

# AJAX TRONG PHP

# Giới thiệu Ajax

- Ajax là bộ máy trung gian, xử lý tương tác dữ liệu giữa client và server, với dữ liệu trả về từ server có thể là html, json, xml, text... tùy theo cách chúng ta muốn lấy ra dữ liệu.
- Hiểu đơn giản là ajax giúp cho website load nhanh hơn, thao tác được mượt mà hơn và không phải load lại trang

# Phương thức trong Ajax

- Để gửi dữ liệu đi và lấy dữ liệu về trong Ajax chúng ta có 3 phương thức:

1. **Load**
2. **Post**
3. **Get**

*// Để sử dụng Ajax chúng ta cần nhúng thư viện jquery*

## Phương thức Load

- Giúp bạn có thể đứng ở trang này mà vẫn hiển thị (load data) được từ một trang khác về.

- **Cú pháp:**

**`$("selector").load('data.php');`**

# Phương thức GET

- Giúp bạn có thể đứng ở trang này gửi dữ liệu đi một trang khác để xử lý và sau đó thì lấy dữ liệu về sau khi đã được xử lý ở một trang khác.

- **Cú pháp:**

```
$.get('url', { data1 : 'value1', data2 : {value2 }}, function(datas){
```

```
    // code
```

```
});
```

# Phương thức POST

- Giống GET, giúp bạn có thể đứng ở trang này gửi dữ liệu đi một trang khác để xử lý và sau đó thì lấy dữ liệu về sau khi đã được xử lý ở một trang khác.
- **Cú pháp:**

```
$.post('url', { data1 : 'value1', data2 : {value2 }}, function(datas){
```

```
// code
```

```
});
```

## Ưu nhược điểm

- Qua trên ta đã tìm hiểu về Ajax với các hàm: \$.get(), \$.post(), \$.load().
- Với những hàm trên thì nó sẽ hoạt động nếu request thành công, không xảy ra vấn đề gì khi request.
- Nhưng vấn đề, khi một request thất bại hay chúng ta cần nhiều sự kiểm soát hơn cho mỗi lần ta sử dụng Ajax thì các hàm trên không đáp ứng được. Vậy giải pháp lúc đó là chúng ta sẽ sử dụng sang: \$.ajax();

# Hàm \$.ajax()

- Hàm \$.ajax() của JQuery được sử dụng để thực hiện các request HTTP bất đồng bộ.
- Ba hàm \$.get(), \$.post() và \$.load() được đề cập ở trên có thể coi là một hàm \$.ajax() với những thiết lập có sẵn.
- Cú pháp tổng quát của hàm \$.ajax():
  - \$.ajax(url[, options]);



# Cú pháp tổng quát của hàm \$.ajax():

\$.ajax({options}); // trong đó option sẽ là rất nhiều các lựa chọn, ở đây chúng ta sẽ tìm hiểu một số option thường hay sử dụng.

```
$.ajax({  
  url: '/path/to/file',  
  type: 'default GET (Other values: POST)',  
  dataType: 'default: Intelligent Guess (Other values: xml, json,  
    script, or html)',  
  data: {param1: 'value1'},  
})  
.done(function() {  
  console.log("success");  
})  
.fail(function() {  
  console.log("error");  
})  
.always(function() {  
  console.log("complete");  
});
```

# Tìm hiểu về Json

- Khi sử dụng JQuery Ajax, dữ liệu chúng ta có thể lấy về là: html, json, text...
- JSON là viết tắt của cụm từ "JavaScript Object Notation", là cách thức để mô tả object trong xử lý của java script.
- **Tại sao phải lấy dữ liệu bằng json mà không phải là HTML, TEXT:**

*Thông thường khi lấy danh sách tin tức hay danh sách gì ở database, kết quả trả về của Ajax ta thường dùng JSON, bởi vì khi chúng ta trả về JSON thì trong kết quả sẽ không có những thẻ HTML nên sẽ có tốc độ nhanh hơn, tuy nhiên tại Javascript bạn sẽ phải lặp kết quả đó để in ra.*

# json\_encode

- Để conver giá trị chỉ định thành định dạng JSON, người ta sử dụng hàm `json_encode`.

```
<?php
$arr = array(
    'sv01' => array(
        'id'    => 'sv01',
        'name'  => 'Tung TK 1'
    ),
    'sv02' => array(
        'id'    => 'sv02',
        'name'  => 'Tung TK 2'
    ),
    'sv03' => array(
        'id'    => 'sv03',
        'name'  => 'Tung TK 3'
    )
);

echo json_encode($arr);
```

# json\_decode

- Để nhận và giải mã chuỗi đã mã hóa JSON, người ta sử dụng hàm `json_decode`. Giải mã nói một cách đơn giản là khôi phục dữ liệu đã được mã hoá trở về bản gốc.

```
$json = json_encode($arr);  
var_dump(json_decode($json));
```

# Chuyển dữ liệu dạng Json sang HTML

- Chúng ta cần lặp dữ liệu trả ra dưới dạng json để hiển thị ra các thẻ HTML, sử dụng each();

Ví dụ:

```
$.each(datas, function(key, value) {  
    result += "<tr><td>" + value['id'] + "</td><td>" + value['name'] + "  
        </td></tr>";  
  
});
```

## Ví dụ

- Phân biệt Json và HTML sau khi ajax trả ra dữ liệu
- Demo về Ajax