# Introduction to AJAX with jQuery

C273 – Advanced Web Application Development in PHP

# What is AJAX?

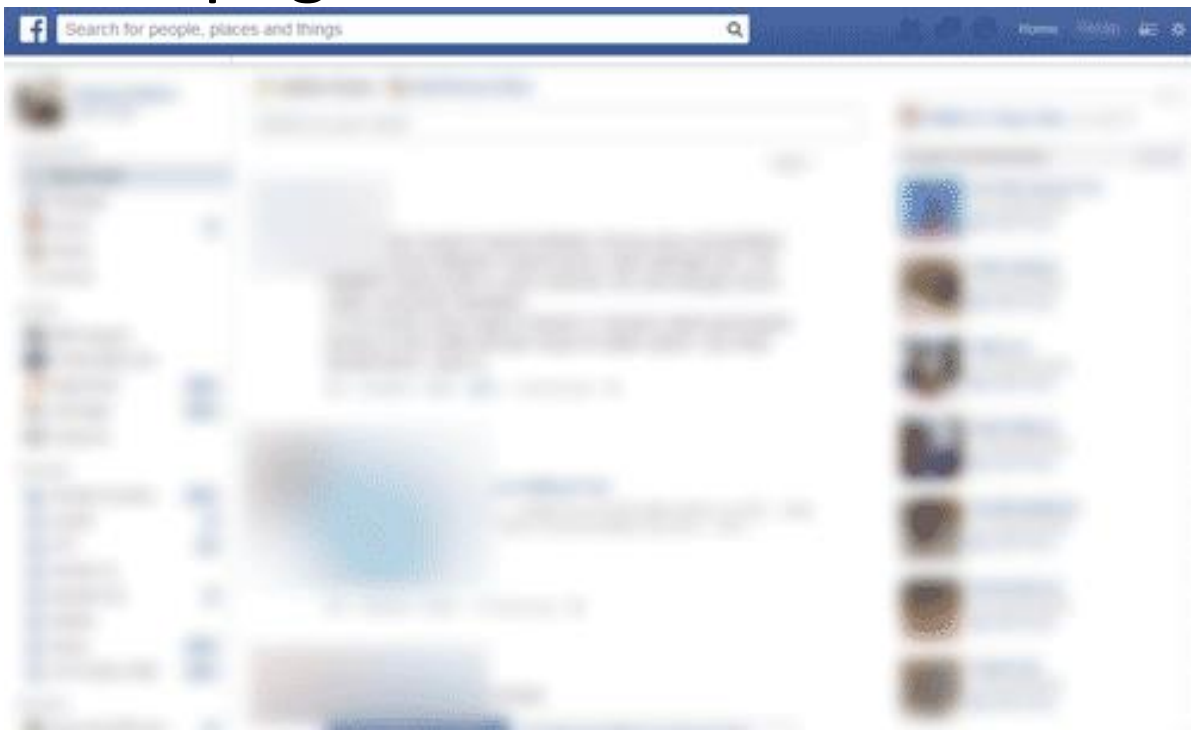
- Asynchronous JavaScript and XML.
- Not a stand-alone language or technology.
- It creates faster, more interactive and user friendly web pages.

# Purpose of ajax

- Prevents unnecessary reloading of a page.
- When we submit a form or refresh a page, the whole page is typically reloaded from the server.  This causes very long waiting times and waste of bandwidth.
- AJAX aims at loading only the necessary information, and making only the necessary changes on the current page without reloading the whole page.

# Example - ajax on facebook

- Facebook Wall uses AJAX to auto-load the older posts without the need for you to reload the entire page.



# Technologies used

AJAX makes use of:

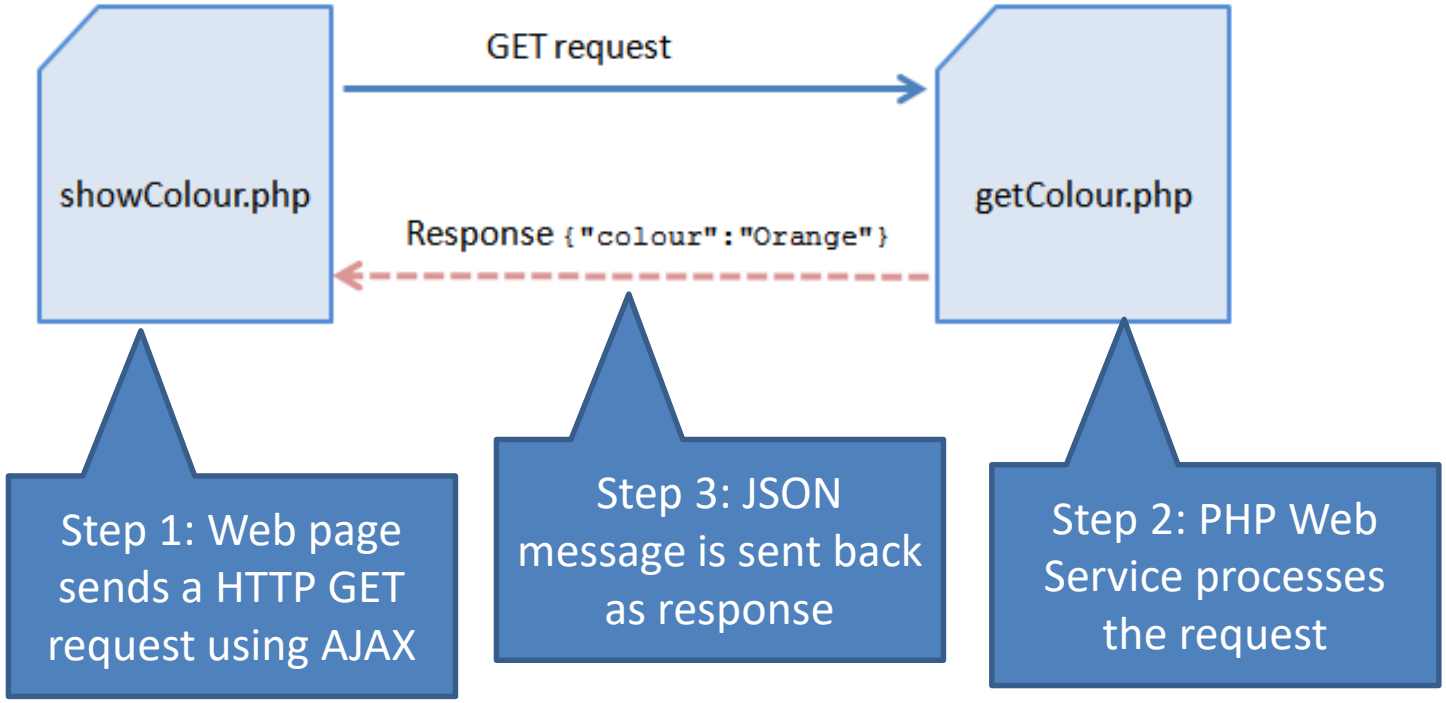
- **jQuery** or JavaScript (for altering the page)
- **JSON** or XML (for data exchange)
- **PHP** / ASP / JSP (server side) is used for creating the web service

Note: This module will be using jQuery, JSON and PHP.

# JSON

- JavaScript Object Notation
- lightweight data-interchange format
- transmit data between a server and web application
- an alternative to XML
- 2 forms
  - Object
    - {"age" : "24","hometown" : "Missoula, MT","gender" : "male"}
  - Array
    - [{"name" : "Jason","age" : "24","gender" : "male"},{"name" : "Kyle","age" : "21","gender" : "male"}]

# AJAX & JSON


showColour.php
GET request
getColour.php
Response {"colour":"Orange"}
Step 1: Web page sends a HTTP GET request using AJAX
Step 3: JSON message is sent back as response
Step 2: PHP Web Service processes the request

# CREATE PHP WEBSERVICE WITH JSON

# PHP: json_encode

* PHP includes the following global function for interacting with JSON data:

| | |
|---|---|
| json_encode(*object*) | returns JSON equivalent for the given object or array or value |

# Exercise 1a

PHP webservice 1a_getFruit.php:

```php
<?php
$fruit['fruit_name'] = "watermelon";
$fruit['image'] = "melon.jpg";
$fruit['description'] = "This fruit is green and red in colour";
echo json_encode($fruit);
?>
```

JSON response

{"fruit_name":"watermelon","image":"melon.jpg","description":"This fruit is green and red in colour"}

# CREATE PHP WEBSERVICE WITH MYSQL AND JSON

# Recap mysqli functions

| Function | Description |
| --- | --- |
| mysqli_connect() | Opens a new connection to the MySQL server |
| mysqli_query() | Performs a query against the database |
| mysqli_fetch_array() | Fetches a result row as an associative, a numeric array, or both |
| mysqli_fetch_assoc() | Fetches a result row as an associative array |
| mysqli_fetch_row() | Fetches one row from a result-set and returns it as an enumerated array |
| mysqli_close() | Closes a previously opened database connection |

# Recap PHP Associative arrays

* An associative array uses text as keys

```
Array
(
    [student_id] => 1234567
    [first_name] => Bob
    [last_name] => Tan
)
```

* A 2-dimensional associative array

```
Array
(
    [0] => Array
        (
            [student_id] => 11111111
            [first_name] => Bob
            [last_name] => Tan
        )

    [1] => Array
        (
            [student_id] => 2222222
            [first_name] => Sally
            [last_name] => Lim
        )

)
```

**Always use mysqli_fetch_assoc to retrieve data from database so that the json_encode function is able to encode the data into name-value pairs**

# JSON encoded data

```
Array
(
    [student_id] => 1234567
    [first_name] => Bob
    [last_name] => Tan
)
```

**json_encode**

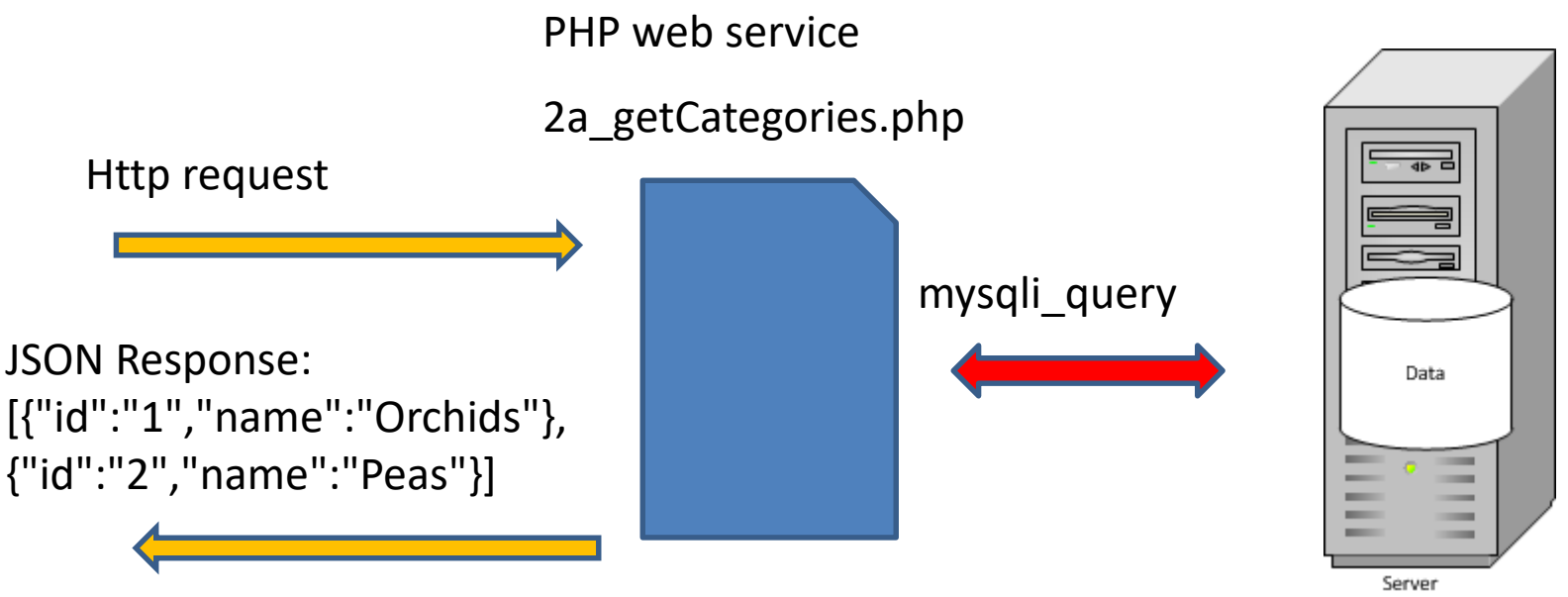
{"student_id":"11111111","first_name":"Bob","last_name":"Tan"}

```
Array
(
    [0] => Array
        (
            [student_id] => 11111111
            [first_name] => Bob
            [last_name] => Tan
        )

    [1] => Array
        (
            [student_id] => 2222222
            [first_name] => Sally
            [last_name] => Lim
        )

)
```

**json_encode**

[{"student_id":"11111111","first_name":"Bob","last_name":"Tan"},{"student_id":"2222222","first_name":"Sally","last_name":"Lim"}]

# Exercise 2a


PHP web service
2a_getCategories.php
Http request
JSON Response:
[{"id":"1","name":"Orchids"},
{"id":"2","name":"Peas"}]
mysqli_query
Data
Server

# Exercise 2a

```php
include ('dbFunctions.php');

$categories = Array();

// SQL query returns multiple database records.
$query = "SELECT id, name FROM flower_categories ORDER BY name";
$result = mysqli_query($link, $query);

while ($row = mysqli_fetch_assoc($result)) {
   $categories[] = $row;
}

echo json_encode($categories);
```

Step1: Write the select query to retrieve data from the flower_category table

Step2: Execute the query using mysqli_query function

Step3: Use mysqli_fetch_assoc function to retrieve the table rows and store into $categories

Step4: Use json_encode function to encode the array into JSON format and echo the output

# AJAX with jQuery

**The $.ajax(*settings*) or $.ajax(*url*, *settings*)**

- Used for sending an Ajax request. The *settings* is an object of key-value pairs:
  - *url*: The request URL
  - *type*: GET or POST.
  - *data*: Request parameters (name=value pairs). Can be expressed as an object (e.g., {name:"peter", msg:"hello"}), or query string (e.g., "name=peter&msg=hello").
  - *dataType*: Expected response data type, such as text, xml, json, script or html.
  - *headers*: an object for request header key-value pairs.

# Exercise 4a


```
$(document).ready(function () {
    $.ajax({
        type: "GET",
        url: "http://localhost/C273_L07Ajax/1a_getFruit.php",
        cache: false,
        dataType: "JSON",
        success: function (response) {
            $('#fruitName').html(response.fruit_name);
            $('#description').html(response.description);
            $('#fruitImage').html(response.image);
        },
        error: function(obj, textStatus, errorThrown) {
            console.log("Error "+textStatus+": "+errorThrown);
        }
    });

});
```

Step 1: AJAX initiates Http GET request and retrieves response as JSON message

Step 2: Retrieve data from object and display using jQuery selector

```
<div id="fruitName">Fruit Name</div>
<div id="description">Description</div>
<div id="fruitImage">Image</div>
```

# What you learnt today

- Identify the purpose of AJAX and how it is used
- Demonstrate understanding of how AJAX works with JSON and PHP
- Demonstrate understanding of json_encode function
- Write a PHP webservice that retrieves data from MySQL database
- Test the PHP webservice using the web browser