

# AJAX and JSON

ผศ. ดร. เกี๊จแกว ฐเนศวร

[kejkaew.tha@kmutt.ac.th](mailto:kejkaew.tha@kmutt.ac.th)

# ตารางเรียน

สัปดาห์	วันที่	หัวข้อ/รายละเอียด	การบ้าน (ผู้ตรวจ)	ส่ง Assignment
1	5/8/2568	Introduction to Web programming, Basic HTML 1	Assignment 1 (KT,KN)	1
	12/8/2568	หยุด		
2	19/8/2568	Basic HTML 2 + CSS 1	Assignment 2 (KT, KN)	1,2
3	26/8/2568	CSS 2	Assignment 3 (KT)	1,2,3
4	2/9/2568	Responsive Web design	Assignment 4 (KT, KN)	2,3,4
5	9/9/2568	JavaScript 1 + DOM	Assignment 5 (KT, KN)	3,4,5
6	16/9/2568	JavaScript 2 + GitHub	Assignment 6 (KT)	4,5,6
7	23/9/2568	JavaScript 3	Assignment 7 (KT, KN)	5,6,7
8	30/9/2568	AJAX + JSON	Assignment 8 (KT, KN)	6,7,8
9	7/10/2568	Node.js 1	Assignment 9 (KT)	7,8,9
10	14/10/2568	Node.js 2	Assignment 10 (KT, KN)	8,9,10
	21/10/2568	หยุด		
11	28/10/2568	Node.js 3	Assignment 11 (KT, KN)	9,10,11
12	4/11/2568	Node.js 4 + MySQL	Assignment 12 (KT, KN)	10,11,12
13	11/11/2568			11,12
14	18/11/2568			12
16	25/11/2568	ส่งโปรเจค (9:00-16:00)	KT	
17	/12/2568	สอบปลายภาค ในตาราง		

# วันนี้เราจะเรียนเรื่อง

---

- AJAX : Asynchronous JavaScript And XML
- JSON: JavaScript Object Notation.
- Assignment 8

# AJAX

---

- Web application: เว็บไซต์แบบไดนามิกที่ให้ความรู้สึกเหมือนแอปพลิเคชันบนเดสก์ท็อป
- AJAX => Asynchronous JavaScript And XML.
- AJAX: a specific style of using JS to call out to the server for more information

# Web Applications

---

- ทุก web page ที่เราทำมาจะประกอบด้วย content, style และ behavior
- Web application คือ web page ที่มีการจัดการข้อมูลของ user ทำให้ผู้ใช้รู้สึกเหมือนกำลังใช้ desktop software หรือ desktop application
- ข้อดีของการทำ web application แทน web page
  - Better user experience
  - Less data sent across the wire
  - Leads to good software architecture:

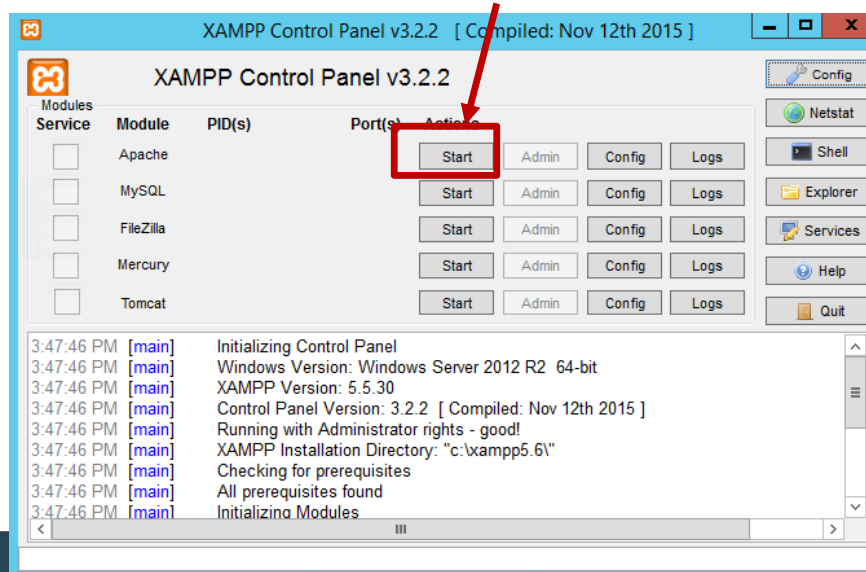
# Local server setup

---

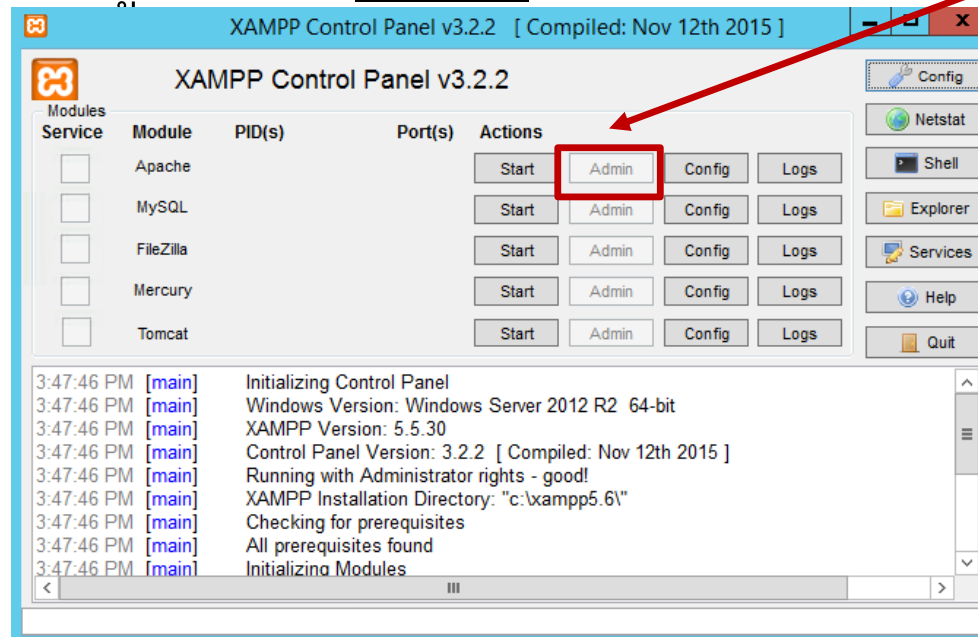
- มี 2 แบบ
  - XAMPP
  - Live server ใน VS code

## ขั้นตอน local server setup: XAMPP

- Download XAMPP (<https://www.apachefriends.org/index.html>)
- Install XAMPP ลงบนเครื่องคอมพิวเตอร์
- ทำการ start Apache ในโปรแกรม XAMPP

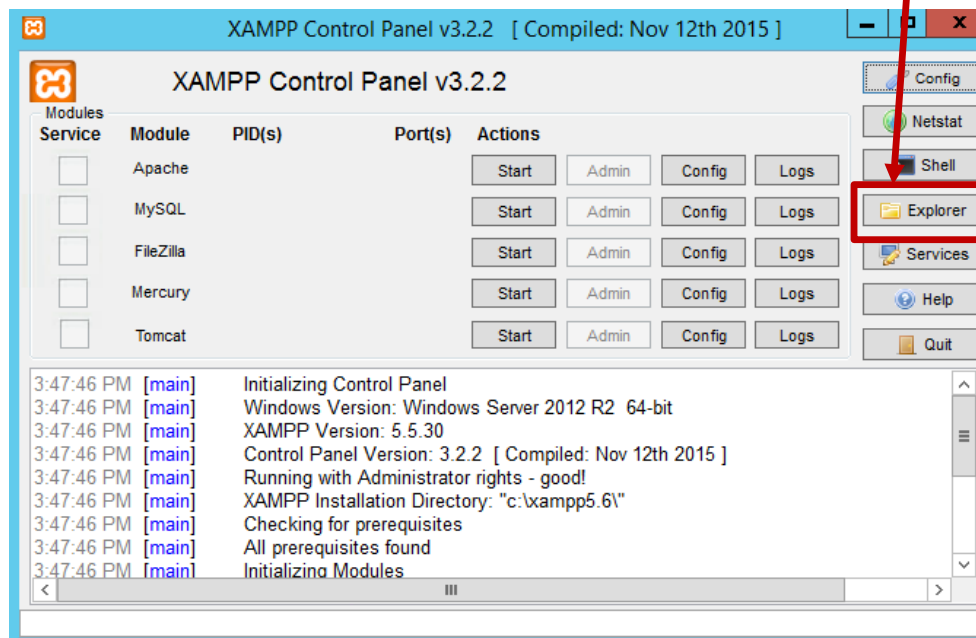


- ทดลอง เรียก <http://localhost> บน browser ของเรา หรือกดที่ Admin ตามรูป ถ้าทำถูกต้องจะมีข้อมูลเกี่ยวกับ XAMPP แสดงขึ้นบน browser





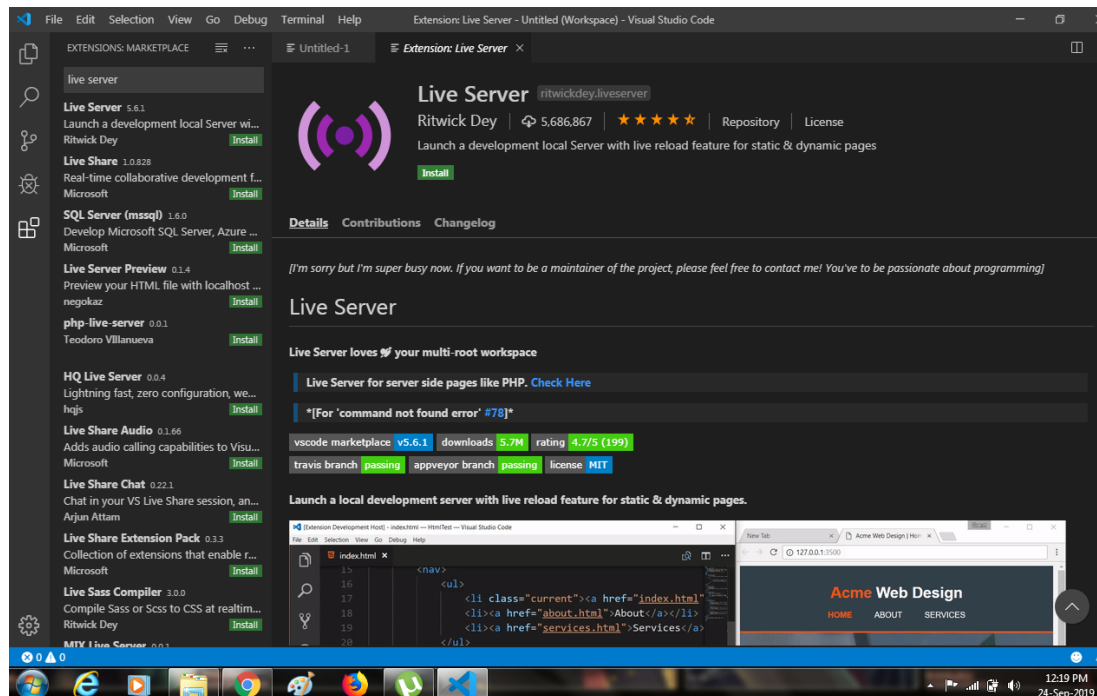
- หา XAMPP folder ที่เราทำการ install ไว้ หรือกดที่ Explorer ตามรูป



- 
- Click ที่ htdocs folder ใน XAMPP folder สร้าง folder ตามชื่อที่เราต้องการ เช่น week8 และทำการ copy or move web files เช่น index.html มาใส่ใน week8
  - เรียก <http://localhost/week8/index.html>

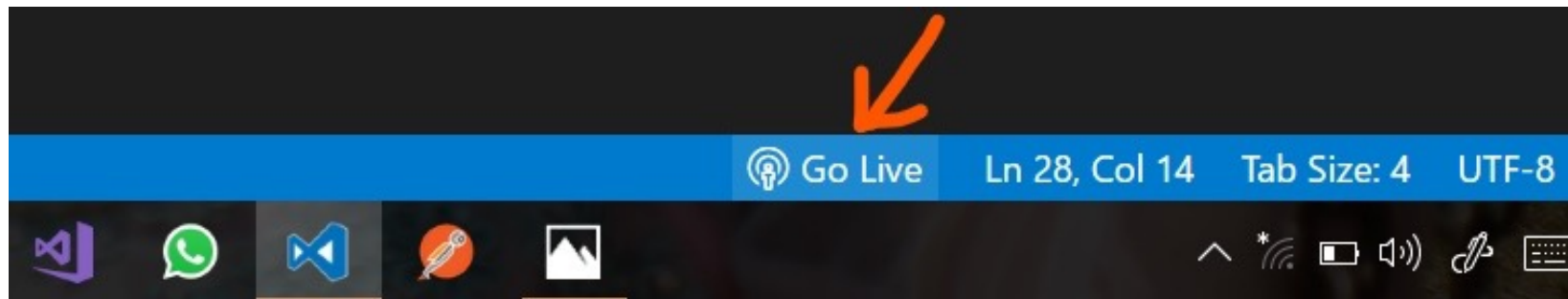
# ขั้นตอน local server setup: Live Server

## ■ Install Live Server extension



# Live Server

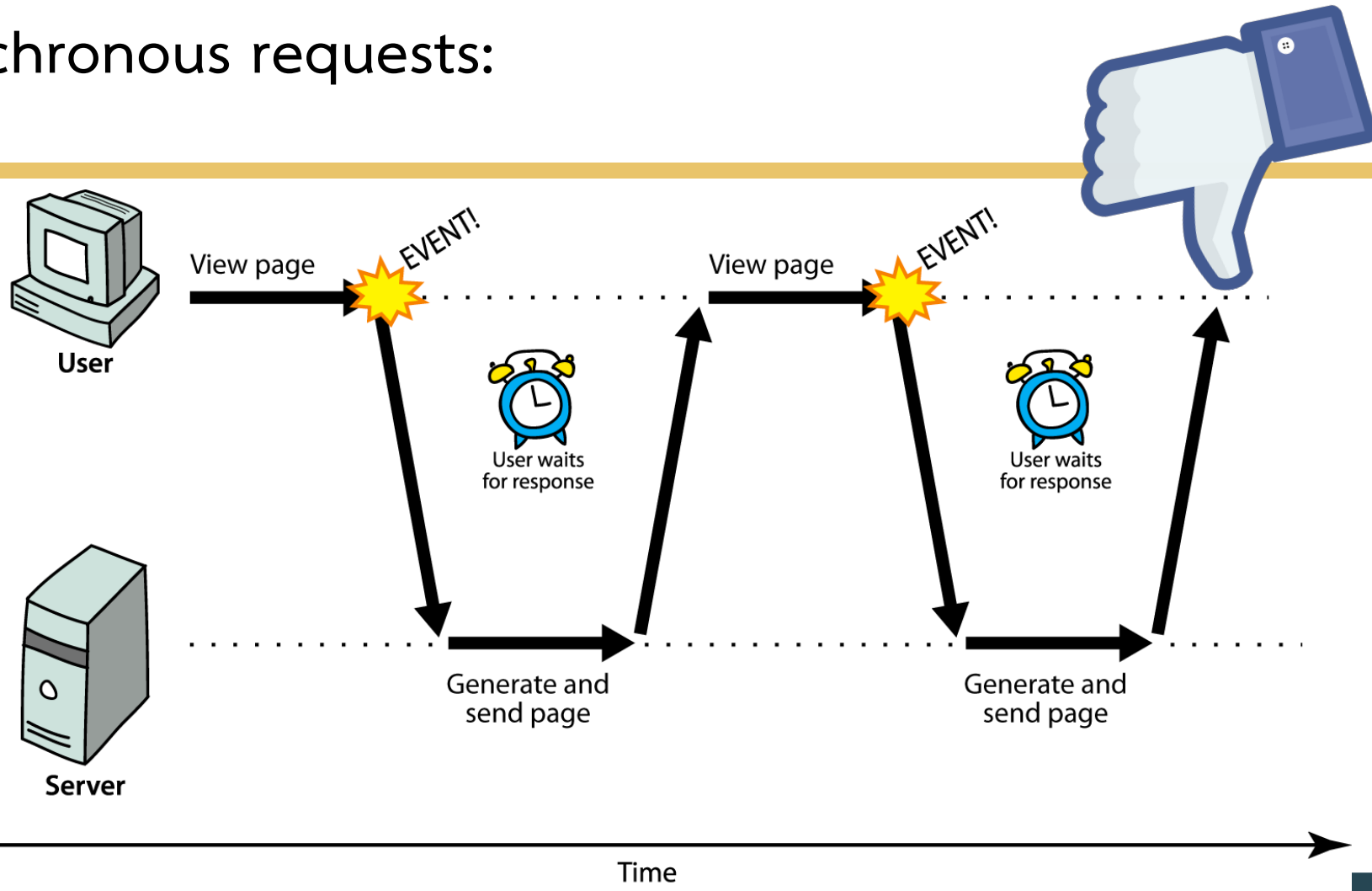
- วิธีการใช้



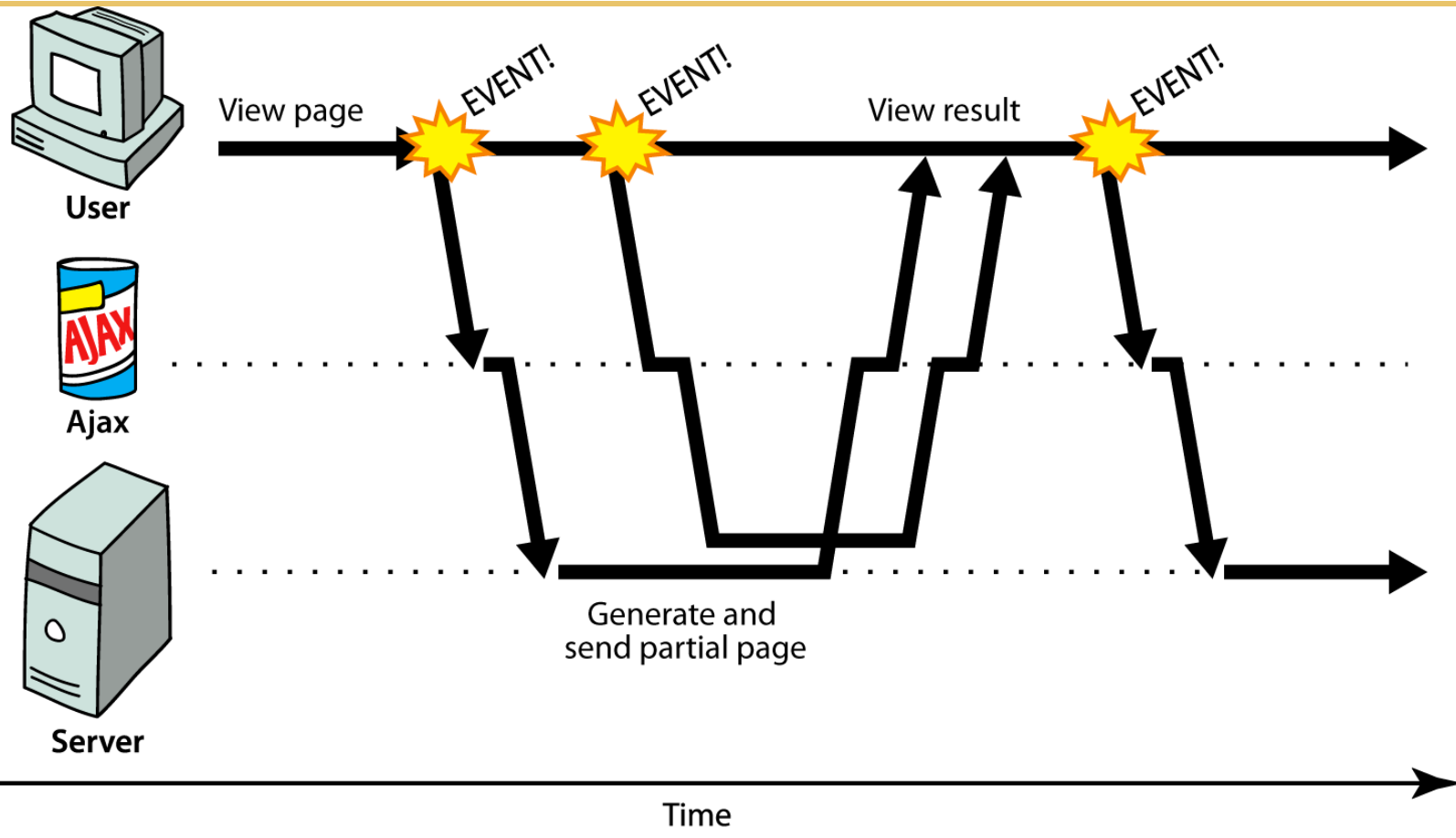
# AJAX

- AJAX คือ การใช้ JavaScript ติดต่อกับ server โดยที่ไม่ต้องทำการ refresh url page หรือ เปลี่ยน url ใหม่
- AJAX ไม่ใช่ programming language.
- หน้าที่ของ AJAX
  - อ่าน data จาก a web server หลังจากที่ page โหลดเสร็จแล้ว
  - สามารถ update web page โดยไม่ต้อง reloading web page ใหม่
  - ส่ง data ไปยัง web server
  - สามารถ ส่งและ update ข้อมูลกับ server แบบ asynchronously

# Synchronous requests:



## Asynchronous requests:



## AJAX - The XMLHttpRequest Object

---

- ในการติดต่อกับ server, AJAX จะใช้ XMLHttpRequest object
- Syntax สำหรับสร้าง XMLHttpRequest object:

```
variable = new XMLHttpRequest();
```

- ตัวอย่างเช่น

```
var xhr = new XMLHttpRequest();
```

//Syntax for creating an XMLHttpRequest object:



# XMLHttpRequest Object Methods

Method	Description
<code>new XMLHttpRequest()</code>	Creates a new XMLHttpRequest object
<code>open(method, url, async, user, psw)</code>	Specifies the request method: the request type GET or POST url: the file location async: true (asynchronous) or false (synchronous) user: optional user name psw: optional password
<code>send()</code>	Sends the request to the server  Used for GET requests

# XMLHttpRequest Object Methods

Method	Description
send(string)	Sends the request to the server. Used for POST requests
onload()	The function called when an <u>XMLHttpRequest</u> transaction completes successfully.
onerror()	The function called when an <u>XMLHttpRequest</u> transaction fails due to an error.

# XMLHttpRequest Object Properties

---

Property	Description
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the <a href="#">Http Messages Reference</a>

## AJAX - Send a Request To a Server

- สำหรับการส่งข้อมูลไปยัง server ใน AJAX เราจะใช้ `open()` and `send()` methods ของ `XMLHttpRequest` object:

```
var xhr = new XMLHttpRequest();
```

```
xhr.open(method(GET/POST), url, [async/sync]);
```

```
xhr.send();
```

- ตัวอย่างเช่น

```
xhr.open("GET", "ajax.txt", true);
```

```
xhr.send();
```

## GET or POST?

- สำหรับ method การส่งข้อมูลเราสามารถใช้ได้ทั้ง GET หรือ POST
- GET ง่ายและประมวลผลได้เร็วกว่า POST
- แต่เราจะใช้ POST เมื่อ
  - ต้องการ update file or database ใน server
  - ส่งข้อมูลที่มีขนาดใหญ่ไปยัง server (POST has no size limitations).
  - ส่ง user input ที่มีอักขระแปลกๆ ไปยัง server

## Asynchronous - True or False?

---

- Server requests ควรมีการส่งแบบ asynchronously.
- ดังนั้น async parameter ใน open() method ควรมีค่าเป็น true
- ส่งแบบ asynchronously คือการที่ JavaScript ไม่จำเป็นต้องรอว่า server จะตอบกลับมาเมื่อไร ทำให้ JS
  - ทำงานอื่นต่อไปได้ในขณะที่รอ server ส่งข้อมูลที่ร้องขอกลับมา
  - หลังจากนั้นค่อยกลับมาจัดการกับข้อมูลที่ server ส่งมา

## AJAX - Server Response

- การจัดการข้อมูลที่ server ส่งมา มีการทำได้ 2 แบบคือ แบบที่ success และ แบบ failure
- Syntax

```
var xhr = new XMLHttpRequest();  
xhr.open(method(GET/POST), url, [async/sync]);  
xhr.onload = function() { /* handle success */ };  
xhr.onerror = function() { /* handle failure */ };
```

## AJAX - Server Response

- ถ้า server ส่งข้อมูลกลับมา เราจะใช้ข้อมูลนั้นได้อย่างไร
- ทำได้ 2 แบบคือ

Property	Description
responseText	get the response data as a string
responseXML	get the response data as XML data

ตัวอย่างในการใช้

```
document.getElementById("demo").innerHTML = xhr.responseText;
```



## สรุปการทำงานของ AJAX

---

```
var xhr = new XMLHttpRequest();  
xhr.open(method, url, [async/sync]);  
//To send a request to a server  
//method: the type of request: GET or POST  
//url: the server (file) location  
//async: true (asynchronous) or false (synchronous)  
xhr.onload = function() { /* handle success */ };  
xhr.onerror = function() { /* handle failure */ };  
xhr.send();  
//Sends the request to the server
```

## ตัวอย่าง ใน main.js

```
window.onload = pageLoad;

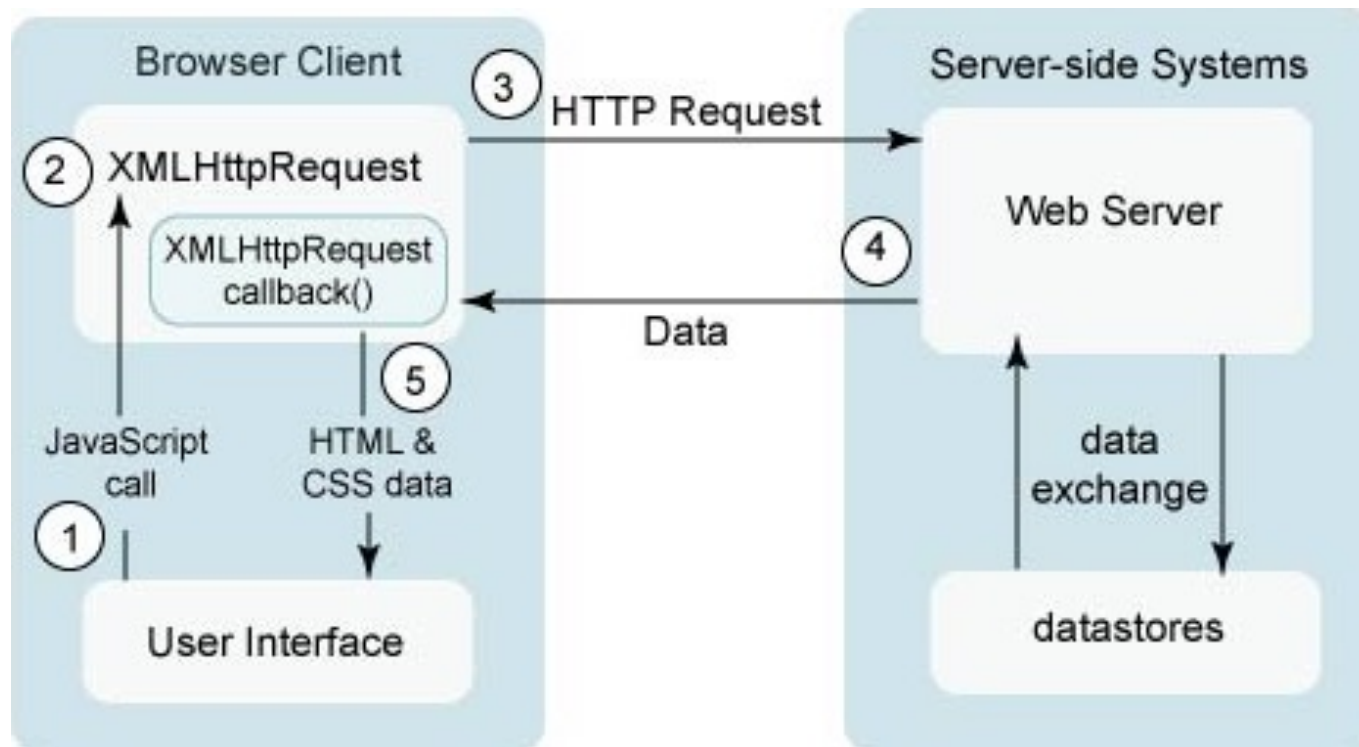
function pageLoad(){
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "data.txt");
    xhr.onload = function() {
        alert(xhr.responseText);
    };
    xhr.onerror = function() {
        alert("ERROR!");
    };
    xhr.send();
}
```

localhost says

Ajax (also AJAX /ˈeɪdʒæks/; short for "Asynchronous JavaScript and XML")<sup>[1][2]</sup> is a set of web development techniques using many web technologies on the client side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows web pages and, by extension, web applications, to change content dynamically without the need to reload the entire page.<sup>[3]</sup> In practice, modern implementations commonly utilize JSON instead of XML.

OK

## สรุป Ajax process



# JSON

---

- JavaScript Object Notation
- คือ data format ที่เป็น text แต่อยู่ในรูปแบบของ JavaScript objects
- ใช้สำหรับแลกเปลี่ยน data ระหว่าง browser กับ server โดยข้อมูลจะเป็น text เท่านั้น
- ทำไมต้องใช้ JSON
  - สามารถเปลี่ยนจาก text เป็น JavaScript object ได้ง่ายแค่ใช้คำสั่งเดียว

# JSON Syntax

---

- JSON syntax เหมือนกับ JavaScript object notation syntax:
  - Data is in name/value pairs
  - Data is separated by commas
  - Curly braces {} hold objects
  - Square brackets [] hold arrays

## ตัวอย่าง JSON file

---

```
{    // no variable assignment (ตัวสีแดงคือ comment)
  "first_name": "Jane",
  "last_name": "Doe",
  "age" : 23, // numbers can be here without quotes
  "female": true  // booleans can be here without quotes
}
```

# JSON Values

In **JSON**, *values* must be one of the following data types:

- a string
- a number
- an object (JSON object)
- an array
- a boolean
- null

In **JavaScript** *values* can be all of the above, plus any other valid JavaScript expression, including:

- a function
- a date
- undefined

In **JSON**, *string values* must be written with double quotes:

## Array as an object

---

- In JavaScript, **arrays** use **numbered** indexes.
- In JavaScript, **objects** use **named** indexes.
- Two ways to initialize an array
- ตัวอย่าง 1

```
51 // array object
52 var person = [];
53 person["firstName"] = "John";
54 person["lastName"] = "Doe";
55 person["age"] = 46;
56 var x = Object.keys(person).length;
57 console.log(x);
```



## ตัวอย่าง 2

---

```
59  var person = {  
60      firstName: "John",  
61      lastName: "Doe",  
62      age: 46  
63  };  
64  console.log(person);
```

# Accessing Object Properties

- เราสามารถ access object properties ได้ 2 แบบ

objectName.propertyName

objectName["propertyName"]

```
59 var person = {  
60     firstName: "John",  
61     lastName: "Doe",  
62     age: 46  
63 };  
64 console.log(person);  
65 console.log(person.lastName);  
66 console.log(person["lastName"]);
```

## JSON - Evaluates to JavaScript Objects

---

- The JSON format เหมือนกับ JavaScript objects.
- ใน JSON keys จะต้องเป็น strings ซึ่งจะอยู่ใน double quotes

JSON

```
{ "name": "John" }
```

- 
- ใน JavaScript, keys สามารถเป็นได้ทั้ง strings, ตัวเลข หรือ ชื่อที่ไม่มี quote  
JavaScript

```
{ name:"John" }
```

- ตัวอย่างของ JS Object
  - DOM elements
  - document, window
  - XMLHttpRequest

# JavaScript Objects and JSON

---

- JSON คือ วิธีเก็บค่าข้อมูล โดยใช้ JavaScript Objects
- สิ่งที่ต้องรู้สำหรับ JSON methods คือ
  - การเปลี่ยน JSON <-> JavaScript objects
- `JSON.parse( /* JSON string */ )`
  - converts JSON string into JavaScript object
- `JSON.stringify( /* Javascript Object */ )`
  - converts a JavaScript object into JSON string

## What is JSON used for?

---

- Web services use JSON to communicate
- Web servers store data as JSON files
- Databases sometimes use JSON to store, query, and return data

## JSON limitations

---

- JSON ไม่สามารถใช้กับ data types บางอย่างได้ ได้แก่
  - Function
  - Date
  - RegExp
  - Error

## ตัวอย่าง JSON file : data\_2.json

```
{
  "people1":{
    "first_name": "Jane",
    "last_name": "Doe",
    "age" : 23,
    "female": true
  },
  "people2":{
    "first_name": "John",
    "last_name": "Dee",
    "age" : 30,
    "female": false
  }
}
```



# ลองใช้ AJAX อ่าน JSON file

Jane Doe Age: 23

John Dee Age: 30

```
function pageLoad(){
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "data_2.json");
    xhr.onload = function() {
        var jsdata = JSON.parse(xhr.responseText);
        console.log(jsdata);
        display(jsdata);
    };
    xhr.onerror = function() { alert("ERROR!"); };
    xhr.send();
}

function display(people){
    console.log(Object.keys(people).length);
    var showdiv = document.getElementById("people")
    var keys = Object.keys(people);
    for(var i =0; i< keys.length;i++){

        var temp = document.createElement("p");
        temp.innerHTML = people[keys[i]].first_name + " "
        + people[keys[i]].last_name + " Age: " + people[keys
        [i]].age;
        showdiv.appendChild(temp);
    }
}
```

## Assignment 8:

---

- Download assignment 8
- ทำการสร้าง JSON file ที่มีข้อมูลสินค้า 9 ข้อมูล โดยสินค้าแต่ละชิ้นจะประกอบไปด้วย ชื่อ รูป brand และราคา เช่น ขายเสื้อ 9 แบบ แต่ละแบบประกอบด้วย รูป brand name และ ราคา
- ใช้ AJAX อ่าน JSON file และนำข้อมูลใน JSON file มาแสดงผลใน index.html



A.K.O.O Clothing

Price: 200 Baht



Abercrombie & Fitch

Price: 300 Baht



Gap Inc

Price: 400 Baht



## Week 8: Classroom game

---

- ทำ classroom game ใน LEB2 ก่อนสัปดาห์หน้า