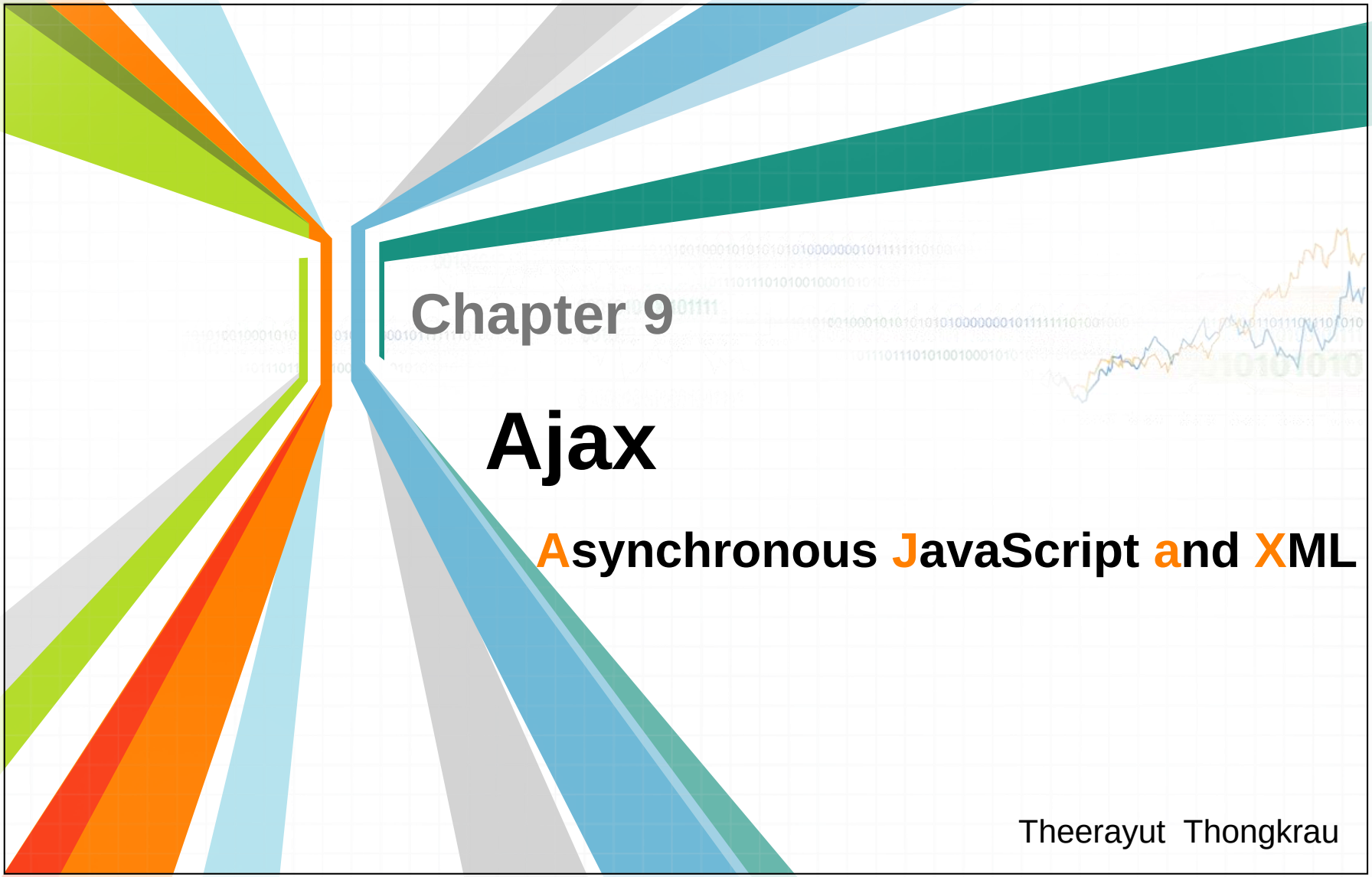


Chapter 9

Ajax

Asynchronous JavaScript and XML

Theerayut Thongkrau

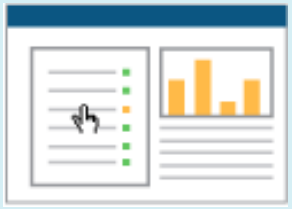
รูปแบบการส่งข้อมูลบนอินเทอร์เน็ต

- **แบบ Synchronous** คือ การส่ง request จาก Browser ไปประมวลผลที่ Server โดยต้องรอให้ Server ประมวลผลเสร็จก่อนและส่ง response กลับมายัง Browser จึงจะแสดงผลได้

- **แบบ Asynchronous** คือ การส่ง request จาก Browser ไปประมวลผลที่ Server โดยไม่ต้องรอให้ Server ประมวลผลเสร็จ Browser จะรับข้อมูลเมื่อ Server ทำเสร็จแล้ว

รูปแบบการส่งข้อมูลบนอินเทอร์เน็ต

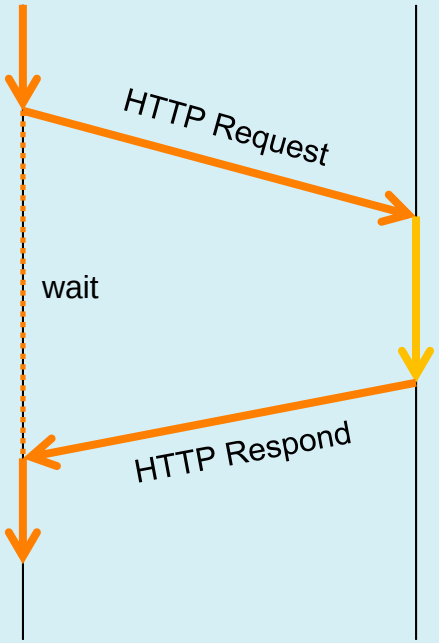
Synchronous



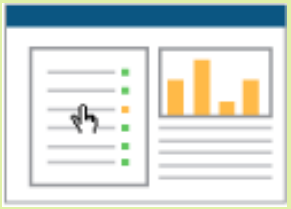
User Action

Screen Reload

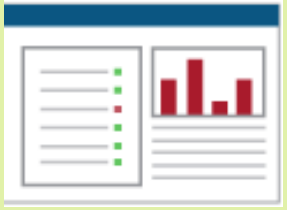
Data Update



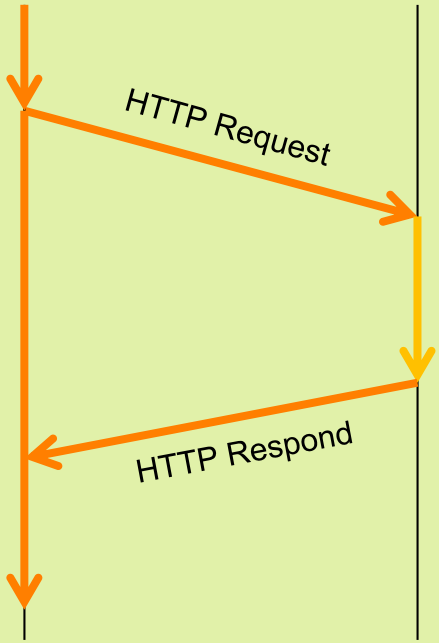
Asynchronous



User Action



Data Update



ปัญหาการส่งข้อมูลแบบ Synchronous

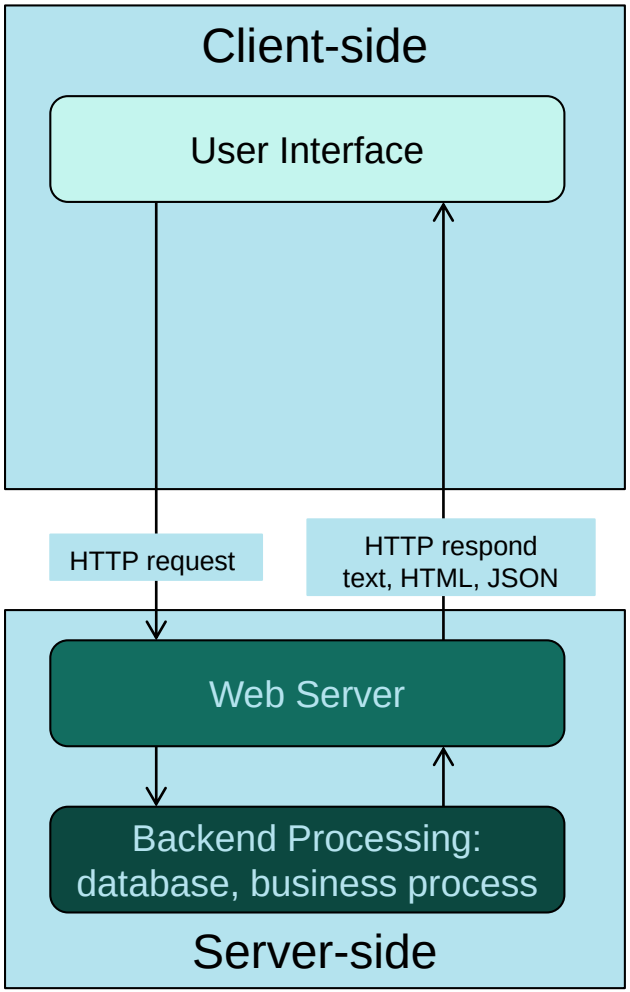
- เมื่อมีการเปลี่ยนแปลงข้อมูลเพียงบางส่วนบนเว็บเพ็จ จะต้องรีโหลดเว็บเพ็จใหม่ทั้งหน้า

- การรอหน้าเว็บเพ็จตอบกลับจาก Server ไม่สามารถทำงานอย่างอื่นขนานกันไปได้

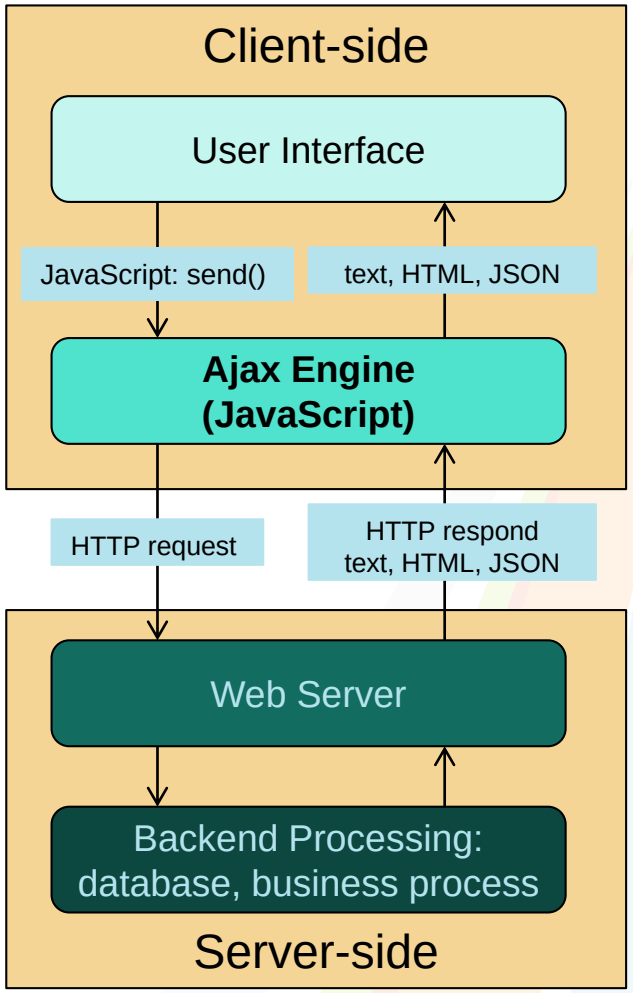
Ajax คืออะไร

- Ajax คือ object มาตรฐานในภาษา JavaScript ที่ใช้ในการรับและส่งข้อมูลกับ Server แบบ Asynchronous
- Ajax รองรับข้อมูลหลากหลายรูปแบบ
  - JSON
  - XML
  - HTML
  - Plain Text

สถาปัตยกรรมเว็บที่ไม่ใช้ และใช้ Ajax



เว็บที่ไม่ใช้ Ajax



เว็บที่ใช้ Ajax

XMLHttpRequest

- XMLHttpRequest คือ object สำหรับใช้ส่งคำร้อง (request) และรับผลตอบกลับ (response) ระหว่าง Browser กับ Server

ชื่อ property	คำอธิบาย
onreadystatechange	ใช้กำหนดชื่อฟังก์ชันที่จะให้ทำงาน เมื่อมีการเปลี่ยนแปลงสถานะ (ค่าใน readyState เปลี่ยนแปลง)
readyState	ค่าบอกลำดับการทำงานของการส่ง request และรับ response
status	รหัสการตอบกลับจาก Server (HTTP response)
responseText	ข้อมูลที่ Server ส่งกลับมายัง Browser (ข้อความที่ถูก echo ใน PHP)

```
<html><head><meta charset="utf-8">
<script>
   var httpRequest;
function send() {
   httpRequest = new XMLHttpRequest();
   httpRequest.onreadystatechange = showResult;

   var a = document.getElementById("a").value;
   var b = document.getElementById("b").value;
   var url= "add.php?a=" + a + "&b=" + b;

   httpRequest.open("GET", url);
   httpRequest.send();
  }

function showResult() {
   if (httpRequest.readyState == 4 && httpRequest.status == 200) {
    document.getElementById("result").innerHTML = httpRequest.responseText;
     }
  }
</script></head>
<body>
    a = <input type="text" id="a"><br>
    b = <input type="text" id="b" onkeyup="send()"><br>
    <span id="result"></span>
</body></html>
```

ดึงผลลัพธ์จาก Server

เมื่อมีการกรอกตัวเลขที่ช่องนี้ จะเรียกฟังก์ชัน send()

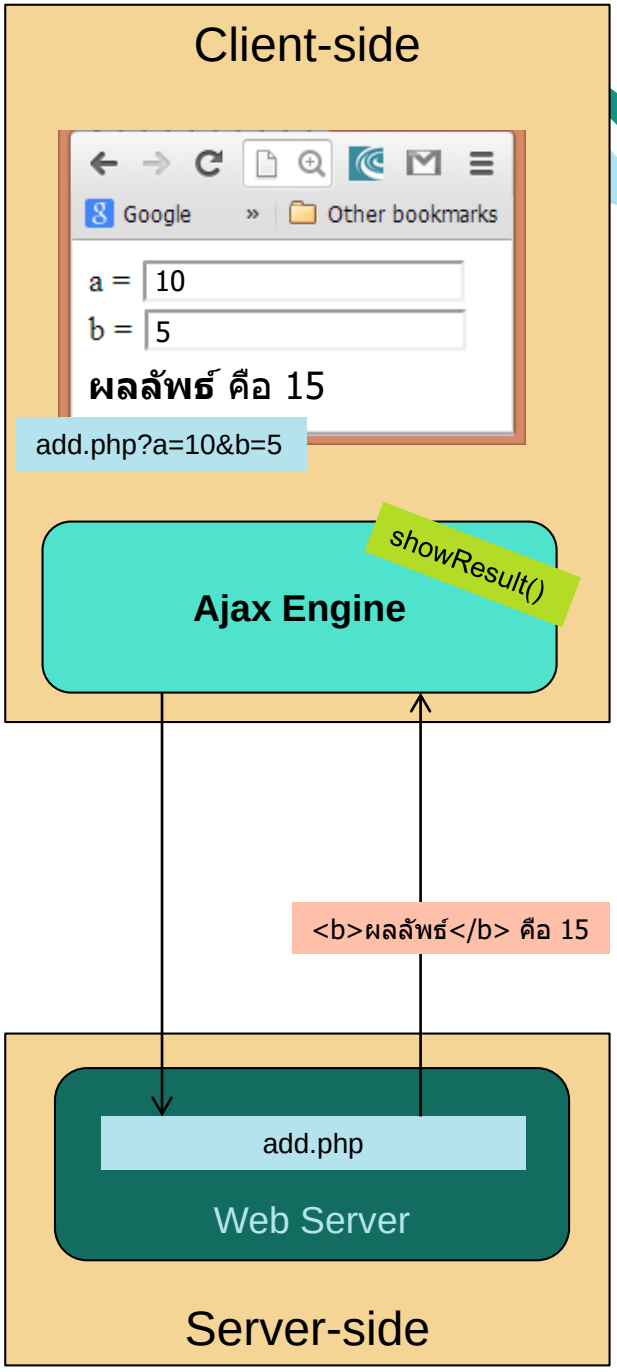
กำหนดจุดที่จะให้แสดงผลลัพธ์

readyState

4

add.php

```
<?php
$a = $_GET["a"];
$b = $_GET["b"];
echo "<b>ผลลัพธ์</b> คือ ";
echo $a + $b;
?>
```



การตรวจสอบสถานะของ request

ค่าบอกลำดับการทำงานของการส่ง request และรับ response จะถูกเก็บใน property ชื่อ readyState มี 4 ค่า ได้แก่

0 – การร้องขอยังไม่ถูกกำหนดขึ้น

1 – การร้องขอได้ถูกกำหนดขึ้นแล้ว

2 – การร้องขอได้ถูกส่งไป

3 – การร้องขอกำลังทำงานอยู่

4 – การร้องขอได้เสร็จสิ้นการทำงานแล้ว

รหัสการตอบกลับจาก Server

รหัสการตอบกลับจาก Server (HTTP response) จะถูกเก็บใน property ชื่อ status ซึ่งเป็นรหัสมาตรฐานของ HTTP

Code	ความหมาย	คำอธิบาย
200	OK	คำร้องที่ต้องการถูกพบบน Server
401	Unauthorized	คำร้องที่ต้องการไม่มีสิทธิ์ในการเข้าถึง
403	Forbidden	คำร้องที่ต้องการเป็นส่วนที่หวงห้ามในการเข้าถึง
404	Not Found	คำร้องที่ต้องการไม่พบบน Server
500	Internal Server Error	Server มีข้อผิดพลาด
503	Service Unavailable	Server ไม่สามารถจัดการกับคำร้องได้

ฟังก์ชันของ XMLHttpRequest

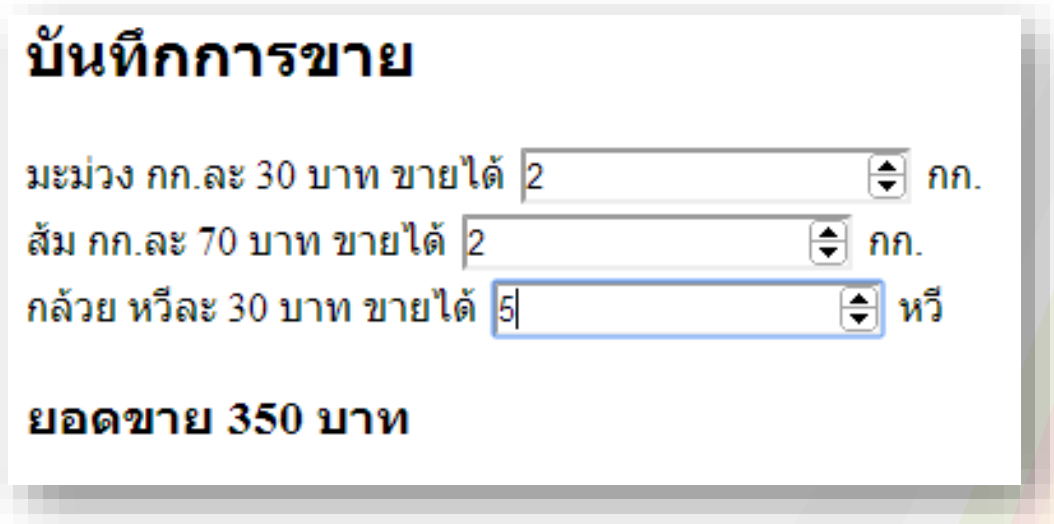
ชื่อฟังก์ชัน	คำอธิบาย
open(method, url, [isAsynchronous])	เปิดการติดต่อกับ Web Server - method คือ การติดต่อแบบ GET หรือ POST - url คือ ไฟล์ PHP ที่ใช้รับและส่งข้อมูล - isAsynchronous มีค่า default เป็น true
send(content)	ส่งการร้องขอไปยัง Server
abort()	ยกเลิกการร้องขอ
getAllResponseHeader()	รับค่าส่วนหัวที่ร้องขอทั้งหมดในรูปแบบ key/value
getResponseHeader(header)	รับค่าส่วนหัวที่ร้องขอตามที่ระบุ
setRequestHeader(header, value)	กำหนดประเภทของข้อมูลที่จะมีการส่งหรือรับ

ขั้นตอนการสร้างเว็บที่มีการใช้ Ajax

1. สร้างส่วนที่เป็นโค้ด html
   - กำหนด event ให้กับแท็กที่จะกระตุ้นให้เริ่มส่งข้อมูลแบบ asynchronous
   - กำหนดจุดแสดงผลเมื่อได้รับการตอบกลับ
2. สร้างส่วนที่เป็นโค้ด JavaScript
   - เขียนฟังก์ชันสำหรับใช้ในการส่งข้อมูล
   - เขียนฟังก์ชันสำหรับใช้ในการรับข้อมูล
3. สร้างส่วนที่เป็นโค้ด PHP บน Server ซึ่งทำหน้าที่คอยรับและส่งข้อมูล

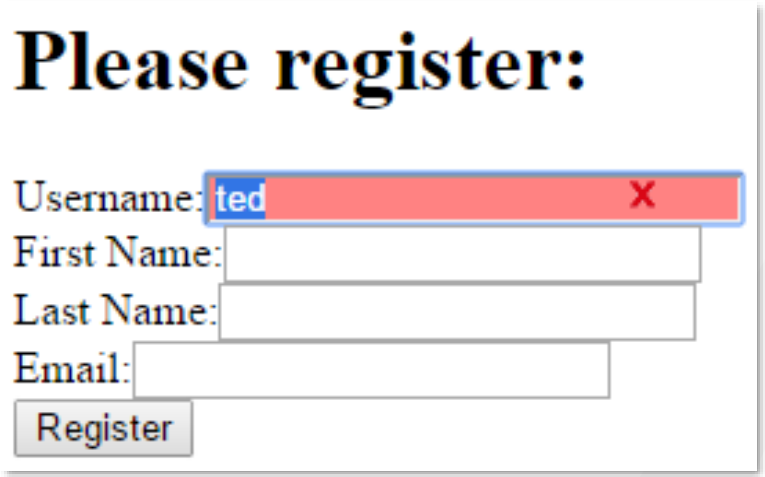
กิจกรรม

- จงเขียนเว็บในการคำนวณยอดขายผลไม้ โดยแบ่งการทำงานออกเป็นสองฝั่ง ได้แก่ ฝั่งเซิร์ฟเวอร์ ทำหน้าที่คำนวณยอดรวม ซึ่งเขียนด้วยภาษา PHP ส่วนฝั่งไคลเอนต์เขียนด้วยภาษา HTML และ JavaScript+Ajax ทำหน้าที่รับค่าจำนวนผลไม้แต่ละชนิดขายได้ส่งไปยังเซิร์ฟเวอร์ และนำผลลัพธ์มาแสดงผลโดยไม่มีการรีโหลดหน้าเว็บใหม่



การตรวจสอบชื่อผู้ใช้ด้วย Ajax

- การลงทะเบียนที่มีการกรอกชื่อผู้ใช้ จะถูกตรวจสอบว่า username ซ้ำกันเมื่อมีการคลิกปุ่มส่งข้อมูลแล้ว แต่การใช้ Ajax จะช่วยตรวจสอบหลังจากที่กรอก username เสร็จได้เลย โดยไม่ต้องรอพิมพ์ให้ครบ



ฟอร์ม HTML

เมื่อผู้ใช้กรอก username เสร็จ และไปยังช่องถัดไป ฟังก์ชัน checkUsername จะถูกทำงาน

```
<form>
    <h1>Please register:</h1>
    Username:<input id="username" type="text" onblur="checkUsername()"><br>
    First Name:<input type="text" name="firstname"><br>
    LastName:<input type="text" name="lastname"><br>
    Email:<input type="text" name="email"><br>
    <input type="submit" value="Register">
</form>
```

ฟังก์ชันส่งและรับข้อมูล

```
<script>
var xmlHttp;
function checkUsername() {
    document.getElementById("username").className = "thinking";

    xmlHttp = new XMLHttpRequest();
    xmlHttp.onreadystatechange = showUsernameStatus;

    var username = document.getElementById("username").value;
    var url = "checkName.php?username=" + username;
    xmlHttp.open("GET", url);
    xmlHttp.send();
}

function showUsernameStatus() {
    if (xmlHttp.readyState == 4 && xmlHttp.status == 200) {
        if (xmlHttp.responseText == "okay") {
            document.getElementById("username").className = "approved";
        } else {
            document.getElementById("username").className = "denied";
            document.getElementById("username").focus();
            document.getElementById("username").select();
        }
    }
}
</script>
```

แสดงภาพตรงช่อง username ว่ากำลังตรวจสอบ

กำหนดฟังก์ชั่นที่ต้องการให้ทำงาน เมื่อมีการเปลี่ยนแปลงสถานะ

ดึงค่า username จากฟอร์ม

กำหนดตำแหน่งสคริป php ที่จะให้รับข้อมูล
พร้อมกับแนบชื่อผู้ใช้ไปด้วย

ถ้าค่าที่ตอบกลับเป็นคำว่า okay จะกำหนด
คลาสให้ textfield ชื่อ approved

CSS สำหรับแสดงสถานะ

```
.thinking {
   background: white url("img/checking.gif") no-repeat;
   background-position: 150px 1px;
}
```

```
.approved {
   background: white url("img/true.gif") no-repeat;
   background-position: 150px 1px;
}
```

```
.denied {
   background: #FF8282 url("img/false.gif") no-repeat;
   background-position: 150px 1px;
}
```

PHP ที่คอยรับและส่งข้อมูล

checkName.php

```
<?php

$takenUsernames = array("bill", "ted");

sleep(1);

if (!in_array( $_GET["username"],  $takenUsernames ) ) {
    echo "okay";
} else {
    echo "denied";

}

?>
```

จำลอง username ที่มีอยู่แล้ว

ให้หยุดรอ 1 วินาที เพื่อจำลองว่ามีการตรวจสอบอยู่

ถ้ามี username ในอาร์เรย์

ส่งคำว่า okay กลับไป