

THỰC HÀNH CÔNG NGHỆ WEB

BUỔI 2: JQUERY

YÊU CẦU CHUNG:

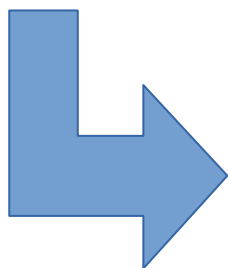
- Mỗi bài thực hành được đặt trong một thư mục riêng. Ví dụ bài làm cho bài thực hành 1 sẽ được đặt trong thư mục “Bai1”.
- Sau khi làm xong các bài thực hành, sinh viên tạo một tập tin nén (zip hoặc rar) chứa tất cả các bài thực hành đã làm trong buổi thực hành với tên tập tin có dạng như sau <MSSV>_<TenSV>_Lab<xx>.zip, trong đó <MSSV>, <TenSV> lần lượt là mã số sinh viên và họ tên sinh viên, <xx> là số thứ tự buổi thực hành. Ví dụ các bài làm trong buổi thực hành thứ 03 của bạn Nguyễn Văn Hải có MSSV 12345 sẽ được đặt trong tập tin nén có tên 12345_NVHai_Lab03.zip. Tập tin zip này sẽ được đặt trong thư mục “D:\CT275”.

Bài 1. Làm việc với một thư viện JavaScript.

- **Mục tiêu:** Biết cách làm việc với một thư viện JavaScript.
- **Yêu cầu:** Trong bài tập này, sinh viên sẽ sử dụng JavaScript DOM API và các hàm từ thư viện *youtube.js* để chuyển đổi một danh sách các đường dẫn video dạng văn bản sang một danh sách các đường dẫn video dạng ảnh thumbnail.

Các đoạn phim hay nhất

- Trip through the 80s
- Otters Holding Hands
- The Ooooh Catttttt



Các đoạn phim hay nhất

- Trip through the 80s



- Otters Holding Hands



- The Ooooh Catttttt



- **Hướng dẫn:**

1) Chọn các đường dẫn (thẻ <a>) trong trang web với:

var videoLinks = document.querySelectorAll(cssSelector).

2) Duyệt qua các thẻ <a>: *for (var i = 0; i < videoLinks.length; i++)*

Đối với mỗi thẻ <a> (*videoLinks[i]*):

- Tìm kiếm giá trị thuộc tính href (đường dẫn đến đoạn phim) của thẻ <a>, lưu giá trị tìm được vào một biến:
var linkURL = videoLinks[i].getAttribute('href'),
- Tạo ra một đường dẫn đến ảnh thumbnail bằng cách gọi hàm *generateThumbnailUrl* như sau:
var thumbnailURL = youtube.generateThumbnailUrl(linkURL).
- Tạo ra một thẻ sử dụng *document.createElement(tagName)*:
var thumbnailImg = document.createElement('img');
- Gán giá trị cho thuộc tính src của thẻ img sử dụng hàm *setAttribute(name, value)*:
thumbnailImg.setAttribute('src', thumbnailURL);
- Gắn thẻ vào thẻ <a> sử dụng *element.appendChild(element)*:
videoLink.appendChild(thumbnailImg);

Bài 2. Làm việc với thư viện jQuery.

- **Mục tiêu:** Biết cách làm việc với thư viện jQuery.
- **Yêu cầu:** Yêu cầu như bài tập 1 nhưng sử dụng jQuery để thao tác với DOM.
- **Hướng dẫn:**

1) Bắt đầu với kết quả của bài tập 1.

2) Tải thư viện jQuery về từ trang jquery.com và lưu vào cùng thư mục với tập tin bài tập. Sau đó thêm thẻ <script> vào tài liệu html tham chiếu đến tập tin jQuery vừa lưu.

3) Thay thế các mã lệnh thao tác với DOM trong bài tập 1 với jQuery.

Bài 3. Xử lý sự kiện và tạo hiệu ứng với jQuery.

- **Mục tiêu:** Biết cách xử lý sự kiện và tạo hiệu ứng đơn giản với jQuery.
- **Yêu cầu:** Từ kết quả bài tập 2, thêm mã lệnh cần thiết để có thể xem các đoạn phim ngay bên trong trang web.

Các đoạn phim hay nhất

- [Trip through the 80s](#)



- [Otters Holding Hands](#)



- [The Ooooh Catttttt](#)



- **Hướng dẫn:**

1) Bắt đầu với kết quả của bài tập 2.

2) Trong thẻ `<div class='row'>` thứ 2 của tài liệu html, tạo một thẻ `<div>` với lớp `col-sm-9` bên dưới thẻ `<div class='col-sm-3'>` để dành chỗ phát các đoạn phim. Đồng thời, gán cho thẻ `<div>` mới này một id.

3) Thêm hàm xử lý sự kiện nhấp chuột (click listener) vào các thẻ `<a>`.

4) Trong hàm xử lý sự kiện nhấp chuột của thẻ `<a>`, đầu tiên ta vô hiệu hóa sự kiện mặc định khi nhấp chọn đường dẫn (sử dụng hàm `e.preventDefault()`). Sau đó, ta tạo thẻ `<iframe>` (`$('#<iframe></iframe>')`). Thiết lập các thuộc tính sau cho thẻ này:

- Thuộc tính 'src' chứa đường dẫn nhúng (embed url) của đoạn phim cần phát (sử dụng hàm `youtube.generateEmbedUrl(videoIdOrUrl)`).
- Thuộc tính 'width' và 'height' để tùy chỉnh kích thước khung phát các đoạn phim.

5) Thêm thẻ `<iframe>` vào thẻ `<div>` đã tạo ở bước 2 (gọi hàm `.html()`).

6) Thêm hiệu ứng “fade in” cho thẻ `<div>` chứa khung phát các đoạn phim.

Bài 4. Làm việc với các plugin của jQuery.

- **Mục tiêu:** Biết cách làm việc với các plugin của jQuery.

- **Yêu cầu:**

1) Sử dụng kết quả của bài tập 3, tạo hiệu ứng bóng đổ ba chiều (3D shadow) cho đề mục trang và cho các đường liên kết sử dụng plugin `funText (jquery.funtext.js)`.

Các đoạn phim hay nhất

- Trip through the 80s



- Otters Holding Hands



- The Ooooh Catttttt



2) jQuery Validation Plugin (<http://jqueryvalidation.org/>) là một trong những plugin được sử dụng phổ biến để tra dữ liệu nhập vào các biểu mẫu. Trong bài tập này, sinh viên được cung cấp một biểu mẫu đăng ký thành viên đơn giản và nhiệm vụ của sinh viên là sử dụng jQuery Validation Plugin để kiểm tra dữ liệu người dùng nhập vào biểu mẫu đã cho với các quy tắc:

- Người dùng phải cung cấp tất cả các thông tin yêu cầu.
- Tên đăng nhập có ít nhất 2 ký tự.
- Mật khẩu có ít nhất 5 ký tự.
- Trường hợp thư điện tử phải là một địa chỉ email hợp lệ.
- Người dùng phải đồng ý với các quy định của tổ chức.
- Trường nhập liệu nào của biểu mẫu không hợp lệ thì (1) làm nổi bật trường nhập liệu bị lỗi và (2) hiển thị một thông báo lỗi phía dưới trường nhập liệu đó.

Tên của bạn	<input type="text"/>	<i>Bạn chưa nhập vào họ của bạn</i>
Họ của bạn	<input type="text"/>	<i>Bạn chưa nhập vào tên của bạn</i>
Tên đăng nhập	<input type="text"/>	<i>Bạn chưa nhập vào tên đăng nhập</i>
Hộp thư điện tử	<input type="text"/>	<i>Hộp thư điện tử không hợp lệ</i>
Mật khẩu	<input type="password"/>	<i>Bạn chưa nhập mật khẩu</i>
Nhập lại mật khẩu	<input type="password"/>	<i>Bạn chưa nhập mật khẩu</i>
<input type="checkbox"/> Đồng ý các quy định của chúng tôi <i>Bạn phải đồng ý với các quy định của chúng tôi</i>		
<input type="button" value="Đăng ký"/>		

Hiện thị thông báo lỗi phía dưới trường thông tin không hợp lệ.

- **Hướng dẫn:**

1) Tài liệu tham khảo cho plugin funText có sẵn tại: <http://briznad.github.io/funText/>

2) Tài liệu tham khảo cho jQuery Validation Plugin có sẵn tại:
<http://jqueryvalidation.org/documentation/>. Các bước làm đề nghị:

- Gán cho các thẻ <form>, <input> một định danh (thuộc tính id). Ngoài ra, jQuery Validation Plugin bắt buộc các thẻ <input> phải có thêm thuộc tính name. Giá trị của thuộc tính name thông thường trùng với giá trị của thuộc tính id. Bên cạnh đó, mỗi thẻ <input> thường đi kèm trước nó một thẻ <label> để hiển thị nhãn miêu tả cho nó. Thẻ <label> này có thuộc tính for có giá trị là id của thẻ <input> mà nó miêu tả. Ví dụ:

```
<label for="firstname">Firstname</label>
<input id="firstname" name="firstname">
```

- Để kiểm tra các trường nhập liệu của biểu mẫu, ta có thể sử dụng hàm `.validate()` cung cấp bởi jQuery Validation Plugin:

```
$('#<id-của-form>').validate ( {
    rules: {
        // Các quy tắc kiểm tra tính hợp lệ. Xem tài liệu tham khảo
        // cung cấp bởi jQuery Validation Plugin.
    },
    messages: {
        // Các thông báo lỗi. Xem tài liệu tham khảo cung cấp bởi
        // jQuery Validation Plugin.
    },
    errorElement: "em", // Thông báo lỗi sẽ được đặt trong thẻ <em>.
    errorPlacement: function (error, element) {
        // Tùy biến vị trí đặt các thông báo lỗi. Tham số đầu tiên là thẻ
        // chứa thông báo lỗi (thẻ <em> dưới dạng một đối tượng
        // jQuery), tham số thứ hai là trường nhập liệu bị lỗi (thẻ
        // <input> dưới dạng đối tượng jQuery).
        // Trong bài tập này ta sẽ thêm lớp 'help-block' vào thẻ <em>
        // (sử dụng .addClass()) và chèn thẻ <em> phía dưới thẻ
        // <input> bị lỗi (sử dụng .insertAfter()).
    },

    highlight: function (element, errorClass, validClass) {
        // Làm thế nào để làm nổi bật trường nhập liệu lỗi: thêm lớp
        // 'has-error', xóa lớp 'has-success' ở thẻ <div> cha của thẻ
        // <input> bị lỗi (tham số element là một đối tượng jQuery đại
        // diện cho thẻ <input> bị lỗi).
    },

    unhighlight: function (element, errorClass, validClass) {
        // Ngược lại với tùy chọn highlight ở trên.
    }
});
```

```
}  
} ); // .validate()
```

Bài 5. Ajax/JSON và jQuery.

- **Mục tiêu:** Biết cách làm việc với Ajax/Json sử dụng jQuery.
- **Yêu cầu:** Trong phần này, sinh viên sẽ sử dụng hàm \$.ajax của jQuery để đọc nội dung một tập tin JSON và chuyển dữ liệu JSON thành các đối tượng DOM. Cụ thể:
 - Đọc hiểu các đoạn mã lệnh đã cung cấp (Bai5.html).
 - Sử dụng hàm \$.ajax của jQuery để đọc nội dung tập tin videos.json. Nếu đọc thành công thì hiển thị các đoạn phim lên trang web sử dụng hàm addVideoToList đã có.
- **Hướng dẫn:** Tham khảo slide bài giảng hoặc tài liệu của jQuery.

Bài 6. Tạo thư viện ảnh tương tác với jQuery.

- **Mục tiêu:**
 - Biết cách xử lý sự kiện trong jQuery.
 - Sử dụng được API cung cấp bởi jQuery để truy xuất các phần tử DOM.
 - Biết cách sử dụng jQuery plugin của Bootstrap.
- **Yêu cầu:** Sinh viên được cung cấp một thư viện ảnh tĩnh dưới dạng một tài liệu html (tương tự như bài thực hành 3 buổi 1). Nhiệm vụ của sinh viên là thêm các mã lệnh JavaScript để tạo các tương tác giữa người dùng với thư viện ảnh nói trên. Cụ thể, khi người dùng nhấp chuột vào bất kỳ ảnh nào trong tài liệu thì ảnh được chọn sẽ xuất hiện trong một hộp thoại (modal). Phía bên dưới hộp thoại, có hai đường dẫn điều hướng: “kế tiếp” và “trước đó”. Khi nhấp chọn “kế tiếp” thì ảnh tiếp theo trong thư viện sẽ được hiển thị trong hộp thoại. Ngược lại, khi nhấp chọn “trước đó” thì ảnh trước đó trong thư viện sẽ được hiển thị. Nếu ảnh đầu tiên trong thư viện được hiển thị trong hộp thoại thì sẽ không có đường dẫn “trước đó”. Tương tự, nếu ảnh cuối cùng trong thư viện được hiển thị thì sẽ không có đường dẫn “kế tiếp”.



g một hộp thoại với
hai đường dẫn điều hướng “trước đó” và “kế tiếp”.

- **Hướng dẫn:**

1) Thêm hàm xử lý sự kiện nhấp chuột cho tất cả các thẻ `` là hậu duệ của thẻ `` (sử dụng bộ chọn `'li img'`). Trong hàm xử lý sự kiện này, đầu tiên ta lấy về đường dẫn đến tập tin ảnh (thuộc tính `src` của thẻ ``). Từ đường dẫn đến tập tin ảnh này, ta tạo ra một thẻ `` mới với lớp là `img-responsive`. Thẻ `` này sẽ được đặt trong thân của hộp thoại modal sẽ được hiển thị.

2) Tiếp theo ta cần xác định vị trí của ảnh trong thư viện. Ta thấy rằng các ảnh nằm trong các thẻ ``. Như vậy, vị trí của thẻ `` cũng chính là vị trí của ảnh trong thư viện:

```
var index = $(this).parent('li').index() + 1;
```

Câu lệnh trên nhằm mục đích xác định vị trí của thẻ cha `` của thẻ `` hiện thời (lưu ý rằng ta đang ở trong hàm xử lý sự kiện nhấp chuột của thẻ ``).

3) Bước kế tiếp, ta cần xây dựng nội dung cho hộp thoại (modal). Nội dung của hộp thoại bao gồm thẻ `` ở bước 1 và hai đường dẫn điều hướng:

```
<img src='...' class='img-responsive' />
```

```
<div style='height:25px;clear:both;display:block;'>
```



```
<a class='controls next' href='<vị trí ảnh kế tiếp>'>kế tiếp &raquo;</a>
```

```
<a class='controls previous' href='<vị trí ảnh trước đó>'>&laquo; trước đó</a>
```

```
</div>
```

4) Sau khi có được nội dung trên, ta tiến thành hiển thị hộp thoại (tham khảo phần hộp thoại modal của Bootstrap). Khi hộp thoại hiển thị thì viết nội dung lên thân hộp thoại (sử dụng hàm .html()). Ngược lại, khi ẩn hộp thoại thì xóa nội dung trong thân hộp thoại đi.

5) Thêm hàm xử lý sự kiện nhấp chuột vào hai đường dẫn điều hướng trong hộp thoại. Lưu ý rằng ở mỗi đường dẫn đều có lưu vị trí của ảnh tương ứng trong thuộc tính href. Như vậy, mỗi khi một đường dẫn điều hướng được nhấp chọn, ta sẽ lấy về thông tin vị trí của ảnh tương ứng và từ đó truy vấn đường dẫn đến tập tin ảnh:

```
var src = $('ul.row li:nth-child(' + index + ') img').attr('src');
```

Câu lệnh trên truy xuất đến đường dẫn tập tin ảnh ở thẻ con của thẻ thứ *index*.

6) Thay thế ảnh hiện thời trong hộp thoại bằng ảnh mới thu được. Đồng thời cập nhật lại các giá trị href của hai đường dẫn điều hướng.

7) Khi giá trị href của đường dẫn “trước đó” bằng 0 thì ẩn đường dẫn “trước đó”. Ngược lại, khi giá trị href của đường dẫn “kế tiếp” bằng (số lượng thẻ + 1) thì ẩn đường dẫn “kế tiếp” (ta có thể đếm số lượng thẻ bằng lệnh sau: \$('ul.row li').length).