



ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

Môn: Phát triển ứng dụng Web

Seminar 01:

THƯ VIỆN JQUERY

Thư viện jQuery

ThS. Ngô Bá Nam Phương - Nguyễn Đức Huy
{nbnphuong, ndhuy}@fit.hcmus.edu.vn

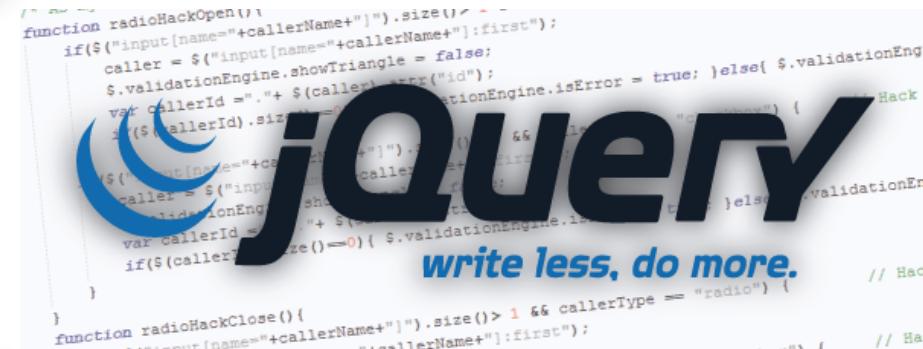
Nội dung





Giới thiệu

Thư viện jQuery



Thư viện jQuery



- Thư viện javascript mã nguồn mở, miễn phí
- Tạo các trang web có khả năng tương tác cao và tương thích trên nhiều trình duyệt.



- Truy xuất các thành phần nội dung trang web với cú pháp tương tự **CSS** (thông qua các bộ chọn selector).
- Hỗ trợ nhiều thao tác xử lý trên tập các element chỉ bằng một dòng lệnh (statement chaining).
`$(“selector”).func1().func2().func3()...;`
- Đơn giản hóa cách viết mã nguồn javascript (***write less, do more***). Tách biệt mã xử lý javascript và thành phần thể hiện HTML.



Cài đặt



- Download: <http://jquery.com/>
- Version mới nhất: **v1.11.0 hoặc v2.1.0**
- Có 2 version:
 - Production (triển khai lên host thật)
 - Development (dùng trong quá trình phát triển, hỗ trợ debug, ..)



Sử dụng JQuery (sự kiện **onload**)



- Xử lý sự kiện **onload** khởi tạo các thành phần trong trang.
- Cách tiếp cận Javascript truyền thống:

```
function onloadHandler()
{
    alert("run after all page contents have been
downloaded, including image");
}
window.onload = onloadHandler;
```
- Với JQuery, hàm xử lý sự kiện onload sẽ gọi ngay sau khi DOM của document đã nạp xong.



Sử dụng JQuery (sự kiện **onload**)



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

```
{
```

```
    alert("hello world");
```

```
}
```

```
);
```

- `$("document").ready` có thể được gọi nhiều lần, các hàm XL sự kiện sẽ được gọi theo thứ tự nó được đăng ký.



Sử dụng JQuery (sự kiện onload)



```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>Untitled Page</title>
    <!-- khai báo sử dụng thư viện jquery -->
    <script type="text/javascript" src="jquery-1.3.2.js" ></script>
    <!-- xử lý sự kiện onload -->
    <script type="text/javascript" >
        $(document).ready( function ()
        {
            alert("hello world");
        });
    </script>
</head>
```



- **Core functionality**: các phương thức core của JQuery và các hàm tiện ích được sử dụng thường xuyên.
- **Selector & Traversal**: chọn, tìm kiếm element, duyệt qua các element trong document.
- **Manipulation & CSS**: thay đổi nội dung các element trong document, làm việc với css.



Các thành phần trong JQuery

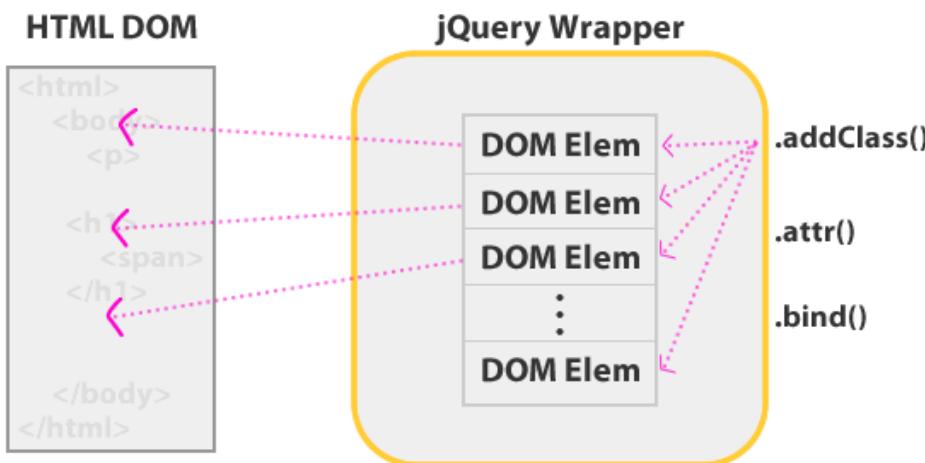


- **Event**: đơn giản hóa việc xử lý event. Cung cấp event helper function đăng ký nhanh các event.
- **Effect & Animation**: cung cấp các hàm hỗ trợ tạo animation & effect.
- **Ajax**: cung cấp các hàm hỗ trợ Ajax
- **User interface**: tập widget với các control: accordion, datepicker, dialog, progressbar, slider, tab
- **Extensibility**: hỗ trợ tạo plugin bổ sung thêm các chức năng mới vào core library.





jQuery Selector



Thư viện jQuery

- Truy xuất nội dung (element) trong document dựa trên biểu thức selector cung cấp. Selector sử dụng cú pháp tương tự CSS.
- Tập kết quả do Selector và Filter trả về: JQuery objects (không phải DOM objects).



● Cú pháp và cách chọn tương tự CSS

SELECTOR	Ý NGHĨA
TAGNAME	Chọn tất cả các element có tên là TAGNAME
#IDENTIFIER	Chọn tất cả các element có ID là IDENTIFIER
.className	Chọn tất cả các element với thuộc tính class có giá trị là className
Tag.className	Chọn tất cả các element thuộc loại Tag, với thuộc tính class có giá trị là className
*	Chọn tất cả các element trên document.



JQuery Selector



Ví dụ:

```
<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>
```

```
<script type="text/javascript">
    $("document").ready(function () {
        $("p").css("border", "1px solid red");
    });
</script>
```

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

this is paragraph 1

this is paragraph 2

this is paragraph 3

this is paragraph 4

JQuery Selector



Ví dụ:

```
<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>this is paragraph 4</p>
```

this is paragraph 1

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

this is paragraph 2

```
    $( "#para2" ).css("border", "1px solid red"); this is paragraph 3
```

this is paragraph 4

```
});
```

JQuery Selector



Ví dụ:

```
<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 () {
    $("li.a").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

- Chọn element dựa trên mối quan hệ phân cấp giữa các element

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 Child element là con trực tiếp của Parent
Ancestor Descendant	Chọn tất cả các Descendant element là con cháu của Ancestor (chứa bên trong Ancestor)
Prev + Next	Chọn tất cả các Next element nằm kế tiếp Prev element
Prev ~ Siblings	Chọn tất cả các element anh em khai báo sau Prev và thỏa Sibling selector

JQuery Selector



Ví dụ:

```
<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 () {

  $("li.a,p.a,p#para2").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

JQuery Selector



Ví dụ:

```
<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 () {
  $(".a,.b").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

JQuery Selector



Ví dụ:

```
<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>
    Hello
    <a href="#">world</a>
    <span>
        <a href="#">2009</a>
    </span>
</p>

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

• item 1
• item 2
• item 3
• item 3

Hello world 2009



JQuery Selector



Ví dụ:

```
<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>
    Hello
    <a href="#">world</a>
    <span>
        <a href="#">2009</a>
    </span>
</p>

$( "document" ).ready(function () {
    $( "ul li.b, p a" ).css("border", "1px solid red");
});
```

The code illustrates the use of jQuery selectors. It defines an

 element with two items, both of which have the class "a". It also defines a

element containing a greeting and two elements, one of which is wrapped in a .

The jQuery code uses the `$("document").ready(function () { ... })` pattern to ensure the code runs after the DOM is fully loaded. Inside, it uses the selector `$("ul li.b, p a")` to select all - elements with class "b" and all elements within the

element.

styling: The selected elements have a border of 1px solid red applied to them.

JQuery Selector



Ví dụ:

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

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

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

Hello [world](#) [2009](#) [2010](#)

abc

JQuery Selector



Ví dụ:

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

```
$( "document" ).ready(function () {
    $("#link1 ~ span").css("border", "1px solid red");
```

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

Hello world 2009 2010

abc



Form Selector



BỘ LỌC	Ý NGHĨA
:input	Chọn tất cả thẻ input, textarea trên Form
:text	Chọn tất cả text field trên Form
:password	Chọn tất cả password field
:radio	Chọn tất cả radio button
:checkbox	Chọn tất cả checkbox
:submit	Chọn tất cả button submit
:reset	Chọn tất cả button reset
:image	Chọn tất cả image
:button	Chọn tất cả generalized button
:file	Chọn tất cả control upload file



Form Selector



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

First Name

Last Name

Disabled Text Field

Gender

 M F

What products are you interested in?

- Widgets
- Hibbity Jibbities
- SplashBangers
- Whatzits

Comments:

Optional life story file

Form Selector



```
$( "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?

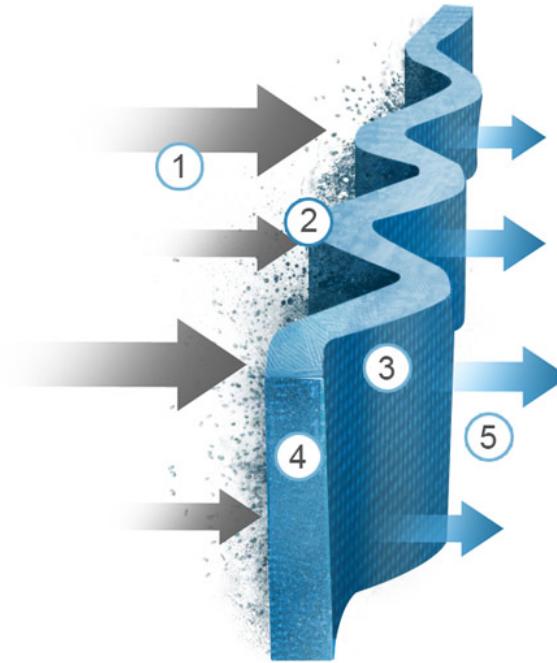
Comments:

Optional life story file



jQuery Filter



Thư viện jQuery



- 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 element
 - Attribute: lọc dựa trên thuộc tính của element
 - Child: lọc dựa trên mối QH với element cha
 - Form: lọc trên các thành phần khai báo trên Form

Basic JQuery Filter



Bộ lọc	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

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>
```

- 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:odd").css("border", "1px solid red");
});
```



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>
```

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



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>
```

- 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:eq(1)" ).css("border", "1px solid red");
    $( "p:eq(3)" ).css("border", "1px solid red");
});
```

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

$( "document" ).ready(function () {
    $("p:not(p:eq(1))" ).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



Attribute Filter



BỘ LỌC	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.



Attribute 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>
```

- 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[class]").css("border", "1px solid red");
});
```



Attribute 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>
```

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

Attribute 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 () {
    $("p[id^=para]").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

Attribute 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" lang="en-us" >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[class=a] [lang*=us]").css("border", "1px solid red");
});
```

Content & Visibility Filter



BỘ LỌC THEO NỘI DUNG	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 là cha (chứa ít nhất 1 element khác hoặc text)

BỘ LỌC THEO TRẠNG THÁI HIỂN THỊ	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)

Content 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" lang="en-us" >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:contains(2)" ) . css ("border", "1px solid red");
}) ;
```

Content 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" lang="en-us" >this is paragraph 4</p>
<p></p>
<div></div>
```

- 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:empty" ).css("border", "1px solid red");
  $( "div:empty" ).css("border", "1px solid red");
});
```



Content Filter



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

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

Hello world 2009 2010

abc

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



Content Filter



```
<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>
```

Hello world 2009 2010

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

Child Filter



FILTER	PURPOSE
:nth-child(<i>index</i>) :nth-child(even) :nth-child(odd) :nth-child(<i>equation</i>)	Matches elements at <i>index</i> , or even or odd increments, or who match an equation of the form $Xn+M$ (for example, $2n$ or $3n+1$)
:first-child	Matches elements who are the first child of their parent
:last-child	Matches elements who are the last child of their parent
:only-child	Matches elements who are the only child of their parent



Child Filter



BỘ LỌC	Ý NGHĨA
:nth-child(index) :nth-child(even) :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



Child 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>
```

```
$( "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");
});
```

- item 1

- item 2

- item 3

- item 3

- item 1

- item 2

- item 3

- item 3





Thay đổi Nội dung Document

Thư viện jQuery 



Duyệt danh sách các element trong document



BỘ LỌC	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



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 innerHTML = $("p").get(i).innerHTML;
}
```



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>
```

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

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



- Phương thức **\$(“html content”)**, kết quả trả về là 1 JQuery object.

Ví dụ:

```
var h1 = $("<h1>heading 1</h1>"); // tạo thẻ h1
với nội dung
var temp = "<h1>heading 1</h1>";
var newH1 = $(temp); // tạo thẻ h1 từ biến
$("p:eq(0)").html(newH1);
```



Truy cập, thay đổi nội dung trong element

jQuery
write less, do more.

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



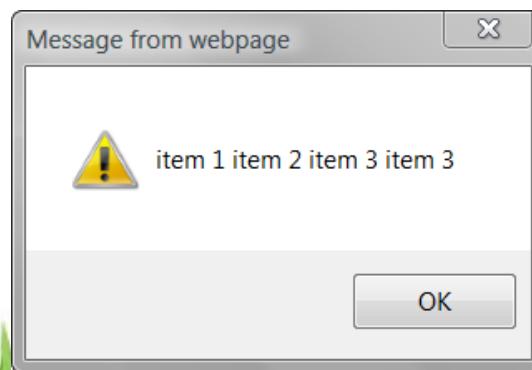
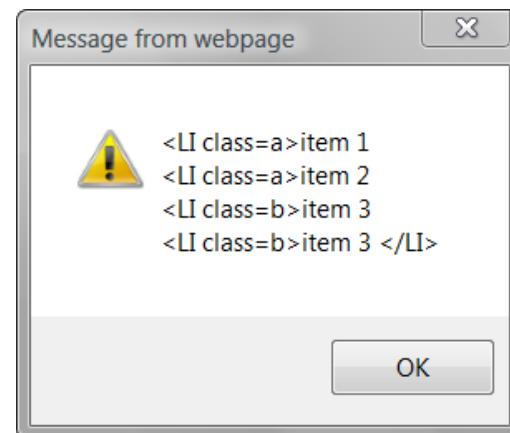
Truy cập, thay đổi nội dung trong element

jQuery
write less, do more.

```
<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());
```



Truy cập, thay đổi nội dung trong element

jQuery
write less, do more.

```
var h1 = $("<h1>heading1</h1>");  
$("#p1").html(h1);  
var img1 = $("<img src='book2.jpg' />");  
$("#p1").html(img1);  
  
$("p:last").text("new content");
```



Thay đổi giá trị thuộc tính



Phương thức	Ý nghĩa
attr(name)	Lấy attribute value của element đầu tiên thỏa selector
attr(properties)	Thiết lập tập attribute cho mọi element thỏa selector. Properties có dạng object-notation syntax.
attr(key, value)	Thiết lập attribute cho mọi element thỏa selector
attr(key, function)	Thiết lập giá trị attribute dựa trên 1 function với mọi element thỏa selector.
removeAttr(name)	Xóa attribute với mọi element



Thay đổi giá trị thuộc tính – ví dụ



```
<a href="trang1.html">Trang 1</a>
```

```
 $("a").attr("href","trang2.html");
```

```
 $("a").text("trang 2");
```

```
<a href="book1.jpg">  
    
</a>
```

```
 $("a").attr("target","_blank");
```

```
 $("a img").attr("src","book2.jpg");
```

```
 $("a").removeAttr("href");
```

```
 $("img").attr( {src:"book2.jpg",alt:"hello world"} );
```



Chèn nội dung



Phương thức	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 nội dung



```
<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>
```

```
$( "li" ).append( "<b>new content</b>" );
```

- item 1 **new content**
- item 2 **new content**
- item 3 **new content**
- item 3 **new content**



Chèn nội dung



```
§ ("p#para2") .appendTo ("li.a");  
• item 1  
this is paragraph 2  
• item 2  
this is paragraph 2  
• item 1  
• item 2  
• this is paragraph 3 • item 3  
item 3 this is paragraph 1  
• this is paragraph 3 this is paragraph 3  
item 3 this is paragraph 4  
this is paragraph 1  
this is paragraph 2  
...  
§ ("p.b") .prependTo ("li.b");
```



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
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



Thay đổi vị trí element



CSS FUNCTIONS	PURPOSE
offset()	Gets the current offset of the first matched element, in pixels, relative to the document
offsetParent()	Returns a jQuery collection with the positioned parent of the first matched element
position()	Gets the top and left position of an element relative to its offset parent
scrollTop()	Gets the scroll top offset of the first matched element
scrollTop(val)	Sets the scroll top offset to the given value on all matched elements
scrollLeft()	Gets the scroll left offset of the first matched element
scrollLeft(val)	Sets the scroll left offset to the given value on all matched elements



Thay đổi vị trí element



Phương thức	Ý nghĩa
offset ()	Lấy vị trí của element đầu tiên thỏa selector so với document
position ()	Lấy vị trí của element đầu tiên thỏa selector so với vị trí của element cha
scrollTop()	Lấy vị trí scroll top của element đầu tiên thỏa selector
scrollLeft()	Lấy vị trí scroll left của element đầu tiên thỏa selector
scrollTop(value)	Thiết lập vị trí scroll top của mọi element thỏa selector
scrollLeft(value)	Thiết lập vị trí scroll left của mọi element thỏa selector



Thay đổi kích thước



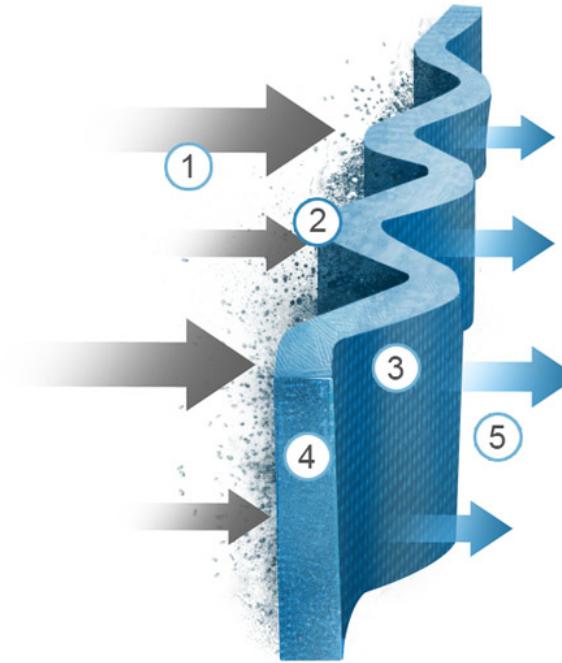
Phương thức	Ý nghĩa
height()	Lấy chiều cao của element đầu tiên thỏa selector
width ()	Lấy chiều rộng của element đầu tiên thỏa selector
height(val)	Thiết lập chiều cao của mọi element thỏa selector
width (val)	Thiết lập chiều rộng của mọi element thỏa selector





jQuery

Sự kiện



Thư viện jQuery



Xử lý sự kiện



Binding/Unbinding

Allows events to be wired up and torn down in a cross-browser way

Unified Event Object

Provides an event object that exposes the most common properties in a cross-browser way

Convenience Features

Provides functions that encapsulate common event features and cross-browser helper routines



```
$(selector).bind(event,data,handler)
$(selector).unbind(event,handler)
```

BIND() PARAMETER	PURPOSE
event	Defines the event that you want to be bound to for each element in the selector's result set. Possible values are blur, focus, load, resize, scroll, unload, beforeunload, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, change, select, submit, keydown, keypress, keyup, error
data	Optional. Defines a piece of data that will be passed to the handler function when the event happens and the handler function is called
handler	Specifies the function that will handle the event

UNBIND() PARAMETER	PURPOSE
event	Defines the event that you want to be disconnected for each element in the selector's result set
handler	Specifies the handler function that was defined to handle the event



Xử lý sự kiện



`$("selector").bind(event,[data],[handler]);`

Tham số	Ý nghĩa
Event	Sự kiện selector xử lý, bao gồm: load, blur, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, submit, keydown, keypress, keyup, ..
Data	Tùy chọn, dữ liệu truyền vào handler khi event xảy ra
Handler	Tên hàm xử lý sự kiện

`$("selector").unbind(event,data,handler);`



Xử lý sự kiện – ví dụ



```
$("div").bind("mouseover",highLight);
$("div").bind("mouseleave",highLight);
$("div").bind("click", function () {
    $("div").unbind("mouseover",highLight);
    $("div").unbind("mouseleave",highLight);
    $("div").html("<p style='color:green;'>turn off</p>");
})
```

```
function highLight(evt)
{
    $("div").toggleClass("highlight");
}
```



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 Func2: hàm xử lý được gọi khi mouse di chuyển ra khỏi selector

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



Đối tượng Event



- 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);
});
```

Ví dụ tổng hợp về event



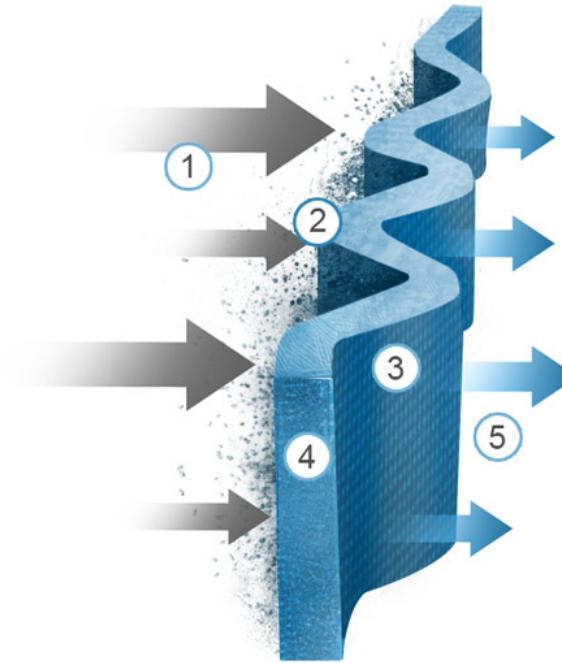
```
$("#theList tr:even").addClass("stripe1");
$("#theList tr:odd").addClass("stripe2");
$("#theList tr").hover(
    function (evt) {
        $(this).toggleClass("highlight");
    },
    function (evt) {
        $(this).toggleClass("highlight");
    }
);
```

Item	Price
Milk	1.99
Eggs	2.29
Butter	3.49
Bread	0.99
Pasta	1.19
Honey	4.39
Cookies	2.99
Apples	0.59
Sugar	1.78
Pepper	1.56



6

jQuery Animation



Thư viện jQuery



- Ẩn, hiện element
- Fade-in, fade-out
- Sliding
- Di chuyển element
- Custom animation effect



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

Phương thức	Ý nghĩa
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.
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.
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.

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");
```



Fade in/fade out



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



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);  
});
```



Sliding



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



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);  
});
```



Custom Animation



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

Tham số	Ý nghĩa
properties	Các thuộc tính xác định 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();
```



Custom Animation



```
$("#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"  
);  
});
```





Thư viện jQuery



Câu hỏi ?