

Module 4

Input type

- Text = ข้อความ
- Password = ใช้ input password ตัวอักษรจะเป็น * และ copy ไม่ได้
- Submit = ใช้ submit form
- Reset = reset to default value
- Radio = ใช้เลือกแค่ 1 อัน จากหลายจุด
- Checkbox = ใช้เลือกหลายคำตอบ
- Button = ปุ่ม
- Color = รับสี
- Date = รับวัน
 - มี attribute max='2018-12-31' เวลาสูงสุดที่เลือกได้
 - min='2000-01-02' คือ เวลาขั้นต่ำที่เลือกได้
- Datetime-local เวลา
 - ไม่ support แล้วส่วนใหญ่ ใน firefox safari ie 12 หรือ ต่ำกว่า
- Email รับ email และจะ varidate รูปแบบเมลให้
- File ใช้ในการส่งไฟล์
- Month รับเดือน
 - ไม่ support แล้วส่วนใหญ่ ใน firefox safari ie 12 หรือ ต่ำกว่า
- Number ตัวเลข
 - มีพารามิเตอร์ min = '1' ต่ำสุด 1
 - Max='5' มากสุด 5'
- Range รับค่าช่วงตัวเลข
 - หากไม่กำหนด min max มันจะ default 0-100
- Search เหมือนช่องข้อความปกติเลย
- Tel ใช้เบอร์
 - ใช้ได้แค่ใน safari 8
- Time ใส่เวลา แบบไม่มีวันและ timezone
- Url ใส่ url และ มันจะ varidate ข้อมูล Url ให้
- Week รับสัปดาห์
 - ไม่ support แล้วส่วนใหญ่ ใน firefox safari ie 11 หรือ ต่ำกว่า

Input attribute

- Disable = ทำให้ เราไม่สามารถทำอะไรกับ input ช่องนั้นได้ (โดน disable ไว้)
- Max = ใช้กับตัวเลข หรือวัน กำหนด ค่ามากที่สุด (html5)
- Maxlength = ตัวอักษรมากที่สุด ใช้กับพวก อะไรก็ตามที่รับเป็น string
- Min = ใช้กับตัวเลข หรือวัน กำหนด ค่าน้อยสุด (html5)

- Pattern = ใช้ในการกำหนด regular expression หรือก็คือ format ในการ input
 - [A-Za-z]{3} คือ a-Z 3 ตัว
 - [0-9]{4} คือ 0-9 4 ตัว
- Readonly กำหนดว่า input ช่องนี้ให้อ่านได้เท่านั้น แก้ไขไม่ได้
- Required บังคับว่าต้องกรอกข้อมูล,
- Size คือขนาดของ ตัวอักษรที่ input
- Step ใช้กับ type number
 - ไม่ support ใน ie9
 - Step 2 คือ รับได้แค่ 2 4 6 8
 - Step 3 คือ 3 6 9 12
- Value คือ กำหนดค่าใน input field

Module 5

Ajax base js (Asynchronous JavaScript and XML)

การที่เว็บจะโต้ตอบกับผู้ใช้ ได้ จะต้องใช้ http request
ใน js จะมี object XMLHttpRequest ในการเรียกใช้ ajax

Support ทุก เบราว์เซอร์

Asynchronous คือ การแบ่ง เธรด cpu ในการประมวลผล แล้วเมื่อที่ประมวลผลเสร็จ เราจะได้ผลลัพธ์กลับมา แต่การทำงานจะไม่ตามลำดับ และ เราจะได้ผลลัพธ์เมื่อตอนที่มันเสร็จแล้วเท่านั้น

ประโยชน์ของการทำ ajax

- ไม่ต้องมานั่งรอการส่ง http request ไปยัง server แล้วโปรแกรมค่อยทำงาน มันสามารถ ส่ง http และยังทำงานอยู่ต่อได้เลย
- ไม่ reload page ใหมทั้งหน้า
- ไม่เปลืองทรัพยากรในการประมวลผล เพราะไม่ reload จึงไม่ต้องโหลดหน้าใหม่ทั้งหมด

ใช้อย่างไร

ส่วนใหญ่จะใช้ในการดึงข้อมูลมาจากหน้าเว็บอื่นๆ และ นำมา render ในหน้า page ด้วย js

ตัวอย่าง

การที่เรา img , script , iframe , video , audio src ก็ เป็นการ http request get เหมือนกัน

ตัวอย่างการใช้ ajax ใน js

```
var request = new XMLHttpRequest();  
var url = "http://contoso.com/resources/...";  
request.open( "GET", url );  
request.send();
```

1 ประกาศ object XMLHttpRequest เก็บใน request

3 ใช้ในการกำหนด method http กับ url ที่จะส่งไป

4 ส่ง ปกติแล้วการส่งมันจะ default asynchronous

หรือ

ถ้าอยากได้แบบ synchronous

```
request.open( "GET", url , false);
```

ถ้าแบบล่าง จะเป็นการ รอ ทำให้เป็น synchronous เพราะปกติ ถ้าเราไม่ใส่ false มันจะเป็น asynchronous โดย default

Handling Http error

```
function tryMyLuck() {  
    var request = new XMLHttpRequest();  
    request.open("GET", "/luckydip/enter");  
    request.send();  
    ...  
    // wait for the request status to be returned  
    ...  
    if (request.status != 200) {  
        alert("Error " + request.status + " - " + request.statusText);  
    }  
    ...  
}
```

สิ่งที่ได้กลับจากการ request ajax

- [Status](#) ของhtml เช่น
 - 404 not found
 - 200 ok
- statusText คือ ข้อความอธิบายถึง status เช่น
 - Not found
 - ok

<https://www.restapitutorial.com/httpstatuscodes.html>

-

- onreadystatechange คือฟังก์ชันที่จะเปลี่ยนแปลงค่าทุกครั้งที่มีการเปลี่ยนแปลง state โดยจะคู่กับ state
- Readystate มี 5 อัน
 - 0 : ยังไม่เริ่มต้นการ request
 - 1: การเชื่อมต่อเซิร์ฟเวอร์ที่จัดตั้งขึ้น แต่ตั้งแต่มยังไม่ร้องขอ
 - 2: ได้รับการร้องขอ (ส่ง request)
 - 3: การประมวลผลคำขอ (process)
 - 4: receive response

```
request.onreadystatechange = function() {
    if (request.readyState === 4) {
        var response = JSON.parse(request.responseText);
        ...
    }
};
```

-
- Try catch ก็ใช้ในการ ดักได้

ตัวอย่าง

```
function loadDoc() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demo").innerHTML =
                this.responseText;
        }
    };
    xhttp.open("GET", "ajax_info.txt", true);
    xhttp.send();
}
```

Response

- .responseText นั้นคือ สิ่งที่เราจะได้รับกลับมาหลังจาก request ผ่าน ajax
- getResponseHeader ใช้ในการ ดูว่า ชนิดของ result ที่เราต้องการหรือที่เรา request ไปเป็นชนิดอะไร เพื่อที่จะนำมาใช้ได้ถูกต้อง

```

...
var type = request.getResponseHeader();
switch( type ) {
    case "text/xml" :
        return request.responseXML;
    case "text/json" :
        return JSON.parse(request.responseText);
    default :
        return request.responseText;
}

```

- - request.responseText
 - request.responseXML
 - JSON.parse(request.responseText)

JSON.parse

ใช้เปลี่ยน string เป็น json

การส่ง data ผ่าน post

```

request.open("POST", url);
request.send(body);

```

Set header ตอนส่งไป

```

request.setRequestHeader("Content-Type", "application/json");

```

```

request.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

```

ใช้ JQuery มาช่วย

มันคือ library ในการเขียน js ช่วยให้เขียนได้ง่ายขึ้นและสั้นลง

```
var response;  
$.get('http://contoso.com/resources/...', function(data) {  
    response = data;  
}).error(function() {  
    alert("error occurred during get operation");  
});
```

```
$.getJSON( url, body, callback );
```

```
$('#container').load( url, body, callback );
```

get() ใช้ ajax แบบ get

```
$.get(url, body, callback);
```

post() ส่ง method post

หรือ Ajax() เลยก็ได้ ajax() เราต้องกำหนดเองว่าจะส่งแบบไหน

\$.getJSON(url, body, callback);

คือ get ข้อมูลที่เป็น json

```
$('#container').load(url, body, callback);
```

หาก load จะ ajax แบบ get มาเลย

ajax() คือเราสร้าง object XMLHttpRequest object ขึ้นมาโดยที่เรายังไม่ได้กำหนดค่าอะไรให้มัน
วิธีกำหนดก็ตามด้านล่าง

```
$.ajax({  
    url: '/luckydip/enter',  
    type: 'GET',  
    timeout: 12000,  
    dataType: 'text'
```

ส่วนของการ error handle

```
}).done(function( responseText ){
    $('#answer').text( responseText );
}).fail(function() {
    alert('An error has occurred - you may not have been entered');
});
```

แบบส่ง data ไปด้วย

```
$.ajax({
    url: '/luckydip/enterWithName',
    type: 'POST',
    timeout: 12000,
    dataType: 'text',
    data: {
        firstName: myForm.fname.value,
        lastName: myForm.lname.value
    }
}).done(...)
).fail(...)
);
```

Error handle

```
var response;
$.get('http://contoso.com/resources/...', function(data) {
    response = data;
}).error(function() {
    alert("error occurred during get operation");
});
```

HTTP method

ใช้กำหนดวิธีการ ที่เราจะดำเนินการกับทรัพยากร

- Get ใช้ดูหรือแสดงข้อมูล

- Post ในการส่งข้อมูล เพื่อ create insert
- Put ใช้ในการ update
- Delete ใช้ในการลบข้อมูล

หาก put delete ไม่มีให้ใช้ ให้ใช้ post แทนได้