

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TR- ỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

**TÌM HIỂU VỀ JQUERY VÀ PHP TRONG VIỆC XÂY
DỰNG CÁC ỨNG DỤNG WEB**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

HẢI PHÒNG - 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

TÌM HIỂU VỀ JQUERY VÀ PHP TRONG VIỆC XÂY DỰNG CÁC ỨNG DỤNG WEB

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Vũ Xuân Chiển

Giáo viên hướng dẫn: Th.S Đỗ Văn Chiểu

Mã số sinh viên: 121289

HẢI PHÒNG - 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
-----o0o-----

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Vũ Xuân Chiến

Mã số: 121289

Lớp: CT1201

Ngành: Công nghệ Thông tin

Tên đề tài:

Tìm hiểu về jQuery và PHP trong việc xây dựng các ứng dụng WEB

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

b. Các yêu cầu cần giải quyết

.....

.....

.....

.....

2. Các số liệu cần thiết để thiết kế, tính toán

.....

.....

.....

.....

3. Địa điểm thực tập

.....

.....

.....

.....

.....

.....

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên:.....

Học hàm, học vị:.....

Cơ quan công tác:.....

Nội dung hướng dẫn:

.....

.....

.....

Người hướng dẫn thứ hai:

Họ và tên:

Học hàm, học vị.....

Cơ quan công tác:

Nội dung hướng dẫn:

.....

.....

.....

Đề tài tốt nghiệp được giao ngày ... tháng năm 20...

Yêu cầu phải hoàn thành trước ngày ... tháng ... năm 20...

Đã nhận nhiệm vụ: Đ.T.T.N

Sinh viên

Đã nhận nhiệm vụ: Đ.T.T.N

Cán bộ hướng dẫn Đ.T.T.N

Hải Phòng, ngàytháng.....năm 20...

HIỆU TRƯỞNG

GS.TS.NGUYỄN Trần Hữu Nghị

PHÂN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Cho điểm của cán bộ hướng dẫn:

(Điểm ghi bằng số và chữ)

.....

.....

.....

.....

.....

Ngày.....tháng.....năm 20...

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

LỜI CẢM ƠN

Trước tiên em xin được bày tỏ sự trân trọng và lòng biết ơn đối với thầy giáo Ths Đỗ Văn Chiêu, giảng viên Khoa Công nghệ thông tin – Trường Đại học dân lập Hải Phòng. Trong suốt thời gian học và làm đồ án tốt nghiệp, thầy đã dành rất nhiều thời gian quý báu để tận tình chỉ bảo, hướng dẫn, định hướng cho em trong việc nghiên cứu, thực hiện đồ án.

Em xin được cảm ơn các thầy cô giáo Trường Đại học Dân lập Hải phòng đã giảng dạy em trong quá trình học tập, thực hành, làm bài tập, đọc và nhận xét đồ án của em, giúp em hiểu thấu đáo hơn lĩnh vực mà em nghiên cứu, những hạn chế mà em cần khắc phục trong việc học tập, nghiên cứu và thực hiện bản luận văn này.

Xin cảm ơn các bạn bè, đồng nghiệp và nhất là các thành viên trong gia đình đã tạo mọi điều kiện tốt nhất, động viên, cổ vũ em trong suốt quá trình học và làm luận văn tốt nghiệp.

Hải Phòng, Tháng 4 năm 2013

Sinh viên

Vũ Xuân Chiến

MỤC LỤC

LỜI CẢM ƠN	9
MỤC LỤC.....	10
MỞ ĐẦU	12
Chương 1: TỔNG QUAN VỀ PHP	13
1.1. Lịch sử phát triển.....	13
1.1.1. PHP.....	13
1.1.2. PHP3.....	13
1.1.3. PHP4.....	13
1.1.4. PHP5.....	13
1.1.5. PHP6.....	14
1.2. Cấu trúc cơ bản của PHP	14
1.2.1. Các cấu trúc cơ bản	14
1.2.2. Xuất giá trị ra trình duyệt	15
1.2.3. Biến, hằng, chuỗi và các kiểu dữ liệu.....	15
1.2.4. Các phương thức được sử dụng trong lập trình PHP	17
1.2.5. Cookie và Session trong PHP.....	18
1.2.6. Cookie và Session trong PHP.....	19
Chương 2: TỔNG QUAN VỀ JQUERY	21
2.1. Giới thiệu về jQuery	21
2.2. jQuery có thể làm gì	21
2.2.1. Hướng tới các thành phần trong tài liệu HTML.....	21
2.2.2. Thay đổi giao diện của một trang web	21
2.2.3. Tương tác với người dùng	21
2.2.4. Tạo hiệu ứng động cho những thay đổi của tài liệu	22
2.2.5. Lấy thông tin từ server mà không cần tải lại trang web	22
2.3. Tại sao jQuery làm việc tốt	22
2.3.1. Tận dụng kiến thức về CSS.....	22
2.3.2. Hỗ trợ Plugin	22
2.3.3. Loại bỏ sự khác biệt giữa các trình duyệt.....	22
2.3.4. Luôn làm việc với Set.....	23
2.3.5. Cho phép nhiều tác vụ diễn ra trên cùng một dòng.....	23
Chương 3 : SỬ DỤNG JQUERY	23
3.1. Các cú pháp jQuery cơ bản.....	23
3.1.1. Cài đặt thư viện jQuery	23
3.1.2. Gọi hàm jQuery	23
3.1.3. jQuery Selector.....	24
3.1.4. jQuery Attribute.....	25
3.1.5. jQuery Events	27
3.2. JQUERY EFFECTS	30

3.2.1. Hide/Show	30
3.2.2. Fading	32
3.2.3. Animate	36
3.2.3. Stop.....	38
3.3. jQuery HTML.....	40
3.3.1. jQueryGet	40
3.3.2. jQuery Set.....	42
3.3.3. jQuery giúp người dùng thao tác với các thành phần của HTML.....	43
3.3.4. jQuery giúp người dùng có thể thao tác, tác động tới CSS.....	44
3.4. jQueryAjax	44
3.4.1. Giới thiệu.....	44
3.4.2. Các phương thức jQuery Ajax.....	45
Chương 4 : CHƯƠNG TRÌNH ỨNG DỤNG WEB SỬ DỤNG JQUERY VA PHP.....	48
4.1. Chương trình ứng dụng web Upload và hiển thị ảnh.	48
4.2. Các chức năng chính:	48
4.2.1. Upload và hiển thị ảnh theo danh sách:.....	48
4.2.2. Tạo menu:	49
4.2.3. Hiển thị ảnh theo slide:.....	50
4.2.3. Thực hiện hiệu ứng lightBox:.....	51
KẾT LUẬN	54

MỞ ĐẦU

Với sự phát triển rất mau lẹ của Internet, người dùng ngày càng quan tâm hơn đến hình thức, tốc độ, tính năng của một trang web. Trước đây một trang web chỉ cần có banner, nội dung và ít footer hời hợt là đã được cho là một trang web hoàn chỉnh. Nhưng bây giờ trang web đó phải có banner bắt mắt, nội dung hay và còn nhiều hiệu ứng lạ mắt khác nữa đồng thời tốc độ xử lý và các tính năng phù hợp thì mới có thể thu hút được người dùng.

Chính vì thế để xây dựng một trang web có đầy đủ các tính năng theo yêu cầu của người dùng, đồng thời đáp ứng được tốc độ xử lý cho phép và một giao diện bắt mắt có thể tương tác với người đọc một cách nhanh nhất và hiệu quả nhất, các nhà phát triển, thiết kế web đã kết hợp việc sử dụng ngôn ngữ PHP - ngôn ngữ lập trình web phổ biến nhất hiện nay – với việc sử dụng thư viện JavaScript mở như jQuery.

Trên cơ sở đó tôi hi vọng rằng việc tìm hiểu về PHP và jQuery trong việc xây dựng các ứng dụng web sẽ giúp nâng cao kiến thức của bản thân trong việc lập trình và xây dựng các ứng dụng web. Đồng thời giúp các bạn sinh viên mới đi sâu về lĩnh vực này có thể hiểu rõ hơn về PHP và jQuery

Đề án gồm 4 chương:

Chương 1: Tổng hợp các kiến thức cơ bản về PHP. Đưa ra cái nhìn tổng quan về ngôn ngữ lập trình web phổ biến này.

Chương 2: Đưa ra cái nhìn tổng quan về jQuery. Giới thiệu các đặc điểm cơ bản của thư viện JavaScript này.

Chương 3: Các cấu trúc của jQuery, một số cách sử dụng cơ bản.

Chương 4: Ứng dụng jQuery vào một trang web PHP

Cuối cùng là kết luận và tài liệu tham khảo.

Chương 1: TỔNG QUAN VỀ PHP

1.1. Lịch sử phát triển

1.1.1. PHP

Được phát triển từ một sản phẩm có tên là PHP/FI. PHP/FI do Rasmus Lerdorf tạo ra năm 1995, ban đầu được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của ông trên mạng. Ông đã đặt tên cho bộ mã kịch bản này là 'Personal Home Page Tools'..

1.1.2. PHP3

PHP 3.0 là phiên bản đầu tiên cho chúng ta thấy một hình ảnh gần gũi với các phiên bản PHP mà chúng ta được biết ngày nay. Nó đã được Andi Gutmans và Zeev Suraski tạo ra năm 1997 sau khi viết lại hoàn toàn bộ mã nguồn trước đó. PHP 3.0 đã chính thức được công bố vào tháng 6 năm 1998, sau thời gian 9 tháng được cộng đồng kiểm nghiệm.

1.1.3. PHP4

Vào mùa đông năm 1998, ngay sau khi PHP 3.0 chính thức được công bố, Andi Gutmans và Zeev Suraski đã bắt đầu bắt tay vào việc viết lại phần lõi của PHP. Một động cơ mới, có tên 'Zend Engine' (ghép từ các chữ đầu trong tên của Zeev và Andi), đã đáp ứng được các nhu cầu thiết kế này một cách thành công, và lần đầu tiên được giới thiệu vào giữa năm 1999. PHP 4.0, dựa trên động cơ này, và đi kèm với hàng loạt các tính năng mới bổ sung, đã chính thức được công bố vào tháng 5 năm 2000, gần 2 năm sau khi bản PHP 3.0 ra đời

1.1.4. PHP5

Sự thành công hết sức to lớn của PHP 4.0 đã không làm cho nhóm phát triển PHP tự mãn. Cộng đồng PHP đã nhanh chóng giúp họ nhận ra những yếu kém của PHP 4 đặc biệt với khả năng hỗ trợ lập trình hướng đối tượng (OOP), xử lý XML, không hỗ trợ giao thức máy khách mới của MySQL 4.1 và 5.0, hỗ trợ dịch vụ web yếu. Những điểm này chính là mục đích để Zeev và Andi viết Zend Engine 2.0, lõi của PHP 5.0. Ngày 29 tháng 6 năm 2003, PHP 5 Beta 1 đã chính thức được công bố để cộng đồng kiểm nghiệm. Đó cũng là phiên bản đầu tiên của Zend Engine 2.0. Phiên bản Beta 2 sau đó đã ra mắt vào tháng 10 năm 2003 với sự xuất hiện của hai tính năng

rất được chờ đợi: Iterators, Reflection nhưng namespaces một tính năng gây tranh cãi khác đã bị loại khỏi mã nguồn. Ngày 21 tháng 12 năm 2003: PHP 5 Beta 3 đã được công bố để kiểm tra với việc phân phối kèm với Tidy, bỏ hỗ trợ Windows 95, khả năng gọi các hàm PHP bên trong XSLT, sửa chữa nhiều lỗi và thêm khá nhiều hàm mới. PHP 5 bản chính thức đã ra mắt ngày 13 tháng 7 năm 2004 sau một chuỗi khá dài các bản kiểm tra thử bao gồm Beta 4, RC 1, RC2, RC3. Mặc dù coi đây là phiên bản sản xuất đầu tiên nhưng PHP 5.0 vẫn còn một số lỗi trong đó đáng kể là lỗi xác thực HTTP.

1.1.5. PHP6

Hiện nay phiên bản tiếp theo của PHP đang được phát triển, PHP 6 bản sử dụng thử đã có thể được download tại địa chỉ <http://snaps.php.net>. Phiên bản PHP 6 được kỳ vọng sẽ lấp đầy những khiếm khuyết của PHP ở phiên bản hiện tại, ví dụ: hỗ trợ namespace (hiện tại các nhà phát triển vẫn chưa công bố rõ ràng về vấn đề này); hỗ trợ Unicode; sử dụng PDO làm API chuẩn cho việc truy cập cơ sở dữ liệu, các API cũ sẽ bị đưa ra thành thư viện PECL...

1.2. Cấu trúc cơ bản của PHP

1.2.1. Các cấu trúc cơ bản

PHP cũng có thể bắt đầu và kết thúc giống với ngôn ngữ HTML. Chỉ khác, đối với PHP chúng ta có nhiều cách để thể hiện.

Cách 1: Cú pháp chính:

```
<?php Mã lệnh PHP ?>
```

Cách 2: Cú pháp ngắn gọn

```
<? Mã lệnh PHP ?>
```

Cách 3: Cú pháp giống với ASP.

```
<% Mã lệnh PHP %>
```

Cách 4: Cú pháp bắt đầu bằng script

```
<script language=php>
```

.....

```
</script>
```

Mặc dù có 4 cách thể hiện. Nhưng đối với 1 lập trình viên có kinh nghiệm thì việc sử dụng cách 1 vẫn là lựa chọn tối ưu.

Trong PHP để kết thúc 1 dòng lệnh chúng ta sử dụng dấu ";"

Để chú thích 1 đoạn dữ liệu nào đó trong PHP ta sử dụng dấu "//" cho từng dòng. Hoặc dùng cặp thẻ "/**/" cho từng cụm mã lệnh.

Ví dụ: <?php echo "Hello "; ?>

1.2.2. Xuất giá trị ra trình duyệt

Để xuất dữ liệu ra trình duyệt chúng ta có những dòng cú pháp sau:

+ echo "Thông tin";

+ printf "Thông tin";

Thông tin bao gồm: biến, chuỗi, hoặc lệnh HTML

```
1 <?php
2 Echo "Hello word";
3 Printf"<br><font color=red>Who Are You ?</font>";
4 ?>
5
```

Nếu giữa hai chuỗi muốn liên kết với nhau ta sử dụng dấu "."

1.2.3. Biến, hằng, chuỗi và các kiểu dữ liệu

a) Biến

Biến được xem là vùng nhớ dữ liệu tạm thời. Và giá trị có thể thay đổi được. Biến được bắt đầu bằng ký hiệu "\$". Và theo sau chúng là 1 từ, 1 cụm từ nhưng phải viết liền hoặc có gạch dưới.

Một biến được xem là hợp lệ khi nó thỏa các yếu tố:

+ Tên của biến phải bắt đầu bằng dấu gạch dưới và theo sau là các ký tự, số hay dấu gạch dưới.

+ Tên của biến không được phép trùng với các từ khóa của PHP.

Trong PHP để sử dụng 1 biến chúng ta thường phải khai báo trước, tuy nhiên đối với các lập trình viên khi sử dụng họ thường xử lý cùng một lúc các công việc, nghĩa là vừa khai báo vừa gán dữ liệu cho biến.

Bản thân biến cũng có thể gán cho các kiểu dữ liệu khác. Và tùy theo ý định của người lập trình mong muốn trên chúng.

```

1 <?
2 $a= 100 // biến a ở đây có giá trị là 100.
3 $a= "PHP is easy" // Biến a ở đây có giá trị "PHP Is easy".
4 Biens=123 //Có lỗi vì bắt đầu 1 biến phải có dấu "$"
5 $123a="PHP" //Có lỗi vì phần tên bắt đầu của biến là dạng số.
6 ?>

```

Biến trong PHP

b) Hằng

Nếu biến là cái có thể thay đổi được thì ngược lại hằng là cái chúng ta không thể thay đổi được. Hằng trong PHP được định nghĩa bởi hàm define theo cú pháp: define (string tên_hằng, giá_trị_hằng).

Cũng giống với biến hằng được xem là hợp lệ thì chúng phải đáp ứng 1 số yếu tố:

- + Hằng không có dấu "\$" ở trước tên.
- + Hằng có thể truy cập bất cứ vị trí nào trong mã lệnh
- + Hằng chỉ được phép gán giá trị duy nhất 1 lần.
- + Hằng thường viết bằng chữ in để phân biệt với biến

```

1 <?
2 define ("C", "COMPANY");
3 define ("YELLOW", "#ffff00");
4 echo "Gia tri cua C la". C;
5 ?>

```

Hằng trong PHP

C) Chuỗi

Chuỗi là một nhóm các ký tự, số, khoảng trắng, dấu ngắt được đặt trong các dấu nháy.

Ví dụ:

‘Hello’

Để tạo 1 biến chuỗi, chúng ta phải gán giá trị chuỗi cho 1 biến hợp lệ.

Ví dụ:

\$firsrt_name= "Nguyen";

\$last_name= ‘Van A’;

Để liên kết 1 chuỗi và 1 biến chúng ta thường sử dụng dấu "."


```

1 <?php
2 $test="QHOnline.Info";
3 echo "welcome to".$test;
4 echo "<br><font color=red>welcome to".$test."</font><br>";
5 ?>
6

```

Liên kết chuỗi và biến trong PHP

d) Kiểu dữ liệu

Các kiểu dữ liệu khác nhau chiếm các lượng bộ nhớ khác nhau và có thể được xử lý theo cách khác nhau khi chúng được theo tác trong 1 script.

Trong PHP chúng ta có 6 kiểu dữ liệu chính như sau:

Kiểu Dữ Liệu	Ví dụ	Mô Tả
Integer	10	Một số nguyên
Double	5.208	Kiểu số thực
String	"How are you ?"	Một tập hợp các ký tự
Boolean	True or False	Giá trị true hoặc false
Object	Hướng đối tượng trong PHP	
Array	Mảng trong PHP, chứa các phần tử.	

Kiểu dữ liệu trong PHP

Chúng ta có thể sử dụng hàm dựng sẵn GETTYPE() của PHP4 để kiểm tra kiểu của bất kỳ biến.

```

1 <?php
2 $a= 5;
3 echo gettype($a); // Integer.
4 $a="qhonline.info";
5 echo gettype($a); //String
6 ?>
7

```

Hàm GETTYPE

1.2.4. Các phương thức được sử dụng trong lập trình PHP

a) Phương thức GET

Phương thức này được dùng để lấy dữ liệu từ *form* nhập liệu. Tuy nhiên nhiệm vụ chính của nó vẫn là lấy nội dung từ trang dữ liệu từ web server.

Ví dụ:

Với url sau: shownews.php?id=50

Vậy với trang shownews ta dùng hàm \$_GET['id'] sẽ được giá trị là 50.

b) Phương thức POST

Phương thức này được sử dụng để lấy dữ liệu từ form nhập liệu. Và chuyển chúng lên trình chủ webserver.

```
<?php
    echo "Welcome ".$_POST['hoten']."!";
?>
<html>
    <form name="test" action="#" method="POST">
        Họ tên <input type="text" name="hoten"/>
        <input type="submit" name="OK" value="OK"/>
    </form>
</html>
```

Phương thức POST

1.2.5. Cookie và Session trong PHP

Cookie và Session là hai phương pháp sử dụng để quản lý các phiên làm việc giữa người sử dụng và hệ thống

a) Cookie

Cookie là 1 đoạn dữ liệu được ghi vào đĩa cứng hoặc bộ nhớ của máy người sử dụng. Nó được trình duyệt gửi ngược lên lại server mỗi khi browser tải 1 trang web từ server.

Những thông tin được lưu trữ trong cookie hoàn toàn phụ thuộc vào Website trên server. Mỗi Website có thể lưu trữ những thông tin khác nhau trong cookie, ví dụ thời điểm lần cuối ta ghé thăm Website, đánh dấu ta đã login hay chưa, v.v...

Cookie được tạo ra bởi Website và gửi tới browser, do vậy 2 Website khác nhau (cho dù cùng host trên 1 server) sẽ có 2 cookie khác nhau gửi tới browser. Ngoài ra, mỗi browser quản lý và lưu trữ cookie theo cách riêng của mình, cho nên 2 browser cùng truy cập vào 1 Website sẽ nhận được 2 cookie khác nhau

1. Để thiết lập cookie ta sử dụng cú pháp:

Setcookie("tên cookie", "giá trị", thời gian sống)

Tên cookie là tên mà chúng ta đặt cho phiên làm việc.

Giá trị là thông số của tên cookie.

Ví dụ: setcookie("name", "admin", time()+3600);

2. Để sử dụng lại cookie vừa thiết lập, chúng ta sử dụng cú pháp:

Cú pháp: \$_COOKIE["tên cookies"]

Tên cookie là tên mà chúng ta thiết lập phía trên.

3. Để hủy 1 cookie đã được tạo ta có thể dùng 1 trong 2 cách sau:

+ Cú pháp: setcookie("Tên cookie")

Gọi hàm setcookie với chỉ duy nhất tên cookie mà thôi

+ Dùng thời gian hết hạn cookie là thời điểm trong quá khứ.

Ví dụ: setcookie("name", "admin", time()-3600);

b) Session

Một cách khác quản lý người sử dụng là Session. Session được hiểu là khoảng thời gian người sử dụng giao tiếp với 1 ứng dụng. Một session được bắt đầu khi người sử dụng truy cập vào ứng dụng lần đầu tiên, và kết thúc khi người sử dụng thoát khỏi ứng dụng. Mỗi session sẽ có được cấp một định danh (ID) khác nhau.

- Để thiết lập 1 session ta sử dụng cú pháp: `session_start()`

Đoạn code này phải được nằm trên các kịch bản HTML. Hoặc những lệnh echo, printf. Để thiết lập 1 giá trị session, ngoài việc cho phép bắt đầu thực thi session. Chúng ta còn phải đăng ký 1 giá trị session. Để tiện cho việc gán giá trị cho session đó.

Ta có cú pháp sau: `session_register("Name")`

4. Giống với cookie. Để sử dụng giá trị của session ta sử dụng mã lệnh sau:

Cú pháp: `$_SESSION["name"]`

Với Name là tên mà chúng ta sử dụng hàm `session_register("name")` để khai báo.

5. Để hủy bỏ giá trị của session ta có những cách sau:

`session_destroy()` // Cho phép hủy bỏ toàn bộ giá trị của session

`session_unset()`//Cho phép hủy bỏ session

1.2.6. Cookie và Session trong PHP

Để giảm thời gian lặp lại 1 thao tác code nhiều lần, PHP hỗ trợ người lập trình việc tự định nghĩa cho mình những hàm có khả năng lặp lại nhiều lần trong Website. Việc này cũng giúp cho người lập trình kiểm soát mã nguồn một cách mạch lạc. Đồng thời có thể tùy biến ở mọi trang. Mà không cần phải khởi tạo hay viết lại mã lệnh như HTML thuần

a) Hàm tự định nghĩa

Cú pháp:

```
function function_name()
```

```
{
```

```
//Lệnh thực thi
```

```
}
```

Tên hàm có thể là một tổ hợp bất kỳ những chữ cái, con số và dấu gạch dưới, nhưng phải bắt đầu từ chữ cái và dấu gạch dưới.

b) Hàm tự định nghĩa với các tham số

Cú pháp:

```
function function_name($gt1,$gt2)
```

```
{
```

```
//Lệnh thực thi
```

```
}
```

c) Hàm tự định nghĩa với giá trị trả về

Cú pháp:

```
function function_name(Có hoặc không có đối số)
```

```
{
```

```
// Lệnh thực thi
```

```
return giatri;
```

```
}.
```

d) Gọi lại hàm

PHP cung cấp nhiều hàm cho phép triệu gọi lại file. Như hàm `include("URL đến file")`, `require("URL Đến file")`.

Ngoài hai cú pháp trên còn có `include_once()`, `require_once()`. Hai hàm này cũng có trách nhiệm gọi lại hàm. Những chúng sẽ chỉ gọi lại duy nhất 1 lần mà thôi.

Chương 2: TỔNG QUAN VỀ JQUERY

2.1. Giới thiệu về jQuery

jQuery là một trong những thư viện JavaScript mở đang được ứng dụng rất rộng rãi và phổ biến, dựa vào nó các nhà phát triển có thể tạo ra các hiệu ứng có thể tương tác trực tiếp với người dùng một cách nhanh chóng và dễ dàng hơn nhiều việc sử dụng thuần JavaScript.

jQuery có cấu trúc rất mạch lạc và theo hệ thống. Cách viết code của jQuery được vay mượn từ các nguồn mà các nhà lập trình web đa phần đã biết như HTML và CSS

2.2. jQuery có thể làm gì

2.2.1. Hướng tới các thành phần trong tài liệu HTML.

Nếu không sử dụng thư viện JavaScript này, bạn phải viết rất nhiều dòng code mới có thể đạt được mục tiêu là di chuyển trong cấu trúc cây (hay còn gọi là DOM = Document Object Model) của một tài liệu HTML và chọn ra các thành phần liên quan. JQuery cho phép bạn chọn bất cứ thành phần nào của tài liệu để sử dụng một cách dễ dàng như sử dụng CSS.

2.2.2. Thay đổi giao diện của một trang web

CSS là công cụ rất mạnh để định dạng một trang web nhưng nó có một nhược điểm là không phải tất cả các trình duyệt đều hiển thị giống nhau. Cho nên jQuery ra đời để lấp chỗ trống này, vì vậy các bạn có thể sử dụng nó để giúp trang web có thể hiển thị tốt trên hầu hết các trình duyệt. Hơn nữa jQuery cũng có thể thay đổi class hoặc những định dạng CSS đã được áp dụng lên bất cứ thành phần nào của tài liệu HTML ngay cả khi trang web đó đã được trình duyệt load thành công. Thay đổi nội dung của tài liệu. JQuery không phải chỉ có thể thay đổi bề ngoài của trang web, nó cũng có thể thay đổi nội dung của chính tài liệu đó chỉ với vài dòng code. Nó có thể thêm hoặc bớt nội dung trên trang, hình ảnh có thể được thêm vào hoặc đổi sang hình khác, danh sách có thể được sắp xếp lại hoặc thậm chí cả cấu trúc HTML của một trang web cũng có thể được viết lại và mở rộng. Tất cả những điều này bạn hoàn toàn có thể làm được nhờ sự giúp đỡ của API (Application Programming Interface = Giao diện lập trình ứng dụng).

2.2.3. Tương tác với người dùng

Cho dù công cụ bạn dùng có mạnh mẽ đến mấy, nhưng nếu bạn không có quyền quyết định khi nào nó được sử dụng thì công cụ đó cũng coi như bỏ. Với thư viện javascript như jQuery, nó cho bạn nhiều cách để tương tác với người dùng ví dụ như khi người dùng nhấp chuột vào đường link thì sẽ có gì xảy ra. Nhưng cái hay của nó là không làm cho code HTML của bạn rối tung lên chính là nhờ các Event Handlers. Hơn nữa Event Handler API sẽ bảo đảm rằng trang web của bạn tương thích hầu hết với các trình duyệt, điều này đã và đang làm đau đầu rất nhiều các web designer.

2.2.4. Tạo hiệu ứng động cho những thay đổi của tài liệu

Để tương tác tốt với người dùng, các web designer phải cho người dùng thấy được hiệu ứng gì sẽ xảy ra khi họ làm một tác vụ nào đó. JQuery cho phép bạn sử dụng rất nhiều hiệu ứng động như mờ dần, chạy dọc chạy ngang v.v.. và nếu vẫn chưa đủ, nó còn cho phép bạn tự tạo ra các hiệu ứng của riêng mình.

2.2.5. Lấy thông tin từ server mà không cần tải lại trang web

Đây chính là công nghệ ngày càng trở nên phổ biến Asynchronous JavaScript And XML (AJAX), nó giúp người thiết kế web tạo ra những trang web tương tác cực tốt và nhiều tính năng. Thư viện jquery loại bỏ sự phức tạp của trình duyệt trong quá trình này và cho phép người phát triển web có thể tập trung vào các tính năng đầu cuối. Đơn giản hoá các tác vụ javascript. Ngoài những tính năng như đã nêu ở trên, jquery còn cho phép bạn viết code javascript đơn giản hơn nhiều so với cách truyền thống như là các vòng lặp và điều khiển mảng..

2.3. Tại sao jquery làm việc tốt

Người dùng ngày càng quan tâm hơn đến Dynamic HTML, đó cũng là nền móng cho sự ra đời của những javascript Frameworks. Có frameworks thì chỉ tập trung vào một vài tính năng vừa nêu ở trên, có cái thì ráng bao gồm tất cả những hiệu ứng, tập tính và nhồi nhét vào một package. Để đảm bảo là một thư viện “nhanh gọn nhẹ” nhưng vẫn “ngon bổ rẻ” với các tính năng đã nêu ở trên, jquery sử dụng những chiến lược sau:

2.3.1. Tận dụng kiến thức về CSS.

Các jquery Selector hoạt động y chang như CSS Selector với cùng cấu trúc và cú pháp. Chính vì thế thư viện jquery là cửa ngõ cho các web designer muốn thêm nhiều tính năng hơn nữa cho trang web của mình. Bởi vì điều kiện tiên quyết để trở thành một web designer chuyên nghiệp là khả năng sử dụng CSS thuần thục. Với kiến thức có sẵn về CSS, bạn sẽ có sự khởi đầu thuận lợi với jquery.

2.3.2. Hỗ trợ Plugin

Để tránh bị rơi vào trạng thái quá tải tính năng, jquery cho phép người dùng tạo và sử dụng Plugin nếu cần. Cách tạo một plugin mới cũng khá đơn giản và được hướng dẫn cụ thể, chính vì thế cộng đồng sử dụng jquery đã tạo ra một loạt những plugin đầy tính sáng tạo và hữu dụng.

2.3.3. Loại bỏ sự khác biệt giữa các trình duyệt

Một thực tế tồn tại là mỗi một hệ thống trình duyệt lại có một kiểu riêng để đọc trang web. Dẫn đến một điều làm đau đầu các web designer là làm thế nào để cho trang web có thể hiển thị tốt trên mọi trình duyệt. Cho nên đôi khi người ta phải làm hẳn một phần code phức tạp để đảm bảo rằng trang web của họ được hiển thị gần như tương đồng ở các trình duyệt phổ biến. JQuery giúp bạn thêm một lớp bảo vệ cho sự khác biệt của trình duyệt và giúp quá trình này diễn ra dễ dàng hơn rất nhiều.

2.3.4. Luôn làm việc với Set

Ví dụ khi chúng ta yêu cầu jQuery tìm tất cả các thành phần có class là delete và ẩn chúng đi. Chúng ta không cần phải loop qua từng thành phần được trả về. Thay vào đó, những phương pháp như là hide() được thiết kế ra để làm việc với set thay vì từng thành phần đơn lẻ. Kỹ thuật này được gọi là vòng lặp ẩn, điều đó có nghĩa là chúng ta không phải tự viết code để loop nữa mà nó vẫn được thực thi, chính vì thế code của chúng ta sẽ ngắn hơn rất nhiều.

2.3.5. Cho phép nhiều tác vụ diễn ra trên cùng một dòng

Để tránh phải sử dụng những biến tạm hoặc các tác vụ lặp tốn thời gian, jQuery cho phép bạn sử dụng kiểu lập trình được gọi là Chaining cho hầu hết các method của nó. Điều đó có nghĩa là kết quả của các tác vụ được tiến hành trên một thành phần chính là thành phần đó, nó sẵn sàng cho tác vụ tiếp theo được áp dụng lên nó. Những chiến lược được nêu ở trên giúp kích thước của jQuery rất nhỏ bé chỉ khoảng trên dưới 20Kb dạng nén. Nhưng vẫn đảm bảo cung cấp cho chúng ta những kỹ thuật để giúp code trên trang nhỏ gọn và mạch lạc.

jQuery sở dĩ trở nên phổ biến là do cách sử dụng đơn giản và bên cạnh đó còn có một cộng đồng sử dụng mạnh mẽ vẫn ngày ngày phát triển thêm Plugin và hoàn thiện những tính năng trọng tâm của jQuery. Cho dù thực tế là vậy, nhưng jQuery lại là thư viện javascript hoàn toàn miễn phí cho mọi người sử dụng. Ta thể sử dụng nó trong hầu hết các trường hợp kể cả thương mại lẫn cá nhân.

Chương 3 : SỬ DỤNG JQUERY

3.1. Các cú pháp jQuery cơ bản

3.1.1. Cài đặt thư viện jQuery

- Hiện tại đang là phiên bản jQuery1.9.1, ta có thể download từ trang chủ www.jquery.com có 2 phiên bản:

+ Phiên bản product(bản nén) dùng cho sản phẩm có dung lượng thấp khoảng 90,45KB

+ Phiên bản dùng cho nhà phát triển(bản chưa nén) có dung lượng khoảng 262,09 KB

- Cài đặt jQuery:

```
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
```

```
<script src="http://code.jquery.com/jquery-migrate-1.1.1.min.js"></script>
```

3.1.2. Gọi hàm jQuery

- Cú pháp: $$(selector).action()$

+ "\$": Ký hiệu của jQuery, có thể thay thế bằng từ khóa "jQuery"

+ selector: là tên của các đối tượng cần truy cập hoặc con đường truy xuất đến các đối tượng cần truy cập

+ action: thực hiện các hành động trên các selector

3.1.3. jQuery Selector

jQuery selector giúp chúng ta dễ dàng truy vấn đến các phần tử DOM (Document Object Model – Mô hình đối tượng tài liệu) một cách nhanh nhất, code đơn giản và ngắn gọn nhất.

jQuery Selector có thể là : Thẻ html, id, class hoặc một đối tượng

- ID Selector: Bắt đầu bằng dấu “#” Và sau đó là Id element, cú pháp : `$("#id_name")`
- Class: Bắt đầu là dấu “.” và tên class, Cú pháp: `$(".class_name")`
- Một đối tượng:

VD: `var obj = $('#name_id')`

`$(obj).html(' ');`

- Element Selector: Tên của các element như : div, ul, li, table, tr, td , span dựa trên các thẻ html để hình thành con đường truy xuất

Cú pháp : VD: `$(“tag_name”)`

```
$(document).ready(function(){
```

```
  $("button").click(function(){
```

```
    $("p").hide();
```

```
  });
```

```
});
```

VD:

```
<div id = “firstID”>
```

```
  <ul>
```

```
    <li> <a class=“red”>red Link</a></li>
```

```
  </ul>
```

```
</div>
```

Để gọi đến thẻ “a” ta có thể gọi : `$(‘div#first ul li a.red’)`

* jQuery Selector sử dụng cơ chế giống như css selector (cho phép lựa chọn các đối tượng DOM dựa trên nguyên tắc truy vấn CSS)

Một số ví dụ về jQuery

CÚ PHÁP	MÔ TẢ
<code>\$(“*”)</code>	Chọn tất cả các thành phần
<code>\$(this)</code>	Chọn thành phần HTML hiện tại

<code>\$(".p.intro")</code>	Chọn tất cả các thẻ <p> có class là “intro”
<code>\$(".p:first")</code>	Chọn thẻ <p> đầu tiên
<code>\$(".ul li:first")</code>	Chọn thẻ đầu tiên của thẻ đầu tiên
<code>\$(".ul li:first-child")</code>	Chọn thẻ đầu tiên của tất cả các thẻ
<code>\$(".[href]")</code>	Chọn tất cả các thành phần có thuộc tính là href
<code>\$(".a[target='_blank']")</code>	Chọn tất cả các thẻ <a> với thuộc tính target có giá trị: "_blank"
<code>\$(".a[target!='_blank']")</code>	Chọn tất cả các thẻ <a> với thuộc tính target có giá trị khác: "_blank"
<code>\$(".:button")</code>	Chọn tất cả các thẻ <button> và thẻ <input> có thuộc tính type="button"
<code>\$(".tr:even")</code>	Chọn tất cả các thẻ <tr> chẵn
<code>\$(".tr:odd")</code>	Chọn tất cả các thẻ <tr> lẻ

Ngoài ra còn có rất nhiều các selector khác

3.1.4. jQuery Attribute

Những phương pháp jquery nhận và thiết lập các thuộc tính của các yếu tố DOM. Các thuộc tính người lập trình jquery hay dùng.

3.1.4.1. Class

a) `addClass(class)`

- Kiểu trả về: jquery
- Thêm các class đã xác định vào mỗi tập phần tử phù hợp. Nếu có thêm nhiều class thì các class được cách nhau bởi khoảng trắng.
- Ví dụ: Thêm class “styleText” vào các thẻ p cuối cùng

```

<!DOCTYPE html>
<html>
<head>
<style>
p.styleText { text-decoration: underline}
</style>
<script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<body>
<p>Hello</p>
<p>and</p>
<p>then</p>
<p>Goodbye</p>
<script>$(".p:last").addClass("styleText");</script>
</body>
</html>

```

Kết quả

Hello
and
then
Goodbye

b) removeClass(class)

- Kiểu trả về: jQuery
- Loại bỏ tất cả hoặc các class đã xác định khỏi tập phần tử phù hợp.
- Ví dụ: Loại bỏ class “styleText” khỏi thẻ p cuối cùng.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.styleText { text-decoration: underline}
</style>
<script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<body>
<p class="styleText">Hello</p>
<p class="styleText">and</p>
<p class="styleText">then</p>
<p class="styleText">Goodbye</p>
<script>$("p:last").addClass("styleText");</script>
</body>
</html>
```

Kết quả

Hello
and
then
Goodbye

c) toggleClass(class)

- Kiểu trả về: jQuery
- Thêm class nếu class chưa tồn tại hoặc loại bỏ nếu class đã tồn tại.
- Ví dụ: Thêm class “styleText” vào thẻ p nếu class “styleText” chưa tồn tại trong thẻ p hoặc loại bỏ class “styleText” khỏi thẻ p nếu nó tồn tại.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.styleText { text-decoration: underline}
</style>
<script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<body>
<p class="styleText">Hello</p>
<p class="styleText">and</p>
```

```
<p class="styleText">then</p>
<p>Goodbye</p>
<script>$("p").toggleClass("styleText");</script>
</body>
</html>
```

Kết quả

Hello
and
then
Goodbye

3.1.4.2. HTML, Text

a) html()

- Kiểu trả về: String
- Lấy nội dung html (innerHTML) của phần tử.
- Ví dụ: Mỗi khi click vào thẻ p lấy nội dung html của thẻ p đó và thông báo nội dung lấy được.

```
$("p").click(function() {alert($(this).html());});
```

b) html(val)

- Kiểu trả về: jQuery
- Thiết lập nội dung html (innerHTML) cho phần tử.
- Ví dụ: Thiết lập nội dung html cho thẻ div.

```
$("div").html("<b>Chào các bạn!<i> Chúc buổi học hôm nay thú vị.</i></b>");
```

c) text()

- Kiểu trả về: String
- Lấy nội dung text (innerText) của phần tử.
- Ví dụ: Mỗi khi click vào thẻ p lấy nội dung text của thẻ p đó và thông báo nội dung lấy được.

```
$("p").click(function(){alert($(this).text());});
```

d) text(val)

- Kiểu trả về: jQuery
- Thiết lập nội dung text (innerText) cho phần tử.
- Ví dụ: Thiết lập nội dung text cho thẻ div.

```
$("div").text("Chào các bạn! Chúc buổi học hôm nay thú vị");
```

3.1.5. jQuery Events

- JavaScript có một số cách được lập sẵn để phản ứng với những tương tác của người dùng và những sự kiện khác. Để làm cho trang web năng động và tương tác tốt, chúng ta cần phải tận dụng chức năng này, để vào những thời điểm phù hợp, chúng ta có thể sử dụng những kỹ thuật jQuery. Cũng có thể làm với JavaScript, nhưng jQuery nâng cao và mở rộng những cơ chế quản lý sự kiện cơ bản để giúp nó có cú pháp đẹp hơn, tiết kiệm thời gian hơn và tất nhiên cũng mạnh mẽ hơn.

- jQuery cung cấp cho chúng ta khá nhiều các sự kiện đủ để thao tác với các thành phần trên website mà chúng ta mong muốn. Bên cạnh đó, jQuery còn loại bỏ hoàn toàn các event code ra khỏi mã HTML như cách viết javascript thông thường

Trong javascript, để xuất hiện thông báo với người dùng khi click vào liên kết ta phải viết như sau:

```
<a class="click" href="#" onclick="alert('Welcome to Ewebvn.Com')">Click me</a>
```

Ta thấy rằng chúng ta phải viết mã để gọi sự kiện onclick ngay trong liên kết <a>. Nhưng với jQuery thì chúng ta có thể sử dụng bộ selector để thao tác với các sự kiện như sau:

```
$("#a.click").click(function() { alert("Welcome to Ewebvn.Com"); });
```

3.1.5.1. Cách viết các phương thức xử lý sự kiện và các sự kiện phổ biến

- Javascript cung cấp Bộ quản lý sự kiện window.onload cho phép chúng ta thực thi một hàm nào đó. Trong khi đó jQuery cung cấp cho chúng ta bộ quản lý sự kiện \$(document).ready() nhưng với những ưu điểm vượt trội hơn sự kiện window.onload window.onload sẽ được thực thi sau khi trình duyệt tải xong toàn bộ tài liệu cần thiết bao gồm stylesheet, hình ảnh...

Trong khi đó \$(document).ready() sẽ được thực thi ngay khi các phần tử DOM được sẵn sàng mà không cần đợi các tài liệu như stylesheet hay hình ảnh tải xong. Do đó quá trình xử lý vào thao tác sẽ diễn ra nhanh hơn.

- Cú pháp viết hàm sự kiện trong jQuery

```
$(document).ready(function() {  
//Code và các chương trình thực thi  
})
```

Hoặc

```
$(function() {  
//Code và các chương trình thực thi  
})
```

Hoặc

```
$(function() {  
//Code và các chương trình thực thi  
})
```

- Một số sự kiện thường sử dụng trong jQuery

Mouse Events	Keyboard Events	Form Events	Document/Window Events
<i>click</i>	<i>keypress</i>	<i>submit</i>	<i>load</i>
<i>dblclick</i>	<i>keydown</i>	<i>change</i>	<i>resize</i>
<i>mouseenter</i>	<i>keyup</i>	<i>focus</i>	<i>scroll</i>

<i>mouseleave</i>		<i>blur</i>	<i>unload</i>
-------------------	--	-------------	---------------

a) click()

- Phương thức click() gắn với hàm xử lý sự kiện cho một thành phần của HTML

- Hàm sẽ được thực hiện khi người dùng nhấp đúp chuột vào một thành phần của HTML

VD: Khi nhấp chuột vào thẻ p, thẻ này sẽ bị ẩn đi

```
$("#p").click(function(){
    $(this).hide();
});
```

b) dblclick()

- Phương thức dblclick() gắn với hàm xử lý sự kiện cho một thành phần của HTML

- Hàm sẽ được thực hiện khi người dùng nhấp chuột vào một thành phần của HTML

VD: Khi nhấp đúp chuột vào thẻ p, thẻ này sẽ bị ẩn đi

```
$("#p").dblclick(function(){
    $(this).hide();
});
```

c) mouseenter

- Phương thức dblclick() gắn với hàm xử lý sự kiện cho một thành phần của HTML

- Hàm sẽ được thực hiện khi người dùng trỏ chuột vào một thành phần của HTML

```
$("#p1").mouseenter(function(){
    alert("You entered p1!");
});
```

d) mouseleave()

- Hàm sẽ được thực hiện khi người dùng dời con trỏ chuột khỏi một thành phần của HTML

```
$("#p").mouseleave(function(){
    alert("Bye! You now leave p!");
});
```

e) mousedown():

- Hàm sẽ được thực hiện khi nhấn trái chuột vào một thành phần của HTML

```
$("#p1").mousedown(function(){
    alert("Mouse down over p1!");
});
```

f) mouseup()

- Hàm sẽ gọi sự kiện này khi nhả chuột trái

```
$("#p1").mouseup(function(){
    alert("Mouse up over p1!");
});
```

g) **hover()**

- Hàm được gọi khi di chuyển con trỏ chuột đến một thành phần của HTML

```
$("#p1").hover(function(){
    alert("You entered p1!");
},
function(){
    alert("Bye! You now leave p1!");
});
```

h) **focus()**

- Sự kiện xảy ra khi ta đang focus vào 1 element (con nháy trong textbox, viền trên button, radio, checkbox...).

```
$("#input").focus(function(){
    $(this).css("background-color", "#cccccc");
});
```

i) **blur()**

- Ngược lại với focus

```
$("#input").blur(function(){
    $(this).css("background-color", "#ffffff");
});
```

Ngoài ra còn có rất nhiều hàm khác, trong tùy từng các trường hợp cụ thể ta có thể sử dụng với các mục đích khác nhau. Ta có thể vào trang chủ của jQuery để tra cứu thêm.

3.2. JQUERY EFFECTS

Các hiệu ứng động của jQuery sẽ làm cho trang web thêm phần sinh động. JQuery cho phép ẩn hiện, trượt lên trượt xuống các thành phần của trang web. Bạn cũng có thể cho nó xảy ra cùng một lúc hoặc theo thứ tự định trước.

3.2.1. Hide/Show

a) **hide() and show()**

- Phương thức giúp ẩn, hiện các thành phần của HTML.

VD:

```
$("#hide").click(function(){
    $("#p").hide();
});
$("#show").click(function(){
    $("#p").show();
});
```

- Cú pháp

```
$(selector).hide(speed,callback);
```

```
$(selector).show(speed,callback);
```

Ta cũng có thể thêm vào một số tham biến để tùy chọn tốc độ việc ẩn đi hay hiển thị ra các thành phần, và giá trị của chúng có thể là “slow”, “fast” hoặc milliseconds

VD:

```
$("#hide").click(function(){
```

```
  $("p").hide(slow, alert);
```

```
});
```

```
$("#hide").click(function(){
```

```
  $("p").hide(1000);
```

```
});
```

b) toggle()

- jQuery cung cấp thêm một phương thức giúp ta có thể ẩn đi và hiện lên các selector, ta có thể lựa chọn giữa việc sử dụng hàm này hoặc sử dụng phương thức hide() và show() đã giới thiệu ở trên. Nó giúp ta ẩn đi các selector đang hiện và hiện lại các selector đã bị ẩn đi

- Cú pháp: về cơ bản phương thức này cũng có cú pháp tương tự như hide() và show()

```
$(selector).toggle(speed,callback);
```

Ta cũng có thể thêm vào một số tham biến để tùy chọn tốc độ việc ẩn đi hay hiển thị ra các thành phần, và giá trị của chúng có thể là “slow”, “fast” hoặc milliseconds. Đồng thời cũng có thể thêm vào một hành động nào đó để thực hiện sau thực hiện xong toggle()

VD: Sau khi thực hiện việc ẩn/hiện ra các thẻ p, sẽ xuất hiện một thông báo “OK”

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
```

```
</script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
  $("button").click(function(){
```

```
    $("p").toggle("slow", alert('OK'));
```

```
  });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<button>Toggle</button>
```

```
<p>This is a paragraph with little content.</p>
```

```
<p>This is another small paragraph.</p>
```

```
</body>
```

```
</html>
```

3.2.2. Fading

jQuery cung cấp các phương thức hỗ trợ cho việc ẩn hiện các selector, nó cũng hỗ trợ việc hiện thị hoặc ẩn đi thông tin kiểu làm cho thông tin mờ dần rồi biến mất, hoặc ngược lại khi hiện thị ra.

jQuery cung cấp một số phương thức giúp thực hiện các hành động dạng này như sau:

`fadeIn()`, `fadeOut()`, `fadeToggle()`, `fadeTo()`

a) `fadeIn()`

- Phương thức này giúp hiển thị ra các selector bị ẩn đi
- Cú pháp

```
$(selector).fadeIn(speed,callback);
```

Ta cũng có thể thêm vào một số tham biến để tùy chọn tốc độ việc ẩn đi hay hiện thị ra các thành phần, và giá trị của chúng có thể là “slow”, “fast” hoặc milliseconds. Đồng thời cũng có thể thêm vào một hành động nào đó để thực hiện sau khi đã hiện thị ra selector.

VD:

```
<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeIn();
    $("#div2").fadeIn("slow");
    $("#div3").fadeIn(3000);
  });
});
</script>
</head>
<body>
<p>Demonstrate fadeIn() with different parameters.</p>
<button>Click to fade in boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>
```

```

<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>
</body>
</html>

```

b) fadeOut()

- Phương thức này giúp ta ẩn đi các selector
- Cú pháp:

```
$(selector).fadeOut(speed,callback);
```

Ta cũng có thể thêm vào một số tham biến để tùy chọn tốc độ ẩn đi các selector, giá trị của các tham biến này có thể là “slow”, “fast”, hoặc milliseconds.

Đồng thời ta cũng có thể thêm vào các hàm thực hiện hành động sau khi đã ẩn đi các selector.

VD:

```

<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeOut();
    $("#div2").fadeOut("slow");
    $("#div3").fadeOut(3000);
  });
});
</script>
</head>
<body>
<p>Demonstrate fadeOut() with different parameters.</p>
<button>Click to fade out boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>

```

```

<div id="div2" style="width:80px;height:80px;background-color:green;"></div>
<br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>

```

c) fadeToggle()

- Đây là phương thức trung gian giữa fadeIn() và fadeOut(), thực hiện việc ẩn đi/hiển thị ra các selector

- Cú pháp:

```
$(selector).fadeToggle(speed,callback);
```

Ta cũng có thể thêm vào một số tham biến để tùy chọn tốc độ ẩn đi/hiển thị các selector, giá trị của các tham biến này có thể là "slow", "fast", hoặc milliseconds.

Đồng thời ta cũng có thể thêm vào các hàm thực hiện hành động sau khi đã ẩn đi/hiển thị các selector.

VD:

```

<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeToggle();
    $("#div2").fadeToggle("slow");
    $("#div3").fadeToggle(3000);
  });
});
</script>
</head>
<body>
<p>Demonstrate fadeToggle() with different speed parameters.</p>
<button>Click to fade in/out boxes</button>
<br><br>

```

```

<div      id="div1"      style="width:80px;height:80px;background-
color:red;"></div>
<br>
<div      id="div2"      style="width:80px;height:80px;background-
color:green;"></div>
<br>
<div      id="div3"      style="width:80px;height:80px;background-
color:blue;"></div>
</body>
</html>

```

d) fadeTo()

- Đây là phương thức giúp ta có thể làm mờ đi các đối tượng selector đến một mức độ nhất định theo tùy chọn.

- Cú pháp:

```
$(selector).fadeTo(speed,opacity,callback);
```

Ta cũng có thể thêm vào một số tham biến tùy chọn tốc độ làm mờ đi các selector, giá trị của các tham biến này có thể là "slow", "fast", hoặc milliseconds, và làm mờ đến mức nào(chọn từ 0 đến 1).

Đồng thời ta cũng có thể thêm vào các hàm thực hiện hành động sau khi đã ẩn đi các selector.

VD:

```

<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeTo("slow",0.15);
    $("#div2").fadeTo("slow",0.4);
    $("#div3").fadeTo("slow",0.7);
  });
});
</script>
</head>

```

```

<body>
<p>Demonstrate fadeTo() with different parameters.</p>
<button>Click to fade boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div>
<br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>

```

3.2.3. Animate

- jQuery cung cấp phương thức giúp ta hỗ trợ việc tạo hiệu ứng cho các element như tăng kích cỡ, dịch chuyển vị trí bằng cách định dạng thêm các thuộc tính CSS

- Cú pháp:

```
$(selector).animate({params},speed,callback);
```

- Với phương thức này ta có thể thêm một hoặc nhiều thuộc tính CSS, giúp tạo thêm các hiệu ứng như ý muốn

VD:

```

<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      left:'250px',
      opacity:'0.5',

```

```

        height:'150px',
        width:'150px'
    });
});
});
</script>
</head>
<body>
<button>Start Animation</button>

```

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

```

<div
style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>
</body>
</html>

```

- Hoặc ta có thể sử dụng thêm các giá trị tương đối

```

$("button").click(function(){
    $("div").animate({
        left:'250px',
        height:'+=150px',
        width:'+=150px'
    });
});

```

- Ta cũng có thể sử dụng các thuộc tính để thực hiện các hiệu ứng : "show", "hide", or "toggle":

```

$("button").click(function(){
    $("div").animate({

```

```

    height:'toggle'
  });
});
- Ta cũng có thể thực hiện việc gọi ra nhiều hiệu ứng :
$("button").click(function(){
  var div=$("#div");
  div.animate({height:'300px',opacity:'0.4'},"slow");
  div.animate({width:'300px',opacity:'0.8'},"slow");
  div.animate({height:'100px',opacity:'0.4'},"slow");
  div.animate({width:'100px',opacity:'0.8'},"slow");
});

```

3.2.3. Stop

- Khi phương thức này được gọi, các hiệu ứng đang xảy ra sẽ bị dừng lại. Giả sử nếu một element đang bị ẩn đi bởi phương thức slideUp(), thì khi phương thức stop() được gọi, element sẽ vẫn hiển thị ra nhưng chiều cao sẽ giảm đi.

- Cú pháp:

```
$(selector).stop(stopAll,goToEnd);
```

VD:

```

<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown(5000);
  });
});

```

```
    $("#stop").click(function(){
        $("#panel").stop();
    });
});
</script>
<style type="text/css">
#panel,#flip
{
padding:5px;
text-align:center;
background-color:#e5eccc;
border:solid 1px #c3c3c3;
}
#panel
{
padding:50px;
display:none;
}
</style>
</head>
<body>
<button id="stop">Stop sliding</button>
<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>
</body>
</html>
```

Để tạo các hiệu ứng effect, jQuery đã cũng cấp thêm rất nhiều phương thức, ta có thể tìm hiểu thêm các phương thức này ở trang chủ của jQuery. Ngoài ra ta có thể tham khảo thêm về jQuery UI đây là bộ thư viện JavaScripts viết dựa trên nền jQuery.

3.3. jQuery HTML

3.3.1. jQueryGet

a) Lấy ra nội dung:

Để lấy ra nội dung của các thành phần, jQuery cung cấp 3 phương thức cơ bản:

* `text()`: trả về nội dung của đoạn văn bản được lựa chọn

* `html()`: trả về nội dung của thành phần được lựa chọn (bao gồm cả các ký tự HTML)

VD:

```
<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        alert("Text: " + $("#test").text());
    });
    $("#btn2").click(function(){
        alert("HTML: " + $("#test").html());
    });
});
</script>
</head>
<body>
```

```
<p id="test">This is some <b>bold</b> text in a paragraph.</p>
```

```
<button id="btn1">Show Text</button>
```

```
<button id="btn2">Show HTML</button>
```

```
</body>
```

```
</html>
```

* val(): Trả ra giá trị của thành phần được chọn.

VD:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
```

```
</script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
  $("button").click(function(){
```

```
    alert("Value: " + $("#test").val());
```

```
  });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>
```

```
<button>Show Value</button>
```

```
</body>
```

```
</html>
```

b) Lấy ra thuộc tính:

* attr(): Phương thức giúp lấy ra nội dung thuộc tính của thành phần được chọn:

VD:

```
<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert($("#test").val(type));
    });
});
</script>
/head>
<body>
<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>
<button>Show</button>
</body>
</html>
```

3.3.2. jQuery Set

jQuery cung cấp phương thức thêm vào một nội dung cho một đoạn văn bản (có thể đi kèm các ký tự cho HTML), đặt lại giá trị cho một thành phần nào đó trong HTML. Ngoài ra ta cũng có thể sử dụng thêm các hàm để tùy biến với mục đích sử dụng.

```
<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
```

```

    $("#test1").text("Hello world!");
  });
  $("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
  });
  $("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
  });
});
</script>
</head>
<body>
<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>
<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>
<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>
</body>
</html>

```

3.3.3. jQuery giúp người dùng thao tác với các thành phần của HTML

jQuery cung cấp các phương thức giúp ta có thể dễ dàng thêm hoặc bớt một nội dung, hoặc một thành phần trong các thẻ HTML

Một số hàm được cung cấp bởi thư viện jQuery như sau:

- * `append()`: Chèn thêm một nội dung ở cuối của thành phần được chọn:

```
$("#p").append("Some appended text.");
```

- * `prepend()`: Chèn thêm nội dung ở phía đầu phần tử được chọn

```
$("#p").prepend("Some appended text.");
```

- * `after()`: Chèn thêm một nội dung ở phía trước của thành phần được chọn:

```
$("#p").after("Some appended text.");
```

- * `before()`: Chèn thêm nội dung ở phía sau phần tử được chọn

```
$("#p").before("Some appended text.");
```

- * `remove()`: Hàm giúp xóa đi các thẻ, sử dụng tùy theo mục đích người dùng

```
$("#div1").remove();
```

- * `empty()`: giúp loại bỏ các thành phần con của phần tử được lựa chọn

```
$("#div1").empty();
```

3.3.4. jQuery giúp người dùng có thể thao tác, tác động tới CSS

jQuery cung cấp các phương thức giúp người dùng có thể thêm hoặc bớt các class CSS cho thẻ HTML, hoặc thêm hay bớt một thuộc tính CSS cho một Class. Có một số phương thức như sau:

*addClass(): Thêm một class CSS .

```
$("#button").click(function(){  
    $("h1,h2,p").addClass("blue");  
    $("div").addClass("important");  
});
```

*removeClass(): Loại bỏ class khỏi đối tượng được chọn.

```
$("#button").click(function(){  
    $("h1,h2,p").removeClass("blue");  
});
```

* toggleClass() : Phương thức giúp chuyển đổi giữa 2 thao tác add và remove

```
$("#button").click(function(){  
    $("h1,h2,p").toggleClass("blue");  
});
```

* css() : Trả về, hoặc cài đặt thêm thuộc tính CSS()

```
$("#p").css("background-color"); <trả về thuộc tính của css>  
$("#p").css("propertyname", "value"); <cài đặt thuộc tính cho css>
```

3.4. jQueryAjax

3.4.1. Giới thiệu

- AJAX là kỹ thuật phát triển web có tính tương tác cao dựa trên cách kết hợp các công nghệ phát triển web với nhau: VD: HTML (hoặc XHTML) với Css, mô hình DOM thông qua JavaScript..., nó giúp trao đổi dữ liệu với server và cập nhật lại từng phần của trang web mà không phải tải lại toàn bộ trang web.

- jQuery cung cấp một vài phương thức để thực hiện các chức năng cho Ajax. Với các phương thức này, có thể yêu cầu một đoạn văn bản, HTML, XML hoặc JSON từ máy Server cùng với việc sử dụng HTTP Get và HTTP Post và cũng có thể tải dữ liệu trực tiếp vào một thành phần của HTML đã được chọn trên trang Web.

- Việc viết code Ajax có thể đòi hỏi sự cẩn thận và khéo léo, vì với mỗi trình duyệt khác nhau thì lại có các cú pháp Ajax khác nhau. Điều này cũng đồng nghĩa với việc ta sẽ phải viết code nhiều hơn bình thường để kiểm tra với những trình duyệt khác nhau, tuy nhiên thì các nhà phát triển jQuery đã lưu ý trước những điều này, do đó ta có thể viết những hàm Ajax với một vài dòng code

3.4.2. Các phương thức jQuery Ajax

a) jQuery load()

- Đây là phương thức rất đơn giản tuy nhiên nó lại là phương thức Ajax có sức ảnh hưởng rất lớn. Nó sẽ tải dữ liệu từ server về và đặt vào các yếu tố được chọn.

- Cú pháp:

```
$(selector).load(URL,data,callback);
```

URL : chỉ định vị trí đường dẫn đối tượng được load

data : chỉ định chuỗi truy vấn, giá trị hoặc từ khóa được gửi cùng khi gửi request

callback: tên hàm được thực hiện sau khi thao tác load được hoàn thành

- Ví dụ : ta có một file text có tên : “test.txt”, ví dụ sẽ thực hiện việc load nội dung từ đoạn văn bản đó vào một thẻ div được lựa chọn

```
<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").load("demo_test.txt",function(responseTxt,statusTxt,xhr){
      if(statusTxt=="success")
        alert("External content loaded successfully!");
      if(statusTxt=="error")
        alert("Error: "+xhr.status+": "+xhr.statusText);
    });
  });
});
</head>
<body>
<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>
<button>Get External Content</button>
</body>
</html>
```

b) jQuery get() và jQuery post()

jQuery cung cấp phương thức yêu cầu lấy dữ liệu từ máy server cùng với HTTP GET hoặc POST

* **get()**:

- Yêu cầu dữ liệu từ máy server cùng với giao thức HTTP GET

- Cú pháp:

```
$.get(URL,callback);
```

URL: Tham số quy định cụ thể địa chỉ mà người dùng yêu cầu.

callback: hàm được gọi khi phương thức get được thực hiện xong

VD:

Ta có file “test.php” có chứa nội dung:

```
<?php echo "DHDL HaiPhong" ?>
```

Ví dụ như sau:

```
<!DOCTYPE html>
```

```

<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.get("test.php",function(data,status){
      alert("Data: " + data + "\nStatus: " + status);
    });
  });
});
</script>
</head>
<body>
<button>Send an HTTP GET request to a page and get the result
back</button>
</body>
</html>

```

***post():**

- Yêu cầu dữ liệu lấy từ server theo giao thức HTTP POST
- Cú pháp:

```
$.post(URL,data,callback);
```

URL : chỉ định vị trí đường dẫn đối tượng được load

data: dữ liệu được gửi cùng với yêu cầu.

callback: hàm được gọi sau khi thao tác load được thực hiện xong.

VD: Ta có file “test.php”, có nội dung:

```

<?php
  echo 'Name: ' . $_POST["name"]."\n";
  echo 'City: ' . $_POST["city"];
?>

```

Ta sẽ dùng jQuery để gửi dữ liệu bằng phương thức post

```

<!DOCTYPE html>
<html>
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.post("test.php",
    {
      name:"Donald Duck",
      city:"Duckburg"
    },
    function(data,status){
      alert("Data: " + data + "\nStatus: " + status);
    });
  });
});

```

```
    });  
  });/  
});  
</script>  
</head>  
<body>  
<button>Send an HTTP POST request to a page and get the result  
back</button>  
</body>  
</html>
```

Chương 4 : CHƯƠNG TRÌNH ỨNG DỤNG WEB SỬ DỤNG JQUERY VA PHP

4.1. Chương trình ứng dụng web Upload và hiển thị ảnh.

Chương trình ứng dụng web được xây dựng dựa trên nền ngôn ngữ PHP kết hợp với việc sử dụng jQuery. Bao gồm các chức năng :

- Upload ảnh từ các máy client, và thực hiện việc load ảnh ra thành danh sách
- Tạo menu.
- Hiển thị danh sách ảnh theo slide
- Thực hiện hiệu ứng lightBox khi click vào một ảnh trong danh sách

4.2. Các chức năng chính:

4.2.1. Upload và hiển thị ảnh theo danh sách:

Chức năng này được xây dựng bằng PHP.

- Lấy ảnh từ máy client sau đó lưu tên ảnh vào file txt

+ Form Upload ảnh

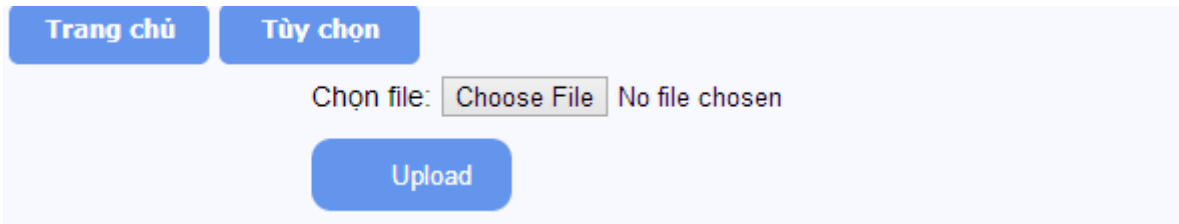
```
<form action="upload_file.php" method="post" enctype="multipart/form-data">
  <div class="content">
    <label for="file">Chọn file:</label>
    <input type="file" name="file" id="file"></br>
  </div>

  <div class="content">
    <input type="submit" name="submit" value="Upload" class="btnSubmit">
  </div>
</form>
```

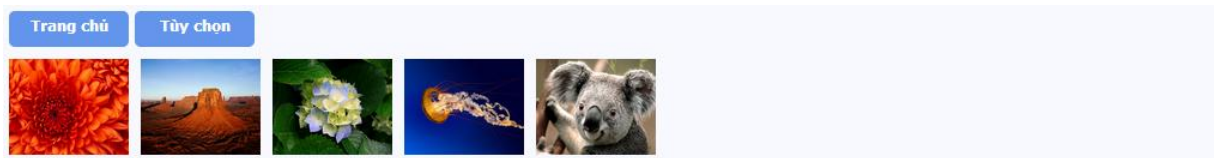
- Nếu upload file ảnh thành công thì sử dụng đoạn code dưới để lấy đường dẫn của ảnh lưu vào file 'directory.txt':

```
$myFile = "directory.txt";
$fh = fopen($myFile, 'a') or die("Can't open file");
$stringData = "upload/" . $_FILES["file"]["name"] . ' ';
fwrite($fh, $stringData);
fclose($fh);
```

- Dùng PHP để load ra danh sách ảnh đã upload, đọc file từ folder upload trên server.



Upload ảnh



Hiển thị ảnh thành danh sách

4.2.2. Tạo menu:

Menu được tạo dựa trên jQuery :

Đoạn script jQuery:

```

$( '.dropdown' ).mouseenter( function () {
    $( '.sublinks' ).stop( false, true ).hide();
    var submenu = $( this ).parent().next();
    submenu.css( {
        position: 'absolute',
        top: $( this ).offset().top + $( this ).height() + 'px',
        left: $( this ).offset().left + 'px',
        zIndex: 1000
    });
    submenu.stop().slideDown( 300 );
    submenu.mouseleave( function () {
        $( this ).slideUp( 300 );
    });
});

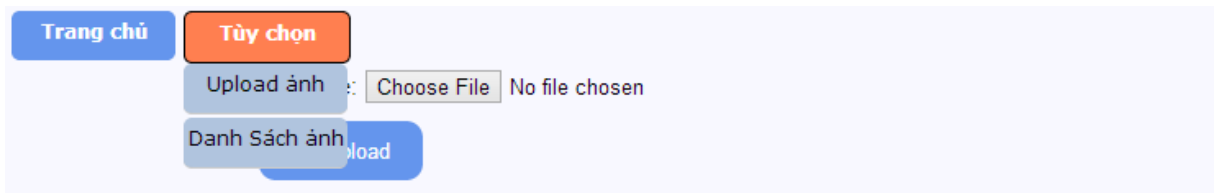
```

Đoạn mã HTML :

```

<div id="container">
    <ul>
        <li id="homepage"><a href="#http://localhost:8080/SlidePicture"
class="dropdown">Trang chủ</a></li>
    </ul>
    <ul>
        <li><a href="#" class="dropdown">Tùy chọn</a></li>
        <li class="sublinks">
            <a href="#" id="uploadImage">Upload ảnh</a>
            <a href="#" id="listImage">Danh Sách ảnh</a>
        </li>
    </ul>
</div>

```



Menu khi di chuột qua

4.2.3. Hiện thị ảnh theo slide:

- Chức năng này được thực hiện bằng plugin jQuery kết hợp với việc sử dụng PHP. Có thể download plugin này theo địa chỉ :

<https://code.google.com/p/rgbm/downloads/detail?name=jquery.cycle.all.min.js&can=2&q=>

- Gọi đoạn script dưới đây:

```
$(document).ready(function () {
    $("#slide").cycle({
        fx:'slideX', //Hiệu ứng - Xem thêm các hiệu ứng tại đây:
        http://jquery.malsup.com/cycle/browser.html
        speed:1000, //Tốc độ diễn ra hiệu ứng
        timeout:2000, //2 giây: Thời gian thay đổi giữa các hình 1/1000
        pager:'#slide-nav' //Tạo phân trang đặt trong thẻ div
    });
});
```

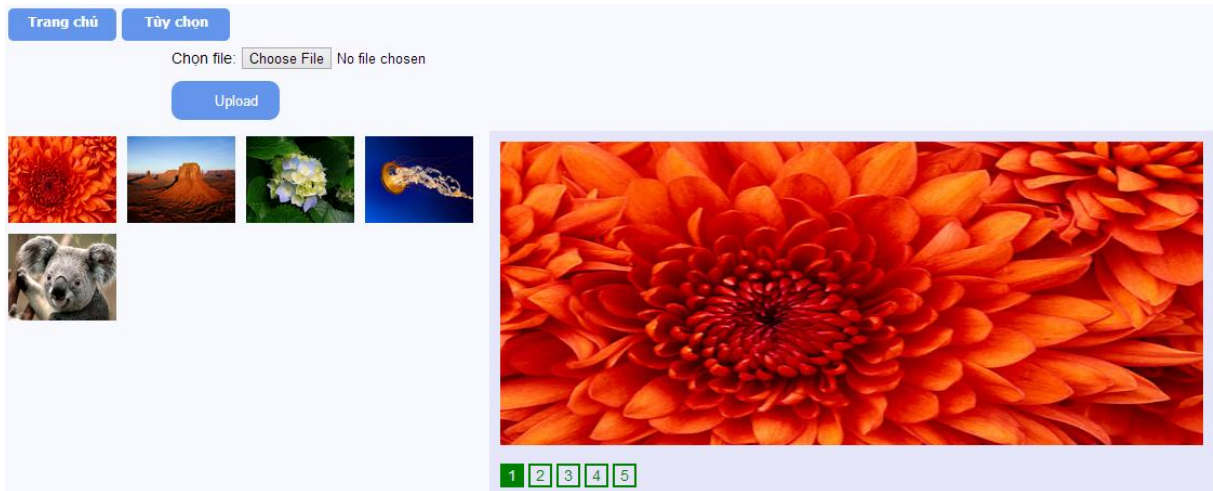
- Dùng PHP để load ra danh sách ảnh và hiện thị theo slide :

```
<div class=content1>
<div id="slide">
<?php
try {
// Đọc file lưu đường dẫn các ảnh và lưu ra thành mảng
$myfile = "directory.txt";
$fh = fopen($myfile, 'r');
$data = fread($fh, filesize($myfile));
fclose($fh);
if (!isset($data)) {
    echo 'Chưa có ảnh cho slide';
} else {
    $arrayDirectory = explode(';', $data);
    for ($j = 0; $j < count($arrayDirectory) - 1; $j++) {
        echo '';
    }
}
} catch (Exception $ex) {
    echo 'Lỗi';
}
?>
```

```

</div>
<div id="slide-nav"></div>
</div>

```

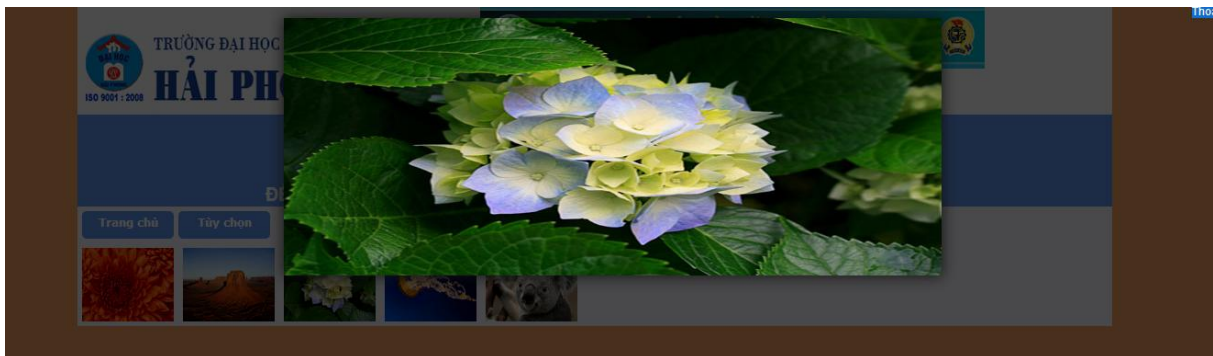


Slide hiển thị ảnh

4.2.3. Thực hiện hiệu ứng lightBox:

Mô tả : Khi click chuột vào một ảnh trong phần danh sách ảnh, thì ảnh sẽ được hiển thị cùng với hiệu ứng lightBox.

Chức năng này được thực hiện dựa trên việc kết hợp giữa plugin jQuery và PHP. Trong đó : PHP thực hiện việc load ảnh ra danh sách, còn jQuery sẽ thực hiện hiệu ứng khi click.



Ảnh được xuất hiện khi click

- Đoạn script jQuery:

```

$('.lightbox_trigger').click(function (e) {
    e.preventDefault();
    var image_href = $(this).attr("href");
    if ($('#lightbox').length > 0) { // #lightbox exists
        $('#content').html('');
    }
}

```

```

        $('#lightbox').show();
    }
    else {
        var lightbox =
            '<div id="lightbox">' +
                '<p>Thoát</p>' +
                '<div id="content">' +
                    '<img src="" + image_href + "" />' +
                '</div>' +
            '</div>';
        $('body').append(lightbox);
    }
});
$('#lightbox').live('click', function () { $('#lightbox').hide();
});
});

```

- Đoạn mã PHP :

```

<div id='listAnh' style="display:none;">
    <?php
        $j = 0;
        $myfile1 = "directory.txt";
        $fh1 = fopen($myfile1, 'r');
        $theData1 = fread($fh1, filesize($myfile1));
        fclose($fh1);
        if (!isset($theData1)) {
            echo 'Chưa có ảnh cho slide';
        } else {
            $arrayDirectory = explode(';', $theData1);

```

```
for ($j; $j < count($arrayDirectory) - 1; $j++) {  
    echo '<a href = "' . $arrayDirectory[$j] . '" class =  
lightbox_trigger ></a>';  
    }  
    }  
    ?>  
</div>
```

KẾT LUẬN

Đồ án đã tìm hiểu được các kiến thức cơ bản và chuyên sâu về jQuery và PHP. jQuery là một thư viện JavaScript mở, được ứng dụng vào việc tạo các ứng dụng Web cũng như các hiệu ứng, cách thao tác với các thành phần cơ bản của HTML và CSS. Trong khi đó PHP là một ngôn ngữ lập trình web phổ biến nhất vào thời điểm hiện tại, được sử dụng vào rất nhiều các trang web quản lý, tin tức, các cổng thông tin điện tử ...

Việc kết hợp giữa jQuery và PHP được sử dụng trong rất nhiều các ứng dụng web hiện nay bởi chúng không chỉ tạo ra những hiệu ứng rất nổi bật mà còn cải thiện tốc độ xử lý cho website, tăng khả năng trải nghiệm người dùng, hiệu năng cho hệ thống.

Đồng thời, qua quá trình làm đồ án, em đã học thêm nhiều kiến thức thực tế và biết vận dụng kiến thức đã học để giải quyết một bài toán đặt ra. Tuy nhiên kết quả còn rất hạn chế, cần có sự hỗ trợ rất nhiều của thầy cô. Để có khả năng làm tốt việc vận dụng lý thuyết vào thực hành và có kỹ năng nhất định, em thấy cần phải thực hành nhiều hơn nữa.

Hướng phát triển tiếp theo của đồ án sẽ tiếp tục nghiên cứu sâu hơn về jQuery, kết hợp thêm với việc thao tác với các hệ cơ sở dữ liệu như MySQL và SQLServer; Tìm hiểu thêm về sự kết hợp của jQuery với các hệ ngôn ngữ lập trình khác như ASP, ASP.Net, ...; Triển khai với các ứng dụng trên nền web như các chương trình quản lý trên nền web, các cổng thông tin điện tử, các Website tin tức...

TÀI LIỆU THAM KHẢO

- [1] Tài liệu: *jQuery in Action* - Tác giả: Bear Bibeault và Yehuda Katz.
- [2] Tài liệu: *Essential JavaScript & jQuery Design Patterns* - Tác giả: Addy Osman
- [3] Tài liệu: *jQuery Novice to Ninja* - Tác giả: Earle Castledine và Craig Sharkie