

# jQuery

---

# Nội dung

---

Giới thiệu về jQuery

**Selector** - xử lý luồng lách giữa các đối tượng HTML

**Attributes** - Xử lý các thuộc tính của đối tượng HTML

**DOM** - xử lý Data Object Model

**Effect** - xử lý hiệu ứng

**Event** - xử lý sự kiện

**Form** - xử lý sự kiện liên quan tới form

# Giới thiệu jQuery

---

**jQuery** là thư viện được viết bằng ngôn ngữ JavaScript hỗ trợ cho việc thiết kế giao diện trên Web.

**jQuery** có thể cải thiện hiệu suất của ứng dụng

**jQuery** tương thích hầu hết các trình duyệt

**jQuery** giúp thực hiện UI mà không phải viết nhiều dòng code

**jQuery** bạn có thể tạo ra các hiệu ứng đẹp mắt cho website.

# Thực thi jQuery

---

Bước 1: khai báo thư viện jQuery

Bước 2: Xác định thành phần cần xử lý bằng jQuery

Bước 3: xác định các thuộc tính, giá trị cần thay đổi

Bước 4: xác định sự kiện xảy ra trên từng thành phần của trang HTML

# Sự kiện onload jQuery

```
$(document).ready(function(){  
    /*Nội dung bên trong*/  
});
```

```
<!DOCTYPE html>  
<html>  
    <head>  
        <title>jQuery - onload </title>  
        <script language="javascript" src="jquery-2.1.4.js"></script>  
    </head>  
    <body>  
        <script language="javascript">  
            $(document).ready(function(){  
                alert('Welcome to jQuery');  
            });  
        </script>  
    </body>  
</html>
```

B1: Khai báo thư viện jQuery

# Ví dụ 1

```
<!DOCTYPE html>
<html>
  <head>
    <title>Chạy chương trình jquery đầu tiên</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <script language="javascript" src="jquery-2.1.4.js"></script>
  </head>
  <body>
    <h1>jQuery</h1>
    <input type="text" id="id-textbox" value=""/> <br/>
    <input type="button" id="id-button" value="Alert"/>
    <script language="javascript">
      $('#id-button').click(function()
      {
        // Cách 1: Khai báo biến và lấy giá trị
        // var textbox = $('#id-textbox');
        // alert(textbox.val());

        // Cách 2: Truy xuất trực tiếp
        alert($('#id-textbox').val());
      });
    </script>
  </body>
</html>
```

Khai báo thư viện jQuery

Gán sự kiện click cho thẻ HTML có id= "id-button"

Lấy giá trị của thẻ HTML có id= "id-textbox"



Alert

# Selector

---

Selector được sử dụng để chọn một hoặc nhiều phần tử HTML. Selector bắt đầu với ký hiệu đô la và cặp dấu ngoặc đơn `$()`.

| STT | Selector & Mô tả                                     | Ví dụ  |
|-----|--|--|
| 1   | Tag Name<br>Biểu diễn một tên thẻ có sẵn             | <code>\$('p')</code> chọn tất cả đoạn văn có thẻ p trong trang tài liệu.                           |
| 2   | Tag ID<br>Biểu diễn một tên thẻ có sẵn với ID đã cho | <code>\$('#some-id')</code> chọn tất cả phần tử đơn trong tài liệu mà có một ID là some-id         |
| 3   | Tag Class<br>Biểu diễn một thẻ có sẵn với lớp đã cho | <code>\$('.some-class')</code> chọn tất cả các phần tử trong tài liệu mà có một lớp là some-class. |

# Selector

---

| STT | Cú pháp                                  | Ý nghĩa  |
|-----|--|--|
| 1   | <code>\$('*')</code>                     | Chọn tất cả các thẻ HTML   |
| 2   | <code>\$('.class-name')</code>           | Chọn thẻ HTML có class class-name  |
| 3   | <code>\$('element')</code>               | Chọn tất cả thẻ có tên là element, ví dụ <code>\$('a')</code> tức là chọn tất cả thẻ a                               |
| 4   | <code>\$('#id-name')</code>              | Chọn thẻ có id là id-name  |
| 5   | <code>\$('#id-name, .class-name')</code> | Chọn thẻ có id là id-name hoặc có class là class-name. Đây là truy vấn dựa vào nhiều dấu hiệu cách nhau bởi dấu phẩy |



<style>

```
body{  
    margin: 20px;  
}  
a{display: block;  
border:solid 1px blue;  
padding: 5px; height: 30px;  
line-height: 30px;  
text-decoration: none;  
margin-bottom: 20px;}
```

id = idtest

class=classtest

Không có id, không có class

class=classtest

Kiểm tra

</style>

▼<body>

```
<a href="#" id="idtest">id = idtest</a>  
<a href="#" class="classtest">class=classtest</a>  
<a href="#">Không có id, không có class</a>  
<a href="#" class="classtest">class=classtest</a>  
<button id="button">Kiểm tra</button>
```

</body>

```
$(document).ready(function()
{
    $('#button').click(function()
    {
        // Tất cả các thẻ được thêm vào class selected
        $('*').css("background-color", "yellow");
        // Tất cả các thẻ có class=classtest được thêm vào class selected
        //$('.classtest').css("background-color", "yellow");
        // Tất cả các thẻ id=idtest được thêm vào class selected
        //$('#idtest').css("background-color", "yellow");
    });
});
```

id = idtest

class=classtest

Không có id, không có class

class=classtest

Kiểm tra

```
$(document).ready(function()
{
    $('#button').click(function()
    {
        // Tất cả các thẻ được thêm vào class selected
        //$('*').css("background-color", "yellow");
        // Tất cả các thẻ có class=classtest được thêm vào class selected
        $('.classtest').css("background-color", "yellow");
        // Tất cả các thẻ id=idtest được thêm vào class selected
        //$('#idtest').css("background-color", "yellow");
    });
});
```

id = idtest

class=classtest

Không có id, không có class

class=classtest

Kiểm tra

```
$(document).ready(function()
{
    $('#button').click(function()
    {
        // Tất cả các thẻ được thêm vào class selected
        //$('*').css("background-color", "yellow");
        // Tất cả các thẻ có class=classtest được thêm vào class selected
        //$('.classtest').css("background-color", "yellow");
        // Tất cả các thẻ id=idtest được thêm vào class selected
        $('#idtest').css("background-color", "yellow");
    });
});
```

id = idtest

class=classtest

Không có id, không có class

class=classtest

Kiểm tra

# Attributes

---

Hàm [addClass\(\)](#) có tác dụng thêm class cho đối tượng Object HTML nào đó.

Hàm này [removeClass\(\)](#) có tác dụng xóa class cho đối tượng Object HTML nào đó.

Hàm [attr\(\)](#) dùng để **lấy giá trị** hoặc **gán giá trị** cho các thuộc tính của một hoặc nhiều thẻ HTML

Hàm [removeAttr\(\)](#) xóa thuộc tính cần xóa.

# Attributes

---

```
<h1 style="font-style: initial" class>Xin chào các bạn</h1>  
<input type="button" id="addclass-button" value="addclass">  
<input type="button" id="removeclass-button" value="removeclass">
```

▼<style>

```
.active{  
  border: solid 1px;  
  color:red;  
  padding: 10px;  
}
```

</style>

**Xin chào các bạn**

addclass

removeclass

# Attributes

```
<script language="javascript">
```

```
// Sự kiện onload trong jquery
$(document).ready(function()
{
    // addClass
    $('#addclass-button').click(function(){
        // Thêm class active vào thẻ h1
        $('h1').addClass('active');
    });

    // removeClass
    $('#removeclass-button').click(function(){
        // Xóa class active ra khỏi thẻ h1
        $('h1').removeClass('active');
    });
});
});
```

```
</script>
```

**Xin chào các bạn**

addclass

removeclass

**Xin chào các bạn**

addclass

removeclass

# Attributes

---

Hàm `prop()` dùng để lấy giá trị các thuộc tính của thẻ (các thuộc tính có giá trị `true/false`)

Hàm `hasClass()` dùng để kiểm tra một đối tượng HTML có tồn tại className nào đó hay không?

`$('selector').hasClass(className)`

**Kết quả:**

- **TRUE** nếu đối tượng có class truyền vào
- **FALSE** nếu đối tượng không có class truyền vào



# Attributes

---

```
<h1>Ví dụ 1</h1>
<strong>Sở thích của bạn là gì? </strong>
<br>
<input type="radio" id="an" name="sothich" value="1">
" Ăn "
<br>
<input type="button" id="view1" value="Xem Name và Type">
<input type="button" id="view2" value="Đổi type thành textbox">
<input type="button" id="view3" value="Đổi type thành radio">
<input type="button" id="view4" value="Check chọn hay không">
```

## Ví dụ 1

Sở thích của bạn là gì?

☐ Ăn

Xem Name và Type

Đổi type thành textbox

Đổi type thành radio

Check chọn hay không

# Attributes

---

```
// Bắt đầu code jquery
$(document).ready(function(){

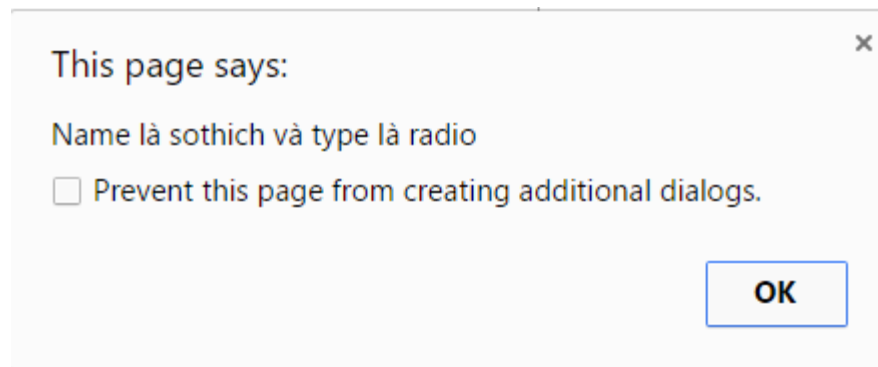
    // Khi click vào button có id = view1
    $('#view1').click(function(){

        // Lấy tên của checkbox có id là an
        var name = $('#an').attr('name');

        // lấy type của checkbox
        var type = $('#an').attr('type');

        alert('Name là ' + name + ' và type là ' + type);

    });
```



# Attributes

---

```
// Khi click vào button có id = view2
$('#view2').click(function(){
    // Thay đổi kiểu thành textbox
    $('#an').attr('type', 'textbox');
});
```

## Ví dụ 1

Sở thích của bạn là gì?

Ăn

Xem Name và Type

Đổi type thành textbox

Đổi type thành radio

Check chọn hay không

# Attributes

---

```
// Khi click vào button có id = view3
$('#view3').click(function(){
    // Thay đổi kiểu thành radio
    $('#an').attr('type', 'radio');

});
```

## Ví dụ 1

Sở thích của bạn là gì?

☐ Ăn

Xem Name và Type

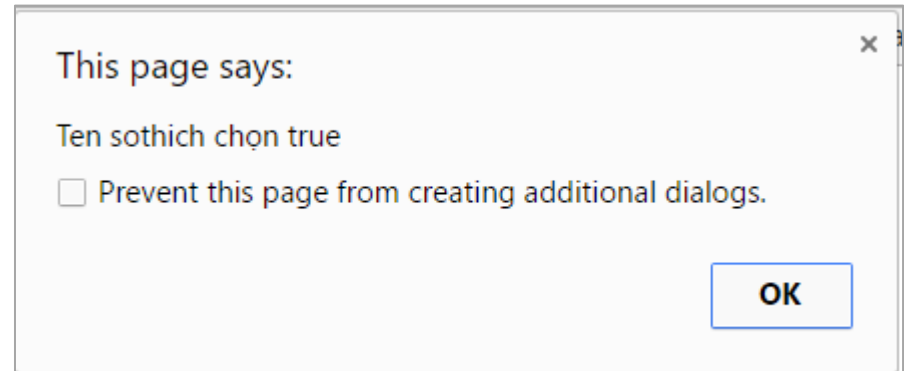
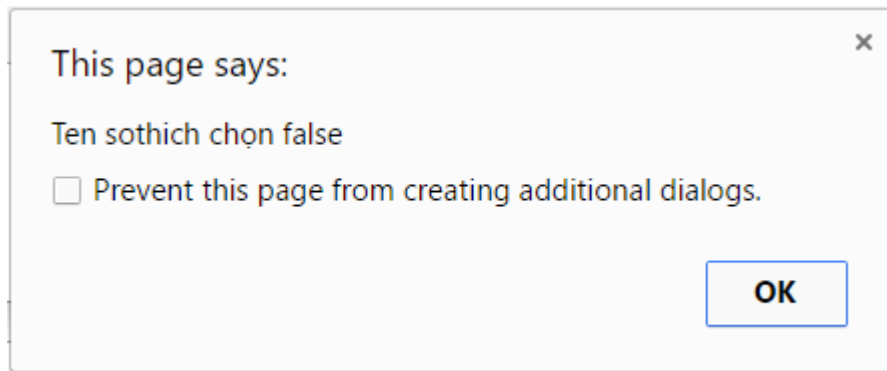
Đổi type thành textbox

Đổi type thành radio

Check chọn hay không

# Attributes

```
// Khi click vào button có id = view4 lấy tên và giá trị check hay ko?  
$('#view4').click(function(){  
    // lấy tên  
    var name= $('#an').prop('name');  
    var checked= $('#an').prop('checked');  
    alert ('Ten '+name + ' chọn '+ checked);  
});
```



# Attributes

---

```
    // Khi click vào button có id =  
view4 xóa thuộc tính checked của radio id=an?  
    $('#view4').click(function(){  
  
        $('#an').removeAttr('checked');  
  
    });  
  
});
```

## Ví dụ 1

Sở thích của bạn là gì?

☐ Ăn

Xem Name và Type

Đổi type thành textbox

Đổi type thành radio

Check chọn hay không

# Selector theo quan hệ phân cấp

| SELECTOR                  | Ý NGHĨA  |
|---------------------------|--|
| Selector1, ..., selectorN | Chọn tất cả các element được xác định bởi tất cả các Selector                                  |
| .class1,.class2           | Chọn tất cả các element có khai báo class1 hoặc class2   |
| Parent > Child            | Chọn tất cả các <b>Child element</b> là con trực tiếp của Parent                               |
| Ancestor Descendant       | Chọn tất cả các <b>Descendant element</b> là con cháu của Ancestor ( chứa bên trong Ancestor ) |
| Prev + Next               | Chọn tất cả các <b>Next element</b> nằm kế tiếp Prev element                                   |
| Prev ~ Siblings           | Chọn tất cả các element anh em khai báo sau Prev và thỏa Sibling selector                      |

Ví dụ:

## Định dạng list-item tầng 1

### CSS

```
.highlight {  
    border-bottom: 1px  
    solid #e6db55;  
    padding: 5px;  
}
```

### JQUERY

```
$(document).ready(function() {  
    $('#nav > li').addClass('highlight');  
});
```

Highlight chỉ cho list items ở tầng 1

[Home Page](#)

---

[About Me](#)

---

[Forum](#)

---

[Ebooks](#)

- [jQuery](#)
  - [CSS](#)
  - [HTML](#)
- 

[Tutorials](#)

---

[Photoshop](#)

- [Action](#)
  - [Effect](#)
  - [Plugins](#)
- 

[Email](#)

---



Ví dụ:

## Định dạng list-item tầng 2

### CSS

```
.background {  
    color: #e6db55; }
```

### JQUERY

```
$(document).ready(function() {  
    $('#nav> li').addClass('highlight');  
    $('#nav li:not(.highlight)').addClass('background');  
});
```

[Home Page](#)

---

[About Me](#)

---

[Forum](#)

---

[Ebooks](#)

- [jQuery](#)
- [CSS](#)
- [HTML](#)

[Tutorials](#)

---

[Photoshop](#)

- [Action](#)
- [Effect](#)
- [Plugins](#)

[Email](#)

---

# Form Selector

| SELECTOR  | Ý NGHĨA                                       |
|-----------|---|
| :input    | Chọn tất cả các thẻ input, textarea trên form |
| :text     | Chọn tất cả các textfield trên form           |
| :password | Chọn tất cả các password field                |
| :radio    | Chọn tất cả các radio button                  |
| :checkbox | Chọn tất cả các checkbox                      |
| :submit   | Chọn tất cả các button submit                 |
| :reset    | Chọn tất cả các button reset                  |
| :image    | Chọn tất cả các image                         |
| :button   | Chọn tất cả các generalized button            |
| :file     | Chọn tất cả các control upload file           |

# Form Selector

---

```
<script src="Scripts/jquery-1.4.1.min.js" type="text/javascript"></script>
<script type="text/javascript">
    /*sau khi trang web được tải thành công thì sự kiện ready sẽ xảy ra*/
    $(document).ready(function () {

        /*đặt độ rộng và kẻ đường viền cho :text, :password và textarea*/
        $(":text, :password, textarea").css(
            { "width": "350px", "border": "1px dotted red" });
        $(":button, :reset, :submit").css({ "width": "80px", "height":
            "25px", "color": "red", "font-weight": "bold" });

    });
</script>
```

# Bộ chọn các phần tử form

```
$(":text, :password, textarea").css(  
    { "width": "350px", "border": "1px dotted red" });  
$(":button, :reset, :submit").css({ "width": "80px", "height":  
    "25px", "color": "red", "font-weight": "bold" });
```

Họ và tên:

Mật khẩu:

Giới tính: ☐ Nam ☐ Nữ

Sở thích: ☐ Đọc sách ☐ Du lịch ☐ Thể thao ☐ Âm nhạc

Quốc tịch:

Hình ảnh:  No file chosen

Ghi chú:

```
$ ("form :text").css ("border", "1px solid red");
```

First Name

Last Name

Disabled Text Field

Gender

☒ M ☐ F☒ Widgets☐ Hibbity Jibbities☒ SplashBangers☐ Whatzits

What products are you interested in?

Comments:

Optional life story file

Browse...

submit

reset

# Bộ lọc Selector

---

JQuery Selector thường trả về 1 tập đối tượng. ***JQuery Filter được dùng để lọc trên kết quả chọn của JQuery Selector.***

Có 6 loại Filter:

- Basic: lọc phần tử ở vị trí đầu tiên, cuối cùng, chẵn, lẻ,...
- Content: lọc dựa trên nội dung
- Visibility: lọc dựa trên trạng thái hiển thị của Selector
- Attribute: lọc dựa trên thuộc tính của element
- Child: lọc dựa trên mối quan hệ với element cha
- Form: lọc trên các thành phần khai báo trên Form

# Bộ lọc Selector cơ bản

| Bộ lọc            | Ý nghĩa  |
|-------------------|--|
| :first            | Chọn phần tử <i>đầu tiên</i> trong tập kết quả do Selector trả về  |
| :last             | Chọn phần tử <i>cuối cùng</i> trong tập kết quả do Selector trả về |
| :even             | Chọn phần tử chẵn  |
| :odd              | Chọn phần tử lẻ  |
| :eq ( index )     | Chọn phần tử tại vị trí index                                      |
| :gt ( index )     | Chọn phần tử có vị trí > index                                     |
| :lt ( index )     | Chọn phần tử có vị trí < index                                     |
| :header           | Chọn tất cả header element (H1, H2, .. H6)                         |
| :not ( selector ) | Chọn phần tử không thỏa selector                                   |

## Basic JQuery Filter

```
<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
```

```
<p class="a">this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p>this is paragraph 4</p>
```

```
$("document").ready(function () {
  $("p:odd").css("border", "1px solid red");
});
```

- item 1
- item 2
- item 3
- item 3

this is paragraph 1

this is paragraph 2

this is paragraph 3

this is paragraph 4



## Basic JQuery Filter

```
<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
<p class="a">this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a">this is paragraph 4</p>

$("document").ready(function () {
  $(".a:odd").css("border", "1px solid red");
});
```

- item 1
- item 2
- item 3
- item 3

this is paragraph 1

this is paragraph 2

this is paragraph 3

this is paragraph 4

## Basic JQuery Filter

```
<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
```

- item 1
- item 2
- item 3
- item 3

```
<p class="a" >this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a">this is paragraph 4</p>
```

this is paragraph 1

this is paragraph 2

```
$("document").ready(function () {
  $("p:eq(1)").css("border", "1px solid red");
  $("p:eq(3)").css("border", "1px solid red");
});
```

this is paragraph 3

this is paragraph 4

```

<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
<p class="a">this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a">this is paragraph 4</p>

```

- item 1
- item 2
- item 3
- item 3

this is paragraph 1

this is paragraph 2

this is paragraph 3

this is paragraph 4

```

$("document").ready(function () {
  $("p:gt(1),p:lt(1)").css("border","1px solid red");
});

$("document").ready(function () {
  $("p:not(p:eq(1))").css("border","1px solid red");
});

```

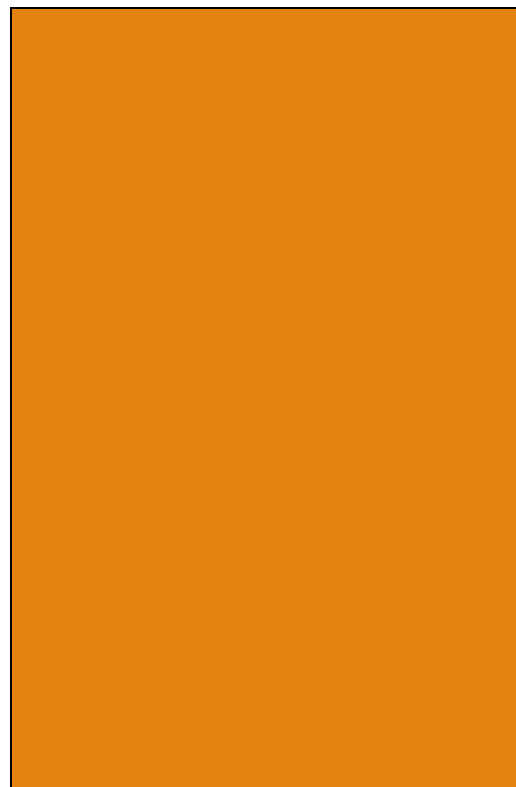
# Bộ lọc Selector thuộc tính

| Cú pháp                                   | Ý nghĩa  |
|---|--|
| <code>\$('[thuộc tính]')</code>           | Tìm thẻ HTML chỉ cần CÓ thuộc tính là được               |
| <code>\$('[thuộc tính*="value"]')</code>  | Tìm thẻ HTML có thuộc tính CHỨA chuỗi value              |
| <code>\$('[thuộc tính^="value"]')</code>  | Tìm thẻ HTML có thuộc tính phải BẮT ĐẦU bằng chuỗi value |
| <code>\$('[thuộc tính\$="value"]')</code> | Tìm thẻ HTML có thuộc tính KẾT THÚC bằng chuỗi value     |
| <code>\$('[thuộc tính="value"]')</code>   | Tìm thẻ HTML có thuộc tính = value                       |
| <code>\$('[thuộc tính!="value"]')</code>  | Tìm thẻ HTML có thuộc tính có giá trị KHÁC value         |

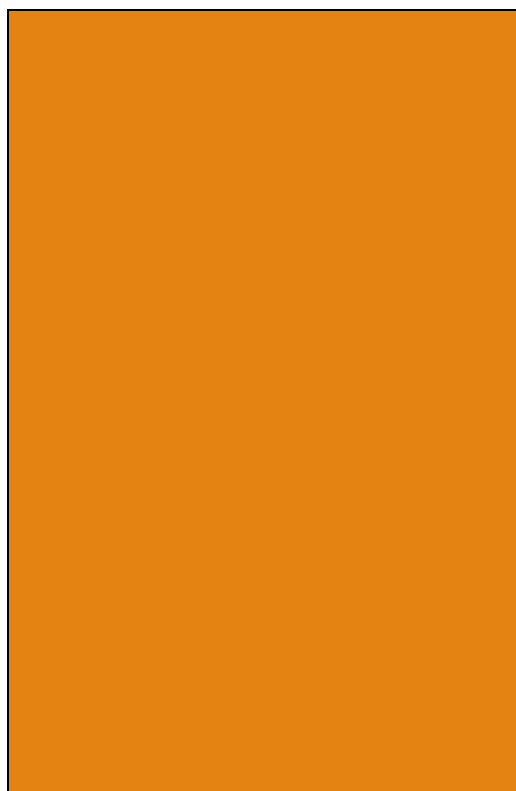
```
<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
<p class="a">this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a">this is paragraph 4</p>
```

```
$( "document" ).ready(function () {
  $( "p[class]" ).css("border","1px solid red");

});
```



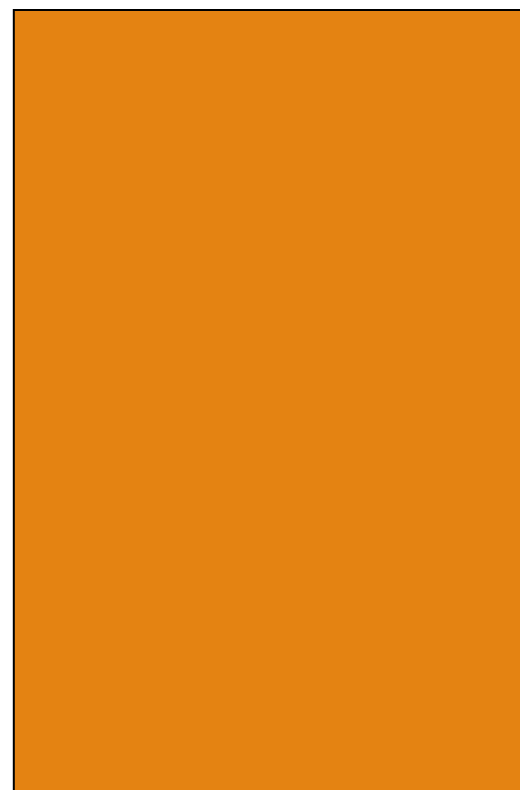
```
<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
<p class="a">this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a">this is paragraph 4</p>
```



```
$("document").ready(function () {
  $("li[class=b]").css("border", "1px solid red");
});
```

```
<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
<p class="a">this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a">this is paragraph 4</p>
```

```
$( "document" ).ready(function () {
  $( "p[id^=para]" ).css("border", "1px solid red");
});
```



```
<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
<p class="a" >this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a" lang="en-us" >this is paragraph 4</p>
```

```
$( "document" ).ready(function () {
    $( "p[class=a][lang*=us]" ).css("border","1px solid red");

});
```



# Bộ lọc Selector cơ bản

---

| Filter           | Ý nghĩa   |
|------------------|---|
| :contains (text) | Lọc các phần tử có chứa chuỗi text                            |
| :empty           | Lọc các phần tử rỗng  |
| :has(selector)   | Lọc các phần tử có ít nhất 1 phần tử selector                 |
| :parent          | Lọc ít nhất 1 phần tử có ít nhất 1 con hoặc text (không rỗng) |
| :visible         | Lọc các phần tử có trạng thái hiển thị visible                |
| :hidden          | Lọc các phần tử có trạng thái ẩn hidden                       |

# Các phương thức cho Selector

| Thuộc tính              | Ý nghĩa   |
|-------------------------|---|
| <code>.next()</code>    | Chọn các phần tử kế bên selector đang chọn        |
| <code>.nextAll()</code> | Chọn TẤT CẢ các phần tử kế bên selector đang chọn |
| <code>.prev()</code>    | Chọn các phần tử phía trước selector đang chọn    |
| <code>.prevAll()</code> | Chọn các TẤT CẢ phần tử trước selector đang chọn  |
| <code>.andSelf()</code> | Chọn các selector đang chọn                       |

## Định dạng từng ô cụ thể

```
$(document).ready(function() {  
  $('td:contains(Bruce Wiliss)').next().addClass('red');  
});
```

| Movies           | Actors/ Actresses     | Year Make |
|------------------|-----------------------|-----------|
| Terminator       | arnold schwarzenegger | 1991      |
| Die Hard         | Bruce Willis          | 2000      |
| Speed            | Sandra Bullock        | 1997      |
| Independence Day | Will Smith            | 1999      |
| Armageddon       | Bruce Willis          | 1997      |
| Under Siege      | Steven Seagal         | 1996      |
| Avatar           | Unknown               | 2010      |

```
$(document).ready(function() {
  $('td:contains(Bruce Wiliss)').nextAll().addClass('red');
});
```

| Movies           | Actors/ Actresses     | Year Make |
|------------------|-----------------------|-----------|
| Terminator       | arnold schwarzenegger | 1991      |
| Die Hard         | Bruce Willis          | 2000      |
| Speed            | Sandra Bullock        | 1997      |
| Independence Day | Will Smith            | 1999      |
| Armageddon       | Bruce Willis          | 1997      |
| Under Siege      | Steven Seagal         | 1996      |
| Avatar           | Unknown               | 2010      |

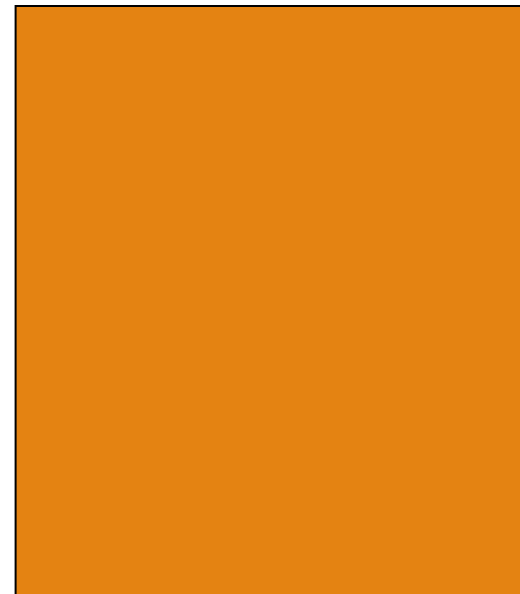
```
$(document).ready(function() {
  $('td:contains(Bruce Wiliss)').nextAll().andSelf().addClass('red');
});
```

| Movies           | Actors/ Actresses     | Year Make   |
|------------------|-----------------------|-------------|
| Terminator       | arnold schwarzenegger | 1991        |
| Die Hard         | <b>Bruce Willis</b>   | <b>2000</b> |
| Speed            | Sandra Bullock        | 1997        |
| Independence Day | Will Smith            | 1999        |
| Armageddon       | <b>Bruce Willis</b>   | <b>1997</b> |
| Under Siege      | Steven Seagal         | 1996        |
| Avatar           | Unknown               | 2010        |

# Bộ lọc Selector cơ bản

---

```
<p>  
  Hello |  
  <a id="link1" href="#">world</a>  
  <span id="abc">  
    <a href="#">2009</a>  
  </span>  
  <span >  
    <a href="#">2010</a>  
  </span>  
</p>
```



```
$( "document" ).ready(function () {  
  $( "span:has(a:contains(2010))" ).css("border","1px solid red");  
});
```

# Bộ lọc Selector cơ bản

---

```
<p>
  Hello
  <a id="link1" href="#">world</a>
  <span id="abc">
    <a href="#">2009</a>
  </span>
  <span >
    <a href="#">2010</a>
  </span>
  <span>
  </span>
</p>
```

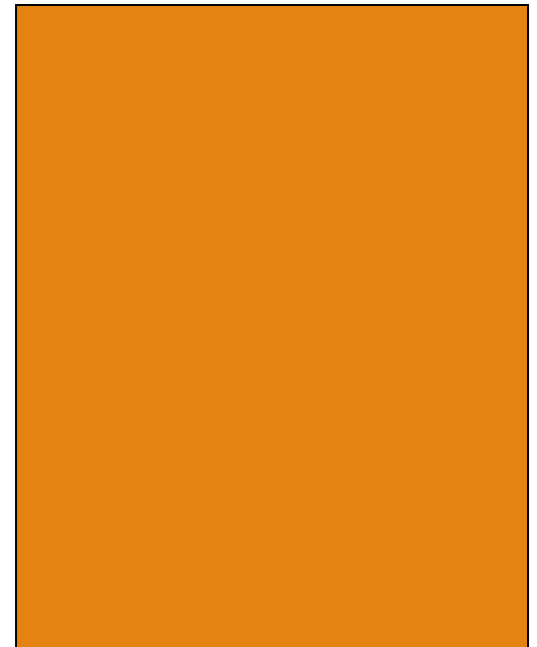


```
$( "document" ).ready(function () {
    $( "span:parent" ).css("border", "1px solid red");
});
```

# Bộ lọc Selector cơ bản

---

```
<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
<p class="a">this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a" lang="en-us">this is paragraph 4</p>
<p></p>
<div></div>
```



```
$("document").ready(function () {
  $("p:empty").css("border", "1px solid red");
  $("div:empty").css("border", "1px solid red");
});
```



# Bộ lọc Selector con cháu

| BỘ LỌC   | Ý NGHĨA  |
|--|--|
| :nth-child(index)<br>:nth-child(even)<br>:nth-child(odd) | Lọc các phần tử theo vị trí so với cha của nó                                  |
| :nth-child(equation)                                     | Lọc phần tử theo vị trí ( vị trí thỏa phương trình tham số ) so với cha của nó |
| :first-child   | Lấy phần tử đầu tiên so với cha của nó   |
| :last-child  | Lấy phần tử cuối cùng so với cha của nó  |
| :only-child  | Lấy phần tử nếu phần tử này là con duy nhất so với cha của nó                  |

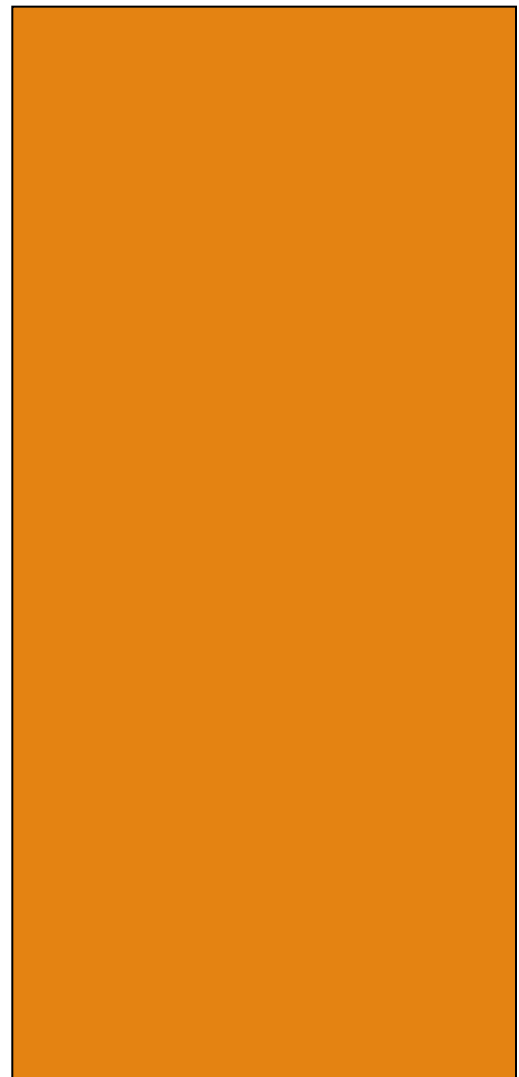
```
<ul id="list1">
  <li class="a">item 1</li>
  <li class="a">item 2</li>
  <li class="b">item 3</li>
  <li class="b">item 3</li>
</ul>
```

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

    $("li:nth-child(2)").css("border","1px solid red");

});
```

```
$("document").ready(function () {
    $("li:nth-child(2n+1)").css("border","1px solid red");
});
```



# DOM

---

| BỘ LỘC           | Ý NGHĨA  |
|------------------|--|
| size(), length   | Lấy số phần tử trong tập kết quả của Selector                            |
| get()            | Lấy tập DOM elements trong tập kết quả của Selector                      |
| get(index)       | Lấy DOM element ở vị trí index   |
| find(expression) | Lấy các element con cháu thỏa expression                                 |
| each()           | Gọi thực thi phương thức với từng element trong tập kết quả của Selector |

# DOM

## Duyệt danh sách các element trong document

```
<p class="a" >this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a" lang="en-us" >this is paragraph 4</p>
```

```
alert( $("p").size() );           // 4
for(var i=0 ; i < $("p").size() ; ++i )
{
    var name = $("p").get(i); // DOM element
    var innerText = $("p").get(i).innerText;
}
```

# DOM

---

## Truy cập, thay đổi nội dung trong element

| Phương thức                         | Ý nghĩa  |
|-------------------------------------|--|
| <code>html()</code>                 | Lấy nội dung html bên trong element đầu tiên thỏa selector   |
| <code>html( newContent )</code>     | Thay đổi nội dung html bên trong mọi element thỏa selector ( tương tự <code>innerHTML</code> trong DOM ) |
| <code>text()</code>                 | Lấy nội dung text bên trong element đầu tiên   |
| <code>text( newTextContent )</code> | Thay đổi nội dung text bên trong mọi element thỏa selector ( tương tự <code>innerText</code> )           |

# DOM

---

```
<p class="a" >this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a" lang="en-us" >this is paragraph 4</p>
```

```
$("#ul").find("li.a").css("border","1px solid red");
```

```
var i = 1;
$("p").each(function () {
    $(this).html("Custom paragraph " + i);
    i++;
});
```

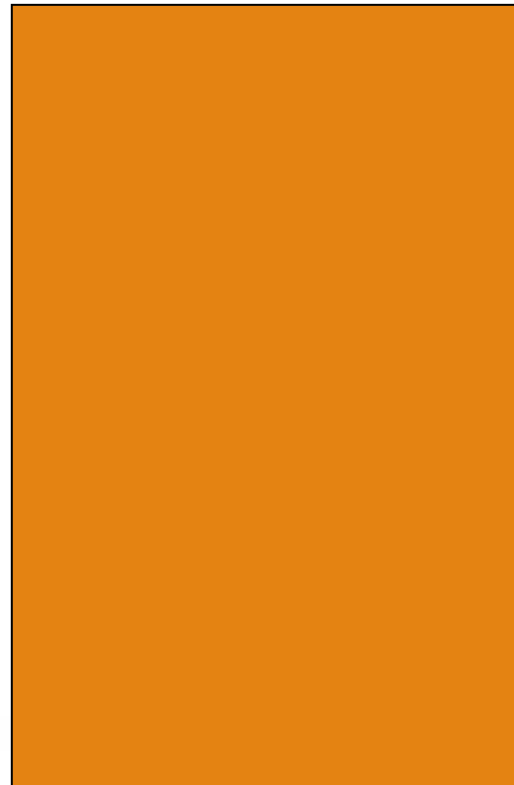
# DOM

---

```
<ul id="list1">  
  <li class="a">item 1</li>  
  <li class="a">item 2</li>  
  <li class="b">item 3</li>  
  <li class="b">item 3</li>  
</ul>
```

```
alert($("#ul").html());
```

```
alert($("#ul").text());
```



# Thêm nội dung DOM

| Phương thức                            | Ý nghĩa   |
|--|---|
| <code>append(nội dung)</code>          | Chèn nội dung vào sau nội dung selector đang chọn           |
| <code>appendTo(selectorthamso)</code>  | Chèn nội dung của selector đang chọn vào sau selectorthamso |
| <code>prpend(content)</code>           | >< <code>append()</code>                                    |
| <code>prependTo(selectorthamso)</code> | >< <code>appendTo(selectorthamso)</code>                    |
| <code>after (content)</code>           | Chèn sau,   |
| <code>before (content)</code>          | CHèn trước  |



```
$ ("p#para2") .appendTo ("li.a") ;
```

```
$ ("p.b") .prependTo ("li.b") ;
```

```
<style>
  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
```

```
<p>Click on any square below:</p>
<span id="result"> </span>
```

```
<div class="div" style="background-color:blue;"></div>
<div class="div" style="background-color:green;"></div>
<div class="div" style="background-color:red;"></div>
```

```
$(document).ready(function() {
  $("div").click(function () {
    $(this).before('<div class="div"></div>' );
  });
});
```

Click on any square below:



Click on any square below:



```
<style>
  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
```

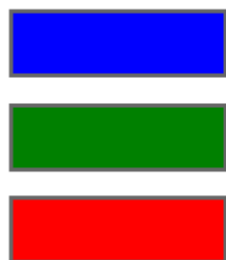
<p>Click on any square below:</p>

<span id="result"> </span>

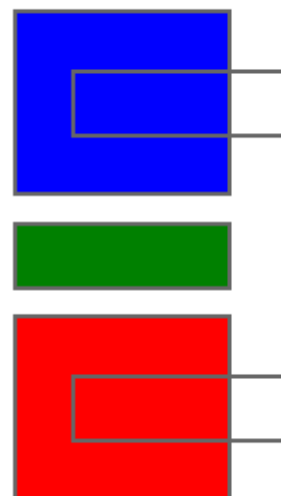
```
<div class="div" style="background-color:blue;"></div>
<div class="div" style="background-color:green;"></div>
<div class="div" style="background-color:red;"></div>
```

```
$(document).ready(function() {
  $("div").click(function () {
    $(this).append('<div class="div"></div>');
  });
});
```

Click on any square below:



Click on any square below:



## Làm việc với CSS

| Phương thức             | Ý nghĩa  |
|-------------------------|--|
| css ( name )            | Lấy giá trị thuộc tính name của element đầu tiên thỏa selector   |
| css ( properties )      | Thiết lập tập thuộc tính css đối với mọi element thỏa selector   |
| css ( property, value ) | Thiết lập giá trị 1 thuộc tính đối với mọi element thỏa selector |

## Làm việc với CSS

```
<p class="a" >this is paragraph 1</p>
<p id="para2">this is paragraph 2</p>
<p class="b">this is paragraph 3</p>
<p class="a" lang="en-us" >this is paragraph 4</p>
```

```
$("#p#para2").css({ "color" : "red" , "background-color" : "green" })
```

this is paragraph 2

```
$("#p.a").css("color", "blue");
```

this is paragraph 1

this is paragraph 2

this is paragraph 3

this is paragraph 4

## Làm việc với CSS

| Phương thức    | Ý nghĩa  |
|----------------|--|
| height ( )     | Lấy chiều cao của element đầu tiên thỏa selector   |
| width ( )      | Lấy chiều rộng của element đầu tiên thỏa selector  |
| height ( val ) | Thiết lập chiều cao của mọi element thỏa selector  |
| width ( val )  | Thiết lập chiều rộng của mọi element thỏa selector |

# Hiệu ứng: Effects

---

# Hiệu ứng: Effects

| Phương thức                            | Ý nghĩa  |
|--|--|
| <code>show ( )</code>                  | Hiển thị các element thỏa selector nếu trước đó bị ẩn  |
| <code>show( speed, callback )</code>   | Hiển thị các element thỏa selector nếu trước đó bị ẩn, speed xác định tốc độ hiển thị. Sau khi hiển thị xong, phương thức callback sẽ được thực thi. |
| <code>hide ( )</code>                  | Ẩn element nếu trước đó đang hiển thị.   |
| <code>hide ( speed, callback )</code>  | Ẩn element nếu trước đó đang hiển thị, tham số có ý nghĩa tương tự phương thức show.   |
| <code>toggle ( )</code>                | Chuyển qua lại trạng thái ẩn/hiện các element.   |
| <code>toggle ( speed, callback)</code> | Chuyển qua lại trạng thái ẩn/hiện các element, tham số có ý nghĩa tương tự phương thức show.   |

- **speed**: tốc độ hiệu ứng quy định bởi các giá trị: “slow”, “normal”, “fast” hoặc **millisecond**



# Hiệu ứng: Effects

Ẩn, hiện element

```
$("#div1").show("normal");
```

```
$("#div1").hide("slow");
```

```
$("#div1").hide(4000); // ẩn trong 4 giây
```

```
// thay đổi luân phiên trạng thái ẩn/hiện
```

```
$("#div1").toggle("fast");
```

# Hiệu ứng: Effects

Fade-in, fade-out

| Phương thức                                    | Ý nghĩa   |
|--|---|
| <code>fadeIn(speed, callback)</code>           | Hiển thị element bằng cách tăng dần độ trong suốt.  |
| <code>fadeOut(speed, callback)</code>          | Ẩn element bằng cách giảm dần độ trong suốt về 0, sau đó thiết lập style display là none. |
| <code>fadeTo(speed, opacity, callback )</code> | Thay đổi độ trong suốt của element.   |

# Hiệu ứng: Effects

Fade-in, fade-out

```
$("#button_fadein").bind("click",function () {  
    $("#div1").fadeIn("normal");  
});
```

```
$("#button_fadeout").bind("click",function () {  
    $("#div1").fadeOut("slow");  
});
```

```
$("#button_fadeto3").bind("click",function () {  
    $("#div1").fadeTo("slow",0.3,function () {  
        alert("finished");  
    });  
});
```

```
$("#button_fadeup").bind("click",function () {  
    $("#div1").fadeTo("slow",1.0);  
});
```

# Hiệu ứng: Effects

## Sliding

| Phương thức                                | Ý nghĩa                                    |
|--|--|
| <code>slideDown(speed, callback)</code>    | Hiển thị element bằng cách tăng chiều cao. |
| <code>slideUp(speed, callback)</code>      | Ẩn element bằng cách giảm chiều cao.       |
| <code>slideToggle( speed, callback)</code> | Chuyển đổi trạng thái ẩn/hiện element.     |

# Hiệu ứng: Effects

## Sliding

```
$("#button_slideup").bind("click",function () {  
    $("#div1").slideUp("normal");  
});
```

```
$("#button_slidedown").bind("click",function () {  
    $("#div1").slideDown("slow");  
});
```

```
$("#button_toggleslide").bind("click",function ()  
    {  
        $("#div1").slideToggle(3000);  
    });
```

# Hiệu ứng: Effects

## Custom animation effect

```
$("selector").animate(properties,[duration],  
[easing],[callback]);
```

| Tham số    | Ý nghĩa  |
|------------|--|
| properties | Các thuộc tính xác trạng thái hiển thị sau khi animate.                |
| duration   | Animate kéo dài trong bao lâu ( "slow", "normal", "fast", milisecond ) |
| easing     | Hiệu ứng xóa : swing, linear   |
| Callback   | Hàm được gọi sau khi animate xong                                      |

```
$("selector").stop();
```

# Hiệu ứng: Effects

## Custom animation effect

```
$("#button_growright").click(function () {  
    $("#div1").animate({width:"800"},"normal");  
});  
  
$("#button_growleft").click(function () {  
    $("#div1").animate({width: "100"},"fast");  
});  
  
$("#button_bigtext").click(function () {  
    $("#div1").animate({fontSize:"40"},2000);  
});  
  
$("#button_movediv").click(function () {  
    $("#div1").animate( { left : "500", fontSize: "50"  } , 1000  
    , "linear" );  
});
```

# Sự kiện: events

- Trong javascript, để xuất hiện thông báo với người dùng khi click vào liên kết bạn phải viết như sau:  
`<a class="click" href="#" onclick="alert('Welcome to Ewebvn.Com')">Click me</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"); });`



# Sự kiện: events

- Javascript cung cấp Bộ quản lý sự kiện *window.onload* cho phép chúng ta thực thi một hàm nào đó.
  - *window.onload* sẽ được thực thi sau khi trình duyệt tải xong toàn bộ tài liệu cần thiết bao gồm stylesheet, hình ảnh...
- 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*
  - *\$(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.

# Sự kiện: events

Cách đăng ký bộ quản lý sự kiện trong jQuery

Cách 1:

```
$(document).ready(function() {  
    //Code và các chương trình thực thi 3})
```

Cách 2:

```
$().ready(function() {  
    //Code và các chương trình thực thi 3})
```

Cách 3:

```
$(function() {  
    //Code và các chương trình thực thi 3})
```

# Sự kiện: events

Một số sự kiện thường sử dụng trong jQuery

## 1. Các sự kiện đối với trình duyệt

- **error()**: Xảy ra khi trình duyệt gặp lỗi trong quá trình tải tài liệu, ví dụ liên kết đến hình ảnh bị gãy, ...
- **resize()**: khi người dùng thay đổi kích thước cửa sổ trình duyệt
- **scroll()**: Xử lý tác tác vụ khi người dùng cuộn trang

# Sự kiện: events

Một số sự kiện thường sử dụng trong jQuery

## 2. Các sự kiện với Form

- `focus()`: khi con trỏ chuột đang tác động tại phần tử Form
- `blur()`: Thực hiện khi click chuột ra ngoài phần tử form
- `change()`: khi giá trị của phần tử form được thay đổi
- `select()`: khi phần tử form được chọn
- `submit()`: Khi form được submit

# Sự kiện: events

Một số sự kiện thường sử dụng trong jQuery

## 3. Sự kiện với bàn phím

- `focusin()`: tương tự `focus()`
- `focusout()`: tương tự `blur()`
- `keyup()`, `keydown()`, `keypress()`: Xảy ra khi đang nhập liệu vào Form

# Sự kiện: events

Một số sự kiện thường sử dụng trong jQuery

## 4. Sự kiện đối với chuột

- **click()**: Khi người dùng click chuột
- **hover()**: di chuyển chuột lên đối tượng
- **toggle()**: Thực hiện theo các lần click chuột của người dùng, ví dụ thay đổi giữa trạng thái ẩn hiện của phần tử

# Sự kiện: events

Xử lý nhanh một số sự kiện thường gặp

| Phương thức           | Ý nghĩa   |
|-----------------------|---|
| click( func )         | Xử lý sự kiện click của 1 selector. Một số hàm khác: blur, mousedown, mouseover, mouseout, submit, ..                         |
| hover ( func1, func2) | Func1: hàm xử lý được gọi khi mouse di chuyển trên selector<br>Func2: hàm xử lý được gọi khi mouse di chuyển ra khỏi selector |

```
$("#div").hover( highlight , highlight );  
function highlight(evt)  
{  
    $("#div").toggleClass("highlight");  
}
```

# Sự kiện: events

Cung cấp các thông tin về event để xử lý

| Thuộc tính / Phương thức | Ý nghĩa  |
|--------------------------|--|
| type                     | Loại event xảy ra, ví dụ: "click"  |
| target                   | Element mà event xảy ra  |
| data                     | Dữ liệu được truyền vào handler bởi phương thức bind                         |
| pageX, pageY             | Tọa độ chuột khi event xảy ra  |
| preventDefault ( )       | Ngăn trình duyệt không thực thi xử lý mặc định, ví dụ khi click vào liên kết |

```
$("#div").click(function (evt)
{
    $(this).html("pageX:" + evt.pageX + ", pageY:" +
    evt.pageY + ",
    type:" + evt.type + ", target:" + evt.target);
});
```



## Một số Ví dụ

### 1. Click

### 2. mousedown và mouseup

- Tình huống: Nhấn trái chuột vào ô vuông xám bên trái thì ô vuông trắng bên phải sẽ hiển thị dòng chữ tương ứng với sự kiện(mouse down hoặc mouse up).

### 3. mouseenter, mouseleave và mousemove:

- Tình huống: Di chuyển chuột vào ô vuông xám bên trái thì ô vuông trắng bên phải sẽ hiển thị số lần sự kiện xảy ra (mouse enter và mouse leave), riêng sự kiện mouse move thì sẽ hiển thị tọa độ của con trỏ chuột trong ô xám

### 4. Một số thuộc tính thường sử dụng của ObjectEventHandle trong mouse event

- pageX, pageY: Tọa độ trên trang web.
- clientX, clientY: Tọa độ trên browser.
- screenX, screenY: Tọa độ trên màn hình máy tính.

| EVENT               | Ý NGHĨA         | VÍ DỤ  |
|---------------------|-----------------|--|
| click(func)         | Nhấp chuột      | <code>\$(“a:first”).click(function(){alert(“Hello”)});</code>                      |
| dblclick(func)      | Nhấp đúp chuột  | <code>\$(“:button”).click(function(){alert(“Hello”)});</code>                      |
| mouseover(func)     | Chuột đi vào    | <code>\$(“img”).mouseover(function(){\$(“div”).toggleClass(“highlight”);});</code> |
| mouseout(func)      | Chuột đi ra     | <code>\$(“img”).mouseout(function(){\$(“div”).toggleClass(“highlight”);});</code>  |
| mousedown(func)     | Đề chuột        | <code>\$(“:text”).mousedown(function(){\$(this).css(“color”, “red”)});</code>      |
| mouseup(func)       | Nhả chuột       | <code>\$(“:text”).mouseup(function(){\$(this).css(“color”, “black”)});</code>      |
| mousemove(func)     | Di chuyển chuột | <code>\$(“#b”).mousemove(function(){alert(“Hello”)});</code>                       |
| keydown(func)       | Đề phím         | <code>\$(“:text”).keydown(function(){\$(this).css(“color”, “red”)});</code>        |
| keypress(func)      | Gõ phím         | <code>\$(“#b”).click(function(){alert(“Hello”)});</code>                           |
| keyup(func)         | Nhả phím        | <code>\$(“:text”).mouseup(function(){\$(this).css(“color”, “black”)});</code>      |
| submit(func)        | Submit form     | <code>\$(“#b”).submit(function(){alert(“Good luck”)});</code>                      |
| blur(func)          | Mất tích cực    | <code>\$(“#b”).blur(function(){alert(“Hello”)});</code>                            |
| focus(func)         | Tích cực        | <code>\$(“#b”).focus(function(){alert(“Hello”)});</code>                           |
| hover(func1, func2) | Vào/ra          | <code>\$(“#b”).hover(function(){alert(“Hi”)}, function(){alert(“Bye”)});</code>    |

# Form validation

---

```
<script src="jquery.validate.js" type="text/javascript"></script>
```

```
$("#form1").validate(  
{  
    rules: {<khai báo luật bắt lỗi cho các trường>},  
    messages: {<định nghĩa các thông báo lỗi>},  
    errorLabelContainer: "<khai báo thẻ chứa lỗi>",  
    wrapper: "<khai báo thẻ bọc lỗi>",  
    submitHandler: <hàm xử lý submit>  
});
```

```

<script src="../../Scripts/jquery-1.4.1.js" type="text/javascript"></script>
<script src="jquery.validate.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function () {
    $("#form1").validate(
        {
            rules:
            {
                txtName: { required: true, minlength: 3 },
                txtAge: { required: true, digits: true, range: [25,65] }
            },
            messages:
            {
                txtAge: { digits: "Nhập số !" },
                txtName: { required: "Không để trống !", minlength: "Ít nhất
3 ký tự !" }
            },
            errorLabelContainer: "#myError",
            wrapper: "li",
            submitHandler: function (form) {
                if (confirm("Dữ liệu form đã hợp lệ. Bạn có muốn submit không
?")) {
                    form.submit();
                }
            }
        }
    );
});

```

# Luật validation

| RULE       | VALUE                   | Ý NGHĨA   | VÍ DỤ                     |
|------------|-------------------------|---|---------------------------|
| range      | [minvalue, maxvalue]    | Giá trị từ min đến max                              | range:[10,100]            |
| email      | boolean                 | Định dạng email                                     | email:true                |
| url        | boolean                 | Định dạng url                                       | <a href="#">url:true</a>  |
| date       | boolean                 | Định dạng ngày javascript                           | date:true                 |
| dateISO    | boolean                 | Định dạng ngày ISO                                  | dateISO:true              |
| number     | boolean                 | Số thực   | number:true               |
| digits     | boolean                 | Số nguyên   | digits:true               |
| creditcard | boolean                 | Định dạng creditcard                                | creditcard:true           |
| accept     | list of file extensions | Kiểu mở rộng file                                   | accept:"doc,xsl,pdf"      |
| equalTo    | Selector                | So sánh giá trị của phần tử và giá trị của selector | equalTo:"#RetypeSelector" |

# Q & A

---

