

Buổi 9

Tổng quan về jQuery

I. Khái niệm

jQuery là 1 thư viện (*library*) của JS với mục đích đơn giản hóa các đoạn lệnh JS, giúp cho việc cài đặt JS trong trang web nhanh gọn và dễ dàng hơn. Ngoài ra, jQuery cũng cung cấp một số hàm được viết sẵn phục vụ 1 chức năng nào đó, cung cấp các hiệu ứng chuyển động cho trang web...

Có rất nhiều thư viện và framework của JS, nhưng jQuery là một trong những thư viện phổ biến nhất và được mở rộng nhiều nhất. jQuery được sử dụng rất nhiều bởi các công ty, tập đoàn lớn như Google, Microsoft, IBM, Netflix...

1. Các extension cần thiết cho VS Code

Để sử dụng jQuery thuận tiện và nhanh chóng hơn, nên cài extension sau đây cho VS Code: jQuery Code Snippets (keyword: donjayamanne.jquerysnippets). Để sử dụng extension này, gõ snippet jq trong mã nguồn JS.

2. Thêm thư viện jQuery vào trang web

Để sử dụng jQuery, đầu tiên cần thêm thư viện jQuery vào trang web theo 1 trong 2 cách sau:

a) Tải thư viện jQuery từ trang chủ <http://www.jquery.com>:

Có 2 phiên bản jQuery:

- Production: Phiên bản khuyên dùng cho việc phát triển website, dung lượng nhỏ hơn nhưng mã nguồn khó đọc hơn.
- Development: Phiên bản dùng cho việc mở rộng jQuery, dung lượng lớn hơn nhưng mã nguồn dễ đọc hơn.

Sau khi tải về file jquery-3.4.1.min.js, chèn file này vào trang web như 1 file JS thông thường bằng đoạn code sau:

```
<head>  
  <script src="js/jquery-3.4.1.min.js"></script>  
</head>
```

b) Dùng thư viện jQuery từ CDN:

CDN (*Content Delivery Network*) cung cấp các file thư viện, framework trực tuyến, do đó chúng ta có thể sử dụng trực tiếp jQuery từ các trang này mà không cần phải tải về.

Để sử dụng Google CDN cho jQuery, dùng đoạn code sau:

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">  
</script>
```

Để sử dụng Microsoft CDN cho jQuery, dùng đoạn code sau:

```
<script  
src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.4.1.min.js">  
</script>
```

Việc sử dụng jQuery từ các CDN sẽ giúp giảm dung lượng cũng như thời gian tải trang web.

3. Cú pháp jQuery

Toàn bộ mã nguồn jQuery được đặt trong 1 file .js như mã nguồn JS thông thường. jQuery xoay quanh việc chọn phần tử HTML và thực hiện thao tác trên phần tử đó.

Cú pháp của jQuery:

```
$(selector).action();
```

Trong đó:

- Ký tự \$ là bắt buộc để sử dụng jQuery.
- selector dùng để truy cập tới đối tượng (tương tự như cách viết selector trong CSS).
- action() là thao tác muốn thực hiện trên đối tượng, có thể là phương thức có sẵn hoặc cài đặt sự kiện bằng hàm tự viết.

Ví dụ:

- Ẩn tất cả thẻ có class là demo:

```
$(".demo").hide();
```
- Ẩn tất cả thẻ <p> có class là demo:

```
$("p.demo").hide();
```
- Ẩn thẻ có id là test:

```
$("#test").hide();
```

Để tránh việc mã nguồn jQuery được chạy khi trang web chưa tải xong, toàn bộ mã nguồn jQuery phải được đặt trong 1 sự kiện là document.ready (snippet: jqDocReady):

```
$(document).ready(function() {  
    // Code jQuery đặt ở đây  
});
```

Sự kiện document.ready còn có thể được viết ngắn gọn như sau (snippet: jqDocReadyShort):

```
$(function() {  
    // Code jQuery đặt ở đây  
});
```

II. Xử lý sự kiện

1. Phương thức sự kiện

Các sự kiện (event) của đối tượng HTML được xử lý bằng các phương thức tương ứng trong jQuery theo cú pháp sau:

```
$(selector).eventName(function() {  
    // Code xử lý cho sự kiện  
});
```

Trong đó, eventName là tên của sự kiện muốn cài đặt.

Một số sự kiện thông dụng gồm có:

Tên event	Ý nghĩa	Snippet
blur	Đối tượng bị mất focus	jqBlur
change	Nội dung của đối tượng bị thay đổi	jqChange
click	Click vào 1 đối tượng	jqClick
dblclick	Double-click vào 1 đối tượng	
focus	Đối tượng có focus	jqFocus
keydown	Nhấn 1 phím	jqKeyDown

keyup	Thả 1 phím	jqKeyUp
keypress	Nhấn và thả 1 phím	jqKeyPress
mouseenter	Con trỏ chuột đi vào 1 đối tượng	jqMouseEnter
mouseleave	Con trỏ chuột đi ra khỏi 1 đối tượng	jqMouseLeave
hover	Con trỏ chuột di chuyển trên 1 đối tượng	jqHover

Ví dụ:

- Khi click chuột vào nút có id là btnDangKi, hiển thị hộp thoại thông báo "Xin chào!":

```
$("button#btnDangKi").click(function() {
    alert("Xin chào!");
});
```

- Khi click chuột vào thẻ <p>, ẩn tất cả thẻ <p>:

```
$("p").click(function() {
    $("p").hide();
});
```

- Khi click chuột vào thẻ <p>, ẩn thẻ <p> đó:

```
$("p").click(function() {
    $(this).hide();
});
```

Đối tượng \$(this) trong jQuery là đối tượng đang gọi sự kiện hiện tại.

2. Phương thức on()

Khi có nhiều sự kiện cùng được cài đặt trên 1 đối tượng, chúng ta có thể sử dụng cách truyền thống là cài đặt từng sự kiện riêng biệt như sau:

```
$("div#demo").click(function() {
    // Code xử lí
});
$("div#demo").mouseenter(function() {
    // Code xử lí
});
$("div#demo").mouseleave(function() {
    // Code xử lí
});
```

Với cách này, các phương thức sự kiện có thể sẽ nằm rải rác trong mã nguồn jQuery, dẫn đến khó quản lí và theo dõi code. Thay vào đó, chúng ta có thể gộp tất cả các sự kiện trên cùng 1 đối tượng bằng cách sử dụng phương thức on() như sau:

```
$("div#demo").on({
    click: function() {
        // Code xử lí cho sự kiện click
    },
    mouseenter: function() {
        // Code xử lí cho sự kiện mouseenter
    },
    mouseleave: function() {
        // Code xử lí cho sự kiện mouseleave
    }
});
```

Với cách này, tất cả các sự kiện trên cùng 1 đối tượng sẽ nằm chung 1 chỗ, dễ quản lí hơn. Sau này, khi cần cài thêm sự kiện khác, chúng ta sẽ bổ sung tiếp tục vào phương thức `on()` này.

Ngoài ra, phương thức `on()` vẫn có thể dùng để cài đặt 1 sự kiện như sau:

```
$("#div#demo").on("click", function() {  
    // Code xử lí  
});
```

3. Sự kiện hover

Sự kiện hover là 1 sự kiện đặc biệt, nó bao gồm 2 sự kiện nhỏ là `mouseenter` và `mouseleave`. Do đó, cách cài đặt sự kiện hover sẽ có khác biệt như sau:

```
$("#div#demo").hover(function() {  
    // Code xử lí cho sự kiện mouseenter  
},  
function() {  
    // Code xử lí cho sự kiện mouseleave  
});
```

III. Thay đổi HTML với jQuery

1. Get và Set

Cho các thẻ HTML sau:

```
<div id="demo"><b>Xin chào</b></div>  
<input id="txt" type="text" value="admin" />
```

jQuery cung cấp 4 phương thức sau đây để lấy (*get*) và gán (*set*) giá trị cho 1 thẻ HTML:

Phương thức	Ý nghĩa	Snippet	Ví dụ	Kết quả
text()	Lấy và gán nội dung (text)	jqTextGet	<code>\$("#demo").text();</code>	Xin chào
		jqTextSet	<code>\$("#demo").text("<i>Hello</i>");</code>	<i>Hello</i>
html()	Lấy và gán nội dung (HTML)	jqHtmlGet	<code>\$("#demo").html();</code>	Xin chào
		jqHtmlSet	<code>\$("#demo").html("<i>Hello</i>");</code>	<i>Hello</i>
val()	Lấy và gán giá trị form control	jqValGet	<code>\$("#txt").val();</code>	admin
		jqValSet	<code>\$("#txt").val("customer");</code>	
attr()	Lấy và gán giá trị thuộc tính	jqAttrGet	<code>\$("#txt").attr("type");</code>	text
		jqAttrSet	<code>\$("#txt").attr("type", "password");</code>	

Ngoài ra, phương thức `attr()` còn cho phép thay đổi giá trị nhiều thuộc tính của thẻ HTML cùng lúc theo cách sau:

```
$("#a").attr({  
    "href": "http://www.google.com/",  
    "title": "Google"  
});
```

2. Add và Remove

Cho các thẻ HTML sau:

```
<b>Hello</b>
```

Kết quả hiển thị ra trang web: **Hello**

jQuery cung cấp 4 phương thức sau để thêm (*add*) và xóa (*remove*) nội dung của 1 thẻ HTML:

- Phương thức `prepend()` (snippet: `jqPrepend`) sẽ thêm nội dung vào đầu thẻ HTML:
`$("b").prepend("Oh! ");` → Kết quả: **Oh! Hello**
- Phương thức `append()` (snippet: `jqAppend`) sẽ thêm nội dung vào cuối thẻ HTML:
`$("b").append(" World");` → Kết quả: **Hello World**
- Phương thức `before()` (snippet: `jqBefore`) sẽ thêm nội dung vào trước thẻ HTML:
`$("b").before("Oh! ");` → Kết quả: **Oh! Hello**
- Phương thức `after()` (snippet: `jqAfter`) sẽ thêm nội dung vào sau thẻ HTML:
`$("b").after(" World");` → Kết quả: **Hello World**

Cả 4 phương thức trên đều có thể nhận vào tham số chứa cả mã nguồn HTML, và có thể nhận vào nhiều tham số cùng lúc, ví dụ:

`$("b").append(" World", " <i>War</i> Z");` → Kết quả: **Hello World War Z**

IV. Tương tác với CSS

1. Thay đổi class

jQuery cung cấp các phương thức sau để thay đổi class cho 1 đối tượng HTML:

- `addClass()`: Thêm class.
- `removeClass()`: Loại bỏ class.
- `toggleClass()`: Bật/tắt class.

Ví dụ: Xem ví dụ 9.1: Giả sử có đoạn mã HTML và CSS sau:

– HTML:

```
<h1>Đây là thẻ h1</h1>
<h2>Đây là thẻ h2</h2>
<p>Đây là thẻ p</p>
<div>Đây là thẻ div</div>
```

– CSS:

```
.blue {
    color: blue;
}

.important {
    font-weight: bold;
    font-size: xx-large;
}
```

- Để thêm class "blue" cho các thẻ `<h1>`, `<h2>`, `<p>` và thêm class "important" cho thẻ `<div>`, chúng ta dùng câu lệnh sau:

```
$("h1, h2, p").addClass("blue");
$("div").addClass("important");
```

- Để loại bỏ class "blue" khỏi các thẻ `<h1>`, `<h2>`, `<p>` và loại bỏ class "important" khỏi thẻ `<div>`, chúng ta dùng câu lệnh sau:

```
$("h1, h2, p").removeClass("blue");
$("div").removeClass("important");
```

- Để bật/tắt class "blue" cho các thẻ `<h1>`, `<h2>`, `<p>`, chúng ta dùng câu lệnh sau:

```
$("#h1, h2, p").toggleClass("blue");
```

Lưu ý: Có thể thêm, loại bỏ hoặc bật tắt cùng lúc nhiều class cho 1 đối tượng HTML, ví dụ:

```
$("#p").addClass("important blue");
```

2. Thay đổi CSS

jQuery cung cấp phương thức `css()` để lấy (get) và gán (set) giá trị cho 1 thuộc tính CSS theo cú pháp sau:

```
$(selector).css("tên thuộc tính"); → Lấy giá trị thuộc tính
```

```
$(selector).css("tên thuộc tính", "giá trị"); → Gán giá trị cho thuộc tính
```

Ví dụ:

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

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

Có thể dùng phương thức `css()` để gán giá trị cho nhiều thuộc tính CSS theo cú pháp sau:

```
.css({"tên thuộc tính 1": "giá trị 1", "tên thuộc tính 2": "giá trị 2", ...});
```

Ví dụ: Xem ví dụ 9.1:

```
$("#p").css({  
    "color": "red", "font-weight": "bold", "font-size": "large"  
});
```

V. Tạo hiệu ứng chuyển động

1. Hide & Show

jQuery cung cấp 2 phương thức `hide()` và `show()` để ẩn và hiện 1 đối tượng. 2 phương thức này có thể nhận 1 tham số là thời gian hiệu ứng (ms).

Ví dụ: Xem ví dụ 9.2:

```
$("#p").hide();
```

```
$("#p").hide(1000);
```

```
$("#p").show();
```

```
$("#p").show(2000);
```

Ngoài ra, phương thức `toggle()` giúp bật/tắt việc ẩn hay hiện đối tượng.

2. Fade

jQuery cung cấp 2 phương thức `fadeOut()` và `fadeIn()` để ẩn và hiện 1 đối tượng theo hiệu ứng fade. 2 phương thức này có thể nhận 1 tham số là thời gian hiệu ứng (ms).

Ví dụ: Xem ví dụ 9.2:

```
$("#p").fadeOut();
```

```
$("#p").fadeOut(1000);
```

```
$("#p").fadeIn();
```

```
$("#p").fadeIn(2000);
```

Ngoài ra, phương thức `fadeToggle()` giúp bật/tắt việc ẩn hay hiện đối tượng.

3. Slide

jQuery cung cấp 2 phương thức `slideUp()` và `slideDown()` để ẩn và hiện 1 đối tượng theo hiệu ứng slide. 2 phương thức này có thể nhận 1 tham số là thời gian hiệu ứng (ms).

Ví dụ: Xem ví dụ 9.2:

```
$("#p").slideUp();
```

```
$("#p").slideUp(1000);
```

```
$("#p").slideDown();
```

```
$("#p").slideDown(2000);
```

Ngoài ra, phương thức `slideToggle()` giúp bật/tắt việc ẩn hay hiện đối tượng.

4. Animation

jQuery cung cấp phương thức `animate()`¹ cho phép biến đổi 1 đối tượng theo 1 đoạn code CSS nào đó.

Ví dụ: Xem ví dụ 9.3:

```
$("#div").animate({
    left: '250px', opacity: '0.5', height: '150px', width: '150px',
    top: '100px'
});
```

Lưu ý: Với phương thức `animate()`, các thuộc tính CSS có dấu gạch ngang phải được viết lại theo kiểu camel-cased, ví dụ: `padding-left` → `paddingLeft`.

5. Dừng hiệu ứng

jQuery cung cấp phương thức `stop()` cho phép dừng 1 hiệu ứng đang chạy:

```
$("#div").stop();
```

VI. Các plug-in cho jQuery

Ngoài cách tự viết jQuery, chúng ta có thể tải thêm các plug-in tại các địa chỉ sau:

- <https://plugins.jquery.com/>: Trang plug-in chính thức từ jQuery.
- <https://jqueryui.com/>: Trang web cung cấp các control cho form, plug-in... hỗ trợ việc thiết kế giao diện người dùng (*UI – User Interface*).

VII. Bài tập

Thực hiện các bài tập sau đây, mỗi bài tập nằm trong 1 trang web riêng.

1. Thiết kế 1 trang web gồm 1 textbox để nhập họ tên và 1 button Đăng nhập.

Yêu cầu: Sau khi nhập họ tên và nhấn nút đăng nhập, hiển thị lời chào "Xin chào abc!" (thay abc bằng họ tên người dùng vừa nhập).

2. Thiết kế 1 trang web gồm 2 textbox để nhập tên tài khoản, mật khẩu, 1 checkbox Hiển thị mật khẩu và 1 button Đăng nhập.

Yêu cầu:

- Khi di chuyển chuột lên textbox Mật khẩu thì hiển thị mật khẩu ở dạng plain text, khi di chuyển chuột ra khỏi textbox Mật khẩu thì hiển thị mật khẩu ở dạng password.
- Khi check vào Hiển thị mật khẩu thì hiển thị mật khẩu ở dạng plain text, khi uncheck vào Hiển thị mật khẩu thì hiển thị mật khẩu ở dạng password như bình thường.
- Sau khi nhập thông tin và nhấn nút Đăng nhập, kiểm tra nếu tên tài khoản là admin và mật khẩu là 123456 thì thông báo "Đăng nhập thành công", ngược lại thông báo "Đăng nhập thất bại".

Gợi ý: Để kiểm tra 1 checkbox có đang được check hay không, sử dụng phương thức `prop()`:

```
if ($("#chk").prop("checked")) {
    // Code xử lí
}
```

Phương thức `prop()` giúp kiểm tra 1 thuộc tính (*property*) có xuất hiện trong đối tượng HTML hay không.

¹ Tham khảo thêm về phương thức `animate()` tại địa chỉ: https://www.w3schools.com/jquery/jquery_animate.asp