

AJAX

6:19, 7:15

Asynchronous JavaScript and XML

- A technology based on: *non-blocking event driven architecture*
 - JavaScript
 - DOM
 - Server side script
- Concept: Client-Driven Communication, Server-Side Processing

Classical case

9:48

เว็บที่ไป

①

Web Page

request

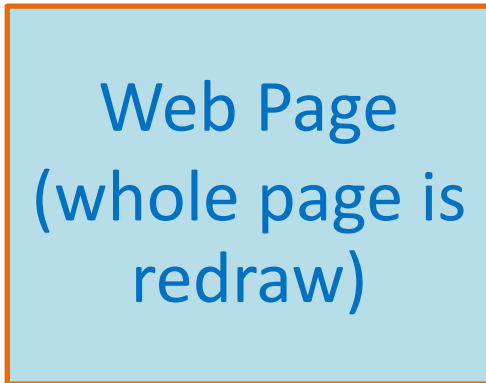
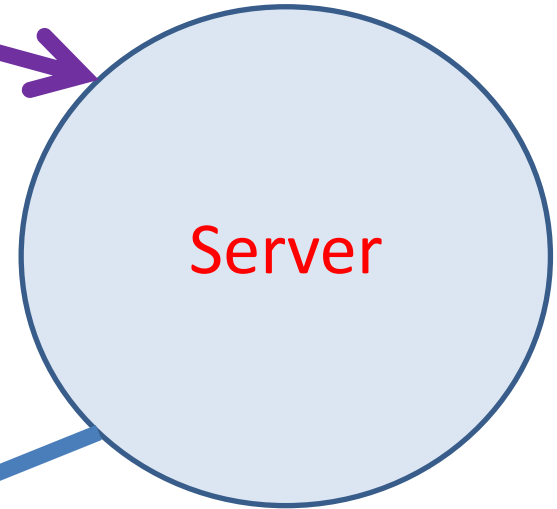
Server

output 9mb html ที่นอน

Web Page
(whole page is
redraw)

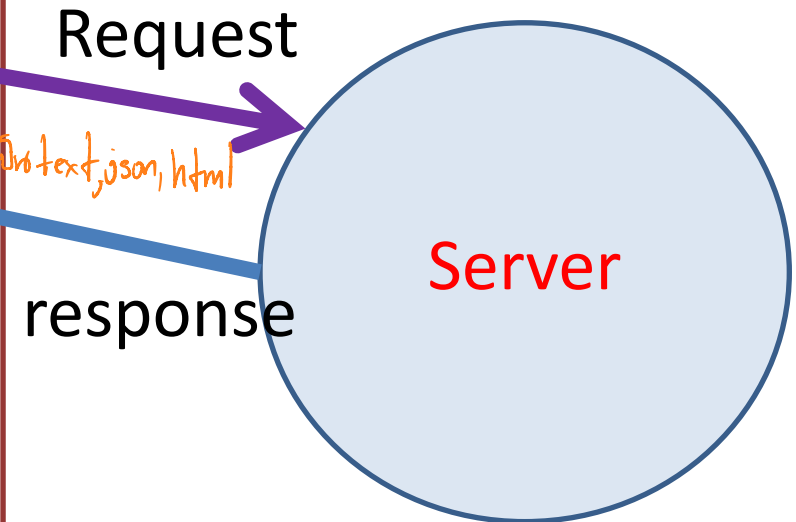
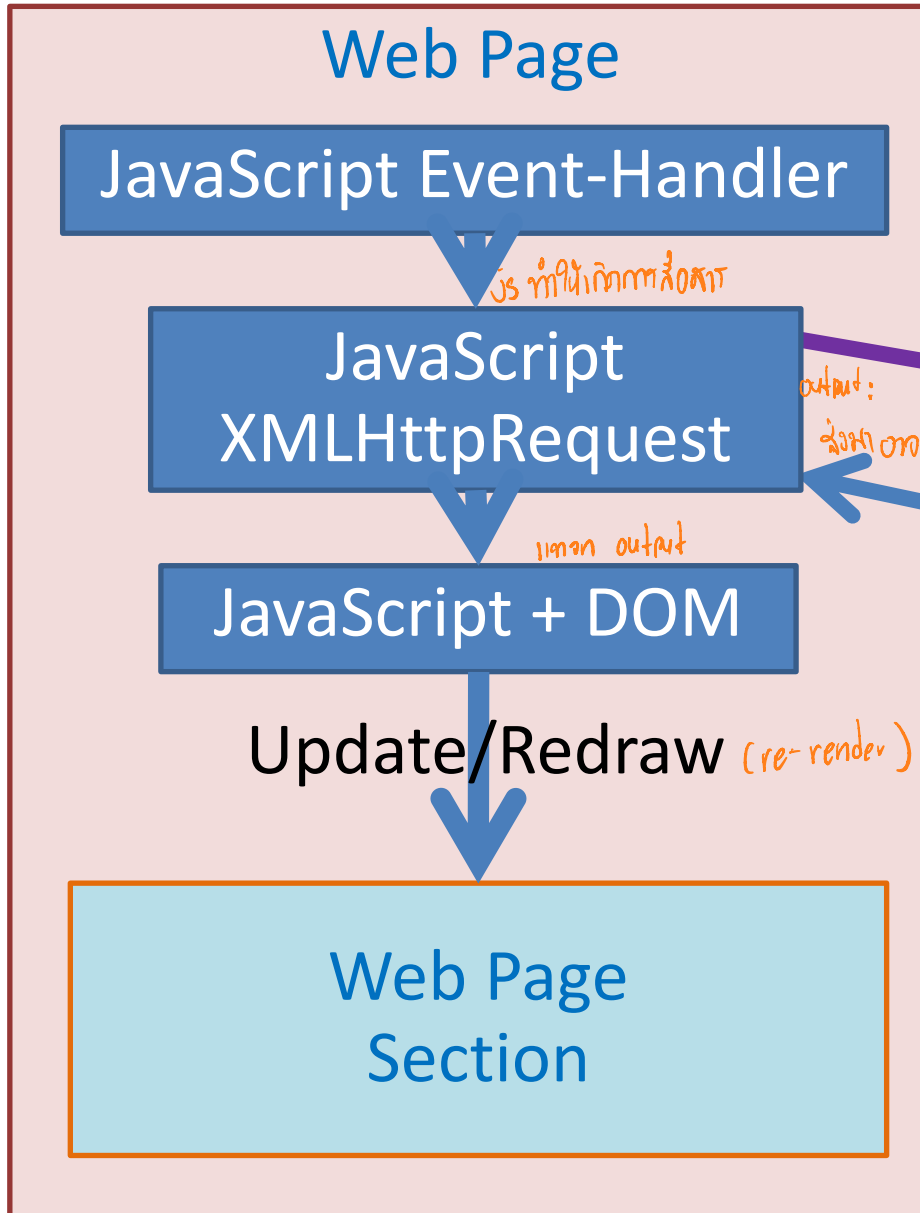
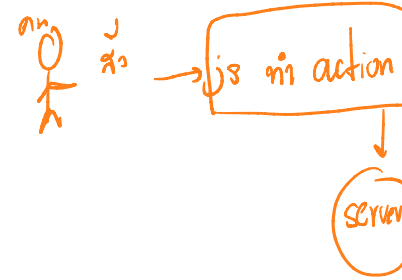
response

②



10:48

Ajax case



Client-Side Communication

- Achieved by
 - XMLHttpRequest Object
 - XMLHttpRequest Object Methods

rough object

Method	Description
abort()	Cancels the current request
getAllResponseHeaders()	Returns all HTTP headers as a String type variable
getResponseHeader()	Returns the value of the HTTP header specified in the method
open()	Specifies the different attributes necessary to make a connection to the server; allows you to make selections such as GET or POST (more on that later), whether to connect asynchronously, and which URL to connect to
setRequestHeader()	Adds a label/value pair to the header when sent
send()	Sends the current request

14.11

XMLHttpRequest

XMLHttpRequest Object Properties

Property	Description
^{เมื่อ state เปลี่ยน} onreadystatechange	Used as an event handler for events that trigger upon state changes
^{สถานะปัจจุบัน (สถานะ: การโหลด)} readyState	Contains the current state of the object (0: uninitialized, 1: loading, 2: loaded, 3: interactive, 4: complete) ^{สถานะปัจจุบัน / สถานะ}
^{ได้รับมาหรือไม่? สถานะ? สถานะ?} responseText	Returns the response in string format
^{ส่งคืน XML} responseXML	Returns the response in proper XML format
^{status http} status	Returns the status of the request in numerical format (regular page errors are returned, such as the number 404, which refers to a not found error)
^{ข้อความอธิบาย status http} statusText	Returns the status of the request, but in string format (e.g., a 404 error would return the string Not Found)

20:08

เช็คตอนนี้สิ dix

- Cross Browser Usage
 - IE and non-IE

code ใช้ browser

```
var xmlhttp = false;

try {
    xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
    alert ("You are using Microsoft Internet Explorer.");
} catch (e) {
    xmlhttp = false;
}
```

ไปดูไฟล์เครื่องตัวเอง

```
if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
    xmlhttp = new XMLHttpRequest();
    alert ("You are not using Microsoft Internet Explorer");
}
```

๑๓:๔๙

code เก็บ browser อีจีส ใช้ตามงานเว็บ web browser ฝั่ง

```
var xmlhttp;
```

```
if (window.ActiveXObject) {
```

```
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
```

```
} else {
```

```
    xmlhttp = new XMLHttpRequest();
```

```
}
```


20:56 minutes

- Sending a request to Server

```
function makerequest(serverPage, objID) {  
    var obj = document.getElementById(objID);  
    xmlhttp.open("GET", serverPage);  
    xmlhttp.onreadystatechange = function() {  
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200)  
        {  
            obj.innerHTML = xmlhttp.responseText;  
            // update text in xml text field obj  
        }  
    }  
    xmlhttp.send(null);  
}
```

The .innerHTML is useful for returning or replacing the content of HTML elements

****innerHTML is not part of W3C DOM but it is supported by all major browser****

39.52

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title>Sample 2_1</title>
```

```
<meta http-equiv="Content-Type"
```

```
content="text/html; charset=iso-8859-1" />
```

```
<script type="text/javascript">
```

```
<!--
```

```
var xmlhttp = false;
```

```
try {
```

```
    xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
```

```
    alert ("You are using MS IE.");
```

```
} catch (e) {
```

```
    xmlhttp = false;
```

```
}
```

```
if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
```

```
    xmlhttp = new XMLHttpRequest();
```

```
    alert ("You are not using MS IE");
```

```
}
```

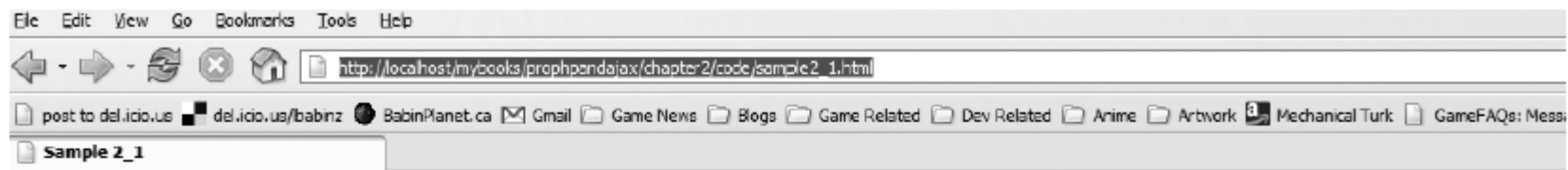
```
function makerequest(serverPage, objID) {  
    var obj = document.getElementById(objID);  
    xmlhttp.open("GET", serverPage);  
    xmlhttp.onreadystatechange = function() {  
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200)  
        {  
            obj.innerHTML = xmlhttp.responseText;  
        }  
    }  
    xmlhttp.send(null);  
}  
//-->  
</script>  
</head>
```

33:38

```
<body onload="makerequest ('content1.html','hw')">
  <div align="center">
    <h1>My Webpage</h1>
    <a href="content1.html" onclick=
      "makerequest('content1.html','hw');
      return false;"> Page 1</a> |
    <a href="content2.html" onclick=
      "makerequest('content2.html','hw');
      return false;">Page 2</a> |
    <a href="content3.html" onclick=
      "makerequest('content3.html','hw');
      return false;">Page 3</a> |
    <a href="content4.html" onclick=
      "makerequest('content4.html','hw');
      return false;">Page 4</a>
    <div id="hw"></div>
  </div>
</body>
</html>
```

id is update

link href is func string



My Webpage

[Page 1](#) | [Page 2](#) | [Page 3](#) | [Page 4](#)

Page 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ASP.Net Core MVC & Ajax Example



Page 3

Controller

```
public IActionResult Message02()
{
    string msg = "<ul><li>topic01</li><li>topic02</li><li>topic03</li><li>topic04</li><li>topic05</li></ul>";
    return Content(msg, "text/plain", Encoding.UTF8);
}
```

ASP.Net Core MVC & Ajax Example

View

```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Ajax testing</title>
  <script>
    var xmlhttp = new XMLHttpRequest();
    ...
    function MakeReq(serverpage, objID) {
      let obj = document.getElementById(objID);
      xmlhttp.open("GET", serverpage);
      xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
          obj.innerHTML = xmlhttp.responseText;
        }
      }
      xmlhttp.send(null);
    }
  </script>
</head>
<body>
  <div>
    <h2>Part I</h2>
    click here ==>
    <a href="" onclick="MakeReq('https://localhost:5001/Ajax/Message01', 'p01'); return false;">Hello</a>
    <div id="p01"></div>
  </div>
  <div>
    <h2>Part II</h2>
    click here ==>
    <a href="" onclick="MakeReq('https://localhost:5001/Ajax/Message02', 'p02'); return false;">Test</a>
    <div id="p02"></div>
  </div>
</body>
</html>
```