

Ajax và JQuery

Giảng viên: **Phạm Nhân Nghĩa**

Email: **nghiapn89@gmail.com**

Giới thiệu

- Ajax (Asynchronous JavaScript And XML) không phải là ngôn ngữ lập mà là kỹ thuật để tạo ra ứng dụng web tốt, nhanh và giao tiếp thân thiện.
- Ajax giúp gửi yêu cầu đến sever và nhận phản hồi mà không cần nạp lại toàn bộ trang.
- Các request có thể thực hiện bất đồng bộ (trong quá trình thực hiện request lên sever, người dùng vẫn có thể tương tác với trang web).
- Một số ứng dụng sử dụng AJAX: Google Maps, Gmail, Youtube, Facebook,...

Ajax làm việc như thế nào?

- AJAX hoạt động dựa trên các chuẩn Internet, nó là sự kết hợp của các kỹ thuật sau:
 - XMLHttpRequest object: trao đổi dữ liệu với server
 - JavaScript/DOM: dùng để hiển thị dữ liệu, tương tác với người dùng. DOM(Document Object Model): Truy xuất, thao tác dữ liệu của tài liệu có cấu trúc(HTML,XML), thể hiện tài liệu dưới dạng cấu trúc cây phân cấp.
 - CSS: trình bày dữ liệu
 - XML: định dạng dữ liệu cho mục đích truyền dữ liệu

Ajax thuần Javascript

- Tạo đối tượng XMLHttpRequest
- Request đến server
- Xử lý phản hồi từ server.

Ajax thuần Javascript

- Tạo đối tượng xmlhttp: dùng để gửi yêu cầu và nhận kết quả trả về từ sever.

```
var xmlhttp;  
if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari  
    xmlhttp = new XMLHttpRequest();  
}  
else { // code for IE6, IE5  
    if(window.ActiveXObject)  
    {  
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  
    }  
    else  
    {  
        alert("Browse not support");  
    }  
}
```

Ajax thuần Javascript

- Gửi request đến server.
 - `open(method, url, async)`:
 - Method: phương thức gửi dữ liệu(GET hoặc POST)
 - url: Đường dẫn vị trí tập tin trên sever.
 - Async: true(bất đồng bộ) hoặc false(đồng bộ)
 - `send(string)`: gửi yêu cầu đến sever.
 - String: dữ liệu gửi trong phần thân của thông điệp.chỉ dùng cho POST
 - VD:

```
xmlhttp.open("POST", "demo.asp", true);  
xmlhttp.send("fname=Henry&lname=Ford");
```

```
xmlhttp.open("GET", url, true);  
xmlhttp.send();
```

Ajax thuần Javascript

- Xử lý response: Nhận thông điệp trả lời từ server và xử lý thông tin nhận được.
 - responseText: nội dung (phần thân) của thông điệp trả lời
 - responseXML: chứa một XML DOM document nếu thông điệp trả về có kiểu nội dung (content type) là "text/xml" hay "application/xml"

```
xmlhttp.onreadystatechange = function () {  
    if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {  
        document.getElementById("ajax").innerHTML = xmlhttp.responseText;  
    }  
    else  
    {  
        document.getElementById("ajax").innerHTML = "File not found";  
    }  
}
```

Ajax thuần Javascript

- `onreadystatechange`: chứa hàm được gọi mỗi khi thuộc tính `readyState` thay đổi giá trị.
- `readyState`: xác định trạng thái của `XMLHttpRequest`. Các giá trị có thể nhận:
 - 0: chưa được khởi tạo. `Open()` chưa được gọi
 - 1: đã khởi tạo, `open()` đã thực thi, `send()` chưa được gọi
 - 2: đã gửi, `send()` đã gọi nhưng chưa nhận được thông điệp trả lời
 - 3: đang nhận dữ liệu
 - 4: hoàn tất, đã nhận được tất cả dữ liệu trả về
- `Status`: mã trạng thái của thông điệp trả lời.
 - 200 : "OK"
 - 404: Page not found.

JQuery

- Giới thiệu JQuery
- Selector
- Sự kiện
- Hiệu ứng
- Thay đổi content
- Filter
- AJAX

jQuery Ajax

- jQuery là một thư viện của Javascript, “viết ít hơn, làm nhiều hơn”.
- jQuery Ajax cung cấp các phương thức giúp thực hiện Ajax trên trang web.
- Yêu cầu dữ liệu trên server bằng GET , POST hoặc tải các dữ liệu từ bên ngoài trực tiếp vào các thành phần HTML được chọn trên trang web.
 - GET: yêu cầu dữ liệu từ nguồn tài nguyên trên server.
 - POST: gửi dữ liệu lên để xử lý trên nguồn tài nguyên trên server.

jQuery Selector

- Cú pháp: `$('<tên thẻ | id | class>')`

Ví dụ:

`$('h3')` : Lấy đối tượng thẻ **`h3`**.

`$('#my_id')`: Lấy đối tượng có **`id='my_id'`**.

`$('.my_class')`: Lấy tất cả các đối tượng của lớp **`class = 'my_class'`**.

Thư viện hàm trong jQuery

Cú pháp inline của JQuery

\$(document).ready(function){

//nội dung các action đặt tại đây

});

.click(fn): hàm thể hiện hành động click vào 1 element

Ví dụ:

\$('.menu').click(function(){

window.alert('Hello world');

});

Thư viện hàm trong jQuery

.bind('event',fn): hàm cho phép gắn sự kiện cho element. Đồng thời, hành động cho sự kiện sau khi đã gắn

Ví dụ:

```
$('.bottom').bind('click',function(){  
    $('#block1').css('background-color', 'yellow');  
});
```

.addClass('class_name'): hàm cho phép thêm class cho element.

Ví dụ:

```
/* Thêm class 'paragraph' cho div */  
$('div').addClass('paragraph');
```

Thư viện hàm trong jQuery

.removeClass(): loại bỏ tất cả các class từ element

Ví dụ 1: `$('body').removeClass();` /*Loại tất cả các class từ body*/

Ví dụ 2: /* Loại tất các class trước đó từ body. Sau đó, thêm class mới 'wrapper' cho body*/

`$('body').removeClass().addClass('wrapper');`

.hide() hoặc **.hide(mins,fn)**: hàm thực hiện ẩn nội dung của 1 element trên trang

Ví dụ:

```
$('.bottom').click(function(){  
    $('#para').hide();  
});
```

Thư viện hàm trong jQuery

.show() hoặc **.show(mins,fn)**: hàm cho phép hiện nội dung đã bị ẩn.

Ví dụ:

```
$('.bottom').click(function(){  
    $('#para').show();  
});
```

.fadeIn() hoặc **.fadeIn(mins, fn)**: hàm cho phép hiện nội dung đã bị ẩn. Giống hàm *show()*, tuy nhiên cách thứ hiện ra có hiệu ứng khác với hàm *show()*.

Thư viện hàm trong jQuery

.fadeOut() hoặc **.fadeOut(mins, fn)**: Giống như hàm *hide()*.

.toggle(): Thay đổi trạng thái ẩn/hiện

.hover(fnIn, fnOut): hàm sinh ra hành động khi người dùng di chuyển chuột lên element và di chuyển chuột ra ngoài element đó.

- **fnIn**: hàm sinh ra khi di chuyển chuột lên

- **fnOut**: hàm sinh ra khi di chuyển chuột ra ngoài

Vd: `$('#switcher .button').hover(function() {
 $(this).addClass('hover');}, function() {
 $(this).removeClass('hover');});`

Thư viện hàm trong jQuery

.slideDown() và **.slideUp()**: giống *fadeOut()*, *fadeIn()*.

.css('property', 'value') hoặc **.css({'property1': 'value1', 'property2': 'value2'})**: hàm cho phép thay đổi 1 hoặc nhiều thuộc tính CSS của element.

.animate({properties_css, duration, easing, complete}): hàm nâng cao cho tạo hiệu ứng xuất hiện nội dung trang.

```
$('#show').click(function(){  
    $("#block").animate({  
        height:200, width:400, opacity: .5}, 1000, "linear",  
        function(){alert("all done");});  
});
```

Thư viện hàm trong jQuery

.toggleClass('class_name'): hàm xử lý tính năng ẩn/hiện giống như sự kết hợp của 2 hàm *show()*, *hide()*.

Ví dụ:

```
$('.button').click(function(){  
    $('.chapter').toggleClass('hidden');  
});
```

.text() và **.text(val)**: hàm xử lý get, set nội dung bên trong element

Ví dụ:

```
<p class = 'block'>Blog content</p>  
var valElement = $('p.block').text();  
$('p.block').text('value');
```

Thư viện hàm trong jQuery

.empty(): hàm xóa tất cả các phần tử bên trong element

Ví dụ:

//Loại tất cả các phần tử trong element có id = 'dictionary'.

```
$('#dictionary').empty();
```

.append(content): hàm xử lý đưa nội dung trong 'content' vào 1 element

Trong đó: content là 1 biến chứa nội dung .

Ví dụ:

```
$('#dictionary').append(<p>Hello <br>world!</p>);
```

Thư viện hàm trong jQuery

.each(): cú pháp xử lý duyệt (lặp) các phần tử trong mảng.

.html(value) và **.html()**: hàm xử lý set và get nội dung có định dạng html.

.find(expr): hàm xử lý tìm ra element trong file xml

Trong đó: **expr** - element cần tìm trong file. Kết hợp với **.text()** để lấy nội dung của element.

jQuery Ajax

- Các phương thức gọi Ajax:-
 - Load(): tải dữ liệu từ máy chủ và đặt lại dữ liệu vào selector.
`$(selector).load(URL,data,callback);`
Vd: `$("#div1").load("demo_test.txt");`
 - Get(): sử dụng để lấy dữ liệu, yêu cầu có thể được lưu trữ
`$.get(URL,callback);`
Vd: `$.get("demo_test.asp",function(data,status){
 alert("Data: " + data + "\nStatus: " + status);
});`

jQuery Ajax

- Post(): gửi dữ liệu để được xử lý, không được lưu trữ, đánh dấu.

```
$.post(URL,data,callback);  
$.post("demo_test_post.asp",  
{  
  name:"NSK",  
  city:"Ha Noi"  
},  
function(data,status){  
  alert("Data: " + data + "\nStatus: " + status);  
});
```

- Callback: phương thức được thực hiện nếu yêu cầu thành công.