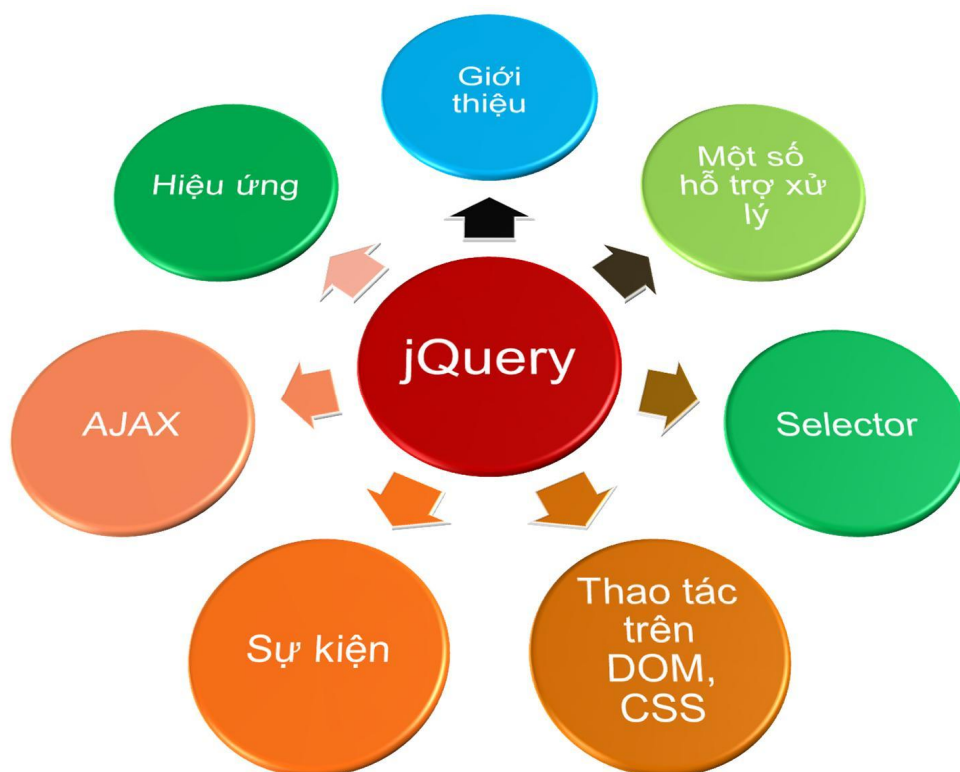


Seminar thực hành
jQuery
Hoàng Anh Tú

Nội dung



Giới thiệu



là một thư viện Javascript

- Nhỏ gọn (29KB – 212KB)
- Dễ học
- Đơn giản hóa lập trình javascript
- Cung cấp nhiều plugins
- Tương thích nhiều trình duyệt
 - IE6+, Firefox 2.0+, Safari 3.0+, Opera 9.0+, Chrome

❖ Version hiện tại: **1.5.1**

❖ Nhiều tổ chức đang sử dụng

- Google, DELL, Mozilla, Wordpress, Drupal, ...



Hoàng Anh Tú

3

Javascript

```
<html>
<head>
<script type="text/javascript">

    function HelloWorld()
    {
        alert("Hello World !!");
    }

</script>
</head>
<body>
    <h1> Javascript - Hello world !!! </h1>
    <hr />

    <input type="button" value="Click me" onClick="HelloWorld();">

</body>
</html>
```



Hoàng Anh Tú

4

Hello jQuery

```
<html>
<head>
<script type="text/javascript" src="jQuery/jquery-1.3.2.js"></script>

<script type="text/javascript">
    $(document).ready(function()
    {
        $("#btnHello").click(function() {
            alert("Hello world !!!");
        });
    });
</script>
</head>
<body>
    <h1> JQuery - Hello world !!! </h1>
    <hr />
    <input type="button" id="btnHello" value="Click me">
</body>
</html>
```



Hoàng Anh Tú

5

Syntax

\$(selector).action()

Chọn ra thẻ
có id là btnHello

Xử lý sự kiện click

```
$("#btnHello").click(function(){
    alert("Hello World!!!");
});
```

Gán sự kiện click
cho tất cả thẻ input

```
$("input").click(function(){
    alert("Hello World!!!");
});
```



Hoàng Anh Tú

6

How to use

- ❖ Download (www.jquery.com) và thêm vào trang html

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

- ❖ Sử dụng trực tiếp thư viện online

```
<script type="text/javascript"  
    src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js">  
</script>
```

```
<script type="text/javascript"  
    src="http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.min.js">  
</script>
```

Hoàng Anh Tú



SELECTORS

Hoàng Anh Tú



Find me – Using selectors

❖ Cho phép tìm kiếm các thẻ trong trang HTML

- Tìm theo tên thẻ

```
$("p").css("background-color", "yellow");
```

- Tìm theo ID thẻ

```
$("#txtTextBox").css("background-color", "yellow");
```

- Tìm theo class của thẻ

```
$(".text").css("background-color", "yellow");
```

- Kết hợp nhiều selector

```
$("div,span,#txtTextBox").css("border","3px solid red");
```



Hoàng Anh Tú

9

Kết hợp nhiều điều kiện

- ❖ \$("*"): tìm tất cả các thẻ

- ❖ \$("p.intro"): tìm tất cả thẻ **p** có **class** là **intro**

- ❖ \$("ul li:first"): tìm thẻ **li** đầu tiên nằm trong thẻ **ul**

- ❖ \$("[href]"): tìm tất cả các thẻ có thuộc tính **href**

- ❖ \$("[href!='#']"): tìm tất cả các thẻ có thuộc tính **href** khác '#'

- ❖ \$("[href\$='.jpg']"): tìm tất cả các thẻ có giá trị của thuộc tính **href** kết thúc là '.jpg'

- ❖ \$("div#intro .head"): tìm tất cả các thẻ có **class** là **.head** nằm trong thẻ **div** có **id** là **#intro**



Hoàng Anh Tú

10

Filter

Filter	Ý nghĩa
:header	Lấy những thẻ header
:first	Lấy thẻ đầu tiên trong danh sách
:last	Lấy thẻ cuối cùng trong danh sách
:not	Lấy những thẻ không thỏa điều kiện của selector
:even	Lấy những thẻ ở vị trí chẵn trong danh sách
:odd	Lấy những thẻ ở vị trí lẻ trong danh sách
:animated	Lấy những thẻ đang thực hiện animation
:eq(index)	Lấy thẻ ở vị trí index trong danh sách
:gt(index)	Lấy những thẻ lớn hơn index trong danh sách
:lt(index)	Lấy những thẻ nhỏ hơn index trong danh sách



Hoàng Anh Tú

11

Filter

```
$("#tr:first").css("font-style", "italic");
```

```
<table>
```

```
<tr><td>Row 1</td></tr>
```

```
<tr><td>Row 2</td></tr>
```

```
<tr><td>Row 3</td></tr>
```

```
</table>
```

Row 1

Row 2

Row 3



```
$("#input:not(:checked) + span").css("background-color", "yellow");
```

```
$("#input").attr("disabled", "disabled");
```

```
<div>
```

```
<input type="checkbox" name="a" />
```

```
<span>Mary</span>
```

```
</div>
```

```
<div>
```

```
<input type="checkbox" name="b" />
```

```
<span>lcm</span>
```

```
</div>
```

```
<div>
```

```
<input type="checkbox" name="c" checked="checked" />
```

```
<span>Peter</span>
```

```
</div>
```



<input type="checkbox"/>	Mary
<input type="checkbox"/>	lcm
<input checked="" type="checkbox"/>	Peter



Hoàng Anh Tú

12

Filter

```
$("td:gt(4)").css("text-decoration", "line-through");
```

```
<table border="1">
```

```
<tr><td>TD #0</td><td>TD #1</td><td>TD #2</td></tr>
```

```
<tr><td>TD #3</td><td>TD #4</td><td>TD #5</td></tr>
```

```
<tr><td>TD #6</td><td>TD #7</td><td>TD #8</td></tr>
```

```
</table>
```



TD #0	TD #1	TD #2
TD #3	TD #4	TD #5
TD #6	TD #7	TD #8

Hoàng Anh Tú



13

Content Filter

Filter	Ý nghĩa
:content(text)	Lấy những thẻ có nội dung chứa text
:empty	Lấy thẻ không có thẻ con bên trong
:has(selector)	Lấy thẻ có chứa thẻ thỏa selector
:parent	Lấy những thẻ có thẻ con bên trong

```
$("div:contains('John')").css("text-decoration", "underline");
```

```
<div>John Resig</div>
```

```
<div>George Martin</div>
```

```
<div>Malcom John Sinclair</div>
```

```
<div>J. Ohn
```



John Resig

George Martin

Malcom John Sinclair

J. Ohn

Hoàng Anh Tú



14

Các filter khác

Visibility Filters:

[:hidden](#)

Matches all elements that are

[:visible](#)

Attribute Filter

Note: In jQuery 1.12.1, simply remove the `attribute` filter.

[\[attribute\]](#)

Matches elements with the specified attribute. This filter is deprecated as of jQuery 1.12.1.

[\[attribute=value\]](#)

Matches elements with the specified attribute and value.

[\[attribute!=value\]](#)

Matches elements with the specified attribute, but not with the specified value.

[\[attribute^=value\]](#)

Matches elements with the specified attribute, starting with the specified value.

[\[attribute\\$=value\]](#)

Matches elements with the specified attribute, ending with the specified value.

[\[attribute*=value\]](#)

Matches elements with the specified attribute, containing the specified value.

[\[attributeFilter\]](#)

[\[attributeFilter\]](#)

Forms:

[:input](#)

Returns: [Array< Element\(s\)>](#)

Matches all input, textarea, select and button elements.

[:text](#)

Returns: [Array< Element\(s\)>](#)

Matches all input elements of type text.

[:password](#)

Returns: [Array< Element\(s\)>](#)

Matches all input elements of type password.

[:radio](#)

Matches all input elements of type radio.

[:checkbox](#)

Matches all input elements of type checkbox.

[:submit](#)

Matches all input elements of type submit.

[:image](#)

Matches all input elements of type image.

[:reset](#)

Returns: [Array< Element\(s\)>](#)

Matches all input elements of type reset.

[:button](#)

Returns: [Array< Element\(s\)>](#)

Matches all button elements and input elements of type button.

[:file](#)

Returns: [Array< Element\(s\)>](#)

Form Filters:

[:enabled](#)

Returns: [Array< Element\(s\)>](#)

Matches all elements that are enabled.

[:disabled](#)

Returns: [Array< Element\(s\)>](#)

Matches all elements that are disabled.

[:checked](#)

Returns: [Array< Element\(s\)>](#)

Matches all elements that are checked.

[:selected](#)

Returns: [Array< Element\(s\)>](#)

Hoàng Anh Tú



15

THAO TÁC TRÊN HTML, CSS

Hoàng Anh Tú



16

Một số thao tác cơ bản

- ❖ `html()`, `html(val)`: lấy chuỗi html trong thẻ
- ❖ `text`, `text(val)`: lấy chuỗi text trong thẻ

```
$("#p").click(function () {  
    var htmlStr = $(this).html();  
    $(this).text(htmlStr);  
});
```

```
<p>  
    <b>Click</b> to change the <span id="tag">html</span>  
</p>
```



```
<p>  
    &lt;b&gt;Click&lt;/b&gt; to change the&lt;span id="tag"&gt;html&lt;/span&gt;  
</p>
```

Click to change the html → `Click to change the html`



Hoàng Anh Tú

17

Một số thao tác cơ bản

- ❖ `val()`, `val(val)`: lấy giá trị trong thuộc tính **value** của thẻ (hoặc gán bằng **val**)
- ❖ `attr(name)`, `attr(name, value)`: lấy giá trị trong thuộc tính có **name** (hoặc gán thuộc tính **name** bằng giá trị **value**)



Hoàng Anh Tú

18

Một số thao tác trên HTML

❖ Thêm thẻ html vào trong trang

```
$("#p").append("<strong>Hello</strong>");
```

<p>I would like to say: </p>

<p>I would like to say: Hello</p>

I would like to say: Hello

❖ appendTo(selector): chèn tất cả các thẻ vào thẻ được lấy bởi selector

```
$("#span").appendTo("#foo");
```



Hoàng Anh Tú

19

Một số thao tác trên HTML

❖ prepend(string / element / jquery): chèn thẻ vào trang HTML nhưng vào đầu của danh sách

❖ prependTo(selector): ngược lại của prepend

<p> is what was said.</p>Hello

\$("#p").prepend(\$("#b"));

\$("#p").append(\$("#b"));

<p> is what was said. Hello</p>

<p>Hello is what was said.</p>



Hoàng Anh Tú

20

Thêm thẻ cùng cấp

- ❖ after(string / element / jquery)
- ❖ before(string / element / jquery)

<p> is what was said.</p>

`$("p").after("Hello");`

<p> is what was said.</p>Hello

`$("p").before("Hello");`

Hello<p> is what was said.</p>

- ❖ insertAfter(selector)
- ❖ insertBefore(selector)

Hoàng Anh Tú



21

Thay thế thẻ

- ❖ replaceWith(content)
- ❖ replaceAll(selector)

`$("p").replaceWith("Paragraph. ");`

<p>Hello</p>

<p>cruel</p>

<p>World</p>



Paragraph.

Paragraph.

Paragraph.

`$("Paragraph. ").replaceAll("p");`

Hoàng Anh Tú



22

Xóa thẻ

❖ `empty()`: xóa tất cả các thẻ con

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

❖ `remove(expr)`: xóa các thẻ thỏa điều kiện trong **expr**

- `$("p").remove()`: xóa tất cả thẻ **p**
- `$("p").remove(":contains('Hello')")`: xóa tất cả các thẻ **p** có nội dung chứa chữ **'Hello'**



Sao chép thẻ qua vị trí khác

❖ `clone()`: sao chép thẻ qua vị trí mới

- `$("b").clone().prependTo("p")`: sao chép thẻ **b** và chèn vào đầu thẻ danh sách thẻ con của **p**

`Hello<p>, how are you?</p>`

`Hello<p>Hello, how are you?</p>`

❖ `clone(true)`: `true` nếu muốn sao chép kèm luôn cả các **sự kiện** của thẻ cũ



Thao tác với CSS

- ❖ `css(name)`: lấy thuộc tính css có tên name
 - `$("p").css("background-color")`: lấy giá trị màu nền của các thẻ p
- ❖ `css(name, value)`: gán thuộc tính css tên name bằng giá trị value
 - `$("p").css("background-color", "yellow")`: gán màu nền của các thẻ p thành màu vàng
- ❖ `css({name1: value1, name2: value2,...})`: gán nhiều thuộc tính css
 - `$("p").css({'background-color':'yellow', 'font-weight':'bolder'})`: gán màu nền và độ đậm



Thao tác với CSS

- ❖ `height()`, `height(val)`: xem/gán giá trị chiều cao
- ❖ `width()`, `width(val)`: xem/gán giá trị chiều rộng



SỰ KIỆN

Hoàng Anh Tú



Các bước xử lý sự kiện

❖ **Bước 1:** Sử dụng \$ với selector phù hợp để chọn đúng đối tượng cần thêm sự kiện.

■ Ví dụ: \$(" #btnSelect") hoặc \$(document) ...

❖ **Bước 2:** Gọi các phương thức bắt sự kiện

```
$("#btnSelect").click(function() {  
    //code xử lý khi sự kiện xảy ra  
});
```

```
$("#btnSelect").bind('click', function(){  
    //code xử lý khi sự kiện xảy ra  
});
```

```
$(document).ready(function() {  
    //code xử lý khi sự kiện xảy ra  
});
```

```
$(document).bind('ready', function(){  
    //code xử lý khi sự kiện xảy ra  
});
```

Hoàng Anh Tú



Một số hàm gán sự kiện(bind)

❖ bind(type, data, fn): gán hàm xử lý sự kiện cho 1 sự kiện

- type: tên sự kiện 'ready', 'click', 'focus', 'dbclick',...
- data: dữ liệu truyền thêm khi xử lý sự kiện
- fn: hàm xử lý sự kiện

```
$('#foo').bind('click', function() {  
    alert('User clicked on "foo."');  
});
```

anonymous function

❖ Có thể sử dụng các hàm click, dbclick, focus,... để code xử lý sự kiện ngắn hơn

Hoàng Anh Tú



29

Một số hàm gán sự kiện(bind)

❖ Truyền thêm dữ liệu vào hàm xử lý sự kiện

```
function handler(event) {  
    alert(event.data.foo)  
}
```

```
$("#p").bind("click", {foo: "bar"}, handler)
```

Hoàng Anh Tú



30

Bỏ xử lý sự kiện (unbind)

❖ `unbind(type, fn)`: bỏ xử lý sự kiện trong hàm

- `$("p").unbind()`: bỏ tất cả hàm xử lý sự kiện của tất cả các thẻ p
- `$("p").unbind("click")`: bỏ tất cả hàm xử lý sự kiện click của tất cả các thẻ p

```
foo = function() {  
    // code xử lý event  
}  
$("p").bind("click", foo)  
$("p").unbind("click", foo)
```

Bỏ hàm xử lý sự kiện foo
của sự kiện click
trong tất cả các thẻ p

Hoàng Anh Tú

31



Live Events

❖ `live(type, data, fn)`: tạo hàm xử lý sự kiện cho các thẻ hiện tại và tương lai thỏa điều kiện trong selector

```
$('.clickme').live('click', function()  
{  
    // code xử lý sự kiện  
});
```

```
$('body').append('<div class="clickme">Another target</div>');
```

click trên thẻ div mới vẫn gọi hàm xử lý sự kiện trên

Hoàng Anh Tú

32



Hủy Live Events

❖ `die(type, fn)`: hủy sự kiện live

Does nothing...

Bind Click

Unbind Click

```
$("#bind").click(function() {  
    $("#theone").live("click", aClick)  
    .text("Can Click!");  
});  
$("#unbind").click(function() {  
    $("#theone").die("click", aClick)  
    .text("Does nothing...");  
});
```

Hoàng Anh Tú



33

Gọi sự kiện 1 lần duy nhất

❖ `one(type, data, fn)`: gán sự kiện chỉ chạy duy nhất 1 lần. Sau đó, những lần sau sẽ gọi hàm xử lý sự kiện của bind hoặc live.

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

chỉ hiện alert box trong
lần đầu tiên click vào thẻ p

Hoàng Anh Tú



34

Kích hoạt sự kiện

❖ trigger(event, data): kích hoạt sự kiện

- \$('#foo').trigger('click'): kích hoạt sự kiện click của thẻ có id là 'foo'

'Custom'

'Event'

```
$('#foo').bind('custom', function(event, param1, param2) {  
    alert(param1 + "\n" + param2);  
});
```

```
$('#foo').trigger('custom', ['Custom', 'Event']);
```

❖ Có thể kích hoạt sự kiện bằng các hàm click(), focus(), hover(), dblclick(),...



Hoàng Anh Tú

35

Một số hàm sự kiện khác

❖ hover(over, out): thêm xử lý cho sự kiện rê chuột vào & ra khỏi thẻ nào đó

```
$("#li").hover(
```

```
function () {  
    $(this).append($("#span ***</span>"));  
},
```

```
function () {  
    $(this).find("span:last").remove();  
}
```

```
);
```

```
$("#li.fade").hover(function() { $(this).fadeOut(100);  
$(this).fadeIn(500); });
```

- Milk
- Bread
- Chips ***
- Socks

```
<ul>  
  <li>Milk</li>  
  <li>Bread</li>  
  <li class='fade'>Chips</li>  
  <li class='fade'>Socks</li>  
</ul>
```



Hoàng Anh Tú

36

Một số hàm sự kiện khác

❖ focus, blur click, dbclick, ...

Text1
Text2

```
$(document).ready(function() {  
    $("input").focus(  
        function() {  
            $(this).css("color", "blue");  
        }  
    );  
  
    $("input").blur(  
        function() {  
            $(this).css("color", "red");  
        }  
    );  
});
```

Hoàng Anh Tú



37

AJAX

Hoàng Anh Tú



38

Kiến trúc Client – Server



AJAX



AJAX – Hạn chế

- ❖ Tăng thời gian phản hồi từ server
- ❖ Khó xác định địa chỉ nội dung trang web
 - Khó trở về trang trước đó
 - Không thể lưu URL nội dung trang web
- ❖ Không thể tìm thấy bằng các search engine
- ❖ Tạo nên các lỗ hổng bảo mật



AJAX & Javascript

Tạo đối tượng request

```
if (window.XMLHttpRequest)
{
    // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
} else {
    // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
```



AJAX & Javascript

Gửi request **bất đồng bộ**

```
xmlhttp.onreadystatechange=function()
{
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        // handle response data
        document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
    }
}
```

```
xmlhttp.open("GET","http://localhost/ajax.aspx",false);
xmlhttp.send();
```



AJAX & Javascript

Gửi dữ liệu **đồng bộ**

```
xmlhttp.open("GET","http://localhost/ajax.aspx",true);
xmlhttp.send();
```

```
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
```



AJAX & Javascript

Xử lý dữ liệu trả về

```
document.getElementById("myDiv").innerHTML=xmlhttp.responseText
```

```
xmlDoc=xmlhttp.responseXML;  
var txt="";  
x=xmlDoc.getElementsByTagName("ARTIST");  
for (i=0;i<x.length;i++)  
{  
    txt=txt + x[i].childNodes[0].nodeValue + "<br />";  
}  
document.getElementById("myDiv").innerHTML=txt;
```



AJAX Request & jQuery

- ❖ `ajax(option)`: hàm gốc, đòi hỏi khai báo đầy đủ tham số của AJAX (get/post, url, data, sự kiện khi thành công/thất bại, loại dữ liệu trả về: txt, xml, json,...)
- ❖ `load(url, data, callback)`: hàm tạo request gửi lên server, sau đó lấy kết quả trả về gắn vào nội dung thẻ
- ❖ `get/post(url, data, callback, type)`: hàm tạo request với phương thức get/post



AJAX Request & jQuery

```
$("#btnCalc").bind("click", function() {  
    var n1 = $("#txtNumber1").attr("value");  
    var n2 = $("#txtNumber2").attr("value");  
    var serverURL = "AJAX.php?number1=" + n1 + "&number2=" + n2;  
    $("#divResult").load(serverURL);  
});
```

Number 1 <input id="txtNumber1" type="text" />

Number 2 <input id="txtNumber2" type="text" />

<input id="btnCalc" type="button" value="Calc"/>

Result is:

Hoàng Anh Tú



47

Tạo chuỗi tham số từ form

```
<form>  
<div><input type="text" name="a" value="1" id="a" /></div>  
<div><input type="text" name="b" value="2" id="b" /></div>  
<div><input type="hidden" name="c" value="3" id="c" /></div>  
<div>  
    <textarea name="d" rows="8" cols="40">4</textarea>  
</div>  
<div><select name="e">  
    <option value="5" selected="selected">5</option>  
    <option value="6">6</option>  
    <option value="7">7</option>  
</select></div>  
<div>  
    <input type="checkbox" name="f" value="8" id="f" />  
</div>  
<div>  
    <input type="submit" name="g" value="Submit" id="g" />  
</div>  
</form>
```

```
$('#form').submit(function() {  
    alert($(this).serialize());  
    return false;  
});
```

a=1&b=2&c=3&d=4&e=5

Hoàng Anh Tú



48

Một số hàm tạo request khác

Ajax Requests:

[jQuery.ajax\(options \)](#)

Returns: XMLHttpRequest

Load a remote page using an HTTP request.

[load\(url, data, callback \)](#)

Returns: jQuery

Load HTML from a remote file and inject it into the DOM.

[jQuery.get\(url, data, callback, type \)](#)

Returns: XMLHttpRequest

Load a remote page using an HTTP GET request.

[jQuerygetJSON\(url, data, callback \)](#)

Returns: XMLHttpRequest

Load JSON data using an HTTP GET request.

[jQuery.getScript\(url, callback \)](#)

Returns: XMLHttpRequest

Loads and executes a JavaScript file using an HTTP GET request.

[jQuery.post\(url, data, callback, type \)](#)

Returns: XMLHttpRequest

Load a remote page using an HTTP POST request.



Hoàng Anh Tú

49

Sự kiện trong AJAX

Ajax Events:

[ajaxComplete\(callback \)](#)

Returns: jQuery

Attach a function to be executed whenever an AJAX request completes. This is an [Ajax Event](#).

[ajaxError\(callback \)](#)

Returns: jQuery

Attach a function to be executed whenever an AJAX request fails. This is an [Ajax Event](#).

[ajaxSend\(callback \)](#)

Returns: jQuery

Attach a function to be executed before an AJAX request is sent. This is an [Ajax Event](#).

[ajaxStart\(callback \)](#)

Returns: jQuery

Attach a function to be executed whenever an AJAX request begins and there is none already active. This is an [Ajax Event](#).

[ajaxStop\(callback \)](#)

Returns: jQuery

Attach a function to be executed whenever all AJAX requests have ended. This is an [Ajax Event](#).

[ajaxSuccess\(callback \)](#)

Returns: jQuery

Attach a function to be executed whenever an AJAX request completes successfully. This is an [Ajax Event](#).



Hoàng Anh Tú

50

HIỆU ỨNG

Hoàng Anh Tú



51

Hiệu ứng trong jQuery

Basics

```
show()  
show( speed, callback )  
hide()  
hide( speed, callback )  
toggle()  
toggle( switch )  
toggle( speed, callback )
```

Sliding

```
slideDown( speed, callback )  
slideUp( speed, callback )  
slideToggle( speed, callback )
```

Custom

```
animate( params, duration, easing, callback )  
animate( params, options )  
stop( clearQueue, gotoEnd )
```

Fading

```
fadeIn( speed, callback )  
fadeOut( speed, callback )  
fadeTo( speed, opacity, callback )
```

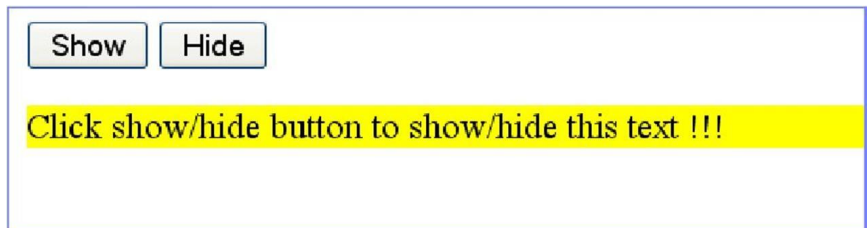
Hoàng Anh Tú



52

Sự kiện đơn giản

- `show()` `$(document).ready(function(){`
- `show(speed, callback)` `$("#btnShow").click(function (){`
`$("#p").show("slow");`
- `hide()` `$("#btnHide").click(function (){`
- `hide(speed, callback)` `$("#p").hide("slow");`
`});`
`});`



MỘT SỐ HÀM HỖ TRỢ XỬ LÝ

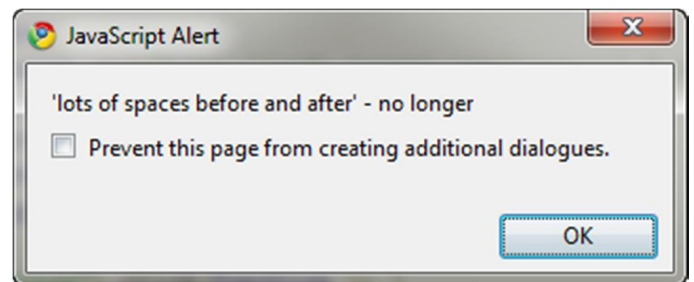
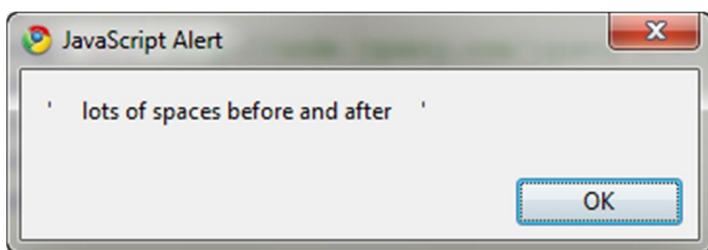


Hỗ trợ xử lý chuỗi

❖ trim(string): xóa khoảng trắng ở 2 đầu chuỗi

```
$("#button").click(function () {  
    var str = "    lots of spaces before and after    ";  
    alert("'' + str + '');
```

```
    str = jQuery.trim(str);  
    alert("'' + str + ' - no longer');  
});
```



Hoàng Anh Tú

55

Hàm hỗ trợ xử lý mảng

❖ each(): lặp qua từng phần tử của mảng

- \$.each()

```
0: 52    $.each([52, 97], function(index, value) {  
1: 97      alert(index + ':' + value);  
          });
```

- \$(selector).each()

```
<ul>  
    <li>foo</li>  
    <li>bar</li>  
</ul>
```

```
$('#li').each(function(index) {  
    alert(index + ':' + $(this).text());  
});
```

```
0: foo  
1: bar
```

Hoàng Anh Tú

56

❖ join(seperator): nối các phần tử của mảng lại bằng seperator

```
var arr= ["a", "b", "c", "d", "e"];  
$("div").text(arr.join(","));
```

a, b, c, d, e

❖ merge(array1, array2): nối mảng 2 vào mảng 1

```
$.merge([0,1,2],[3,4,5]);
```

❖ unique(array): loại bỏ các dòng trùng trong mảng



Hoàng Anh Tú

57

Thank You !

