




TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

ỨNG DỤNG WEB

Nguyễn Thị Mai Trang

1




TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Mục tiêu

- Trình bày được hiệu quả của jQuery trong xử lý trang web.
- Khai báo và sử dụng được jQuery trong thiết kế trang web.
- Sử dụng được bộ chọn CSS để truy cập các phần tử trên trang web theo cú pháp jQuery.
- Sử dụng được cú pháp trong jQuery để thao tác với các thành phần trong trang web

3




TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Chương 5

jQuery

2




TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Nội dung

- Giới thiệu jQuery
- Xử lý sự kiện
- Làm việc với phần tử trong DOM
- Duyệt cây DOM
- Thiết lập hiệu ứng với jQuery


4


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
 Cơ hội học tập cho mọi người

5.1 Giới thiệu jQuery

- Giới thiệu jQuery
- Download và sử dụng
- Chọn phần tử trong jQuery

5


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
 Cơ hội học tập cho mọi người


5.1.2 Download và sử dụng

- Khai báo jQuery vào trang web có hai cách
 - **Sử dụng jQuery CDN (Content Delivery Network)** trực tuyến:
 - Google CDN


```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
</head>
```
 - Microsoft CDN


```
<head>
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js"></script>
</head>
```


7


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
 Cơ hội học tập cho mọi người

5.1.1 Giới thiệu jQuery

- jQuery là một thư viện các hàm JavaScript
- Giúp đơn giản hóa mã lệnh JS.
- Giúp thao tác với mô hình DOM đơn giản và linh hoạt hơn.
- Thư viện jQuery cho phép thao tác với:
 - HTML/DOM
 - CSS
 - Xử lý sự kiện trong HTML
 - Hiệu ứng, đa phương tiện
 - AJAX
 - Utilities

6


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
 Cơ hội học tập cho mọi người

Download và sử dụng (tt)

- Khai báo jQuery (tt)
 - Download thư viện jQuery từ jquery.com, tham chiếu đến tập tin trong phần tử **<head></head>**

```
<head>
<script src="jquery-3.3.1.min.js"></script>
</head>
```

8

5.1.3 Chọn phần tử trong jQuery

- **jQuery Selectors:** là bộ chọn của jQuery, được sử dụng để chọn một hoặc nhiều phần tử HTML.
- **Cú pháp: \$(selector).action()**
 - \$(): chọn một phần tử.
 - selector: phần tử được chọn theo cú pháp CSS.
 - action(): hành động được thực hiện trên phần tử selector
- Ví dụ:
 - \$(this).hide(): ẩn phần tử hiện hành.
 - \$("p").hide(): ẩn tất cả các phần tử <p>
 - \$(".test").hide(): ẩn tất cả các phần tử có class="test".
 - \$("#test").hide(): ẩn tất cả các phần tử có id="test".

9

9

Chọn phần tử trong jQuery (tt)

- Ví dụ một số bộ chọn jQuery (tt)

Cú pháp	Mô tả
\$("#li:first-child")	Chọn phần tử đầu tiên của mỗi phần tử
\$("#li:eq(3)")	Chọn phần tử thứ 4 trong danh sách ul (phần tử đầu có chỉ số (index) = 0)
\$("#li:gt(3)")	Chọn các phần tử có chỉ số (index) lớn hơn 3
\$("#li:lt(3)")	Chọn các phần tử có chỉ số (index) nhỏ hơn 3
\$("#:header")	Chọn tất cả các phần tử <h1>, <h2>,...

11

11

Chọn phần tử trong jQuery (tt)

- Ví dụ một số bộ chọn jQuery

Cú pháp	Mô tả
\$("#*")	Chọn tất cả
\$("#p")	Chọn tất cả các phần tử <p>
\$("#p.intro")	Chọn tất cả phần tử <p> có class="intro"
\$("#p#intro")	Chọn tất cả phần tử <p> có id="intro"
\$(this)	Chọn phần tử hiện hành
\$("#[href]")	Chọn các phần tử có thuộc tính href
\$("#a[target='_blank']")	Chọn các phần tử <a> có thuộc tính target = "_blank"
\$("#a[target!='_blank']")	Chọn các phần tử <a> có thuộc tính target != "_blank"

10

10


Chọn phần tử trong jQuery (tt)

- Ví dụ một số bộ chọn jQuery (tt)

Cú pháp	Mô tả
\$("#p:first")	Chọn phần tử <p> đầu tiên
\$("#p:last")	Chọn phần tử <p> cuối cùng
\$("#tr:even")	Chọn tất cả các phần tử <tr> ở dòng chẵn
\$("#tr:odd")	Chọn tất cả các phần tử <tr> ở dòng lẻ
\$("#p:only-child")	Chọn tất cả các phần tử <p> là con duy nhất của phần tử khác
\$("#p:only-of-type")	Chọn tất cả các phần tử là con duy nhất của phần tử khác trong nhóm các phần tử p

12

12




Trường Đại học Mở TP. HCM
Cơ hội học tập cho mọi người

Chọn phần tử trong jQuery (tt)

- Ví dụ một số bộ chọn jQuery (tt)

Cú pháp	Mô tả
\$("#:parent")	Chọn các phần tử có chứa phần tử khác
\$("#p:hidden")	Chọn các phần tử <p> ở trạng thái ẩn
\$("#:input")	Chọn tất cả các phần tử <input>
\$("#:text")	Chọn các phần tử <input type = "text">
\$("#:button")	Chọn tất cả các phần tử <button> và <input type = "button">
\$("#:password")	Chọn các phần tử <input type = "password">
\$("#:file")	Chọn các phần tử <input type = "file">

13




Trường Đại học Mở TP. HCM
Cơ hội học tập cho mọi người

Nội dung

- Giới thiệu jQuery
- **Xử lý sự kiện**
- Làm việc với phần tử trong DOM
- Duyệt cây DOM
- Thiết lập hiệu ứng với jQuery

15




Trường Đại học Mở TP. HCM
Cơ hội học tập cho mọi người

Chọn phần tử trong jQuery (tt)

- Ví dụ một số bộ chọn jQuery (tt)

Cú pháp	Mô tả
\$("#:focus")	Chọn các phần tử đang nhận focus
\$("#:empty")	Chọn các phần tử rỗng
\$("#input:not(:empty)")	Chọn tất cả phần tử <input> có giá trị (không rỗng)
\$("#:disabled")	Chọn các phần tử <input> ở trạng thái bị vô hiệu hóa
\$("#:enabled")	Ngược với \$("#:disabled")
\$("#:checked")	Chọn các phần tử <input> ở trạng thái được check
\$("#:contains('Hello')")	Chọn các phần tử có chứa chuỗi "Hello"
\$("#:selected")	Chọn các phần tử đang ở trạng thái selected

14



Trường Đại học Mở TP. HCM
Cơ hội học tập cho mọi người

5.2 Xử lý sự kiện

- Giới thiệu
- Sự kiện trên cửa sổ trình duyệt và tài liệu
- Sự kiện trên form
- Sự kiện chuột
- Sự kiện bàn phím

16

5.2.1 Giới thiệu

- Các sự kiện trong jQuery có thể được tổ chức phân nhóm theo loại thành các danh mục giúp quản lý sự kiện dễ dàng hơn.
- Các sự kiện đều có thể được cài đặt và xóa một cách dễ dàng.
- Khai báo và xử lý sự kiện trong jQuery:
 - Viết code trong hàm xử lý (function) khi sự kiện xảy ra theo cú pháp:


```
$(selector).event_name(function() {
    //code xử lý
});
```
 - Gắn một hoặc nhiều sự kiện cho một bộ chọn


```
$(selector).on(event,childSelector,data,function,map)
```

17

17

Giới thiệu (tt)

- Sự kiện mức trang (Document/Window):
 - **load**: xảy ra khi một phần tử và tất cả các phần tử con đã được nạp hoàn toàn (hình ảnh, tập lệnh, khung, iframe, đối tượng cửa sổ)
 - **ready**: xảy ra khi cây DOM của tài liệu HTML được nạp hoàn toàn (ngoại trừ hình ảnh), tương tự sự kiện load nhưng xảy ra sớm hơn.
 - **resize**: xảy ra khi kích thước của cửa sổ trình duyệt bị thay đổi.
 - **scroll**: xảy ra khi người dùng cuộn trong phần tử, áp dụng cho đối tượng cửa sổ và các phần tử có thuộc tính overflow bằng "scroll" hoặc "auto".
 - **unload**: xảy ra khi người dùng rời khỏi trang web: click vào liên kết URL, nhập URL mới trong thanh địa chỉ, click các nút back, forward, đóng cửa sổ trình duyệt, tải lại trang.
 - **error**: xảy ra khi có lỗi trên trang web.

19

19

5.2.1 Giới thiệu

- Đối tượng **event**
 - **event.target**: trả về phần tử DOM phát sinh sự kiện
 - **event.type**: trả về loại sự kiện (mouseout, mouseover, keydown, click, dblclick...)
 - **event.pageX**: vị trí chuột so với cạnh trái của tài liệu
 - **event.pageY**: vị trí chuột so với cạnh trên của tài liệu
 - **event.keycode**: mã ascii của ký tự (sự kiện bàn phím)
 - **event.which**: mã unicode của ký tự (sự kiện bàn phím)
 - **event.preventDefault()**: ngăn chặn xử lý sự kiện mặc định của trình duyệt

18


18

Giới thiệu (tt)

- Sự kiện trên form và các phần tử trong form:
 - **submit**: xảy ra khi form được submit.
 - **change**: xảy ra khi nội dung trong phần tử nhập liệu (<input>, <textarea>) bị thay đổi hay phần tử mất focus, hoặc thay đổi lựa chọn đối với các phần tử như radio, checkbox, select.
 - **focus**: xảy ra khi phần tử như <input>, <select>, <a>, ... nhận focus (người dùng click chuột vào hoặc chuyển đến bằng phím Tab).
 - **blur**: xảy ra khi phần tử mất focus.
 - **select**: xảy ra khi người dùng chọn một vùng văn bản trong phần tử (<input type="text">, <textarea>).

20

20




Giới thiệu (tt)

- Sự kiện chuột:
 - **click**: xảy ra khi người dùng click chuột trái trên phần tử.
 - **contextmenu**: xảy ra khi người dùng click chuột phải trên phần tử.
 - **dblclick**: xảy ra khi người dùng nhấp đúp chuột trái trên phần tử.
 - **mousedown**: xảy ra khi người nhấn chuột trên phần tử.
 - **mouseup**: xảy ra khi người dùng nhả chuột trên phần tử.
 - **mouseenter, mouseover**: xảy ra khi con trỏ chuột đi vào phần tử.
 - **mouseleave, mouseout**: xảy ra khi con trỏ chuột rời khỏi phần tử.
 - **mousemove**: xảy ra khi con trỏ chuột di chuyển trên phần tử.
 - **hover**: xảy ra khi con trỏ chuột đi vào hay rời khỏi phần tử.

21

21



5.2.2 Sự kiện trên cửa sổ trình duyệt và tài liệu


- Sự kiện load:


```
$(selector).load(function() {
    // code xử lý
});
```

```
$("#img.userIcon").load(function() {
    if ($(this).height() > 100) {
        $(this).addClass("bigImg");
    }
});
```

23

23




Giới thiệu (tt)

- Sự kiện bàn phím:
 - **keydown**: xảy ra đối với phần tử đang nhận focus, khi một phím được nhấn.
 - **keyup**: xảy ra đối với phần tử đang nhận focus, khi thả một phím đang được nhấn.
 - **keypress**: tương tự keydown, nhưng không xử lý được đối với các phím non-character như Shift, Ctrl, Esc, Tab, CapsLock, Alt, Backspace. Nếu cài đặt cả hai sự kiện thì keypress được ưu tiên hơn

22

22



Sự kiện trên cửa sổ trình duyệt và tài liệu (tt)

- Sự kiện ready:


```
$(document).ready(function() {
    //code
});
```

– Hoặc viết ngắn gọn:

```
$.ready(function() {
    //code
});
```

```
<script>
$(document).ready(function(){
    $("#p").click(function(){
        $(this).hide();//ẩn phần tử p khi click lên phần tử
    });
});
</script>
```

– Thông thường, tất cả các code xử lý trên trang đều được viết bên trong function này.

24

24

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Sự kiện trên cửa sổ trình duyệt và tài liệu (tt)

- Sự kiện **resize**:
`$(window).resize(function(){
//code xử lý
});`

```
<script>
var width ,height;
$(document).ready(function(){
    $(window).resize(function(){
        width = $(this).innerWidth();
        height =$(this).innerHeight();
        $("p").html(" " + width + " , " + height + " ");
    });
});
</script>
```

25

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Sự kiện trên form (tt)

- Sự kiện **change**:
`$(selector).change(function(){
//code xử lý
});`

```
$( "select" ).change(function () {
    var str = "";
    $( "select option:selected" ).each(function() {
        str += $(this).text() + " ";
    });
    $( "#Monhoc" ).text("Bạn đã chọn: " + str );
});
$( "input[type='text']" ).change(function() {
    $( "#Ten" ).text("Tên bạn là: " +$( this ).val());
});
```

27

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

5.2.3 Sự kiện trên form

- Sự kiện **submit**:
`$("#form_id").submit(function() {
//code xử lý
});`

```
$("#myForm").submit(function(event) {
    var number = parseInt($("#input:first").val());
    if ( number >= 1 && number <= 10 ) {
        $("span").text( "Bạn đã nhập: " + number ).show();
    }
    else{
        $( "span" ).text("Giá trị nhập không hợp lệ!").show();
    }
    event.preventDefault();
});
```


26

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

5.2.4 Sự kiện chuột

- Cú pháp chung của các sự kiện chuột:
`$(selector).event_name(function() {
//code xử lý
});`
- Đối tượng event:
 - event.pageX: vị trí chuột so với cạnh trái của tài liệu
 - event.pageY: vị trí chuột so với cạnh trên của tài liệu


28

 **TRƯỜNG ĐẠI HỌC MỞ TP. HCM**
Cơ hội học tập cho mọi người

Sự kiện chuột (tt)

```
<div class="overout"><p></p></div> <div class="in enterleave"><p></p></div>
<script>
  $("div.overout").mouseover(function() {
    $("p",this).text( "mouse over" );
  })
  .mouseout(function() {
    $("p", this ).text( "mouse out" );
  });
  $("div.enterleave").mouseenter(function() {
    $("p", this ).text( "mouse enter" );
  })
  .mouseleave(function() {
    $("p", this ).text( "mouse leave" );
  });
</script>
```


29

 **TRƯỜNG ĐẠI HỌC MỞ TP. HCM**
Cơ hội học tập cho mọi người

5.2.6 Bật – tắt sự kiện với on(), off()

- on():** gắn một hoặc nhiều sự kiện cho bộ chọn **\$(selector).on(event, childSelector, data, function, map)**
 - event: một hoặc nhiều sự kiện
 - childSelector: bộ chọn các phần tử bên trong, tùy chọn
 - data: dữ liệu cung cấp cho hàm xử lý, tùy chọn
 - function: hàm xử lý
 - map: sơ đồ sự kiện dạng event map ({event:function, event:function, ...})

31


 **TRƯỜNG ĐẠI HỌC MỞ TP. HCM**
Cơ hội học tập cho mọi người

5.2.5 Sự kiện bàn phím

- Cú pháp chung của các sự kiện bàn phím:
\$(selector).keydown(function) {
 // code xử lý
});

```
$(document).ready(function(){
  $(document).keypress(function(event){
    var key = (event.keyCode ? event.keyCode : event.which);
    if(key == '13')
      $("h3#keycode").text('Enter');
    else
      $("h3#keycode").text(key);
  });
});
```

30

 **TRƯỜNG ĐẠI HỌC MỞ TP. HCM**
Cơ hội học tập cho mọi người


Bật – tắt sự kiện với on(), off() (tt)

- on()**

```
<p>Di chuyển chuột vào, ra trên khung và click ...</p>
<p>Click chuột lên <span>đồng chữ này</span></p>
```

```
$( "p:first" ).on({
  mouseover: function(){
    $("body").css("background-color", "lightgray");
  },
  mouseout: function(){
    $("body").css("background-color", "lightblue");
  },
  click: function(){
    $("body").css("background-color", "yellow");
  }
});
$( "p:last" ).on("click","span",function(){
  $(this).css("font-size","2em");
});
```

32


Trường Đại học Mở TP. HCM
 Cơ hội học tập cho mọi người

Bật – tắt sự kiện với on(), off() (tt)


- off():** hủy sự kiện được tạo bởi on()

```

$(document).ready(function(){
  $("p").on("click", function(){
    $(this).css("background-color", "pink");
  });
  $("button").click(function(){
    $("p").off("click");
  });
});

```


33


Trường Đại học Mở TP. HCM
 Cơ hội học tập cho mọi người

5.3 Làm việc với phần tử trong DOM

- Truy cập nội dung phần tử
- Truy cập thuộc tính phần tử
- Chèn nội dung/phần tử vào bên trong, trước, sau một phần tử
- Xóa nội dung, xóa phần tử
- Cập nhật CSS
- Thiết lập kích thước cho phần tử


35


Trường Đại học Mở TP. HCM
 Cơ hội học tập cho mọi người

Nội dung

- Giới thiệu jQuery
- Xử lý sự kiện
- Làm việc với phần tử trong DOM**
- Duyệt cây DOM
- Thiết lập hiệu ứng với jQuery


34


Trường Đại học Mở TP. HCM
 Cơ hội học tập cho mọi người

Làm việc với phần tử trong DOM (tt)

- Truy cập nội dung phần tử**
 - text():**
 - text(): trả về chuỗi văn bản trong phần tử.
– Ví dụ: \$("p#p1").text()
 - text(string): thiết lập chuỗi string cho phần tử.
– Ví dụ: \$("p#p1").text("Hello")
 - html():**
 - html(): trả về văn bản HTML trong phần tử.
– Ví dụ: \$("p#p1").html()
 - html(strHtml): thiết lập chuỗi strHtml cho phần tử.
– Ví dụ: \$("p#p1").html("Hello")


36

 **TRƯỜNG ĐẠI HỌC MỞ TP. HCM**
Cơ hội học tập cho mọi người

Làm việc với phần tử trong DOM (tt)

- **Truy cập nội dung phần tử (tt)**
 - **val()**:
 - **val()**: trả về giá trị trong phần tử nhập liệu (<input>, <textarea>, ...)
 - Ví dụ: `$("#input#firstName").val()`
 - **val(string)**: thiết lập chuỗi string cho phần tử nhập liệu
 - Ví dụ: `$("#input#firstName").val("Anna");`


37

 **TRƯỜNG ĐẠI HỌC MỞ TP. HCM**
Cơ hội học tập cho mọi người

Làm việc với phần tử trong DOM (tt)

- **Thêm nội dung mới vào bên trong phần tử**
 - **prepend(content)**: thêm content vào đầu nội dung phần tử được chọn.
 - Ví dụ: `$("#ol").prepend("Phần tử đầu")`
 - **append(content)**: nối content vào cuối nội dung của phần tử được chọn.
 - Ví dụ: `$("#ol").append("Phần tử cuối")`


39

 **TRƯỜNG ĐẠI HỌC MỞ TP. HCM**
Cơ hội học tập cho mọi người

Làm việc với phần tử trong DOM (tt)

- **Truy cập thuộc tính phần tử:**
 - **attr("attr_name")**: trả về giá trị của thuộc tính attr_name.
 - Ví dụ `$("#a.#title").attr("href")`
 - **attr("attr_name", "value")**: thiết lập giá trị value cho thuộc tính attr_name.
 - Ví dụ `$("#a.#title").attr("href", "http://abc.com")`


38

 **TRƯỜNG ĐẠI HỌC MỞ TP. HCM**
Cơ hội học tập cho mọi người

Làm việc với phần tử trong DOM (tt)


- **Chèn nội dung mới vào trước, sau phần tử:**
 - **before(content)**: chèn content vào trước phần tử được chọn.
 - Ví dụ: `$("#p#p2").before("<h3>Chèn trước p2</h3>")`
 - **after(content)**: chèn content vào sau phần tử được chọn.
 - Ví dụ: `$("#p#p2").after("<h3>Chèn sau p2</h3>")`

40


Làm việc với phần tử trong DOM (tt)


- **Xóa nội dung/phần tử**
 - **remove()**
 - **remove()**: xóa phần tử và tất cả các phần tử con.
 - Ví dụ: `$("del").remove()`
 - **remove(selector)**: xóa các phần tử thỏa bộ chọn selector.
 - Ví dụ: `$("p").remove(":contains('Hello')"`)
 - **empty()**: xóa tất cả nội dung bên trong một phần tử
 - Ví dụ: `$("div#container").empty()`

41


Làm việc với phần tử trong DOM (tt)


- **Cập nhật CSS (tt)**
 - **css("property")**: trả về giá trị của thuộc tính "property".
 - Ví dụ: `var textcolor = $("p.p1").css("color")`
 - **css("property", "value")**: thiết lập giá trị "value" cho thuộc tính "property".
 - Ví dụ: `$("p.p1").css("color", "red")`
 - **css({"property1": "value1", "property2": "value2"})**: thiết lập nhiều thuộc tính css.
 - Ví dụ: `$("p.p2").css({"background-color": "yellow", "color": "blue"})`

43


Làm việc với phần tử trong DOM (tt)

- **Cập nhật CSS**
 - **addClass("class_name1 class_name2")**: bổ sung thêm hai class name: "class_name1" và "class_name2" cho phần tử.
 - Ví dụ: `$("div").addClass("border bold")`
 - **removeClass("class_name")**: xóa class="class_name" của phần tử.
 - Ví dụ: `$("div").removeClass("bold")`
 - **toggleClass("class_name")**: chuyển đổi giữa addClass và removeClass
 - Ví dụ: `$("h3").toggleClass("red")`

42


Làm việc với phần tử trong DOM (tt)

- **Thiết lập kích thước cho phần tử**
 - **width(value)/width()**: chiều rộng của phần tử (trừ padding, border, margin).
 - **height(value)/height()**: chiều cao của phần tử (trừ padding, border, margin).
 - **innerWidth()**: chiều rộng của phần tử (bao gồm phần padding).
 - **innerHeight()**: chiều cao của phần tử (bao gồm phần padding).
 - **outerWidth()**: chiều rộng của phần tử (bao gồm phần padding và border).
 - **outerHeight()**: chiều cao của phần tử (bao gồm phần padding và border).
 - **outerWidth(true)**: chiều rộng của phần tử (bao gồm padding, border, margin).
 - **outerHeight(true)**: chiều cao của phần tử (bao gồm padding, border, margin).

44

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Làm việc với phần tử trong DOM (tt)

- Thiết lập kích thước cho phần tử

45

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

5.4 Duyệt cây DOM

- Chọn các phần tử cha
- Chọn các phần tử con
- Chọn các phần tử đồng cấp
- Chọn theo vị trí
- Chọn theo điều kiện

47

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Nội dung

- Giới thiệu jQuery
- Xử lý sự kiện
- Làm việc với phần tử trong DOM
- Duyệt cây DOM**
- Thiết lập hiệu ứng với jQuery

46

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Chọn các phần tử cha

- Phương thức parent():** trả về phần tử cha trực tiếp của phần tử được chọn
- Phương thức parents():** trả về tất cả các phần tử cha và tổ tiên của phần tử được chọn.
- Phương thức parents(selector):** trả về tất cả các phần tử cha và tổ tiên của phần tử được chọn thỏa bộ chọn selector.
- Phương thức parentsUntil (element):** trả về phần tử cha và tổ tiên nằm giữa phần tử được chọn và phần tử element

48

Chọn các phần tử con

- **Phương thức children():** trả về các phần tử con trực tiếp của phần tử được chọn.
- **Phương thức children (selector):** trả về tất cả các phần tử con trực tiếp của phần tử được chọn thỏa bộ chọn selector.
- **Phương thức find(selector):** trả về các phần tử là con cháu của phần tử được chọn thỏa bộ chọn selector.

49

49

Chọn theo vị trí

- **first():** trả về phần tử đầu tiên trong các phần tử được chọn.
- **last():** trả về phần tử cuối cùng trong các phần tử được chọn
- **eq(n):** trả về phần tử thứ n trong các phần tử được chọn (phần tử đầu có chỉ số là 0).

51

51

Chọn các phần tử đồng cấp

- **siblings():** trả về tất cả các phần tử anh em với phần tử được chọn.
- **siblings(selector):** trả về tất cả các phần tử anh em với phần tử được chọn thỏa bộ chọn selector.
- **next():** trả về phần tử anh em kế tiếp của phần tử được chọn.
- **nextAll():** trả về tất cả các phần tử anh em kế tiếp của phần tử được chọn
- **nextUntil(element):** trả về các phần tử anh em kế tiếp nằm giữa phần tử được chọn và phần tử element.
- **prev():** trả về phần tử anh em đứng trước phần tử được chọn
- **prevAll():** trả về tất cả các phần tử anh em đứng trước của phần tử được chọn
- **prevUntil(element):** trả về các phần tử anh em đứng trước nằm giữa phần tử được chọn và phần tử element

50


50

Chọn theo điều kiện

- **filter():** chọn các phần tử thỏa điều kiện tìm kiếm.
- **not():** chọn các phần tử không thỏa điều kiện tìm kiếm.

52


52


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
 Cơ hội học tập cho mọi người

Nội dung

- Giới thiệu jQuery
- Xử lý sự kiện
- Làm việc với phần tử trong DOM
- Duyệt cây DOM
- **Thiết lập hiệu ứng với jQuery**


53


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
 Cơ hội học tập cho mọi người

Hiển thị - ẩn phần tử

- **Sử dụng các phương thức**
 - **show([duration] [, complete])**: hiển thị phần tử.
 - **hide([duration] [, complete])**: ẩn phần tử.
 - **toggle([duration] [, complete])**: chuyển đổi trạng thái giữa show và hide
 - duration: tốc độ, mặc định là 400 mili giây.
 - complete : hàm xử lý tiếp theo sau khi hiển thị đối tượng.

55


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
 Cơ hội học tập cho mọi người

5.5 Thiết lập hiệu ứng với jQuery

- Hiển thị - ẩn phần tử
- Làm mờ - rõ phần tử
- Thiết lập hiệu ứng trượt cho phần tử
- Tạo hoạt hình cho phần tử
- Dừng hiệu ứng
- Kết hợp các hiệu ứng

54


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
 Cơ hội học tập cho mọi người

Hiển thị - ẩn phần tử (tt)

- Ví dụ:

```

$("#show" ).click(function() { $("#rose").show( "slow" ); })
$("#hide" ).click(function() { $("#rose").hide(1000); });
$("#show_hide" ).click(function() {
    $("#rose").toggle(1000, showDisplay());
});
function showDisplay(){
    let str = $("#show_hide").text();
    if(str.search("Show") >=0)
        $("#show_hide").text("Hide");
    else
        $("#show_hide").text("Show");
}
  
```

56

Làm mờ - rõ phần tử

• Sử dụng các phương thức:

- **fadeIn([duration] [, complete]):** làm rõ dần phần tử.
- **fadeOut([duration] [, complete]):** làm mờ dần phần tử
- **fadeToggle([duration] [, complete]):** chuyển đổi trạng thái giữa mờ dần và rõ dần.
- **fadeTo(duration, opacity [, complete]):** làm mờ/rõ đến giá trị opacity.
 - **opacity:** giá trị từ 0 - 1, dùng trong phương thức fadeTo, phần tử sẽ được làm mờ/rõ đến giá trị này.

57

57

Thiết lập hiệu ứng trượt

• Sử dụng các phương thức:

- **slideDown([duration] [, complete]):** tạo hiệu ứng trượt xuống cho phần tử.
- **slideUp([duration] [, complete]):** tạo hiệu ứng trượt lên cho phần tử.
- **slideToggle([duration] [, complete]):** chuyển đổi trạng thái giữa trượt xuống và trượt lên.

59

59

Làm mờ - rõ phần tử (tt)

• Ví dụ:

```
$("#fadeIn").click(function() { $("#rose").fadeIn(1000); });
$("#fadeOut").click(function() { $("#rose").fadeOut(1000); });
$("#fadeTo").click(function() { $("#rose").fadeTo(1000, 0.5); });
$("#fadeInOut").click(function() {
    let str = $(this).text();
    $("#rose").fadeToggle(2000, function() {
        if(str.search("in") >= 0)
            $("#fadeInOut").text("fade out");
        else
            $("#fadeInOut").text("fade in");
    });
});
```

58

58


Thiết lập hiệu ứng trượt (tt)

• Ví dụ:

```
$("#Up").click(function() { $("#banner").slideUp(1000); });
$("#Down").click(function() { $("#banner").slideDown(1000); });
$("#Toggle").click(function() {
    let str = $(this).text();
    $("#banner").slideToggle(1000, function() {
        if(str.search("up") >= 0)
            $("#Toggle").text("slide down");
        else
            $("#Toggle").text("slide up");
    });
});
```


60

60

 **Tạo hoạt hình cho phần tử**

- **Sử dụng các phương thức:**
 - **animate(properties[, duration][, complete])**
 - properties: một hoặc nhiều bộ thuộc tính: giá trị CSS thiết lập cho đối tượng sau khi hoạt hình.
 - **Lưu ý:** để một phần tử có thể di chuyển, thiết lập giá trị cho thuộc tính **position** có giá trị khác **static**.

61

 **Tạo hoạt hình cho phần tử (tt)**

- Ví dụ


```

$("button").click(function(){
    $("div").animate({
        height: 'toggle'
    });
});


```

```

$("button").click(function(){
    $("div").animate({height: '300px', opacity: '0.4'}, "slow");
    $("div").animate({width: '300px', opacity: '0.8'}, "slow");
    $("div").animate({height: '100px', opacity: '0.4'}, "slow");
    $("div").animate({width: '100px', opacity: '0.8'}, "slow");
});

```

63

 **Tạo hoạt hình cho phần tử (tt)**

- Ví dụ:


```

$("button").click(function(){
    $("div").animate({
        left: '200px',
        opacity: '0.5',
        height: '120px'
    });
});


```

```

$("button").click(function(){
    $("div").animate({
        left: '250px',
        height: '+=150px',
        width: '+=150px'
    });
});

```

62

 **Dừng các hiệu ứng**

- **Sử dụng phương thức:**
 - **\$(selector).stop(stopAll, goToEnd)**
 - **stopAll:** true/false: dừng hết/không các hiệu ứng còn trong hàng đợi, mặc định là false
 - **goToEnd:** true/false: hoàn thành hiệu ứng ngay lập tức, mặc định là false
 - Ví dụ: khi click lên phần tử có id là "stop", hiệu ứng đang chạy của phần tử có id là "panel" sẽ dừng ngay tức khắc


```

$("#stop").click(function(){
    $("#panel").stop();
});

```

64

Kết hợp nhiều hiệu ứng

- Để kết hợp nhiều hiệu ứng trên phần tử, liệt kê các hàm nối tiếp nhau, phân cách bằng dấu chấm (“.”)

```
$("#p1").css("color", "red")  
        .slideUp(2000)  
        .slideDown(2000);
```

65