

## BÀI 14. SỬ DỤNG JQUERY

1. JQuery là gì? .....	1
2. Cài đặt và sử dụng .....	2
2.1. Cài đặt.....	2
2.2. Sử dụng.....	2
2.3. Gọi hàm trong thư viện jQuery.....	3
3. Bộ chọn JQuery .....	4
3.1. Bộ chọn phần tử.....	4
3.2. Bộ chọn id.....	5
3.3. Bộ chọn class .....	5
4. Sự kiện trong jQuery .....	6
5. Hiệu ứng jquery .....	7
6. jQuery HTML.....	8
6.1. jQuery thao tác trên DOM .....	8
6.2. Lấy nội dung và các phương thức text(), html(), val() .....	8
6.3. Phương thức attr() – lấy ra các thuộc tính .....	8
6.4. Đặt các giá trị và thuộc tính cho các phần tử HTML. ....	8
6.5. Thêm nội dung vào HTML.....	8
6.6. jQuery thao tác với CSS .....	9
7. jQuery Traversing.....	9
7.1. jQuery Ancestors .....	10
7.2. jQuery Descendants.....	10
7.3. jQuery Siblings.....	11
7.4. jQuery Filterings.....	11

### 1. JQuery là gì?

jQuery là một thư viện JavaScript được thiết kế bởi John Resig vào năm 2006 với phương châm tốt đẹp: viết ít hơn, làm nhiều hơn.

jQuery là một bộ công cụ JavaScript được thiết kế để đơn giản hóa nhiệm vụ khác nhau. Dưới đây là danh sách các tính năng cơ bản quan trọng hỗ trợ bởi jQuery:

- Thao tác với mô hình DOM: jQuery dễ dàng chọn phần tử DOM, duyệt qua và sửa đổi nội dung của nó.
- Xử lý sự kiện: jQuery cung cấp cách nắm bắt các sự kiện, chẳng hạn như một người dùng nhấp vào một liên kết mà không làm xáo trộn HTML.
- Hỗ trợ AJAX: jQuery giúp chúng ta rất nhiều để phát triển một trang web khi hỗ trợ sử dụng công nghệ AJAX.
- Hình ảnh động: jQuery xây dựng rất nhiều các hiệu ứng hình ảnh động mà chúng ta có thể sử dụng trong các trang web của mình.
- Gọn nhẹ: jQuery là thư viện rất gọn nhẹ - kích thước khoảng 19KB.
- Hỗ trợ các loại trình duyệt: jQuery làm việc tốt trên các trình duyệt IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+

## 2. Cài đặt và sử dụng

### 2.1. Cài đặt

Để sử dụng thư viện jQuery, chúng ta làm theo hai bước đơn giản:

+ Vào trang <http://jquery.com/> để lấy phiên bản mới nhất ví dụ như file jquery-1.3.2.min.js.

+ Copy file jquery-1.3.2.min.js vào một thư mục chứa trang web

File jquery-1.3.2.min.js có thể có phiên bản khác. Phiên bản minified là phiên bản ngắn gọn của jquery.

Hoặc: đặt đường dẫn từ trang web trong thẻ <script> như sau:

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"> - với cách này thì máy tính của chúng ta luôn luôn có kết nối internet.

jQuery không cần bất kỳ cài đặt đặc biệt và rất giống với JavaScript, chúng ta không cần bất kỳ sự thiết lập nào để sử dụng jQuery.

### 2.2. Sử dụng

Sử dụng thư viện jquery như sau:

```
<html>
<head>
<title>The jQuery Example</title>
  <script type="text/javascript"
    src="/jquery/jquery-1.3.2.min.js"></script>
  <script type="text/javascript">
    // đoạn code javascript ở đây
  </script>
</head>
<body>
.....
</body>
</html>
```

### 2.3. Gọi hàm trong thư viện jQuery

Hầu hết tất cả mọi công việc chúng ta làm khi sử dụng jQuery là đọc hoặc thao tác với mô hình đối tượng tài liệu (DOM), chúng ta cần phải đảm bảo rằng chúng ta bắt đầu thêm các sự kiện ngay khi DOM đã sẵn sàng.

Để làm điều này, chúng ta đăng ký một sự kiện đã sẵn sàng cho các tài liệu như sau:

```
$(document).ready(function() {  
    // đoạn code khi DOM đã sẵn sàng  
});
```

Đây là ví dụ đầu tiên: Khi click vào dòng “Click on this to see a dialogue box.” một hộp thoại sẽ hiển thị “Hello world”.

```
<html>  
<head>  
<title>The jQuery Example</title>  
    <script type="text/javascript"  
        src="/jquery/jquery-1.3.2.min.js"></script>  
  
    <script type="text/javascript" language="javascript">  
  
        $(document).ready(function() {  
            $("div").click(function() {  
                alert("Hello world!");  
            });  
        });  
  
    </script>  
</head>  
<body>  
<div id="newdiv">  
Click on this to see a dialogue box.  
</div>  
</body>  
</html>
```

Chúng ta có thể đặt đoạn code trên sang một file khác, giả sử đặt trong file custom.js:

```
$(document).ready(function() {  
    $("div").click(function() {  
        alert("Hello world!");  
    });  
});
```

Và file html nội dung sẽ như sau:

```
<html>  
<head>  
<title>The jQuery Example</title>
```

```

<script type="text/javascript"
src="/jquery/jquery-1.3.2.min.js"></script>
<script type="text/javascript"
src="/jquery/custom.js"></script>
</head>
<body>
<div id="newdiv">
Click on this to see a dialogue box.
</div>
</body>
</html>

```

### 3. Bộ chọn JQuery

Query selectors cho phép bạn chọn và thao tác các phần tử HTML.

jQuery selectors được dùng để "tìm" (hoặc chọn) HTML elements dựa trên id, class, loại, thuộc tính, giá trị của thuộc tính và nhiều hơn nữa. Nó dựa trên CSS Selectors, và ngoài ra nó còn có thêm một số selectors riêng.

Tất cả các **selectors** trong jQuery bắt đầu với kí hiệu dollar và dấu ngoặc đơn: **\$()**.

#### 3.1. Bộ chọn phần tử

Với **jQuery selector** bạn có thể chọn tất cả các phần tử **<p>** trên website giống như sau:

```
$("p")
```

Ví dụ:

Khi người dùng click chuột vào button, tất cả phần tử **<p>** sẽ được ẩn đi

```

<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
</head>
<body>
<h2>Đây là thẻ H2 </h2>
<p>Đây là thẻ p.</p>
<p>Đây là thẻ p.</p>
<button>Click để xoá thẻ p</button>
</body>
</html>

```

Kết quả:

## Đây là thẻ H2

Đây là thẻ p.

Đây là thẻ p.

Click để xoá thẻ p

### 3.2. Bộ chọn id

jQuery **#id** selector được dùng để tìm thuộc tính id của 1 thẻ HTML cụ thể.

Thuộc tính id thì nên được mô tả duy nhất cho 1 thẻ HTML trên 1 trang, vì vậy bạn nên dùng #id selector khi bạn muốn tìm 1 phần tử duy nhất trên website.

Ví dụ muốn tìm thẻ có cùng id="test" bạn làm như sau :

```
$("#test")
```

Ví dụ:

Khi người dùng click vào button, phần tử xác định có id="test" sẽ được ẩn đi

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide();
    });
});
</script>
</head>
<body>
<h2>Đây là thẻ H2</h2>
<p>Đây là thẻ p.</p>
<p id="test">Đây là thẻ p có id là test.</p>
<button>Click để xoá id test</button>
</body>
</html>
```

### 3.3. Bộ chọn class

jQuery class selector được dùng để tìm các phần tử có cùng 1 class cụ thể

Ví dụ muốn tìm tất cả các thẻ có cùng class="test" bạn làm như sau :

```
$(".test")
```

Khi người dùng click vào button, tất cả các thẻ có cùng class='test' sẽ được ẩn đi.

Ví dụ:

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

```

$("button").click(function(){
    $(".test").hide();
});
});

```

\*Các ví dụ về jQuery Selectors

Cú pháp	Mô tả
<code>\$("*")</code>	Chọn tất cả phần tử
<code>\$(this)</code>	Chọn HTML phần tử hiện tại
<code>\$("p.intro")</code>	Chọn tất cả phần tử <code>&lt;p&gt;</code> với <code>class="intro"</code>
<code>\$("p:first")</code>	Chọn phần tử <code>&lt;p&gt;</code> đầu tiên
<code>\$("ul li:first")</code>	Chọn phần tử <code>&lt;li&gt;</code> đầu tiên của <code>&lt;ul&gt;</code> đầu tiên
<code>\$("ul li:first-child")</code>	Chọn phần tử <code>&lt;li&gt;</code> đầu tiên của mọi <code>&lt;ul&gt;</code>
<code>\$("[href]")</code>	Chọn tất cả phần tử có thuộc tính <code>href</code>
<code>\$("a[target='_blank']")</code>	Chọn tất cả thẻ <code>&lt;a&gt;</code> có thuộc tính <code>target</code> là <code>"_blank"</code>
<code>\$("a[target!='_blank']")</code>	Chọn tất cả thẻ <code>&lt;a&gt;</code> có thuộc tính <code>target</code> khác <code>"_blank"</code>
<code>\$(":button")</code>	Chọn tất cả <code>&lt;button&gt;</code> và <code>&lt;input&gt;</code> elements có <code>type="button"</code>
<code>\$("tr:even")</code>	Chọn tất cả phần tử chẵn <code>&lt;tr&gt;</code>
<code>\$("tr:odd")</code>	Chọn tất cả phần tử lẻ <code>&lt;tr&gt;</code>

#### 4. Sự kiện trong jQuery

Tất cả những hành động khác nhau của người dùng tác động lên trang web, và nhận được phản hồi từ trang web được gọi là sự kiện (event)

Ví dụ:

- Di chuyển chuột lên một thành phần của website
- Chọn vào một mục nào đó
- Hoặc là click vào một button

Dưới đây là những sự kiện DOM chung :

Sự kiện chuột	Sự kiện bàn phím	Sự kiện Form	Document/ Window Events
click	keypress	submit	Load
dblclick	keydown	change	Resize

mouseenter	keyup	focus	Scroll
mouseleave		blur	Unload

## 5. Hiệu ứng jquery

Phương thức	Mô tả
<a href="#"><u>animate()</u></a>	Chạy hoạt ảnh tùy chỉnh trên các phần tử đã chọn.
<a href="#"><u>clearQueue()</u></a>	Loại bỏ tất cả các hàm đang được xếp còn lại khỏi các phần tử đã chọn.
<a href="#"><u>delay()</u></a>	Đặt độ trễ cho tất cả các hàm đang được xếp trên các phần tử đã chọn.
<a href="#"><u>dequeue()</u></a>	Loại bỏ hàm tiếp theo khỏi hàng đợi, sau đó thực thi hàm
<a href="#"><u>fadeIn()</u></a>	Làm mờ dần các phần tử đã chọn lúc đi vào
<a href="#"><u>fadeOut()</u></a>	Làm mờ dần các phần tử đã chọn lúc đi ra
<a href="#"><u>fadeTo()</u></a>	Làm mờ dần các phần tử đã chọn đến một độ mờ nhất định
<a href="#"><u>fadeToggle()</u></a>	Chuyển đổi giữa các phương thức fadeIn () và fadeOut ()
<a href="#"><u>finish()</u></a>	Dừng, xóa và hoàn thành tất cả các hoạt ảnh được xếp hàng đợi cho các phần tử đã chọn.
<a href="#"><u>hide()</u></a>	Ẩn các phần tử đã chọn
<a href="#"><u>queue()</u></a>	Hiển thị các hàm được xếp hàng đợi trên các phần tử đã chọn
<a href="#"><u>show()</u></a>	Hiển thị các phần tử đã chọn
<a href="#"><u>slideDown()</u></a>	Trượt xuống (hiển thị) các phần tử đã chọn
<a href="#"><u>slideToggle()</u></a>	Chuyển đổi giữa các phương thức slideUp () và slideDown ()
<a href="#"><u>slideUp()</u></a>	Trượt lên (ẩn) các phần tử đã chọn
<a href="#"><u>stop()</u></a>	Dừng hoạt ảnh hiện đang chạy cho các phần tử đã chọn
<a href="#"><u>toggle()</u></a>	Chuyển đổi giữa các phương thức hide () và show ()

## 6. jQuery HTML

### 6.1. jQuery thao tác trên DOM

Một phần rất quan trọng của jQuery chính là khả năng thao tác trên DOM. jQuery có những phương thức thật sự mạnh mẽ để thay đổi cũng như thao tác lên các phần tử HTML và thuộc tính của chúng.

DOM là viết tắt của Document Object Model – Mô hình đối tượng tài liệu.

DOM được định nghĩa là 1 chuẩn cho việc truy xuất vào tài liệu HTML cũng như XML:

*"Document Object Model (DOM) là một nền tảng và ngôn ngữ trung lập cho phép chương trình và các kịch bản (scripts) truy cập tự động và cập nhật nội dung, cũng như cấu trúc, style của một tài liệu"*

### 6.2. Lấy nội dung và các phương thức text(), html(), val()

Có 3 phương thức đơn giản, dễ sử dụng cho việc thao tác DOM, đó là:

1. **text()** - Set hoặc trả về (returns) nội dung text của một phần tử được chọn
2. **html()** - Set hoặc trả về (returns) nội dung HTML của một phần tử được chọn (gồm cả HTML markup)
3. **val()** - Set hoặc trả về (returns) giá trị của một trường (form field) như textbox chẳng hạn

### 6.3. Phương thức attr() – lấy ra các thuộc tính

Phương thức attr() được dùng để lấy giá trị của thuộc tính.

### 6.4. Đặt các giá trị và thuộc tính cho các phần tử HTML.

Ví dụ sau sẽ cho chúng ta biết cách thực hiện với những phương thức text(), html(), và val() :

```
$("#btn1").click(function(){
    $("#test1").text("Ha noi");
});
$("#btn2").click(function(){
    $("#test2").html("<b>Tp Ho chi minh</b>");
});
$("#btn3").click(function(){
    $("#test3").val("Viet nam");
});
```

### 6.5. Thêm nội dung vào HTML

1. **append()** - Thêm nội dung vào cuối các phần tử được chọn
2. **prepend()** - Thêm nội dung tại đầu của phần tử được chọn



3. **after()** - Thêm nội dung ở sau phần tử được chọn
4. **before()** - Thêm nội dung ở trước phần tử được chọn

## 6.6. jQuery thao tác với CSS

1. **addClass()** - Thêm một hoặc nhiều class đến phần tử được chọn
2. **removeClass()** - Xoá một hoặc nhiều class từ phần tử được chọn
3. **toggleClass()** - Chính là sự chuyển đổi kết hợp giữa thêm/xoá class từ phần tử được chọn
4. **css()** - Gán hoặc trả về thuộc tính style (style attribute)

Ví dụ về CSS:

Đây cũng là CSS sẽ được sử dụng cho tất cả những ví dụ trong bài học này :

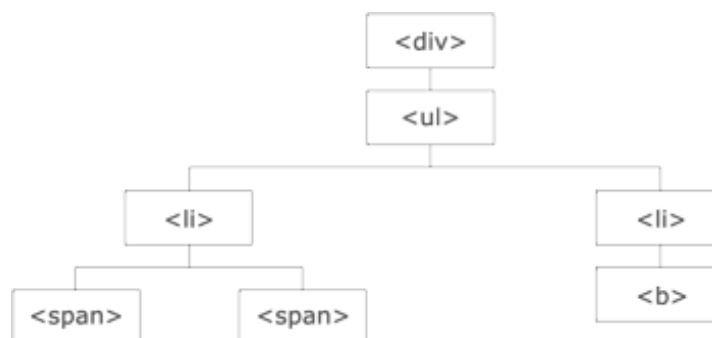
```
.important{
    font-weight:bold;
    font-size:xx-large;
}
.blue{
    color:blue;
}
```

## 7. jQuery Traversing

**jQuery Traversing:** Tìm phần tử (element) dựa trên mối quan hệ với các phần tử khác.

**jQuery traversing** có nghĩa là "di chuyển qua" được dùng để tìm hoặc lựa chọn phần tử HTML dựa trên mối quan hệ của chúng với những phần tử khác. Bắt đầu với một sự lựa chọn và di chuyển vượt qua cho đến khi đến được phần tử mà chúng ta muốn

Hình ảnh dưới đây mô tả về một **family tree** (cây gia đình). Với *jQuery traversing*, chúng ta có thể dễ dàng di chuyển lên phần tử cha của phần tử hiện tại, xuống phần tử hiện tại (phần tử con), di chuyển đến những phần tử ngang hàng (anh, chị, em) của phần tử hiện tại.



**Giải thích sơ đồ cây gia đình:**

1. Thẻ `<div>` là cha (**parent**) của `<ul>`, và tổ tiên (**ancestor**) của tất cả những thẻ khác mà nó chứa bên trong như `<li>` `<span>` `<b>`
2. Thẻ `<ul>` là cha (**parent**) của cả hai thẻ `<li>`, và là con (**child**) của `<div>`
3. Thẻ `<li>` bên trái là cha (**parent**) của `<span>`, là con (**child**) của `<ul>` và là một hậu duệ (**descendant**) của `<div>`
4. Thẻ `<span>` là con (**child**) của thẻ `<li>` bên trái và cháu (**descendant**) của `<ul>` và `<div>`
5. Hai thẻ `<li>` là anh chị em (**siblings**) vì cùng cha là thẻ `<ul>`
6. Thẻ `<li>` bên phải là cha (**parent**) của `<b>`, con của `<ul>` và một hậu duệ của `<div>`
7. Thẻ `<b>` là một con của `<li>` phải và một hậu duệ của `<ul>` và `<div>`

### 7.1. jQuery Ancestors

jQuery Ancestors - tìm và chọn các phần tử cha mẹ, tổ tiên.

Có 3 phương thức chính để thực hiện việc đi qua cây DOM để tìm tổ tiên của phần tử đó là:

1. `parent()`
2. `parents()`
3. `parentsUntil()`

Phương thức `parent()`: Trả về trực tiếp phần tử cha mẹ của phần tử được chọn. Và này chỉ đi qua 1 cấp độ đơn trên cây DOM (DOM tree)

Phương thức jQuery `parents`: sẽ trả về tất cả tổ tiên, ông bà, cha mẹ của phần tử được lựa chọn, tất cả theo hướng đi lên điểm gốc của trang (`<html>`).

### 7.2. jQuery Descendants

jQuery Descendants: Tìm và chọn các phần tử con cháu.

Với **jQuery**, các chúng ta có thể dễ dàng thao tác với các thành phần DOM trong cấu trúc HTML, đi lên trên, xuống dưới.

Trong jQuery, có 2 phương thức hữu ích để thực hiện việc đi xuống dưới cây gia đình DOM (DOM tree) đó là :

1. `children()`
2. `find()`

Phương thức `children()`: Trả về tất cả các phần tử con trực tiếp của phần tử được chọn. Nói cách khác thì `children()` chỉ đi qua một cấp độ duy nhất trong cây DOM.

Phương thức `find()`: Được dùng để tìm tất cả các phần tử con của phần tử được chọn với một tham số xác định.

### 7.3. jQuery Siblings

jQuery Siblings: Tìm và chọn các phần tử anh chị ngang hàng

Mục đích của jQuery Siblings chính là thao tác di chuyển trong DOM cấu trúc HTML để tìm kiếm, chọn lựa những phần tử ngang hàng của một phần tử được xác định. Ngang hàng ở đây có nghĩa là cùng thuộc 1 phần tử bố mẹ, hay nói cách khác là những phần tử anh chị em với nhau.

Không những thế *jQuery Siblings* còn có khả năng giúp chúng ta chọn những phần tử ngang hàng mà chúng ta muốn trong một giới hạn được chính chúng ta đặt ra.

Các phương thức thực hiện việc đi ngang trong cây DOM như sau

1. `siblings()`
2. `next()`
3. `nextAll()`
4. `nextUntil()`
5. `prev()`
6. `prevAll()`
7. `prevUntil()`

+ `siblings()`: Trả về tất cả những phần tử ngang hàng của phần tử được chọn.

+ `next`: Trả về phần tử ngang hàng tiếp theo của phần tử được chọn.

+ `nextAll`: Trả về tất cả phần tử tiếp theo của phần tử được chọn.

+ `nextUntil`: Trả về tất cả những phần tử anh chị em tiếp theo giữa 2 đối số nhất định

+ `prev()`, `prevAll()` & `prevUntil()` : Các phương thức này làm việc tương tự như các phương thức ở trên nhưng trả về kết quả ngược lại là đi theo hướng trước đó.

### 7.4. jQuery Filterings

Có 3 phương thức cơ bản để thực hiện kỹ thuật lọc đó là **`first()`**, **`last()`** và **`eq()`**.

Và thêm 2 phương thức lọc khác như **`filter()`** và **`not()`** sẽ cho phép chúng ta chọn các phần tử phù hợp hoặc không phù hợp theo một tiêu chí nhất định

- \* `first()`: Tìm được phần tử đầu tiên trong nhóm các phần tử.
- \* `last()`: Trả về phần tử cuối cùng của nhóm các phần tử được chọn.
- \* `eq(index)`: Trả về phần tử có chỉ số bằng với chỉ số là `index`.
- \* `filter()`: Lọc phần tử phù hợp với một tiêu chí nào đó.
- \* `not()`: Lọc phần tử không phù hợp với một tiêu chí nào đó.