



AJAX JQUERY

AJAX LÀ GÌ ?

1. Ajax là một kỹ thuật viết tắt của chữ *AJAX = Asynchronous JavaScript and XML*).
2. Ajax là một công nghệ giúp chúng ta tạo ra những trang web động mà hoàn toàn không cần reload lại trang nên rất mượt và đẹp.
3. Ajax được viết bằng ngôn ngữ javascript nên nó chạy trên client, tức mỗi máy user chạy độc lập mà không hoàn toàn ảnh hưởng đến nhau.
4. Hiện nay có nhiều thư viện javascript như jQuery hay Angular JS đều hỗ trợ kỹ thuật này.

AJAX TRONG JQUERY

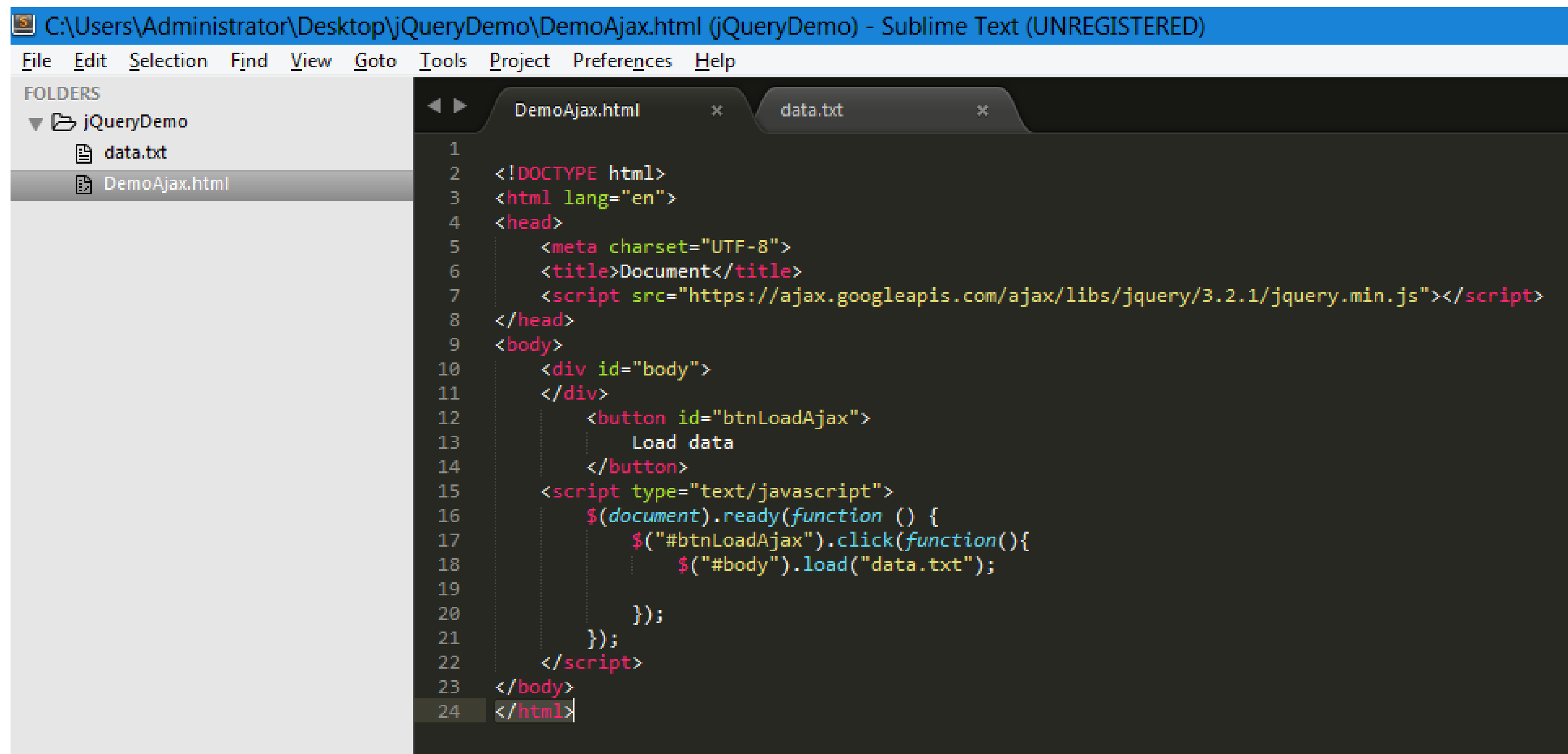
□ *Một số phương thức ajax thông dụng trong jquery.*

- 1. Phương thức jQuery load*
- 2. Phương thức jQuery get*
- 3. Phương thức jQuery post*

❑ Phương thức jQuery load

- Phương thức jQuery load() là một phương pháp AJAX đơn giản, giúp tải dữ liệu từ máy chủ và đặt dữ liệu trả về vào phần tử được chọn.

■ Ví dụ:



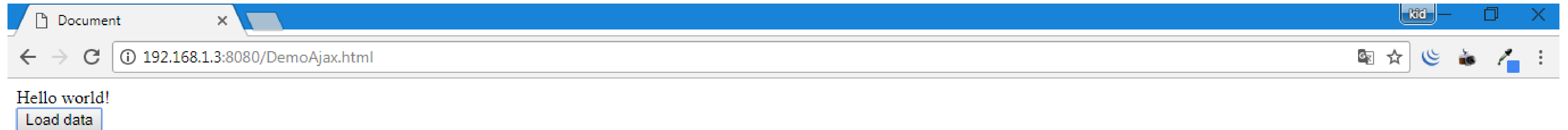
```
C:\Users\Administrator\Desktop\jQueryDemo\DemoAjax.html (jQueryDemo) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  jQueryDemo
    data.txt
    DemoAjax.html
DemoAjax.html
data.txt
1
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <title>Document</title>
7   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
8 </head>
9 <body>
10   <div id="body">
11   </div>
12     <button id="btnLoadAjax">
13       Load data
14     </button>
15     <script type="text/javascript">
16       $(document).ready(function () {
17         $("#btnLoadAjax").click(function(){
18           $("#body").load("data.txt");
19         });
20       });
21     </script>
22   </body>
23 </html>
```

❑ *Phương thức jQuery load*

**Lưu ý: Trước khi sử dụng ajax cần cài đặt server (link hướng dẫn cài server nodejs)*

<https://www.youtube.com/watch?v=vnPemSnnJYY>

■ *Kết quả:*



❑ *Phương thức jQuery get*

- *Phương thức jQuery GET yêu cầu dữ liệu từ một nguồn chỉ định*
- *Truy vấn của GET có thể được lưu lại ở (cached)*
- *Truy vấn bằng GET có những hạn chế về chiều dài dữ liệu*
- *Cú pháp:*

\$.get(URL, callback);

URL: Địa chỉ yêu cầu gửi đến server

callback: Hàm nhận kết quả trả về khi phương thức Get hoàn tất

❑ Phương thức jQuery get

- Ví dụ:

Callback: hàm nhận kết quả trả về từ server

URL: link đến server

```
4 <head>
5 <meta charset="UTF-8">
6 <title>Document</title>
7 <!--Thư viện ajax từ google -->
8 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9 </head>
10 <body>
11 <div id="body">
12
13 </div>
14 <br>
15 <button id="btnGetAjax">
16   Get Ajax
17 </button>
18
19 <script type="text/javascript">
20   $(document).ready(function () {
21     $("#btnGetAjax").click(function(){
22       //Đường dẫn Api trỏ đến server
23       var apiUrl = 'http://597baa9ec577810011876649.mockapi.io/api/demoajax';
24       $.get(apiUrl,function(data,status){
25         //data: Dữ liệu server trả về thông thường json, xml
26         //status: trạng thái (200, 404, 501 ...)
27         for(var i=0;i<data.length;i++)
28         {
29           var content = "ID:" + data[i].id + " - " + data[i].name + "<br />";
30           $("#body").append(content);
31         }
32       });
33     });
34   });
35 </script>
36 </body>
37 </html>
```

192.168.1.6:8080/DemoAjax.html

ID:1 - Name:name 1
ID:2 - Name:name 2
ID:3 - Name:name 3
ID:4 - Name:name 4
ID:5 - Name:name 5
ID:6 - Name:name 6
ID:7 - Name:name 7
ID:8 - Name:name 8
ID:9 - Name:name 9
ID:10 - Name:name 10

Get Ajax

data: dữ liệu trả về từ server dạng json

```
{
  0:{id:1, name:name1},
  1:{id:2, name:name2},
  ...
}
```

❑ *Phương thức jQuery post*

- *Phương thức \$.post() yêu cầu dữ liệu từ máy chủ sử dụng một yêu cầu HTTP POST.*
- *Cú pháp:*

\$.post(URL,data,callback);

URL: *Địa chỉ yêu cầu gửi đến server*

data: *dữ liệu post lên server (ví dụ: {name:cybersoft})*

callback: *Hàm nhận kết quả trả về khi phương thức post hoàn tất*

❑ Phương thức jQuery post

- Ví dụ:

data: data gửi đi với cấu trúc tương ứng.

URL: link đến server

```
10 <body>
11 <div id="body">
12   Name: <input type="text" name="name" id="name">
13   <br /><br />
14 </div>
15 <br>
16 <button id="btnPostAjax">
17   Post Ajax
18 </button>
19 <button id="btnGet">
20   Xem kết quả
21 </button>
22 <script type="text/javascript">
23   $(document).ready(function () {
24     $("#btnPostAjax").click(function(){
25       var nameValue = $("#name").val();
26       //đường dẫn api trở về server
27       var apiUrl = 'http://597baa9ec577810011876649.mockapi.io/api/demoajax';
28       $.post(apiUrl,{name:nameValue},function(data,status){
29         //name là thuộc tính của đối tượng server quy định
30         //name Value là giá trị truyền vào
31         alert(status);
32       });
33     });
34     // Dùng get để lấy dữ liệu về kiểm tra xem phương thức post đã thêm thành công chưa
35     $("#btnGet").click(function(){
36       //đường dẫn api trở về server
37       var apiUrl = 'http://597baa9ec577810011876649.mockapi.io/api/demoajax';
38       $.get(apiUrl,function(data,status){
39         // data: dữ liệu server trả về thông thường là json,xml),
40         // status: trạng thái
41         alert(status)
42         for(var i=0;i<data.length;i++){
43           {
44             var content = "ID:"+data[i].id + " Name:"+data[i].name + "<br />";
45             $("#body").append(content);
46           }
47         }
48       });
49     });
50   }
51 </script>
52 </body>
```

← → ↻ ⓘ 192.168.1.6:8080/DemoAjax.html

Name:

ID:1 Name:name 1
ID:2 Name:name 2
ID:3 Name:name 3
ID:4 Name:name 4
ID:5 Name:name 5
ID:6 Name:name 6
ID:7 Name:name 7
ID:8 Name:name 8
ID:9 Name:name 9
ID:10 Name:name 10
ID:11 Name:cybersoft

Post Ajax

Xem kết quả

callback: hàm nhận kết quả trả về từ server.
(data, status ý nghĩa tương tự \$.get)

❑ Một cách viết khác đầy đủ đối với cả Get, Post

jQuery: \$ajax.get

```
// Dùng get để lấy dữ liệu về kiểm tra xem phương thức post đã thêm thành công chưa
$("#btnGet").click(function(){
    //đường dẫn api trỏ đến server
    var apiUrl = 'http://597baa9ec577810011876649.mockapi.io/api/demoajax';
    $.ajax({
        url : apiUrl, // gửi ajax đến server api
        type : "GET", // chọn phương thức gửi là post
        dataType:"json", // dữ liệu trả về dạng json hoặc text hoặc ...
        contentType: 'application/json; charset=utf-8', // định dạng dữ liệu trả về
        success : function (result,status){
            // result: dữ liệu server trả về (thông thường là json,xml),
            // status: trạng thái
            alert(status)
            for(var i=0;i<result.length;i++)
            {
                var content = "ID:"+result[i].id + " Name:"+result[i].name + "<br />";
                $("#body").append(content);
            }
        }
    });
});
```

jQuery: \$ajax.post

Chú ý kỹ phần **data:Jsonstringify**
khi dùng post dạng này

```
$("#btnPostAjax").click(function(){
    var nameValue = $("#name").val();
    //đường dẫn api trỏ đến server
    var dataJSON = { name: nameValue }; // định nghĩa 1 object json
    var apiUrl = 'http://597baa9ec577810011876649.mockapi.io/api/demoajax';
    $.ajax({
        type : 'POST', // chọn phương thức gửi là post
        url : apiUrl, // gửi ajax đến server api
        dataType:'json', // dữ liệu trả về dạng json hoặc text hoặc ...
        contentType: 'application/json; charset=utf-8', // định dạng dữ liệu trả về
        data : JSON.stringify(dataJSON), // Chuyển từ object json sang string
        success : function (result,status){
            // result: dữ liệu trả về từ server
            // status: trạng thái
            alert(result.name);
        },
        error: function (xhr, ajaxOptions, thrownError) { //Hàm trả về khi post ajax không thành công
            alert(xhr.status);
            alert(thrownError);
        }
    });
});
```

❑ *Một số lưu ý cần nắm về ajax.*

- *Khi sử dụng hình thức post hoặc get đối với dữ liệu là 1 server api thuộc domain khác thì phải được server thuộc domain đó cấp quyền thì mới có thể truy cập được.*
- *Ngoài ra ajax có thể gửi yêu cầu đến server hiện tại thuộc cùng domain một cách đơn giản.*
- *Trong ví dụ trên ta đang dùng api thuộc 1 domain khác.*

☐ *Sortable trong jquery*

- *Sortable là 1 chức năng kéo thả trong jquery được sử dụng rất nhiều ở trang web.*



❑ Sortable trong jquery

- Để thực hiện được tính năng Sortable ta cần thêm 2 thư viện jquery sau:

```
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
```

- Để áp dụng chức năng kéo thả (Sortable) cho selector ta chèn đoạn script sau

Selector(2) liên kết với **Selector(1)**. Các phần tử con của selector(1) có thể dịch chuyển qua **Selector (2)**

Selector(1) chứa các phần tử con có thể kéo thả

True: Cho phép thả phần tử con khi selector đó rỗng

False: Không cho phép thả phần tử con khi selector đó rỗng

```
$( function() {
  $( "ul.droptree" ).sortable({
    connectWith: "ul"
  });

  $( "ul.dropfalse" ).sortable({
    connectWith: "ul",
    dropOnEmpty: false
  });

  $( "#sortable1, #sortable2, #sortable3" ).disableSelection();
} );
```

❑ *Sortable trong jquery*

- *Ngoài connectWidth và dropOnEmpty còn 1 số thuộc tính khác*
 - placeholder: truyền vào classname khi phần tử đó đang dịch chuyển.
 - opacity: độ đậm nhạt của phần tử đang dịch chuyển.
 - items: truyền vào selector phần tử có thể dịch chuyển (các phần tử khác không dịch chuyển được).

```
$( function() {  
  $( "ul.droptrue" ).sortable({  
    connectWith: "ul",  
    placeholder: 'placeholder',  
    opacity: '0.5',  
    items: '.sortable-item'  
  });  
});
```