

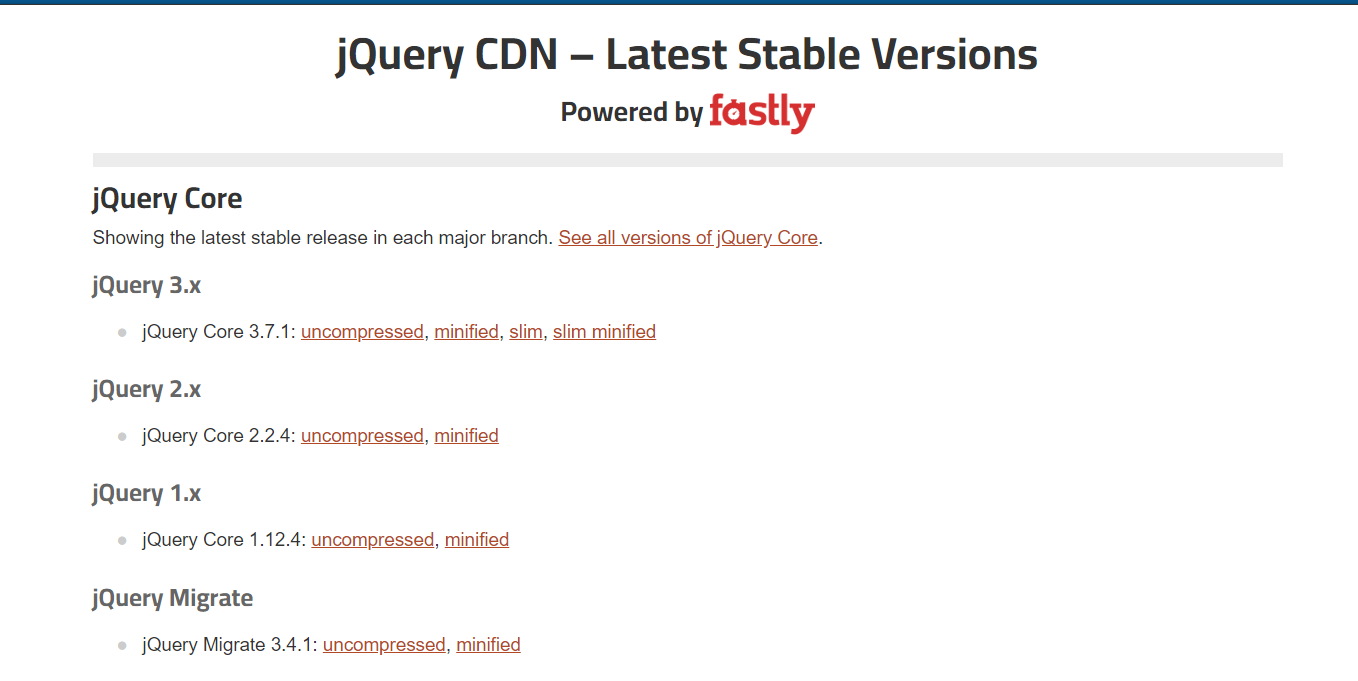
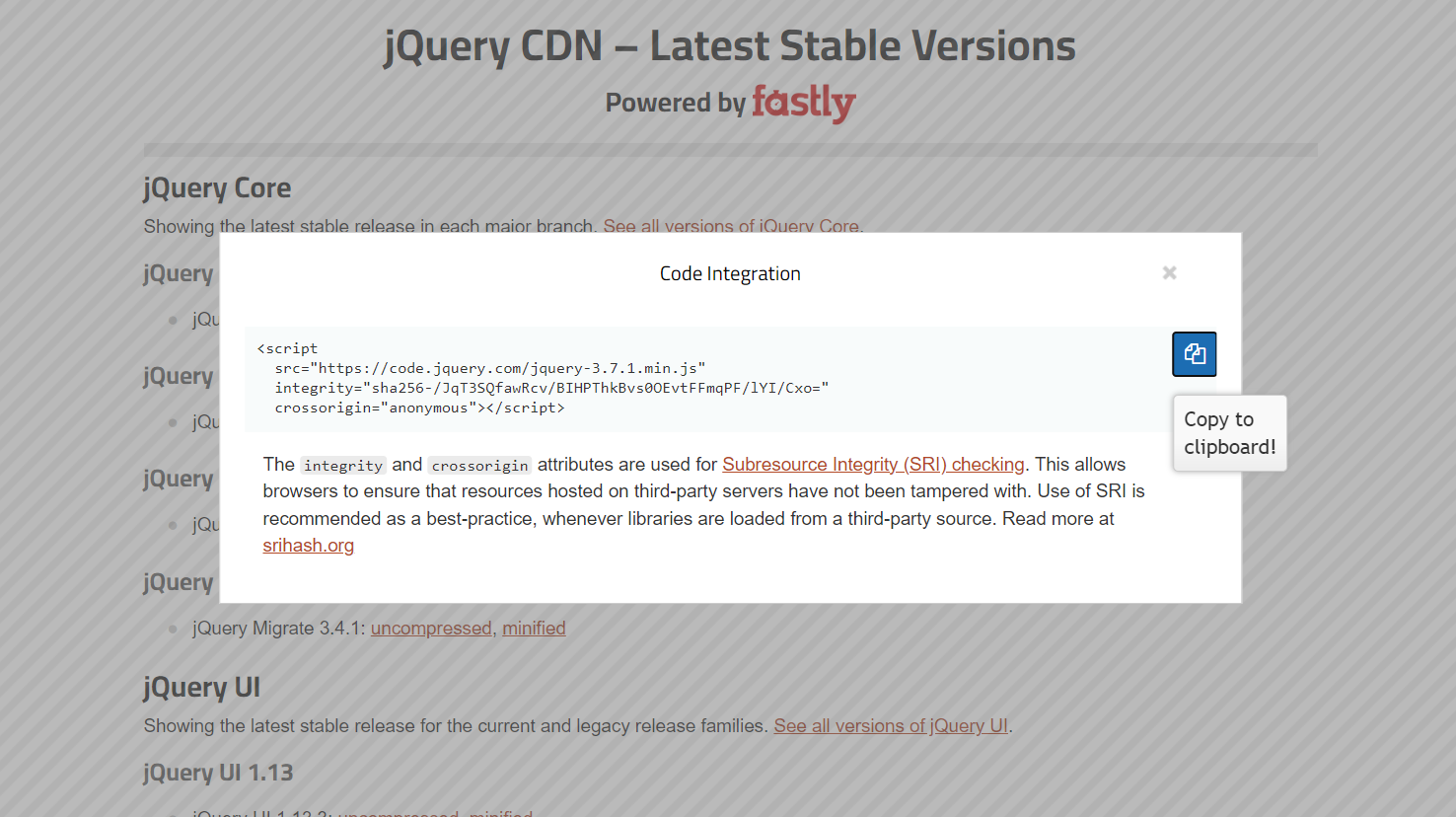
Học Jquery cơ bản

By Phạm Ngọc Bảo An

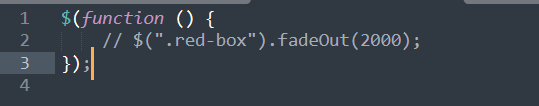
Hiệu ứng ảnh động và bộ lọc

1. Tải jquery  
   vào <https://releases.jquery.com/>

Chọn mini file của 1 trong 3 jquery bất kì rồi coppy link script

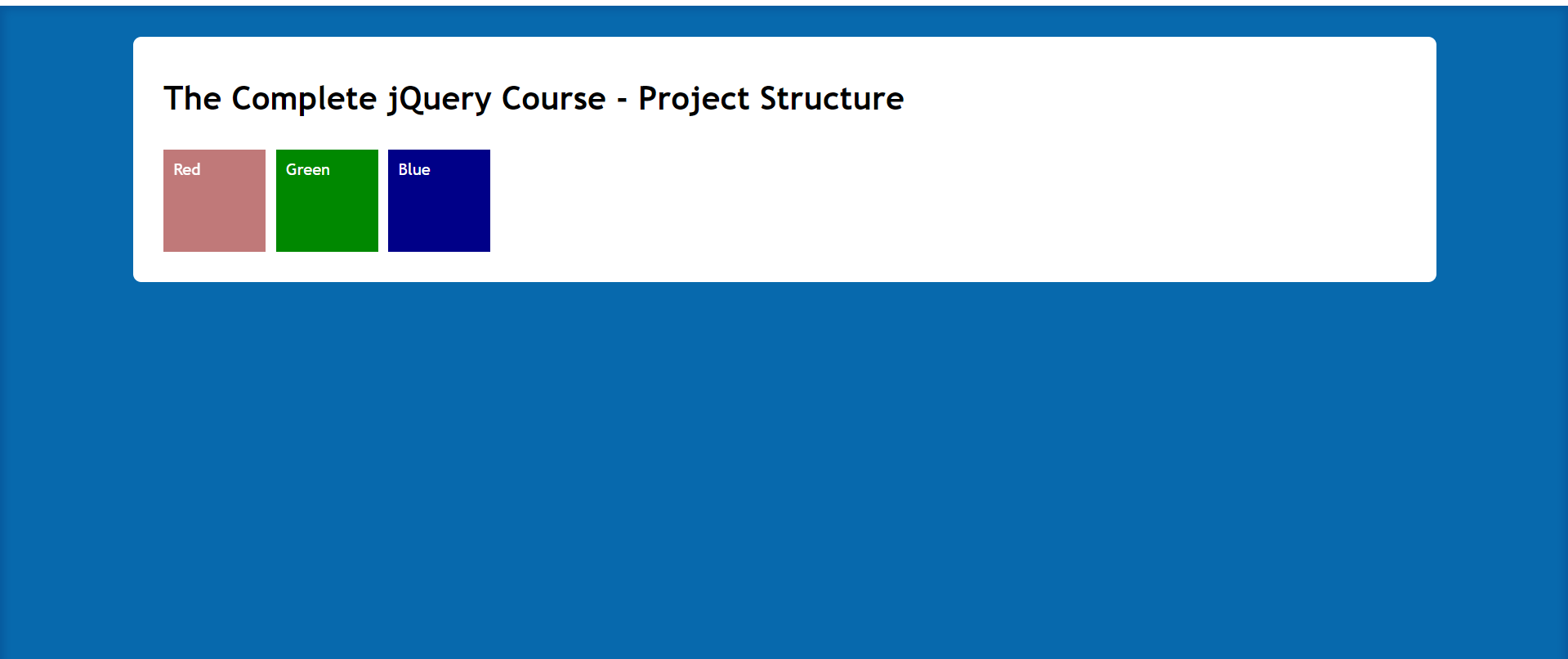
  


1. Tạo hàm

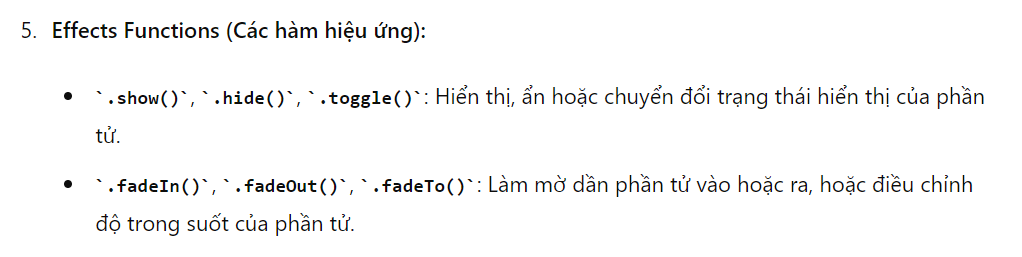


Với $ được hiểu là jquery

1. Truy vấn đến phần tử cần được thao tác, ở đây gọi tới class redbox với thời gian ẩn đi sau 2s

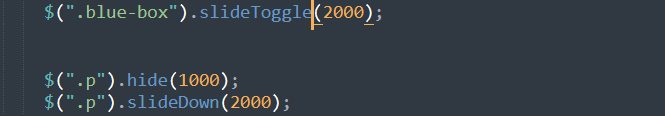


1 số hàm hiệu ứng

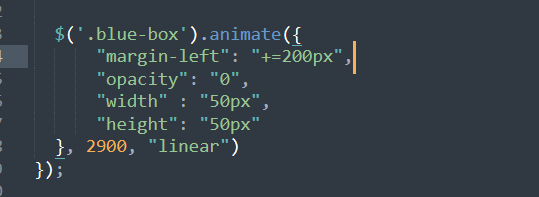


Slide down – slide up

Slide down hợp lí trong các trường hợp muốn load thêm cái gì đó



Custom animation



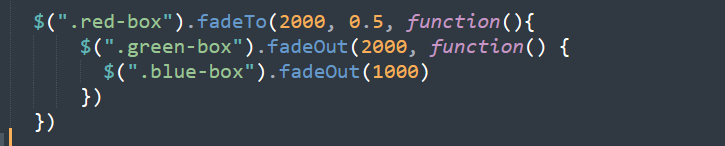
1. Trì hoãn và sâu chuổi các hoạt ảnh

Thêm hàm delay để trì hoãn công việc đi 2s trước khi thực thi các tác vụ khác



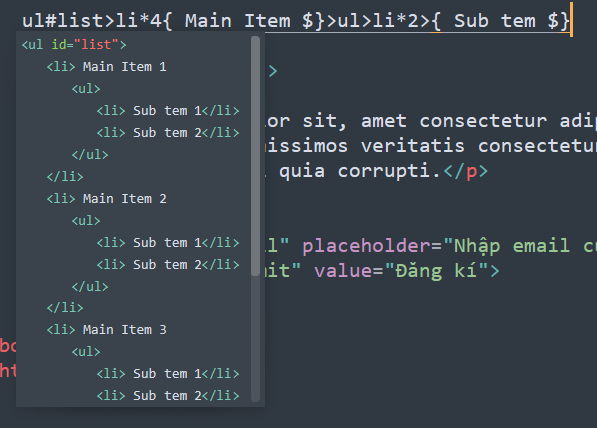
1. Callback, thực thi các thao tác

Sau 2s mờ dần đi 50% thì gọi đến hàm thực thi ẩn dần đi trong 2s và hàm đó tiếp tục gọi hàm trong để ẩn dần đi blue box trong 2s



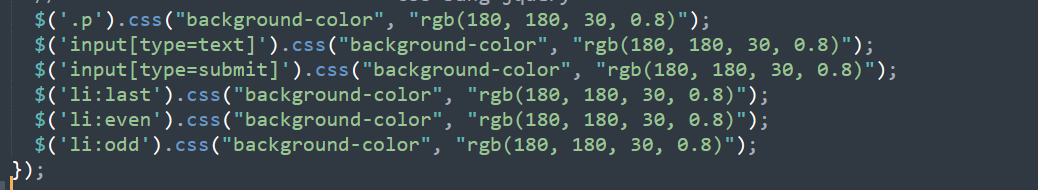
1. Tạo list

Trick tạo list nhanh



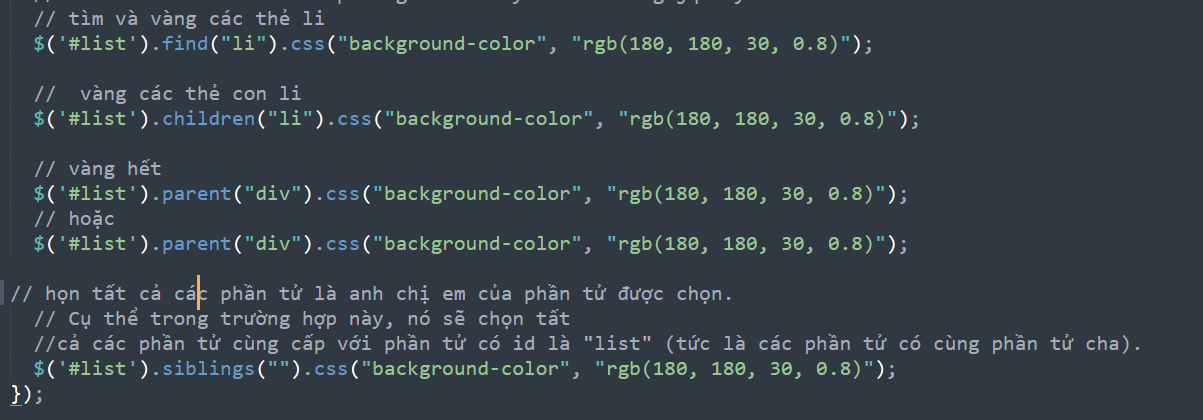
1. Css bằng jquery

Gọi tới class hặc id sau đó .”.css” để thực hiện css lên thẻ đó



1. Các phương thức truyền tải trong jquery

Một số phương thức cơ bản trong jquery như find: tìm, children: thẻ con, parent: thẻ cha, sibling



1. Các phương thức lọc của jquery ( filter )

Không chỉ filter mới tạo hàm mà các hàm khác vẫn có thể truyền hàm vào được



Thao tác với DOM Chèn, thây thến, xóa

1. DOM là gì



DOM là gì

DOM (Document Object Model) là một mô hình đối tượng tài liệu, là giao diện lập trình ứng dụng (API) được sử dụng để truy cập và thao tác với nội dung, cấu trúc và kiểu dáng của các tài liệu web (chẳng hạn như HTML và XML). DOM thể hiện tài liệu dưới dạng cấu trúc cây, trong đó mỗi nút trong cây đại diện cho một phần của tài liệu.

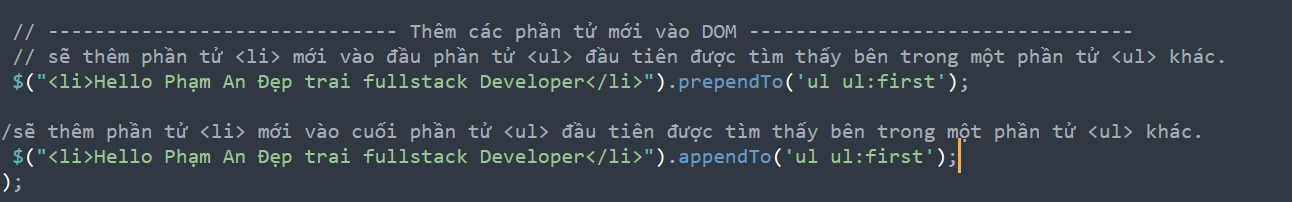
1. **Cấu trúc cây**: Tài liệu được thể hiện dưới dạng một cây phân cấp, trong đó mỗi phần tử, thuộc tính và văn bản trong tài liệu là một nút trong cây đó. Nút gốc của cây là đối tượng document.
2. **Các loại nút**:
   * **Element nodes**: Đại diện cho các thẻ HTML (như <div>, <p>, <a>, v.v.).
   * **Attribute nodes**: Đại diện cho các thuộc tính của thẻ HTML (như id, class, src, v.v.).
   * **Text nodes**: Đại diện cho nội dung văn bản bên trong các thẻ HTML.
   * **Comment nodes**: Đại diện cho các bình luận trong tài liệu HTML.
3. **API**: DOM cung cấp các phương thức và thuộc tính để truy cập và thao tác với các nút trong cây DOM. Ví dụ:
   * document.getElementById(): Trả về một phần tử có id cụ thể.
   * document.querySelector(): Trả về phần tử đầu tiên khớp với selector CSS.
   * element.appendChild(): Thêm một nút con vào phần tử.
   * element.setAttribute(): Đặt giá trị của một thuộc tính.
4. **Truy cập và thao tác**: DOM cho phép các lập trình viên JavaScript truy cập và thay đổi nội dung, cấu trúc và kiểu dáng của tài liệu web. Ví dụ:
   * Thay đổi nội dung của một phần tử: element.innerHTML = "New Content";
   * Thay đổi thuộc tính của một phần tử: element.setAttribute("class", "new-class");
   * Thêm phần tử mới: parentElement.appendChild(newElement);
5. **Sự kiện**: DOM cũng hỗ trợ mô hình sự kiện, cho phép các lập trình viên thêm các trình xử lý sự kiện (event handlers) vào các phần tử trong tài liệu để phản hồi các sự kiện như nhấp chuột, di chuyển chuột, nhập bàn phím, v.v. Ví dụ:
   * element.addEventListener("click", function() { alert("Element clicked!"); });

DOM là một phần quan trọng trong việc phát triển web hiện đại, cho phép các trang web trở nên tương tác và động hơn.

1. Thêm các thẻ mới vào DOM

Hàm append() là một phương thức trong JavaScript được sử dụng để thêm các nút hoặc nội dung mới vào cuối danh sách con của một phần tử HTML.

Thêm các phần tử mới vào DOM

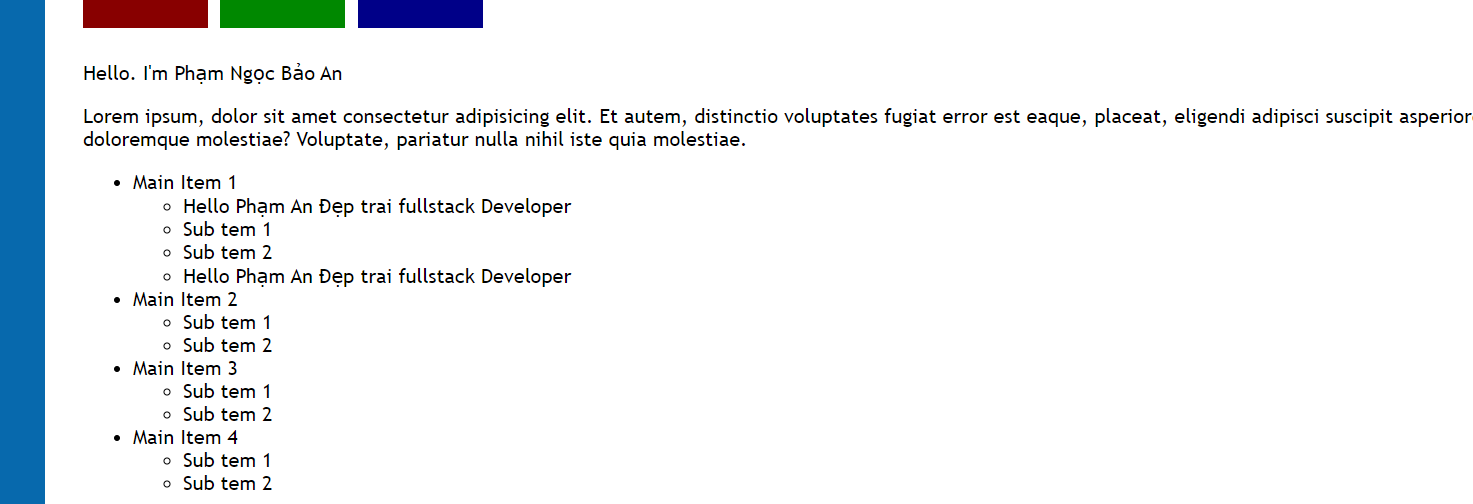


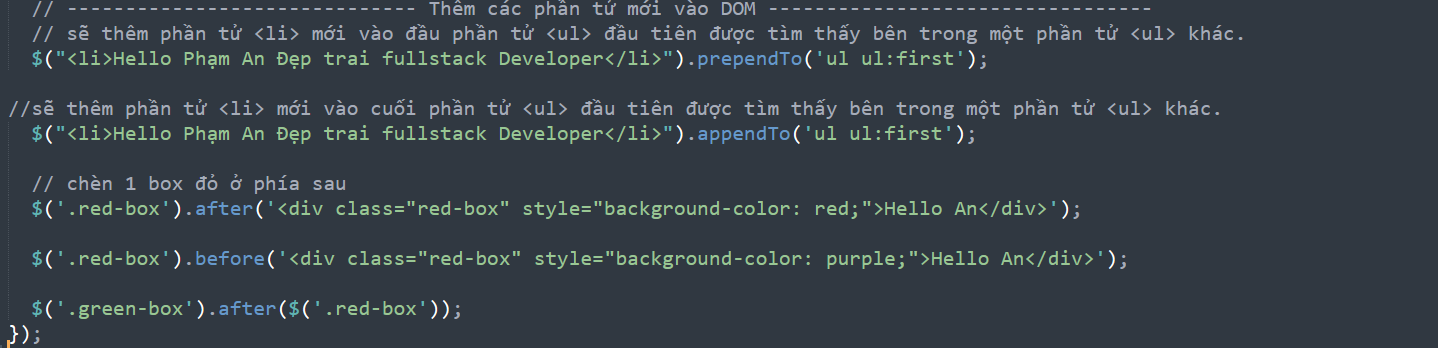
 prependTo(selector):

* Dòng mã này thêm phần tử mới vào **đầu** của phần tử cha được chọn.
* Cụ thể, đoạn mã $("<li>Hello Phạm An Đẹp trai fullstack Developer</li>").prependTo('ul ul:first'); sẽ thêm phần tử <li> mới vào **đầu** phần tử <ul> đầu tiên được tìm thấy bên trong một phần tử <ul> khác.

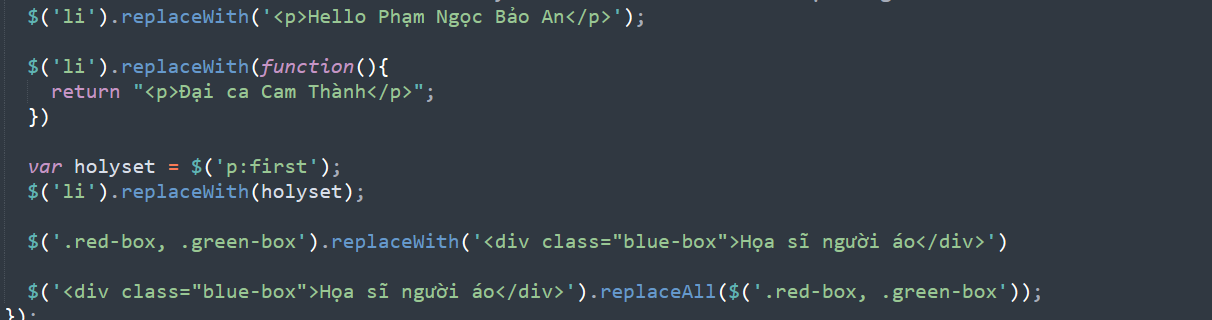
 appendTo(selector) hoặc append(selector):

* Dòng mã này thêm phần tử mới vào **cuối** của phần tử cha được chọn.
* Cụ thể, đoạn mã $("<li>Hello Phạm An Đẹp trai fullstack Developer</li>").appendTo('ul ul:first'); hoặc $("<li>Hello Phạm An Đẹp trai fullstack Developer</li>").append('ul ul:first'); sẽ thêm phần tử <li> mới vào **cuối** phần tử <ul> đầu tiên được tìm thấy bên trong một phần tử <ul> khác.



Còn nếu mà đổi thẻ $(ui ul:first lên trước) thì chỉ cần append hoặc prepend là được, không cần phải có To  
  
1 số hàm như before after chèn trước hoặc sau phần tử được chỉ điểm  


1. THây thế các thẻ element và nội dung



Dòng 1: Sẽ đổi tất cả thẻ li thành thẻ p với nội dung là Hello Phạm Ngọc Bảo An.

Dòng 2: sẽ đổi nội dung tất cả thẻ li thành thẻ p giống dòng 1 nhưng theo kiểu callback.

Dòng 3: tạo 1 biến với dữ liệu là là thẻ p đầu tiên sau đó đổi thẻ li thành thẻ p đó.

Dòng 4: đổi box đỏ và xanh là thành màu dương.

Dòng 5 giống dòng 4 nhưng đổi lại cách viết.

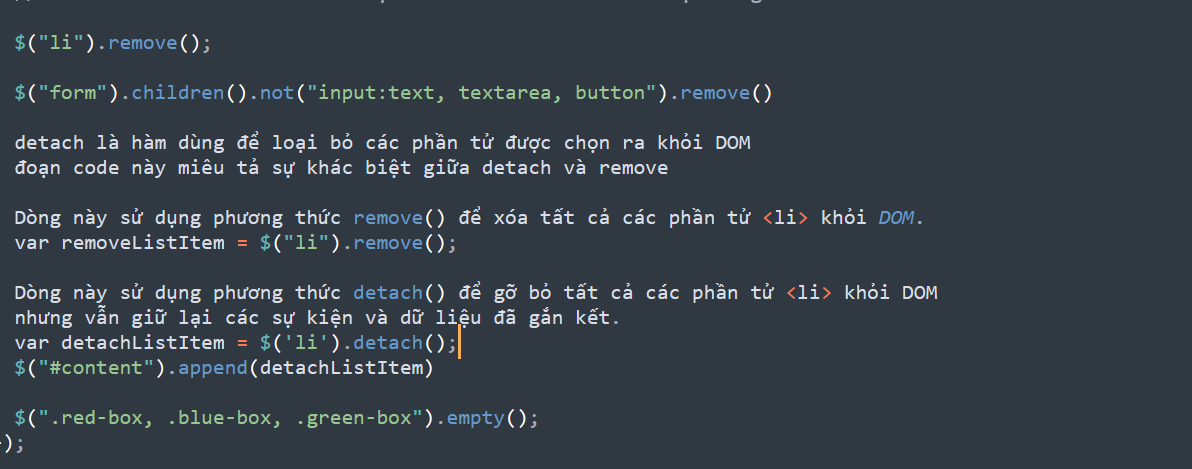
1. Xóa các thẻ element và nội dung

Trong phần này sẽ tập trung vào remove(), detach() và empty()

 **empty()**: Loại bỏ tất cả các phần tử con của phần tử được chọn, nhưng giữ lại phần tử cha trong DOM.

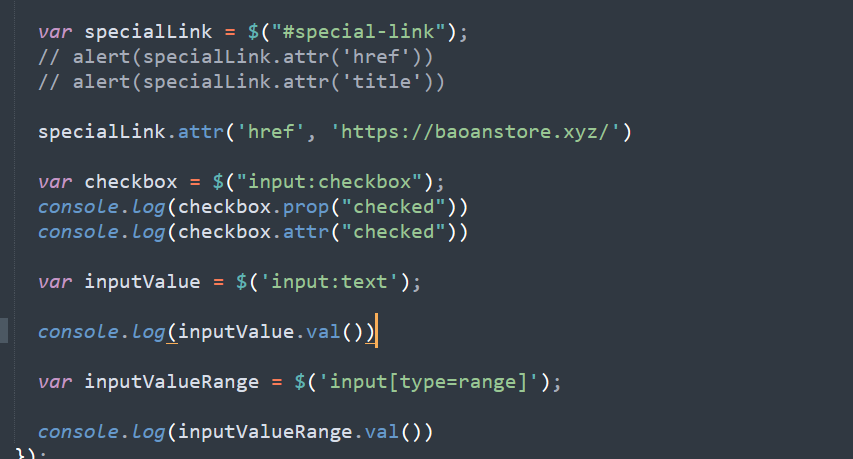
 **detach()**: Gỡ bỏ các phần tử được chọn khỏi DOM, giữ lại các sự kiện và dữ liệu đã gắn kết. Các phần tử này có thể được thêm lại vào DOM sau đó.

 **remove()**: Loại bỏ hoàn toàn các phần tử được chọn khỏi DOM, bao gồm cả các sự kiện và dữ liệu đã gắn kết.



1. Thao tác thuộc tính và thuộc tính

ở dòng đường dẫn , ta thây đổi đường dẫn web bằng 1 đường dẫn khác bằng attr

  
  
  
  
  
  
ở phần checked   
 **Dòng 1:**

var checkbox = $("input:checkbox");

* **Ý nghĩa:** Sử dụng jQuery để chọn tất cả các phần tử <input> có thuộc tính type="checkbox" trong tài liệu HTML. Biến checkbox sẽ chứa một đối tượng jQuery đại diện cho các phần tử được chọn này.

 **Dòng 2:**

console.log(checkbox.prop("checked"));

* **Ý nghĩa:** Lấy giá trị của thuộc tính checked từ đối tượng jQuery checkbox sử dụng phương thức prop().
* **Kết quả:** Giá trị trả về sẽ là true hoặc false tùy thuộc vào trạng thái của checkbox đầu tiên được chọn. Nếu checkbox đầu tiên trong tập hợp được chọn, giá trị sẽ là true, ngược lại là false.

 **Dòng 3:**

console.log(checkbox.attr("checked"));

* **Ý nghĩa:** Lấy giá trị của thuộc tính checked từ đối tượng jQuery checkbox sử dụng phương thức attr().
* **Kết quả:** Giá trị trả về sẽ là undefined, true, hoặc chuỗi "checked" tùy thuộc vào trạng thái và cách checkbox được thiết lập.
  + Nếu thuộc tính checked được thiết lập bằng HTML (ví dụ: <input type="checkbox" checked>), attr("checked") sẽ trả về chuỗi "checked".
  + Nếu checkbox được chọn thông qua JavaScript mà không có thuộc tính checked trong HTML, attr("checked") sẽ trả về undefined.

  
  
Lấy giá trị của input có kiểu là text rồi in ra cửa sổ console  
  
lấy giá trị của input có kiểu là range rồi in ra cửa số console  
  
Tổng kết   
 **prop("checked")** là cách đúng đắn để kiểm tra trạng thái hiện tại của một checkbox (nó có được chọn hay không).

 **attr("checked")** kiểm tra sự hiện diện của thuộc tính checked trong HTML gốc, không phản ánh trạng thái hiện tại nếu trạng thái đã thay đổi sau khi trang được tải.

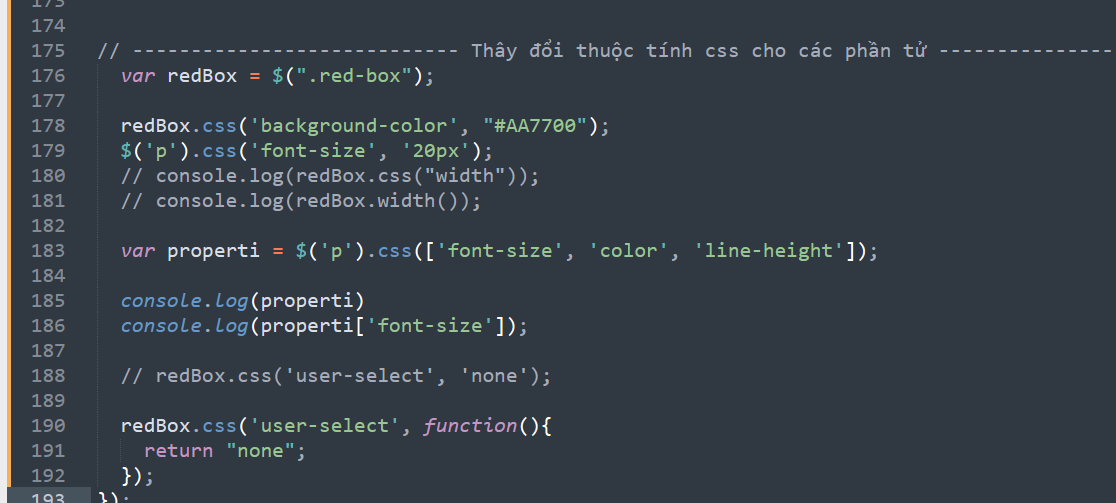
1. Làm slide show cơ bản

### Giải thích từng phần

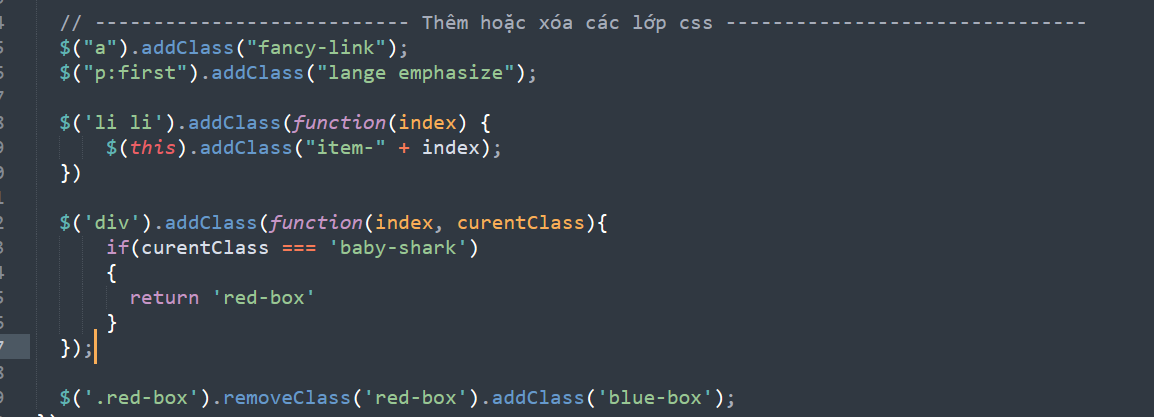
1. **$(function() {...})**:
   * Đây là một cách viết ngắn gọn để đảm bảo rằng đoạn mã jQuery sẽ chỉ chạy sau khi toàn bộ tài liệu HTML đã được tải.\
2. **var imageGallery = $('.gallery').find('img').first();**:
   * Tìm phần tử <img> đầu tiên bên trong phần tử có lớp .gallery và lưu trữ nó vào biến imageGallery.
3. **var images = ['image/image-1.jpg', 'image/image-2.jpg', 'image/image-3.jpg'];**:
   * Tạo một mảng chứa các URL của các hình ảnh sẽ được sử dụng trong thư viện ảnh.
4. **var i = 0;**:
   * Khởi tạo biến i để theo dõi chỉ số của ảnh hiện tại trong mảng images.
5. **setInterval(function() {...}, 2000);**:
   * Thiết lập một hàm sẽ được thực thi mỗi 2 giây (2000 milliseconds).
6. **i = (i + 1) % images.length;**:
   * Tăng chỉ số i lên 1 và sử dụng phép toán modulus để đảm bảo rằng i luôn nằm trong phạm vi của mảng images. Lưu ý rằng đã sửa lỗi logic tại đây. Ban đầu bạn viết i = i + 1 % images.length; là sai, vì toán tử % có độ ưu tiên cao hơn +, dẫn đến kết quả không mong muốn.
7. **imageGallery.fadeOut(function() {...});**:
   * Làm mờ ảnh hiện tại, sau đó thay đổi thuộc tính src của ảnh bằng URL mới từ mảng images và làm cho ảnh xuất hiện lại.

Thây đổi thuộc tính CSS của các phần tử

1. Thây đổi thuộc tính css của phần tử

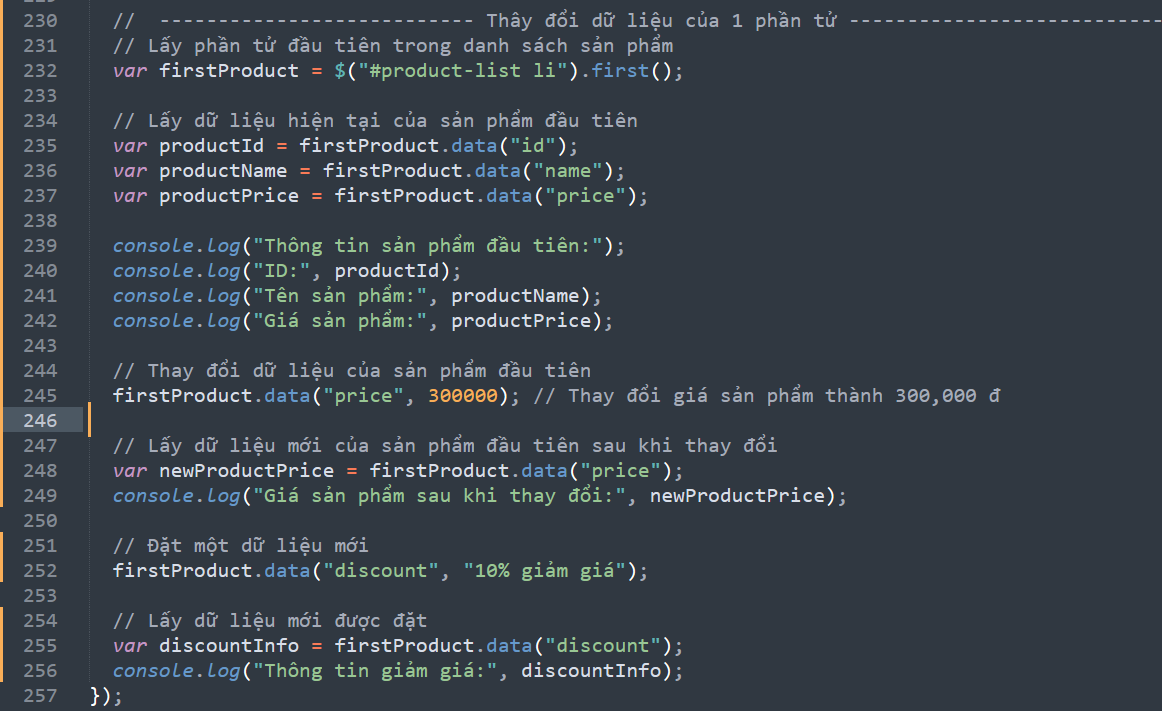


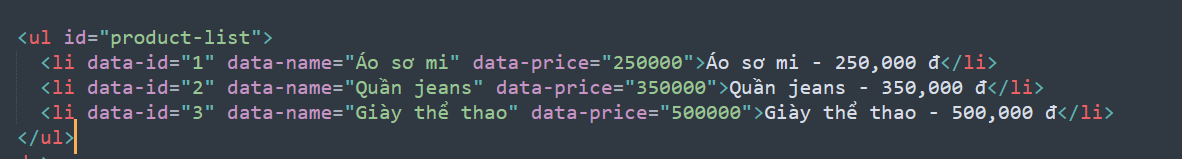
1. Thêm hoặc xóa các lớp css



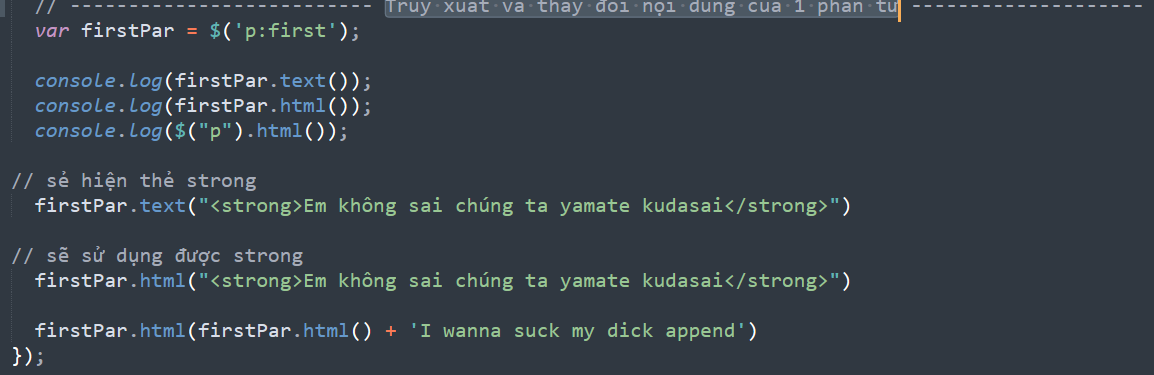
* + - Thêm class có tên là fancy-link
    - Thẻ p đầu tiên add thêm class có tên là lange emphase
    - Thẻ li add thêm class có tên item-index
    - Tìm nếu có class nào có tên baby-shark thì đổi tên thành red box
    - Xóa đi class có tên là red box và thêm vào blue box

3.Thây đổi dữ liệu của 1 phần tử





4. Truy xuất và thây đổi nội dung của 1 phần tử



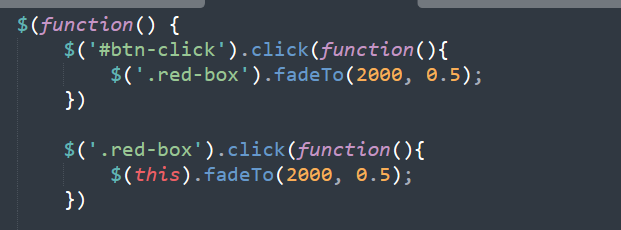
Dòng 1 khởi tạo biến firstPar là thẻ p thứ nhất.

Sau đó log ra màn hình theo 3 kiểu.

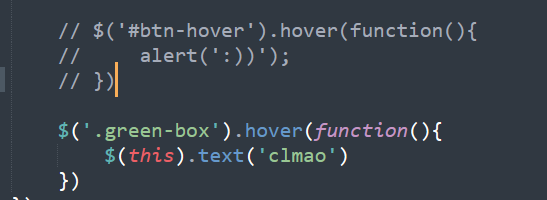
Xem cách hiển thị dữ liệu theo từng hàm.

Sử lí sự kiện với chuột và bàn phím

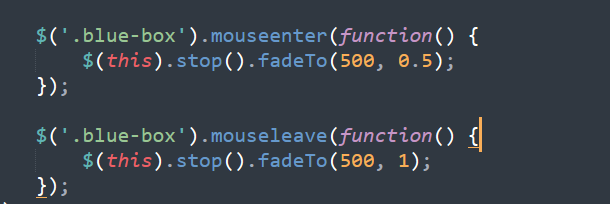
1. Thêm trình sử lí nhấp chuột



* 1. Thêm trình sử lí hover

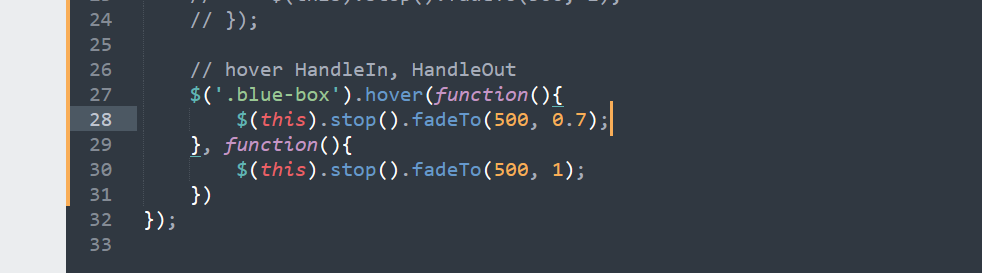


* 1. Thêm trình sử lí Mouse Enter và MouseLeave



 **Không sử dụng .stop():** Nếu bạn nhanh chóng di chuột vào và ra khỏi phần tử nhiều lần, các hiệu ứng .fadeTo() sẽ được xếp chồng lên nhau trong hàng đợi hoạt ảnh của jQuery. Điều này dẫn đến các hiệu ứng chạy tuần tự, gây ra hiện tượng nhấp nháy hoặc kéo dài thời gian hoạt ảnh tổng thể.

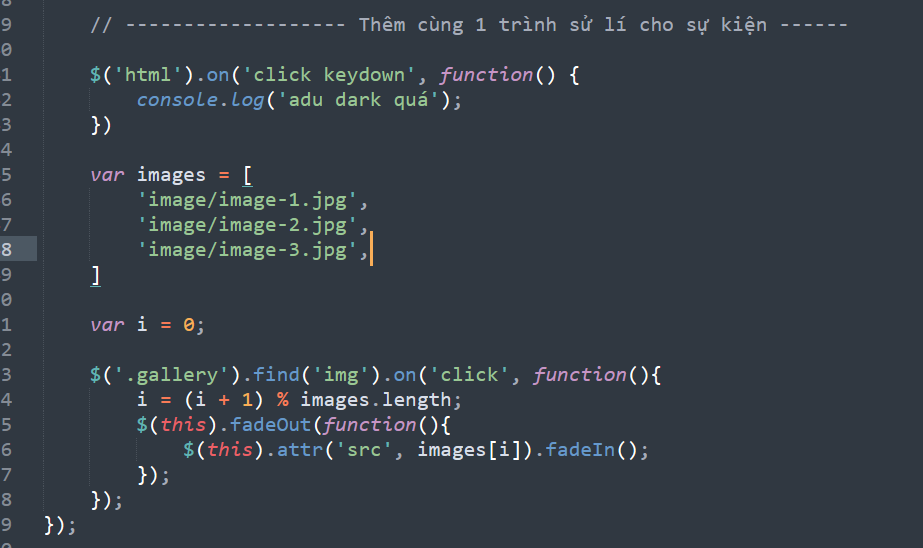
 **Sử dụng .stop():** Phương thức .stop() dừng ngay lập tức tất cả các hoạt ảnh đang chạy hoặc được xếp chồng lên nhau trong hàng đợi trên phần tử đó. Khi bạn sử dụng .stop(), nó sẽ dừng mọi hoạt ảnh hiện tại và bắt đầu thực hiện hoạt ảnh mới ngay lập tức, giúp ngăn chặn các hiệu ứng chồng chéo và đảm bảo hoạt ảnh diễn ra mượt mà hơn.

Cách 2 : 

* 1. Thêm cùng 1 trình sử lí cho sự kiện

SỰ kiện số 1 nhấn vào bất kì đâu trên bàn phím hoặc nhấn chuột sẽ console.log ra

Sự kiện thứ 2 click vào tấm ảnh để slide đổi ảnh



* 1. Modun hóa trình sử lí sự kiện ( Không còn chức năng nội tuyến nữa).

Đơn giản là nó k khác gì ở mục 4 nhưng nó tách ra từng quá trình

* + - Dễ bảo trì
    - Tái sử dụng mã
    - Dễ dàng kiểm tra
    - Đọc hiểu tốt hơn
    - Dễ dàng mở rộng hoặc thây đổi



* 1. Sự kiện được ủy quyền