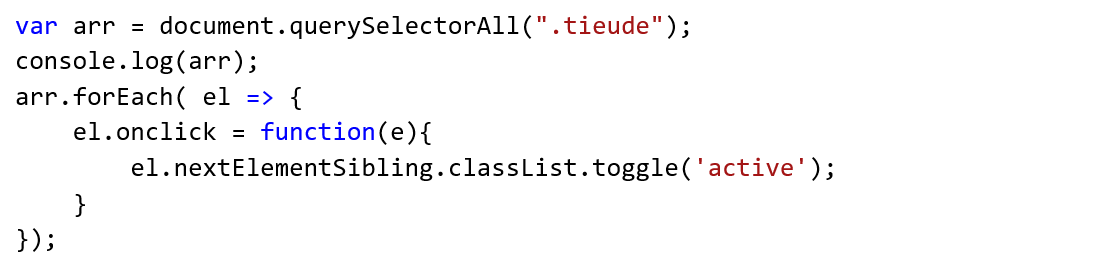
padding: 10px; cursor: pointer; margin:0; }

.noidung { padding: 20px 0;}

.active { background: yellow;}

1. Code javascript thực hiện yêu cầu sau

Khi nhắp vào tên bài hát, thì gán cho div nội dung kế tiếp class css active



1. Sinh viên thực hiện thêm
2. Thay tiêu đề và nội dung các bài hát tùy ý.
3. Khi nhắp tiêu đề bài hát, thì nội dung tương ứng sẽ ẩn/hiện
4. Định dạng thêm cho đẹp

### Bài 2 : Trò chơi Tictactoe

Tạo trang web cho phép 2 người chơi trò tictactoe có giao diện như sau.



Mô tả hoạt động:

* Hiển thị 9 nút button
* Người chơi nhấp vào nút trống
  + Lượt lẻ: tạo nhãn X và nền xanh (aqua)
  + Lượt chẵn: tạo nhãn O nền vàng
  + Vô hiệu hóa nút được click để không click được nữa
* Nếu tổng click 8 lần thì báo kết thúc và load lại trang web

Hướng dẫn thực hiện

1. Tạo file **bai2-trochoi.html** và code html

<head><title></title></head>

<body>

<h1>TICTACTOE GAME</h1>

<table class="container">

    <tr>

        <td><button></button></td>

        <td><button></button></td>

        <td><button></button></td>

    </tr>

    <tr>

        <td><button></button></td>

        <td><button></button></td>

        <td><button></button></td>

    </tr>

    <tr>

        <td><button></button></td>

        <td><button></button></td>

        <td><button></button></td>

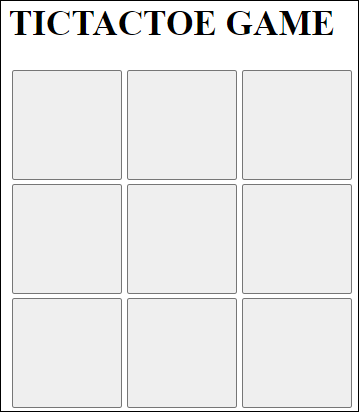
    </tr>

</table>

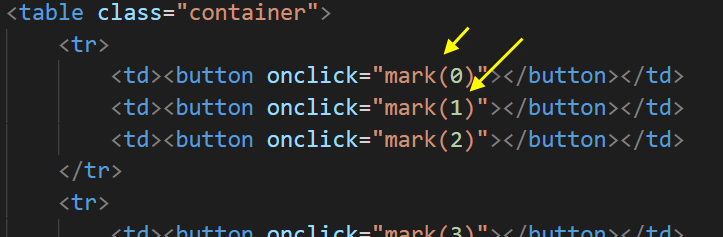
</body>

1. Định dạng css cho các button có width là 100px và height cũng là 100px, font-size là 1.5em

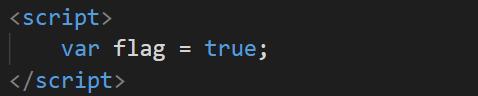
Xem thử trang web:



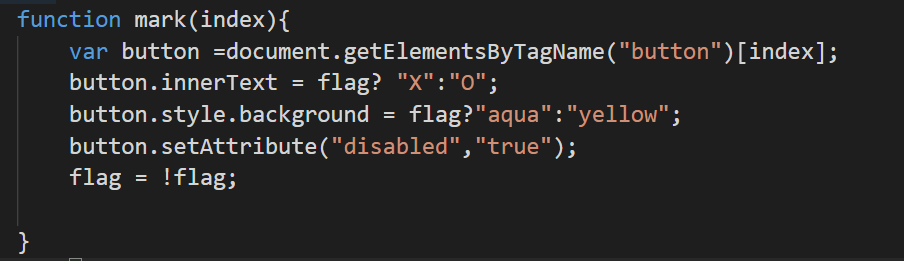
1. Code trong sự kiện click của 9 button: gọi hàm mark và truyền tham số (từ 0 đến 8)



1. Khai báo biến flag



1. Định nghĩa hàm mark để gán text cho button, gán màu nền, khóa button

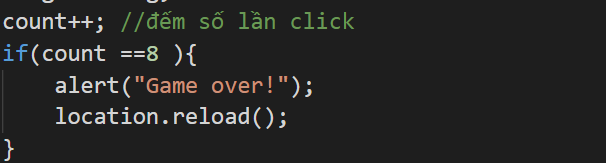


*Chú ý có thể thay lệnh* ***button.setAttribute("disabled","true");*** *bằng* ***button.disabled=true;***

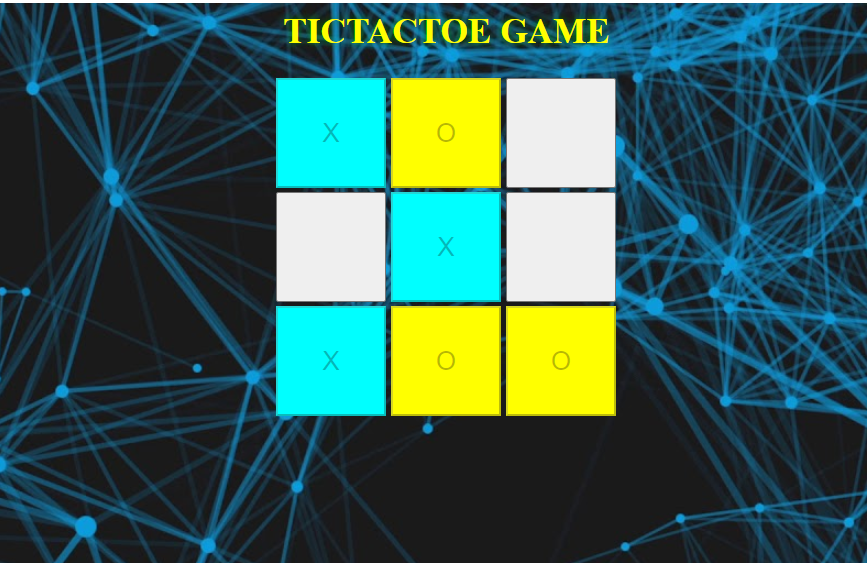
1. Kiểm soát kết thúc game
2. Khai báo biến count đếm số lần nhắp (khai báo sau biến flag)

**var count = 0;**

1. Code thêm trong cuối hàm mark



1. Định dạng cho đẹp: font, color, background, border…

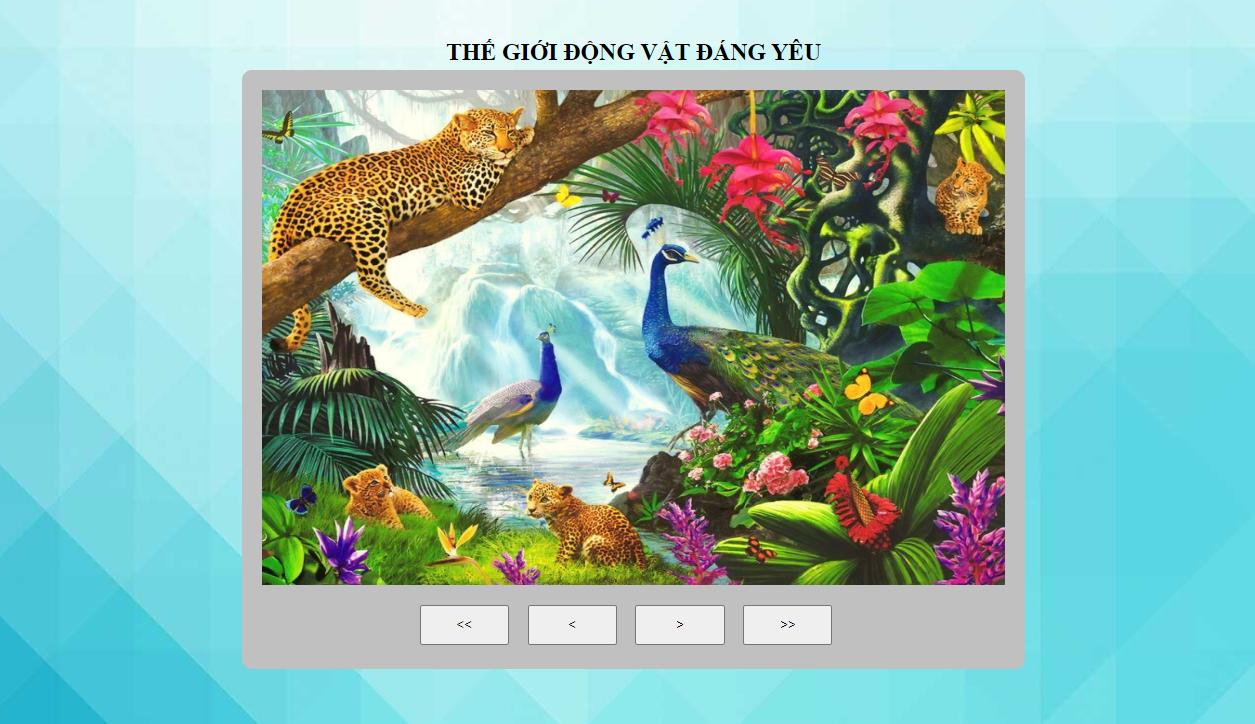


### Bài 3 : Slideshow

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:

*  : hiện hình tiếp sau, nếu đang là hình cuối sẽ trở lại hình đầu tiên
*  : hiện hình kế trước, nếu là hình đầu tiên thì hiện hình cuối
*  : hiện hình đầu tiên
*  : hiện hình cuối cùng



**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

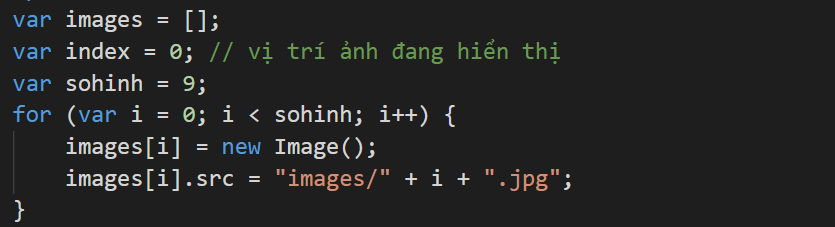
Tìm ít nhất 8 hình cùng kích thước theo 1 chủ đề nào đó (động vật, phong cảnh , hoa, sản phẩm…) và lưu vào folder **images**. Đặt tên file từ 0 trở đi (Vd: 0.jpg, 1.jpg, 2.jpg …)

#### Tạo file

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

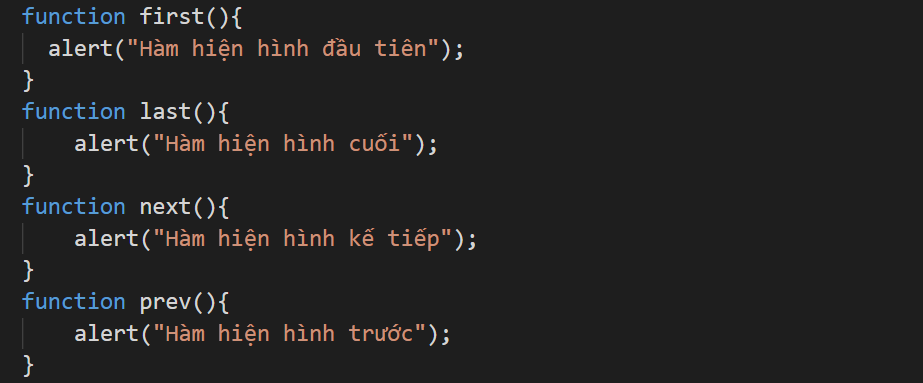


#### Viết mã javascript nạp sẵn các hình vào bộ nhớ



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

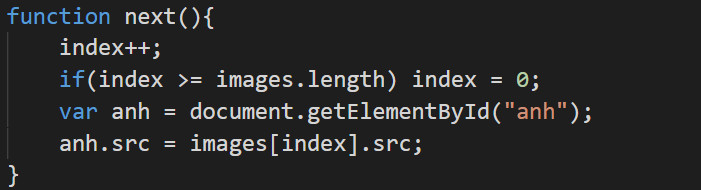
#### Định nghĩa 4 hàm



Test : Nhắp các button trong trang, phải thấy các thông báo tương ứng khi nhắp.

#### Hiện hình kế:

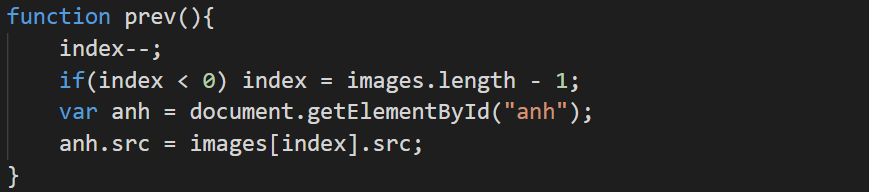
* Code lại bên trong hàm next: tăng biến index và gán địa chỉ hình kế vào ảnh:



* Test: nhắp nút  phải thấy hình kế.

#### Hiện hình trước:

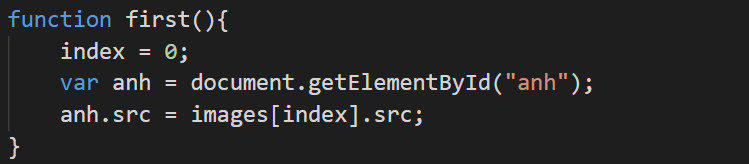
* Code lại bên trong hàm prev: giảm biến index và gán địa chỉ hình trước vào ảnh:



* Test: nhắp nút  phải thấy hình kế

#### Hiện hình đầu tiên:

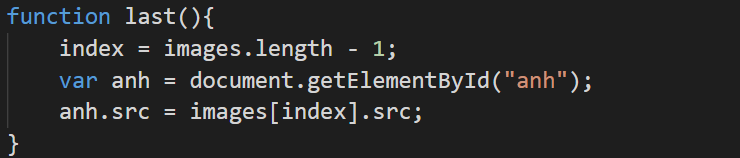
* Code lại bên trong hàm first: gán biến index về 0 và gán địa chỉ hình đầu tiên vào ảnh:



* Test: nhắp nút  phải thấy hình đầu tiên

#### Hiện hình cuối cùng:

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



* Test: nhắp nút  phải thấy hình cuối

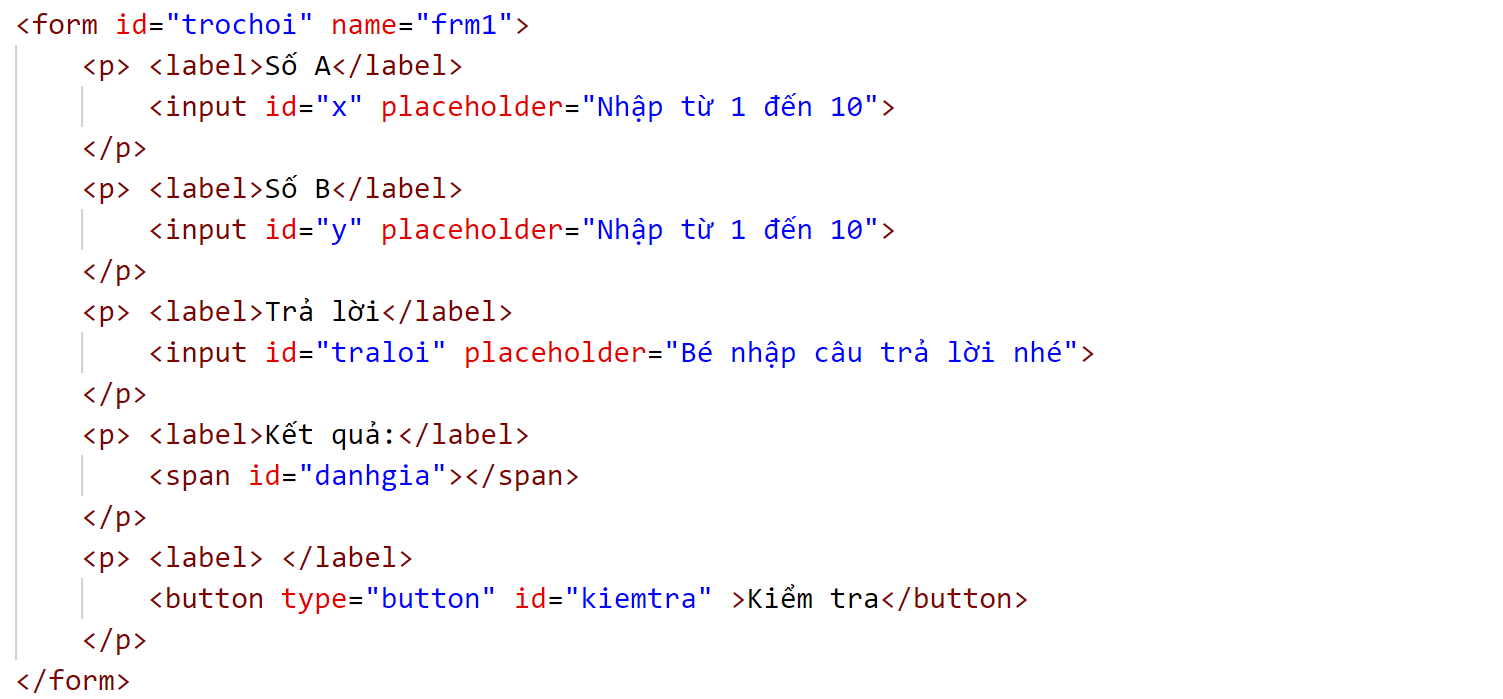
#### Bổ sung (nếu muốn)

* Thêm chữ dòng tiêu đề slideshow cho phù hợp với các hình của bạn
* Định dạng css cho đẹp

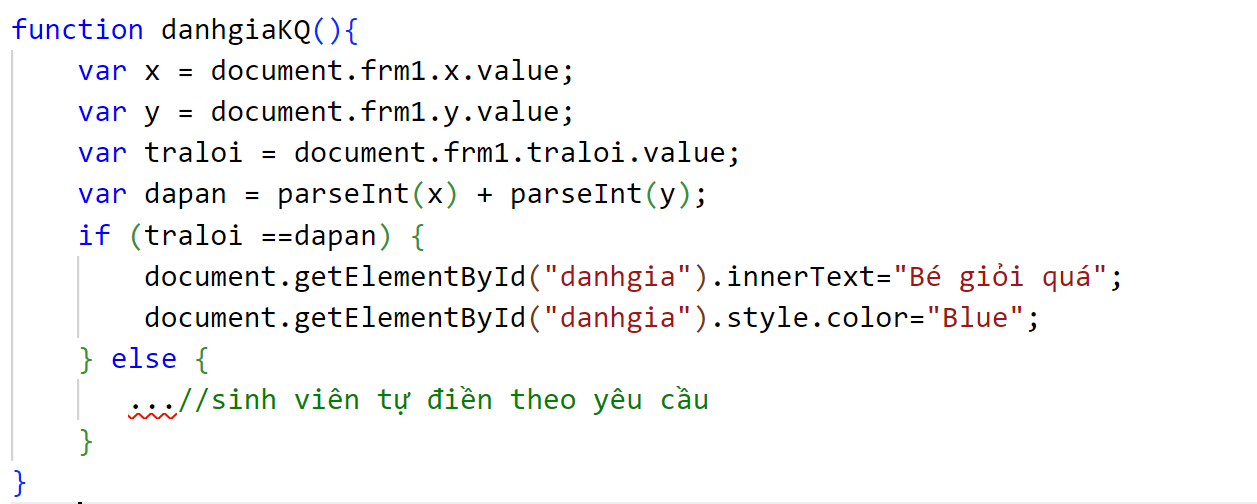
### Bài 4: Bé làm toán

|  |  |
| --- | --- |
|  | **Mô tả**   * Bé nhập 2 số rồi nhập trả lời. Sau đó nhắp nút **Kiểm tra** sẽ hiện kết quả * Nếu trả lời đúng sẽ hiện chữ **Bé giỏi quá** màu xanh * Nếu trả lời sai sẽ hiện chữ **Sai rồi bé** màu đỏ |

1. Tạo file bai4-belamtoan.html
2. Nhập code html trong tag body để tạo form



1. Định nghĩa hàm đánh giá kết quả trong tag head



1. Gọi hàm danhgiaKQ trong sự kiện click của nút Kiểm tra

Sinh viên tự thực hiện

1. Mởi sinh viên làm thêm

* Tạo nút Cho số, khi nhắp vào sẽ hiện ngẫu nhiên 2 số 1-10 trong 2 input A, B
* Định dạng thêm cho đẹp