

ASP.NET Core 2.0 - jQuery



Lương Trần Hy Hiển
hyhien@gmail.com
0989.366.990

 Visual Studio
2017 Enterprise
& C# 7

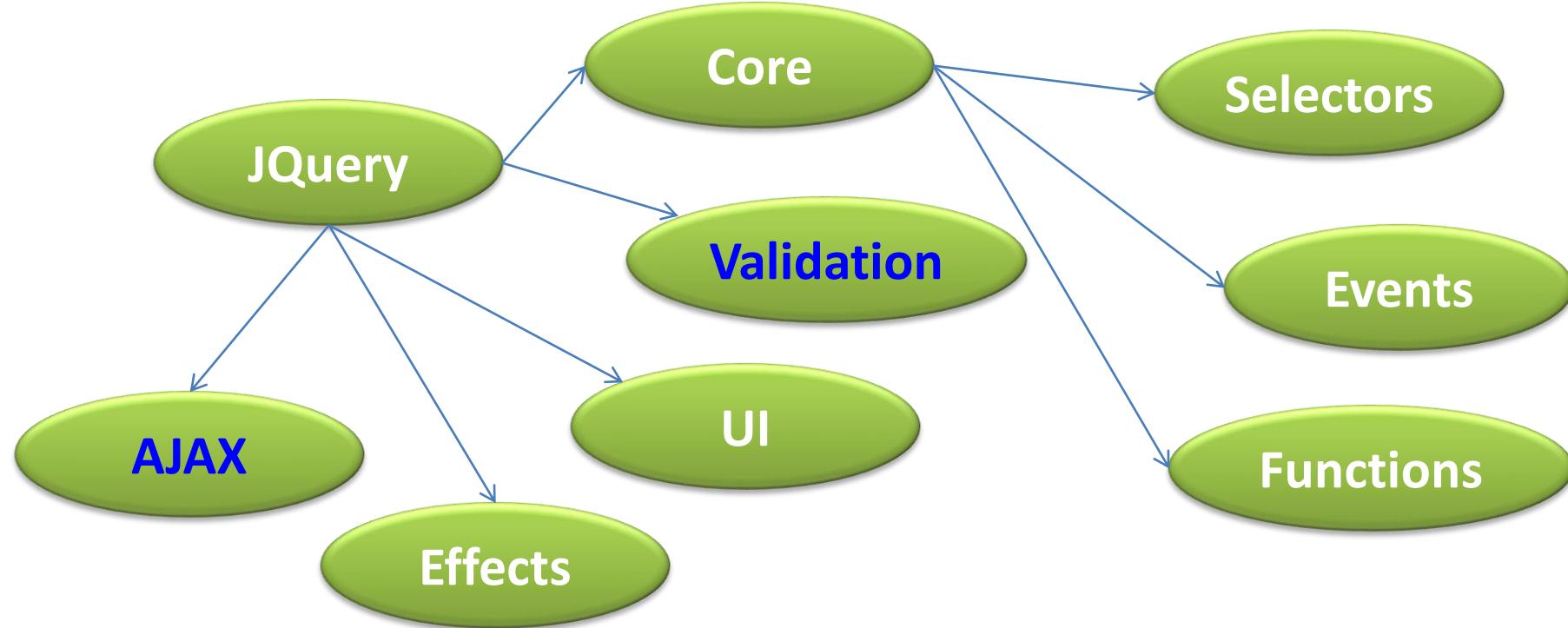
Nội dung

- ❑ Giới thiệu về JQuery
- ❑ JQuery Selector / JQuery Filter
- ❑ Thay đổi nội dung document
- ❑ Xử lý sự kiện
- ❑ Kiểm tra hợp lệ
- ❑ Ajax
- ❑ Giao diện
- ❑ Hiệu ứng & hoạt ảnh

Giới thiệu jQuery

- **jQuery** là thư viện được viết bằng ngôn ngữ JavaScript hỗ trợ cho việc thiết kế giao diện trên Web.
- **jQuery** được thiết kế để thay đổi cách lập trình ở phía client như Javascript truyền thống.
- **jQuery** cung cấp các tính năng:
 - Truy xuất tài liệu HTML (DOM) dễ dàng hơn.
 - Xử lý sự kiện javascript một cách dễ dàng.
 - Các control hỗ trợ trong quá trình thiết kế giao diện web.
 - Cung cấp các hiệu ứng trên web.
 - Đơn giản hóa việc sử dụng công nghệ AJAX.
 - Hỗ trợ các thư viện giao diện.

JQUERY STUDY PATH



Ví dụ mở đầu

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

<script type="text/javascript">

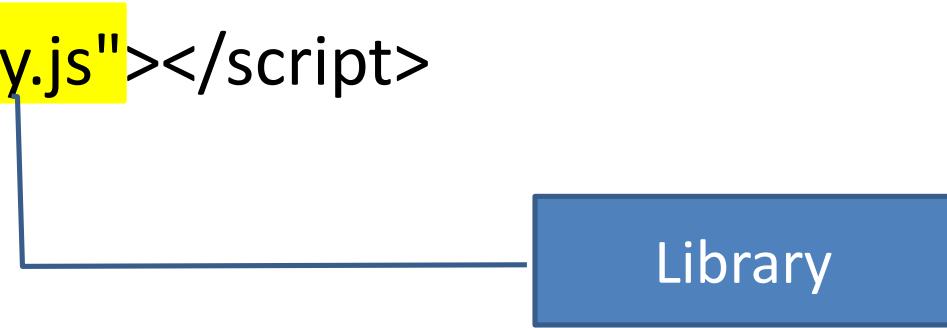
$(document).ready(function()
{
    $("#btnHello").click(function() {
        alert("Hello world !!!");
    });
});

</script>

</head>
<body>
    <h1> Jquery - Hello world !!! </h1>
    <hr />
    <input type="button" id="btnHello" value="Click me">
</body>
</html>
```

JQUERY

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
    $("p").click(function()
    {
        $(this).hide();
    });
});</script>
```



Click vào thẻ <p> nào thì ẩn thẻ <p> đó

Khởi động nhanh với Jquery

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

/*sau khi trang web được tải thành công thì sự kiện ready sẽ xảy ra*/
$(document).ready(function () {

    /*đăng ký sự kiện click cho tất cả các thẻ input có type='button'*/
    $(":button").click(function () {

        /*ẩn nút bị click vào kéo dài 1 giây--*/
        $(this).hide(1000, function () {
            /*hộp thoại thông báo sẽ được hiện ra sau khi nút được ẩn*/
            alert("Nút đã được ẩn");
        });

    }));
});

</script>
```

Bộ chọn (Selector)

- Dùng để chọn 1 hay nhiều thẻ HTML
- Một số cách lựa chọn:
 - `$(“*”)`: tất cả các thẻ
 - `$(“.class”)`: các thẻ với thuộc tính class
 - `$(“#id”)`: các thẻ với thuộc tính id
 - `$(“element”)`: các thẻ được chỉ định với tên thẻ
 - `$(“selector1, selector2, selectorN”)` : áp dụng cho nhiều thẻ được liệt kê.
 - `$(“element.class”)` hoặc `$(“element#id”)` các thẻ được chỉ định với tên thẻ và classname (hoặc id).

`$(“li.asp, #tag”)`

```
<script src="Scripts/jquery-1.4.1.min.js"></script>
<script type="text/javascript">
    $(document).ready(function () {
        /*kẻ đường bao (1px liền nét màu đỏ) cho tất cả thẻ <p>*/
        $("p").css("border", "1px solid red");
    });
</script>
```

Bộ chọn theo quan hệ phân cấp

SELECTOR	VÍ DỤ	Ý NGHĨA
Selector1,selector2...	<code>\$("#basic,.java")</code>	Chọn tất cả các thẻ được xác định bởi tất cả các bộ chọn
Parent > Child	<code> \$("body>.app")</code>	Chọn tất cả các thẻ con trực tiếp của các thẻ thuộc bộ chọn <Parent>
Ancestor Descendant	<code> \$("body .java")</code>	Chọn tất cả các thẻ con, cháu của các thẻ thuộc bộ chọn <Ancestor>
Prev + Next	<code> \$(".java+p")</code>	Chọn một thẻ nằm kế tiếp của cả các thẻ thuộc bộ chọn <Prev>
Prev ~ Siblings	<code> \$("p.java~p")</code>	Chọn tất cả các thẻ em của các thẻ thuộc bộ chọn <Prev>

`$("body > *)")`

`$("body>.java, #basic .app")`

`$("p.java+p")`

`$("p.java~p")`

Bộ chọn các phần tử form

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

Bộ chọn các phần tử form

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:

```
<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 lọc cơ bản

FILTER	Ý NGHĨA
:first	Chọn phần tử đầu tiên trong tập kết quả do Selector trả về
:last	Chọn phần tử cuối cùng 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

\$(".java:not(li:eq(0))")

ASP.NET Cre

Bộ lọc thuộc tính

FILTER	Ý NGHĨA
[attribute]	Lọc các phần tử có khai báo attribute
[attribute=value]	Lọc các phần tử có attribute với giá trị = value
[attribute!=value]	Lọc các phần tử có attribute với giá trị != value
[attribute^=value]	Lọc các phần tử có attribute với giá trị bắt đầu là value
[attribute\$=value]	Lọc các phần tử có attribute với giá trị kết thúc là value
[attribute*=value]	Lọc các phần tử có attribute chứa giá trị value
[attributeFilter1] [attributeFilter2]...	Lọc các phần tử thỏa tất cả các attribute filter.

Một số bộ lọc khác

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 chứa ít nhất 1 element thỏa selector
:parent	Lọc các phần tử có ít nhất 1 con hoặc text (Lấy các thẻ không rỗng)

FILTER	Ý NGHĨA
:visible	Lọc các phần tử có trạng thái là visible (đang hiển thị)
:hidden	Lọc các phần tử có trạng thái hidden (đang ẩn)

Bộ lọc con cháu

FILTER	Ý NGHĨA
:nth-child(index)	Lọc các phần tử theo vị trí so với cha của nó
:nth-child(even)	
:nth-child(odd)	
: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

❑ Ví dụ

- \$("li:nth-child(even)") - xác định các thẻ con tại vị trí chẵn
- \$("li:nth-child(even)") - xác định thẻ là con duy nhất
- \$("li:nth-child(2)") - xác định các thẻ con thứ 3
- \$("li:nth-child(3n+1)") – xác định các thẻ thứ 1, 4, 7, 10...

Xử lý tập kết quả

METHOD	Ý NGHĨA
<code>size()</code>	Lấy số phần tử trong tập kết quả của Selector
<code>get()</code>	Lấy tập DOM elements trong tập kết quả của Selector
<code>get(index)</code>	Lấy DOM element ở vị trí index
<code>find(expression)</code>	Lấy các element con cháu thỏa expression
<code>each()</code>	Gọi thực thi phương thức với từng element trong tập kết quả của Selector

Ví dụ:

- ✓ `$(".li.java").size()` – cho số phần tử `<li class='java'>`
- ✓ `$(".li").get(2)` – cho phần tử `` thứ 3
- ✓ `$(".ul").find("li.app")` – cho các `<li class='app'>` trong ``
- ✓ `$(".java").each(function(){alert($(this).html())})` – thông báo nội dung các thẻ có `class='java'`

Xử lý nội dung

METHOD	Ý NGHĨA	VÍ DỤ
html()	Lấy nội dung html bên trong element đầu tiên thỏa selector	HTML = \$("#basic").html();
html(newContent)	Thay đổi nội dung html bên trong mọi element thỏa selector (tương tự innerHTML trong DOM)	\$(".app").html("Hello");
text()	Lấy nội dung text bên trong element đầu tiên	Text = \$("#basic").text();
text(newTextContent)	Thay đổi nội dung text bên trong mọi element thỏa selector (tương tự innerText)	\$(".app").html("Hello");

Chèn và bổ sung

METHOD	Ý NGHĨA
append(content)	Chèn content vào sau nội dung có sẵn của các element thỏa selector
appendTo(selector)	Chèn element thỏa selector vào sau nội dung có sẵn của các element thỏa selector tham số
prepend(content)	Chèn content vào trước nội dung có sẵn của các element thỏa selector
prependTo(selector)	Chèn element thỏa selector vào trước nội dung có sẵn của các element thỏa selector tham số
after(content)	Chèn content vào sau các element thỏa selector
before (content)	Chèn content vào trước các element thỏa selector

Chèn và bổ sung

GỌI HÀM	TRƯỚC KHI CHÈN	SAU KHI CHÈN
<code>\$(".li").append("Subject");</code>	<code>Java</code>	<code>Java Subject</code>
<code>\$(".p:first").appendTo("li");</code>	<code><p>Subject</p> Java</code>	<code>Java<p>Subject</p></code>
<code>\$(".li").prepend("Subject: ");</code>	<code>Java</code>	<code> Subject : Java </code>
<code>\$(".p:first").prependTo("li");</code>	<code><p>Subject</p> Java</code>	<code><p>Subject</p>Java</code>
<code>\$(".li").after("Subject")</code>		<code>JavaSubject</code>
<code>\$(".li").before ("Subject")</code>		<code>SubjectJava</code>

Xử lý thuộc tính

METHOD	Ý NGHĨA	VÍ DỤ
attr(name)	Lấy attribute value của element đầu tiên thỏa selector	<code>v = \$("#txtId").attr("value")</code>
attr(name, value)	Thiết lập attribute cho mọi element thỏa selector	<code>\$("#txtId").attr("value", "Hello")</code>
attr(name, function)	Thiết lập giá trị attribute dựa trên 1 function với mọi element thỏa selector.	<code>\$("#txtId").attr("value", function(){ return "Hello"; });</code>
attr(properties)	Thiết lập tập attribute cho mọi element thỏa selector. Properties có dạng object-notation syntax.	<code>\$("#txtId").attr({"value": "Hello", "disabled": "true"})</code>
removeAttr(name)	Xóa attribute với mọi element	<code>\$("#txtId").removeAttr("disabled")</code>
val([value])	Viết tắt của attr("value") để thao tác thuộc tính các phần tử form	<code>\$("#txtId").val("Hello"); v=\$("#txtId").val();</code>

Xử lý CSS

METHOD	Ý NGHĨA	VÍ DỤ
css (name)	Lấy giá trị thuộc tính name của element đầu tiên thỏa selector	w = \$("img").css("width");
css (properties)	Thiết lập tập thuộc tính css đối với mọi element thỏa selector	\$(“div:first”).css({“text-align”:”center”, “background-image”:”url(bg.gif)”})
css (property,value)	Thiết lập giá trị 1 thuộc tính đối với mọi element thỏa selector	\$(“tr:odd”).css(“background-color”:”red”})

METHOD	Ý NGHĨA
addClass (class)	Thêm class vào các element thỏa selector
hasClass (class)	Kiểm tra class có tồn tại trong các element thỏa selector
removeClass(class)	Xóa class khỏi các element thỏa selector
toggleClass (class)	Thêm class vào các element thỏa selector nếu class chưa khai báo, ngược lại nếu đã tồn tại rồi, class sẽ bị xóa

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)	Đè 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)	Đè 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 Events

- \$(selector).blur()
- \$(selector).change()
- \$(selector).focus()
- \$(selector).select()
- \$(selector).submit()

Keyboard Events

- \$(selector).focusin()
- \$(selector).focusout()
- \$(selector).keydown()
- \$(selector).keypress()
- \$(selector).keyup()

Hiệu ứng ẩn/hiện

METHOD	Ý NGHĨA	VÍ DỤ
show ()	Hiển thị các element thỏa selector nếu trước đó bị ẩn.	
show(speed,callback)	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>\$("#div1").show("normal", function(){alert("Hello")});</code>
hide ()	Ẩn element nếu trước đó đang hiển thị.	
hide(speed,callback)	Ẩn element nếu trước đó đang hiển thị, tham số có ý nghĩa tương tự phương thức show.	<code>\$("#div1").hide("slow");</code> <code>\$("#div1").hide(4000);</code>
toggle()	Chuyển qua lại trạng thái ẩn/hiện các element.	
toggle(speed,callback)	Chuyển qua lại trạng thái ẩn/hiện các element, tham số có ý nghĩa tương tự phương thức show.	<code>\$("#div1").toggle("fast");</code>

Hiệu ứng fade và slide

METHOD	Ý NGHĨA	VÍ DỤ
fadeIn(speed,callback)	Hiển thị element bằng cách tăng dần độ trong suốt.	\$("#div1").fadeIn("normal");
fadeOut(speed,callback)	Ẩn element bằng cách giảm dần độ trong suốt về 0, sau đó thiết lập style display là none.	\$("#div1").fadeOut("slow");
fadeTo(speed,opacity, callback)	Thay đổi độ trong suốt của element.	\$("#div1").fadeTo("slow",0.3, function() {alert("finished");}); \$("#div1").fadeTo("slow",1.0);

METHOD	Ý NGHĨA	VÍ DỤ
slideDown(speed, callback)	Hiển thị element bằng cách tăng chiều cao.	\$("#div1").slideDown("slow");
slideUp(speed, callback)	Ẩn element bằng cách giảm chiều cao.	\$("#div1").slideUp("normal");
slideToggle(speed, callback)	Chuyển đổi trạng thái ẩn/hiện element.	\$("#div1").slideToggle(3000);

Hiệu ứng tùy biến

THAM SỐ	Ý NGHĨA	VÍ DỤ
properties	Các thuộc tính xác định trạng thái hiển thị sau khi animate.	{width:"100", color:"red"}
duration	Animate kéo dài trong bao lâu ("slow", "normal", "fast", milisecond)	1000
easing	Hiệu ứng xoa : swing, linear	swing
Callback	Hàm được gọi sau khi animate xong	function(){alert("Hello")}

```
$("#button_growright").click(function () {
    $("#div1").animate({ width: "800" }, "normal");
});
```

Giao diện (Jquery UI)

- Datepicker
- Accordion
- Tab
- Dialog

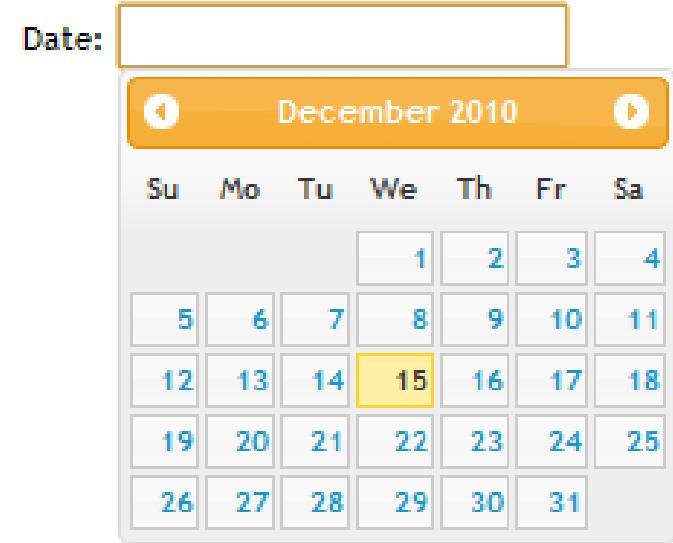
```
// phần thư viện lõi thường dùng trước đây  
  
// css cần thiết cho các thành phần giao diện  
  
// thư viện xử lý các thành phần giao diện  

```

UI-Datepicker

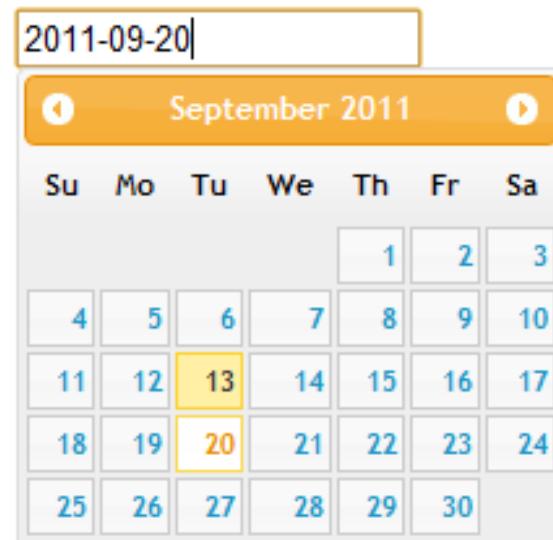
```
<script>
$(function() {
    $("#datepicker").datepicker();
});
</script>
```

```
<div class="demo">
<p>Date: <input type="text" id="datepicker"></p>
</div>
```



UI-Datepicker

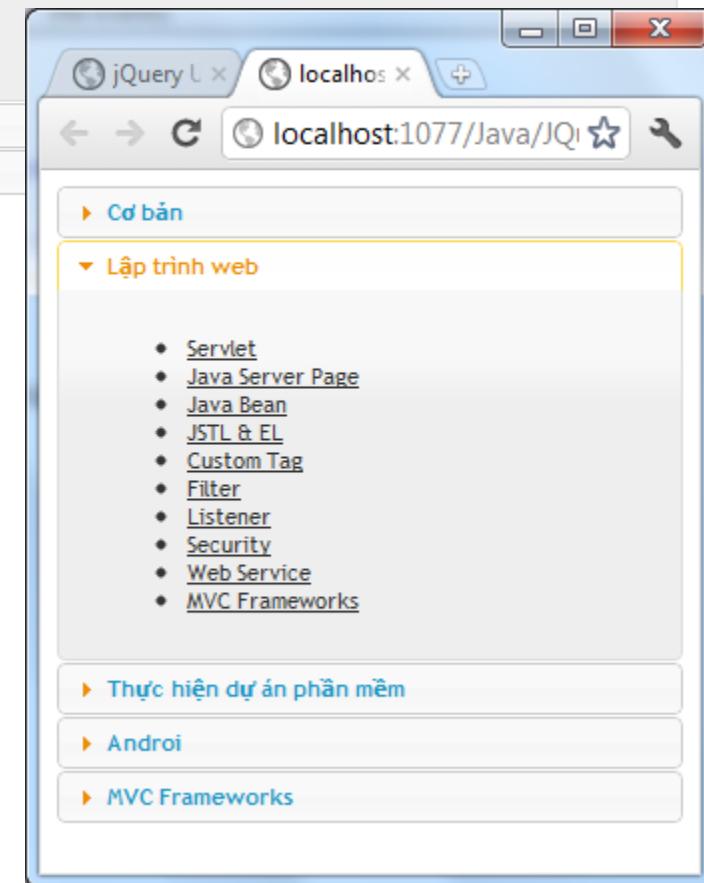
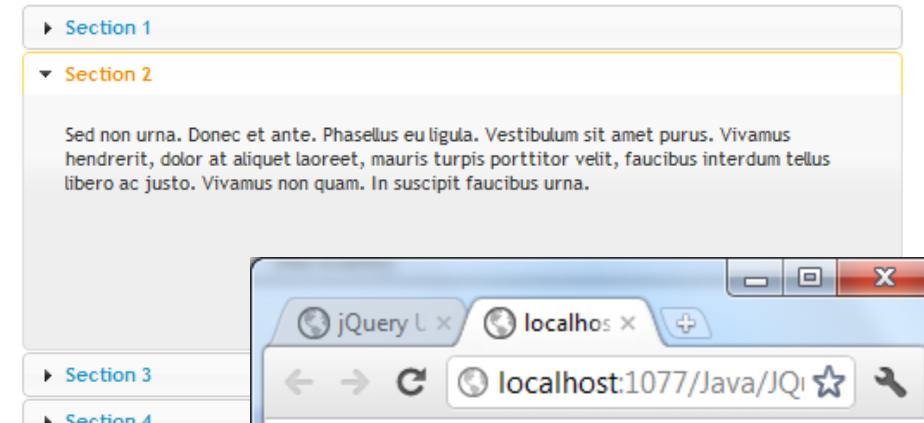
```
<head>
    <link href="jquery-ui-1.8.7.custom.css" rel="stylesheet"/>
    <script src="jquery-1.4.4.min.js"></script>
    <script src="jquery-ui-1.8.7.custom.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $("#birthday, #joined").datepicker({ dateFormat: 'yy-mm-dd' });
        });
    </script>
    <style type="text/css">
        body{font-size: 11px;}
    </style>
</head>
<body>
    Date of birth: <input id="birthday" />
    Date of registration: <input id="joined" />
</body>
```



UI-Accordion

```
<script>
$(function() {
    $( "#accordion" ).accordion();
});
</script>
```

```
<div id="accordion">
    <h3><a href="#">First header</a></h3>
    <div>First content</div>
    <h3><a href="#">Second header</a></h3>
    <div>Second content</div>
</div>
```



UI-AutoComplete

```
<script>
$(function() {
    $("#tags").autocomplete
    ({
        source: ["ActionScript", "AppleScript", "Java", "JavaScript","Lisp",
        "Perl","PHP","Python", "Ruby","Scala","Scheme"]
    });
}>
</script>
```

```
<div class="demo">
    <div class="ui-widget">
        <label for="tags">Tags: </label>
        <input id="tags">
    </div>
</div>
```

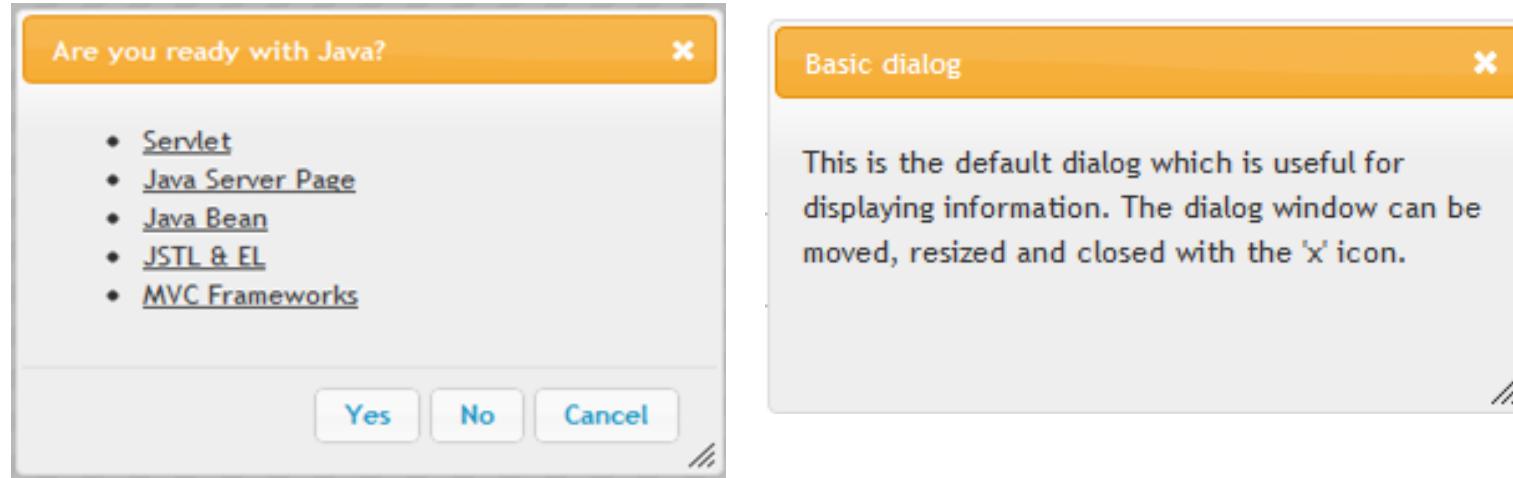
Tags: a

- ActionScript
- AppleScript
- Asp
- BASIC
- Erlang
- Fortran
- Haskell
- Java
- JavaScript
- Scala

UI-Dialog

```
<script>
$(function() {
    $( "#dialog" ).dialog();
});
</script>
```

```
<div class="demo">
    <div id="dialog" title="Basic dialog">
        <p>This is the default dialog which is useful for displaying information. The dialog window can be moved, resized and closed with the 'x' icon.</p>
    </div>
</div>
```



UI-TABS

```
<script>
$(function() {
    $("#tabs").tabs();
});
</script>
```

```
<div class="demo"><div id="tabs">
<ul>
    <li><a href="#tabs-1">Nunc tincidunt</a></li>
    <li><a href="#tabs-2">Proin dolor</a></li>
</ul>
<div id="tabs-1">
    <p>Proin elit arcu, rutrum commodo, vehicula tempus</p>
</div>
<div id="tabs-2">
    <p>Morbi tincidunt, dui sit amet facilisis feugiat</p>
</div></div></div>
```

Nunc tincidunt Proin dolor Aenean lacinia

Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, ut pharetra massa metus id nunc. Duis scelerisque molestie turpis. Sed fringilla, massa eget luctus malesuada, metus eros molestie lectus, ut tempus eros massa ut dolor. Aenean aliquet fringilla sem. Suspendisse sed ligula in ligula suscipit aliquam. Praesent in eros vestibulum mi adipiscing adipiscing. Morbi facilisis. Curabitur ornare consequat nunc. Aenean vel metus. Ut posuere viverra nulla. Aliquam erat volutpat. Pellentesque convallis. Maecenas feugiat, tellus pellentesque pretium posuere, felis lorem euismod felis, eu ornare leo nisi vel felis. Mauris consectetur tortor et purus.

UI-Sortable

```
<script>
$(function() {
    $("#sortable").sortable();
    $("#sortable").disableSelection();
});
</script>
```

```
<div class="demo">
<ul id="sortable">
<li class="ui-state-default">
    <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 1</li>
<li class="ui-state-default">
    <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 2</li>
<li class="ui-state-default">
    <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>Item 3</li>
</ul>
</div>
```



UI-Selectable

```
<style>
#selectable .ui-selecting { background: #FECA40; }
#selectable .ui-selected { background: #F39814; color: white; }
#selectable { list-style-type: none; margin: 0; padding: 0; width: 60%; }
#selectable li { margin: 3px; padding: 0.4em; font-size: 1.4em; height: 18px; }
</style>
```

```
<script>
$(function() {
    $("#selectable").selectable();
});
</script>
```

```
<div class="demo">
<ol id="selectable">
    <li class="ui-widget-content">Item 1</li>
    <li class="ui-widget-content">Item 2</li>
    <li class="ui-widget-content">Item 3</li>
</ol>
</div>
```



Ajax

```
<script src="JQuery/js/jquery-1.4.1.min.js"></script>

<script>

    $.ajax({

        url: "Ajax.jsp",
        data: "name=John&location=Boston",
        success: function (responseText) {
            alert("Đã lưu dữ liệu thành công: " + responseText);
        }
    });
</script>
```

Q&A

