



หน่วยที่ 5

การพัฒนาเว็บไซต์แบบ Dynamic





หน่วยที่ 5

การพัฒนาเว็บไซต์แบบ Dynamic

หัวข้อเรื่อง (Topics)

5.1 การเขียนโปรแกรมเพื่อติดต่อฐานข้อมูลด้วยภาษา SQL เบื้องต้น

5.2 การเขียนโปรแกรมตรวจสอบเงื่อนไขและประมวลผลการทำงานด้วยภาษา PHP

5.3 การเขียนโปรแกรมควบคุมส่วนติดต่อประสานงานกับผู้ใช้งานด้วยภาษา JavaScript



เนื้อหาสาระ (Content)

5.1 การเขียนโปรแกรมเพื่อติดต่อฐานข้อมูลด้วยภาษา SQL เบื้องต้น

5.1.1 คำสั่ง Insert into คือ คำสั่งภาษา SQL ที่ใช้ในการเพิ่มข้อมูล โดยมีรูปแบบดังนี้

```
INSERT INTO user_tb (Username, Password)  
VALUES (User1, User1);
```

Username	Password
Admin	admin



Username	Password
Admin	Admin
User1	User1

จากคำสั่งข้างต้น สามารถอธิบายได้ดังนี้

คือ การเพิ่มข้อมูลลงในตาราง user_tb ซึ่งเป็นการเพิ่มข้อมูลลงในคอลัมน์ Username โดยกำหนดค่าให้เป็น User1 และเพิ่มข้อมูลลงในคอลัมน์ Password โดยกำหนดค่าให้เป็น User1

5.1.2 คำสั่ง Update คือ คำสั่งภาษา SQL ที่ใช้ในการแก้ไขข้อมูล โดยมีรูปแบบดังนี้

```
UPDATE user_tb  
SET Username= 'userSC', Password= 'PassUserSC'  
WHERE id_user=2;
```



id_user	Username	Password
1	Admin	Admin
2	User1	User1



id_user	Username	Password
1	Admin	Admin
2	userSC	PassUserSC

จากคำสั่งข้างต้น สามารถอธิบายได้ดังนี้
คือ การแก้ไขข้อมูลในตาราง user_tb โดยกำหนดให้คอลัมน์ Username มีค่าเท่ากับ userSC และคอลัมน์ Password มีค่าเท่ากับ PassUserSC โดยอ้างอิงคีย์หลัก (Primary Key) ของคอลัมน์ id_user โดยเลือกแถวที่ต้องการแก้ไข คือ แถวที่ 2

5.1.3 คำสั่ง Delete คือ คำสั่งภาษา SQL ที่ใช้ในการลบข้อมูล โดยมีรูปแบบดังนี้

```
DELETE FROM user_tb  
WHERE id_user=2;
```



id_user	Username	Password
1	Admin	Admin
2	userSC	PassUserSC



id_user	Username	Password
1	Admin	Admin



จากคำสั่งข้างต้น สามารถอธิบายได้ดังนี้
คือ การลบข้อมูลของตาราง user_tb ในลักษณะเป็นแนวแถว โดยอ้างอิงคีย์หลัก (Primary Key) ของคอลัมน์ id_user โดยเลือกแถวที่ต้องการลบ คือ แถวที่ 2

5.1.4 คำสั่ง Select คือคำสั่งภาษา SQL ที่ใช้ในการแสดงข้อมูล โดยมีรูปแบบดังนี้
`SELECT * FROM user_tb LIMIT 5;`

id_user	Username	Password
1	Admin	Admin
2	userSC	PassUserSC
3	User1	User1
4	User2	User2
5	User3	User3
6	User4	User4



id_user	Username	Password
1	Admin	Admin
2	userSC	PassUserSC
3	User1	User1
4	User2	User2
5	User3	User3

จากคำสั่งข้างต้น สามารถอธิบายได้ดังนี้
คือ การเลือกข้อมูลจากฐานข้อมูลทั้งหมดของตาราง user_tb แต่ไม่เกิน 5 แถว
เพื่อนำแสดงผล
`SELECT * FROM user_tb;`



id_user	Username	Password
1	Admin	Admin
2	userSC	PassUserSC
3	User1	User1
4	User2	User2
5	User3	User3
6	User4	User4

จากคำสั่งข้างต้น สามารถอธิบายได้ดังนี้
คือ การเลือกข้อมูลจากฐานข้อมูลทั้งหมดของตาราง user_tb โดยไม่ระบุจำนวนแถวเพื่อนำมาแสดงผล

5.1.5 คำสั่ง `mysql_fetch_array`

```
mysql_fetch_array($value);
```



id_user	Username	Password
1	Admin	Admin
2	userSC	PassUserSC
3	User1	User1
4	User2	User2
5	User3	User3
6	User4	User4



จากคำสั่งข้างต้น สามารถอธิบายได้ดังนี้

คือ การเรียกใช้ข้อมูลจากฐานข้อมูลในรูปแบบ Array สามารถเรียกใช้ด้วยชื่อคอลัมน์หรือหมายเลขอินเด็กซ์เพื่อชี้ไปยังตำแหน่งของข้อมูลที่ต้องการนำมาแสดงผล ได้แก่ \$result['Username']; และ \$result[1]; คำสั่ง mysql_fetch_array(); นิยมประยุกต์ใช้กับคำสั่งอื่น ๆ เช่น คำสั่งวนลูป for, while, do while หรือคำสั่ง SQL อื่น ๆ

ตัวอย่างการใช้งาน mysql_fetch_array();

```
$str=mysql_query("SELECT * FROM user_tb WHERE id_user=1");  
$result=mysql_fetch_array($str) or die(mysql_error());  
echo $result['Username']; /*แสดงผลคำว่า Admin*/  
echo $result['Password']; /*แสดงผลคำว่า Admin*/
```

5.2 การเขียนโปรแกรมตรวจสอบเงื่อนไขและประมวลผลการทำงานด้วยภาษา PHP

5.2.1 คำสั่ง isset()

คำสั่ง isset() คือ คำสั่งที่ใช้เพื่อตรวจสอบว่า ตัวแปรนั้น ๆ ได้ถูกกำหนดขึ้น และมีค่าที่ไม่ใช่ null หรือไม่



5.2.2 คำสั่ง SESSION

คำสั่ง SESSION คือ ตัวแปรที่มีการทำงานเหมือนกับคุกกี้ (Cookie) ซึ่งมีการจดจำค่าต่าง ๆ เช่น ตัวเลข ตัวอักษร เป็นต้น ถึงแม้จะมีการเปลี่ยนหน้าเว็บเพจ ตัวแปรแบบ SESSION สามารถเก็บค่านั้น ๆ ไว้ได้จนกระทั่งปิดโปรแกรมท่องเว็บไซท์หรือมีการล้างตัวแปรแบบ SESSION

ตัวอย่างการใช้งาน คำสั่ง SESSION

```
session_start();           /*การประกาศการเริ่มต้นใช้งานตัวแปรแบบ SESSION*/
$_SESSION["favcolor"] = "green"; /*การกำหนดค่าให้กับตัวแปรแบบ SESSION*/
session_destroy();          /*การ Clear การทำงานของตัวแปรแบบ SESSION*/
```

5.2.3 คำสั่ง if-else

คำสั่ง if-else คือ คำสั่งที่ใช้ตรวจสอบเงื่อนไขของตัวแปรหรือค่าพารามิเตอร์ต่าง ๆ เช่น ถ้า x มากกว่า 5 ให้แสดงผลว่า Hello X แต่ถ้าน้อยกว่า 5 ให้แสดงผลว่า So Bad เป็นต้น

```
if ($x>5) {                  /*ถ้า x มากกว่า 5 ให้ทำ*/
    echo "Hello X";          /*แสดงผล Hello X*/
} else {                     /*ถ้าไม่ใช่ ให้ทำ*/
    echo "So Bad";           /*แสดงผล So Bad*/
}
```



5.2.4 คำสั่ง Array

Array คือ ตัวแปรชนิดหนึ่งที่สามารถเก็บค่าไว้หลายค่าภายในตัวแปรเดียวได้ โดยมีการเรียกใช้ค่านั้น ๆ ด้วยหมายเลขอินเด็กซ์ (Index)

ตัวอย่างการใช้คำสั่ง Array

```
$cars = array("Honda", "BMW", "Toyota");
```

จากคำสั่งข้างต้น สามารถอธิบายได้ว่า ตัวแปร cars เก็บข้อมูลยี่ห้อรถ 3 ยี่ห้อ ไว้ในตัวแปรเดียว เพื่อลดหน่วยความจำ แทนการเก็บข้อมูลแบบเดิม เช่น \$car1=Honda; \$car2=BMW; \$car3=Toyota; การกำหนดค่าต่าง ๆ ให้กับตัวแปรแบบ Array จากตัวอย่างสามารถทำได้ดังนี้

```
$cars[0] = "Honda";
```

```
$cars[1] = "BMW";
```

```
$cars[2] = "Toyota";
```



ตัวอย่างการเรียกใช้งานตัวแปรแบบ Array ด้วย Index

```
<?php
```

```
$cars = array("Honda", "BMW", "Toyota"); /*กำหนดค่าให้กับตัวแปรแบบ Array*/
```

```
$arrlength= count($cars); /*นับจำนวนสมาชิก Array ด้วยคำสั่ง count*/
```

```
for($x = 0; $x < $arrlength; $x++) { /*วนรอบเพื่อแสดงค่าที่อยู่ในตัวแปร $cars*/
```

```
    echo $cars[$x]; /*แสดงค่าของ Array ตั้งแต่ index ที่ 0-2*/
```

```
    echo "<br>";
```

```
}
```

```
?>
```

5.2.5 คำสั่งวนซ้ำ (Loop)

คำสั่งวนซ้ำ คือ คำสั่งที่มีการวนซ้ำโดยมีเงื่อนไขกำหนด เช่น เพิ่มค่าไปเรื่อย ๆ จนกว่าจะถึง 10 เป็นต้น คำสั่งวนซ้ำแบ่งเป็น 3 รูปแบบ ได้แก่

1. คำสั่ง **While** มีการทำงานตามเงื่อนไขที่กำหนด เมื่อทำตามเงื่อนไขเรียบร้อยแล้ว คำสั่ง

While หยุดการทำงานทันที

```
while (เงื่อนไข) {
```

```
    คำสั่ง...
```

```
}
```



2. คำสั่ง **for** มีการทำงานตามเงื่อนไขที่กำหนด โดยเงื่อนไขในคำสั่ง **for** แบ่งเป็น 3 ส่วน คือ การกำหนดค่าเริ่มต้น $i=0$; เงื่อนไข $i<10$; และการเพิ่ม-ลดค่า $i++$; ทั้งนี้สามารถกำหนดเป็นค่าอื่น ๆ ได้ตามเงื่อนไขที่ต้องการ

```
for ( $i=0$ ;  $i<10$ ;  $i++$ ) { /*การกำหนดเงื่อนไขของคำสั่งวนซ้ำ*/  
    echo  $i$ ; /*การแสดงผลของตัวแปรค่าเริ่มต้นโดยเริ่มจาก 0 เพราะกำหนดให้  $i=0$ */  
}
```

3. คำสั่ง **do while** มีการทำงานอย่างน้อย 1 ครั้ง ก่อนทำการตรวจสอบเงื่อนไขเพื่อวนซ้ำ ซึ่งมีตัวอย่างการใช้คำสั่ง **do while** ดังนี้

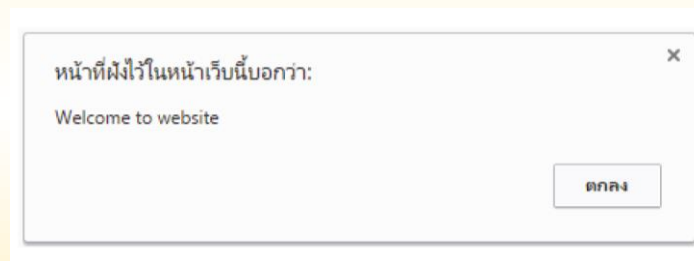
```
 $i=0$ ; /*กำหนดค่าให้กับตัวแปร  $i$  โดยมีค่าเท่ากับ 0*/  
do {  
    echo "Hello"; /*แสดงผลคำว่า Hello*/  
     $i++$ ; /*เพิ่มค่าให้กับตัวแปร  $i$  ที่ละ 1*/  
} while( $i<10$ ); /*ตรวจสอบเงื่อนไข ถ้า  $i$  มีค่าน้อยกว่า 10 ให้วนซ้ำการทำงาน*/
```



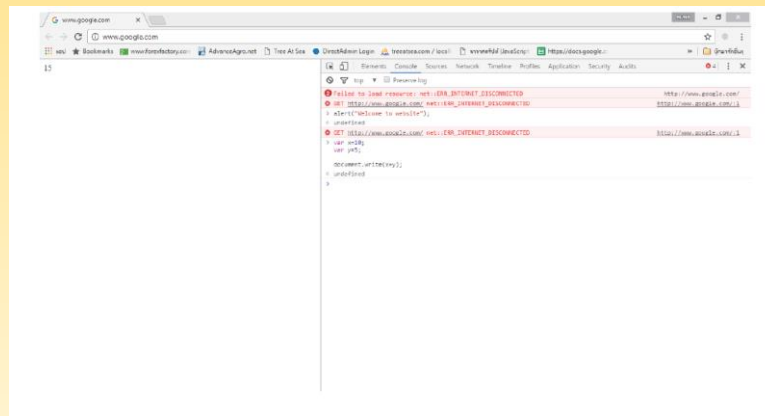
5.3 การเขียนโปรแกรมควบคุมส่วนติดต่อประสานงานกับผู้ใช้งานด้วยภาษา JavaScript

ภาษา JavaScript เป็นภาษาที่สามารถประมวลผลฝั่ง Client ที่มีความสามารถมากมายเหมือนกับภาษาอื่น ๆ เช่น การตรวจสอบเงื่อนไข การใช้งานฟังก์ชัน OOP และการควบคุมการแสดงผลต่าง ๆ บนหน้าจอ ซึ่งการแสดงผลในรูปแบบต่าง ๆ ของภาษา JavaScript มีดังนี้

1. การแสดงผลด้วย Window Alert



2. การแสดงผลด้วยคำสั่งของภาษา JavaScript
3. การแสดงผลโดยอ้างอิง Id ของแท็ก HTML
4. การแสดงผลใน Console Log

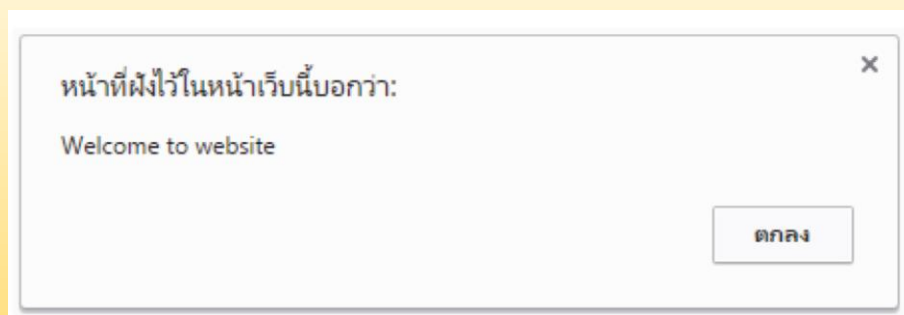


5.3.1 การแสดง Alert

การแสดงผล Alert คือ การแสดง Popup แจ้งเตือนเพื่อแสดงข้อความที่กำหนดไว้ให้ผู้ใช้งานได้ทราบ ซึ่งสามารถนำมาประยุกต์ใช้ในการเตือนหรือแจ้งข้อมูลกระบวนการทำงาน เป็นต้น

รูปแบบคำสั่ง alert ("ข้อความ");

ตัวอย่างการใช้งาน `<script>alert("Welcome to website");</script>`





5.3.2 การใช้งาน Confirm Alert

การใช้งาน Confirm Alert คือ การแสดง Popup เพื่อต้องการยืนยันข้อมูลก่อนประมวลผล ในขั้นตอนถัดไป เช่น การยืนยันการลบ การยืนยันการออกระบบ เป็นต้น ซึ่ง Confirm Alert มีปุ่มให้เลือก 2 ปุ่ม ได้แก่ ปุ่มตกลงและยกเลิก โดยคำสั่งของ Confirm Alert มีดังนี้

รูปแบบคำสั่ง `confirm("Press a button");`

ตัวอย่างการใช้งานคำสั่ง

```
var r = confirm("Do you want to Delete Item?"); /*กำหนดค่าให้กับตัวแปร r ด้วยคำสั่ง  
confirm*/
```

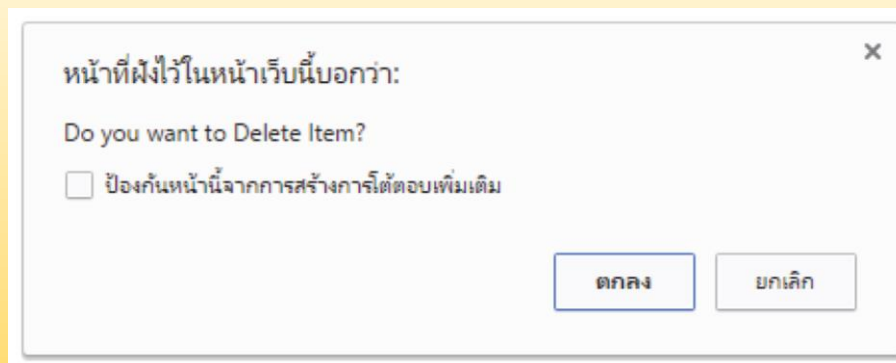
```
if (r == true) { /*ตรวจสอบเงื่อนไข ถ้าตัวแปร r เป็นจริงหรือถูกคลิกตกลง*/
```

```
    alert("Deleting...."); /*ให้แสดง Popup */
```

```
} else { /*ถ้าไม่จริงหรือถูกคลิกปุ่มยกเลิก */
```

```
    alert("Cancel Delete"); /*ให้แสดง Popup */
```

```
}
```





5.3.3 การรับค่าจากผู้ใช้งานด้วย Prompt

Prompt คือ การแสดง Popup พร้อมช่องที่มีไว้สำหรับป้อนข้อมูล เพื่อนำข้อมูลที่ป้อนนำไปประมวลผลในลำดับต่อไป

รูปแบบคำสั่ง prompt ("ข้อความแจ้งเตือน", "ข้อความค่าเริ่มต้น");

ตัวอย่างการใช้งาน

```
var person = prompt("กรณากรอกชื่อ", "ปีเตอร์");  
if (person != null) {  
    alert("กรอกข้อมูลสำเร็จ");  
} else {  
    alert("กรณากรอกข้อมูล");  
}
```



5.3.4 การใช้งาน Event

การใช้งาน Event คือ การกระทำต่อวัตถุด้วยเหตุการณ์ต่าง ๆ เช่น การคลิกปุ่ม หรือการนำเมาส์มาชี้ที่วัตถุเพื่อดูคำอธิบาย เป็นต้น Event ที่นำมาใช้งานมีดังนี้

Event (เหตุการณ์)	ความหมาย
OnClick	เมื่อทำการคลิกวัตถุใด ๆ
Onload	เมื่อทำการโหลดหน้าเว็บเพจ
Onchange	เมื่อมีการเปลี่ยนแปลง
Onmouseover	เมื่อนำเมาส์มาชี้วัตถุ
Onmouseout	เมื่อนำเมาส์ออกจากวัตถุ
Onkeydown	เมื่อกดปุ่มใด ๆ บนคีย์บอร์ด



ตัวอย่างการใช้งาน Event Onclick ของภาษา JavaScript

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<button type="button" onclick="changeText(this)">คลิกเพื่อดูข้อมูล</button>
```

*/*กำหนด Event Onclick ด้วยการสร้าง Function ให้กับปุ่ม*/*

```
<script>
```

```
function changeText(id) { /*สร้างฟังก์ชันเพื่อตอบสนอง Event Onclick*/
```

```
    alert("Hello world"); /*แสดง Popup เมื่อคลิกปุ่ม*/
```

```
}
```

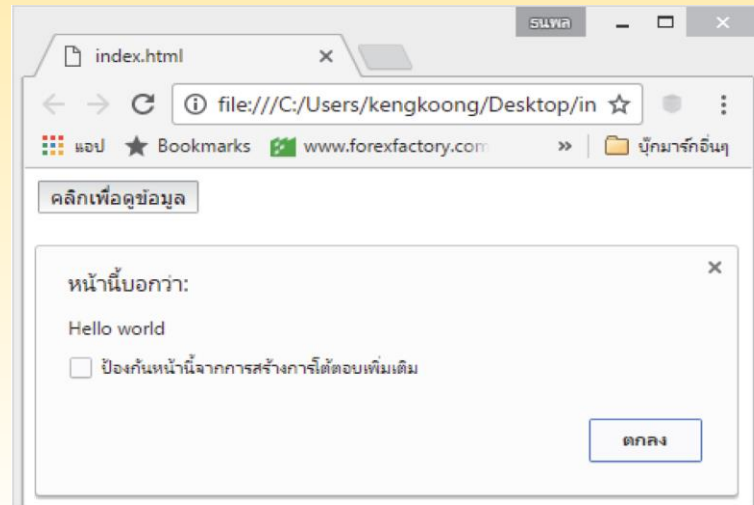
```
</script>
```

```
</body>
```

```
</html>
```



ผลลัพธ์ของการใช้งาน Event Onclick



ตัวอย่างการใช้งาน Event Onmouseover และ Onmouseout

```
<html>
```

```
<head>
```

```
<style>
```

```
.bg-tooltip { /*กำหนด Style ให้กับ class bg-tooltip*/
```

```
background-color:#000;
```

```
color:#fff; width:50px;
```

```
border-radius:8px;
```

```
}  
</style>
```

```
</head>
```

```
<body>
```

```
<p onmouseover="tooltip(1)" onmouseout="tooltip(0)">ชี้เพื่อดูรายละเอียด</p>
```

```
/*กำหนด Event ให้กับแท็ก p*/
```

```
<div id="show" class="bg-tooltip"></div> /*กำหนดจุดแสดงผลด้วย id show และใช้งาน class ของ CSS */
```

```
<script>
```

```
function tooltip(value) { /*function tooltip ตรวจสอบการแสดงผลใน id show*/
```

```
if(value==1) {
```

```
/*ตรวจสอบเงื่อนไขจากค่าที่ส่งมาโดยเก็บที่ตัวแปร value ถ้ามีค่าเท่ากับ 1 ให้แสดงผลคำว่า hello*/
```

```
document.getElementById("show").innerHTML="Hello";
```

```
} else {
```

```
/*ตรวจสอบเงื่อนไขจากค่าที่ส่งมาโดยเก็บที่ตัวแปร value ถ้ามีค่าไม่เท่ากับ 1 ให้แสดงคำว่าว่าง*/
```

```
document.getElementById("show").innerHTML="";
```

```
}
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```





ผลลัพธ์ของการใช้ Event onmouseover และ onmouseout

