

แผนการบริหารการสอนประจำบทที่ 4

เนื้อหาประจำบทที่ 4

บทที่ 4 พื้นฐานการเขียนโปรแกรมภาษาจาวาสคริปต์

1. อะไรคือภาษาสคริปต์
2. ข้อดีของภาษาสคริปต์
3. ภาษาสคริปต์ทำอะไรได้บ้าง
4. วิธีการใช้งานภาษาสคริปต์
5. คำสั่งสำหรับการแสดงผล
6. รูปแบบคำสั่งในภาษาภาษาจาวาสคริปต์
7. การเขียนคำสั่งในภาษาภาษาจาวาสคริปต์
8. ตัวแปรในภาษาภาษาจาวาสคริปต์
9. ข้อกำหนดในการตั้งชื่อตัวแปร
10. ชนิดของข้อมูลในภาษาภาษาจาวาสคริปต์
11. ตัวดำเนินการในภาษาภาษาจาวาสคริปต์
12. พงกชันและการเรียกใช้งาน
13. องค์ประกอบของวัตถุ
14. การตรวจจับเหตุการณ์
15. การจัดการเกี่ยวกับวันและเวลา
16. การกำหนดและตรวจสอบเงื่อนไข
17. การวนรอบตามเงื่อนไข
18. สรุป

จุดประสงค์เชิงพฤติกรรม

เมื่อศึกษาบทที่ 4 แล้ว นักศึกษามารถ

1. บอกความหมายของภาษาสคริปต์ได้
2. บอกข้อดีของภาษาสคริปต์ได้
3. บอกความสามารถของภาษาสคริปต์ได้
4. อธิบายวิธีการใช้งานภาษาสคริปต์ได้
5. เขียนคำสั่งสำหรับการแสดงผลโดยใช้ภาษาภาษาสคริปต์ได้

6. อธิบายรูปแบบคำสั่งในภาษาจาวาสคริปต์ได้
7. อธิบายวิธีการเขียนคำสั่งในภาษาจาวาสคริปต์ได้
8. อธิบายตัวแปรในภาษาจาวาสคริปต์แต่ละชนิดได้
9. บอกข้อกำหนดในการตั้งชื่อตัวแปรได้
10. อธิบายชนิดของข้อมูลในภาษาจาวาสคริปต์ได้
11. บอกความหมายของตัวดำเนินการในภาษาจาวาสคริปต์ได้
12. อธิบายฟังก์ชันและการเรียกใช้งานฟังก์ชันได้
13. อธิบายองค์ประกอบของวัตถุในภาษาจาวาสคริปต์ได้
14. บอกวิธีการตรวจสอบเหตุการณ์ได้
15. อธิบายวิธีการจัดการเกี่ยวกับวันและเวลาได้
16. อธิบายวิธีการกำหนดและตรวจสอบเงื่อนไขได้
17. อธิบายวิธีการเขียนคำสั่งสำหรับการวนรอบตามเงื่อนไขได้

กิจกรรมการเรียนการสอนประจำบท

1. ผู้สอนอธิบายทฤษฎี และซักถามผู้เรียน พัฒนาอธิบายไฟล์เอกสารพาวเวอร์พอยท์ประกอบการบรรยายโดยใช้คอมพิวเตอร์ และโปรเจคเตอร์
2. ผู้สอนเปิดโปรแกรม Notepad++ และอธิบายส่วนประกอบหน้าจอหลักของโปรแกรม
3. ให้ผู้เรียนศึกษาเอกสารคำสอน
4. ให้ผู้เรียนตั้งคำถามเกี่ยวกับเนื้อหาที่สงสัย
5. ให้ผู้เรียนทำแบบฝึกหัดท้ายบท
6. ให้ผู้เรียนทำแบบฝึกปฏิบัติประจำบท

สื่อการเรียนการสอน

1. เอกสารคำสอน
2. หนังสืออ่านประกอบศึกษาค้นคว้าเพิ่มเติม
3. โปรแกรม Notepad++
4. ไฟล์เอกสารพาวเวอร์พอยท์
5. แบบฝึกหัดท้ายบท
6. แบบฝึกปฏิบัติ

การวัดและประเมินผล

1. สังเกตจากการซักถามผู้เรียน
2. สังเกตจากการร่วมกิจกรรมของผู้เรียน
3. สังเกตจากการอภิปรายกลุ่มย่อยและสรุปผลการอภิปราย
4. ประเมินจากแบบฝึกหัดท้ายบท
5. ประเมินจากแบบฝึกปฏิบัติ
6. ประเมินจากการสอบระหว่างภาค

บทที่ 4

พื้นฐานการเขียนโปรแกรมภาษาสคริปต์

การพัฒนาเว็บไซต์โดยทั่วไปจะต้องมีพื้นฐานเกี่ยวกับภาษาพื้นฐานในการพัฒนาเว็บไซต์ ซึ่งประกอบไปด้วย ภาษาอุ๊กอิ๊ก ภาษาอังกฤษ สำหรับเพิ่มรายละเอียดเนื้อหาลงในหน้าเว็บเพจ ภาษาซีเอสเอส สำหรับตกแต่งรูปแบบการแสดงผล ซึ่งเป็นพื้นฐานที่สำคัญในการออกแบบและพัฒนาเว็บไซต์

ภาษาคอมพิวเตอร์อีกประเภทหนึ่งที่ทำให้เว็บไซต์ทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น คือ ภาษาสคริปต์ เป็นภาษาคอมพิวเตอร์สำหรับการพัฒนาเว็บไซต์เพื่อให้สามารถประมวลผลข้อมูลบนหน้าเว็บไซต์ได้ ทำให้เว็บไซต์มีความน่าสนใจมากยิ่งขึ้น ซึ่งภาษาสคริปต์แบ่งออกเป็น 2 ประเภท ได้แก่ ภาษาสคริปต์ที่ประมวลผลในฝั่งเครื่องผู้ร้องขอบริการ (Client side script) และ ภาษาสคริปต์ที่ประมวลผลในฝั่งเครื่องผู้ให้บริการ (Server side script)

อะไรคือภาษาสคริปต์

ภาษาสคริปต์ (JavaScript) เป็นภาษาสคริปต์แบบสั้น ๆ (Lightweight programming language) ที่สามารถประมวลผลในฝั่งเครื่องผู้ร้องขอบริการ สามารถใช้งานร่วมกับเอกสารอุ๊กอิ๊ก ภาษาสคริปต์ เพื่อทำให้เว็บไซต์สามารถตรวจสอบจับการเคลื่อนไหวภายในเว็บไซต์ ทำให้สามารถสนองต่อ หรือโต้ตอบกับผู้ใช้งานได้

ภาษาสคริปต์ทำให้เว็บไซต์ใช้งานได้ง่ายมากยิ่งขึ้น รวมไปถึงสร้างแรงจูงใจในการใช้งานเว็บไซต์ ด้วย เนื่องจากภาษาสคริปต์เป็นภาษาคอมพิวเตอร์ที่ถูกกำกับมาตรฐานโดยองค์กร W3C ซึ่งเป็นองค์กรที่กำหนดมาตรฐานของเว็บไซต์ทั้งหมดที่อุ๊กอิ๊ก และซีเอสเอส จึงทำให้สามารถแสดงผลได้ในทุก ๆ เว็บเบราว์เซอร์ และรองรับการแสดงผลในทุก ๆ อุปกรณ์

ข้อดีของภาษาสคริปต์

1. ภาษาสคริปต์สามารถใช้งานร่วมกับเอกสารอุ๊กอิ๊ก โดยการแทรกคำสั่งลงในเอกสาร และสามารถทดสอบการแสดงผลของคำสั่งได้ทันที (hellomyweb, 2553)
2. ง่ายต่อการใช้งานคำสั่ง โดยผู้พัฒนาไม่จำเป็นต้องมีพื้นฐานโปรแกรมมากนัก
3. ผู้พัฒนาสามารถพัฒนาเป็นโปรแกรมบนเว็บไซต์เพื่อประมวลผลข้อมูลอย่างง่ายได้
4. รองรับการแสดงผลในทุก ๆ เว็บเบราว์เซอร์ ทุกระบบปฏิบัติการ และไม่จำกัดกับอุปกรณ์

5. เป็นภาษาคอมพิวเตอร์ที่เปิดให้ใช้งานได้ฟรี ไม่มีค่าใช้จ่าย
6. ใช้ทรัพยากรเครื่องน้อย โดยผู้พัฒนาสามารถรับทดสอบคำสั่งได้โดยไม่จำเป็นต้องใช้เครื่องคอมพิวเตอร์ที่มีคุณสมบัติสูง

จา瓦สคริปต์ทำอะไรได้บ้าง

1. จา瓦สคริปต์ทำให้ผู้พัฒนาสามารถโปรแกรมได้ และรับทดสอบบนเครื่องผู้พัฒนาได้เลย โดยไม่ต้องใช้ภาษาคริปต์อื่น หรือ ไม่ต้องอัปโหลดขึ้นบนเครื่องแม่ข่าย
2. ใช้สำหรับตรวจสอบข้อมูลที่ได้จากฟอร์มในหน้าเว็บเพจ ซึ่งสามารถกำหนดรูปแบบข้อมูล ที่ต้องการเพื่อป้องกันข้อผิดพลาดของข้อมูลที่ได้จากผู้ใช้งาน
3. จา瓦สคริปต์สามารถสร้างคุกกี้ (Cookies) เพื่อจดจำสถานะการเข้าใช้งานเว็บไซต์
4. จา瓦สคริปต์สามารถสร้างลูกเล่น เพื่อความน่าสนใจของเว็บไซต์ เช่น การสร้างภาพเคลื่อนไหว การตรวจจับการใช้งานเว็บไซต์เพื่อตอบโต้ หรือสนองตอบต่อผู้ใช้งานเว็บไซต์ เพื่อให้เว็บไซต์มีปฏิสัมพันธ์ เป็นต้น
5. จา瓦สคริปต์สามารถปรับเปลี่ยนคุณสมบัติของแท็กคำสั่งเช่น `` หรือ `<input>` ได้โดยไม่ต้องโหลดหน้าเว็บเพจใหม่ ทำให้ตอบสนองต่อการใช้งาน หรือ ตอบสนองต่อการคลิกของผู้ใช้งานในหน้าเว็บเพจ

วิธีการใช้งานจาวาสคริปต์

การใช้งานภาษาจาวาสคริปต์มี 2 รูปแบบ คือ

1. ฝังคำสั่งไว้ภายในเอกสารอโซชีเอนเมล (Internal JavaScript) ซึ่งการแทรกคำสั่นจา华สคริปต์ไว้ในเอกสารอโซชีเอนเมล ซึ่งผู้พัฒนาจะต้องวางรหัสคำสั่งไว้ในระหว่างแท็ก `<script>` และปิดท้ายด้วย `</script>` เป็นการกำหนดขอบเขตของภาษาจา华สคริปต์ที่แทรกลงในหน้าเอกสาร

```
<script>
.....รหัสคำสั่งจา华สคริปต์.....
</script>
```

ผู้พัฒนาสามารถเลือกแทรกคำสั่งในเอกสารอโซชีเอนเมล ซึ่งสามารถวางคำสั่งไว้ได้ทั้งในส่วนหัวของเอกสาร คือ ในระหว่างแท็ก `<head> </head>` หรือ เลือกว่างคำสั่งไว้ในระหว่างแท็ก `<body> </body>` ของเอกสารอโซชีเอนเมลก็ได้ ดังตัวอย่าง

```

<!DOCTYPE html>

<html>
  <head>
    <script>
      .....รหัสคำสั่ง javasCript.....
    </script>
  </head>
  <body>
    <script>
      .....รหัสคำสั่ง javasCript.....
    </script>
  </body>
</html>

```

2. เรียกใช้คำสั่ง javasCript จากภายนอก (External JavaScript) เป็นรูปแบบการเขียน รหัสคำสั่ง javasCript แล้วบันทึกแยกเป็นไฟล์เอกสาร javasCript (*.js) อยู่ภายนอกแยกจากไฟล์เอกสารอื่นๆ เช่น .html .css .php ฯลฯ ซึ่งทำให้สามารถเรียกใช้คำสั่ง javasCript ได้โดยตรง ไม่ต้องระบุชื่อไฟล์ ทำให้แก้ไขรหัสคำสั่งได้ง่าย เพิ่มความเร็วในการโหลด และไม่เปลี่ยนพื้นที่ใน การจัดเก็บหน้าเว็บไซต์ การเรียกใช้งานคำสั่ง javasCript จากภายนอก ดังตัวอย่าง

```

<!DOCTYPE html>

<html>
  <head>
    <script src="myScript.js"></script>
  </head>
  <body>
  </body>
</html>

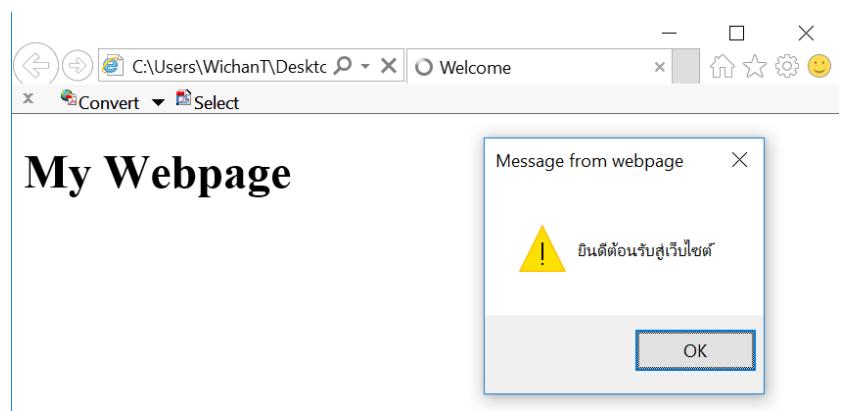
```

คำสั่งสำหรับการแสดงผล

คำสั่งสำหรับการแสดงผลข้อมูล (JavaScript output) โดยใช้รหัสคำสั่ง JavaSciript แบ่งออกเป็น 4 รูปแบบ ดังนี้ (w3schools, 2016)

1. การเขียนข้อมูลลงในกล่องข้อความแจ้งเตือน (alert box) โดยใช้คำสั่ง window.alert() รูปแบบการทำงานจะปรากฏเป็นกล่องข้อความเพื่อแจ้งข้อมูลให้กับผู้ใช้งาน เช่น การแจ้งเตือน ผลของ การประมวลผลข้อมูล ตัวอย่างดังภาพที่ 4.1

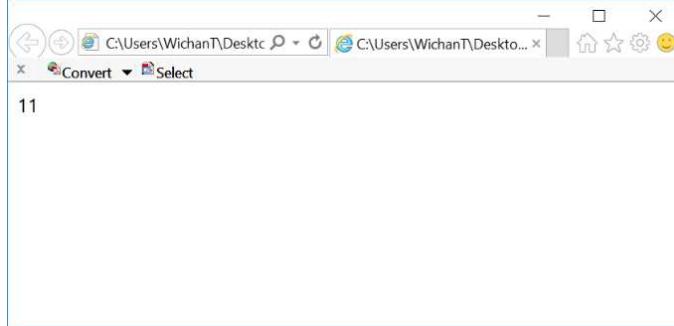
```
<!DOCTYPE html> <html>
<head>
<meta charset="utf-8">
<title>Welcome</title>
</head>
<body>
<h1>My Webpage</h1>
<script>
window.alert("ยินดีต้อนรับสู่เว็บไซต์");
</script>
</body>
</html>
```



ภาพที่ 4.1 การแสดงผลกล่องข้อความแจ้งเตือน

2. การเขียนข้อมูลลงในเอกสารเอกสารเชทีอิมเมล โดยใช้คำสั่ง document.write() เป็นรูปแบบการแสดงผลข้อมูลเพื่อแทรกลงในหน้าเว็บเพจ ซึ่งอาจจะเป็นข้อมูลที่ได้จากการประมวลผล หรือ การแสดงผลข้อความโดยใช้คำสั่ง JavaSciript เป็นต้น ตัวอย่างดังภาพที่ 4.2

```
<!DOCTYPE html><html><body>
<script>
    document.write(8 + 3);
</script>
</body></html>
```



ภาพที่ 4.2 การเขียนข้อมูลลงในเอกสารเอชทีเอ็มแอล

3. การเขียนข้อมูลลงในอิลิเมนต์ของเอกสารเอชทีเอ็มแอล โดยใช้คำสั่ง innerHTML การเข้าถึงอิลิเมนต์ของของเอกสารเอชทีเอ็มแอล สามารถใช้คำสั่ง `document.getElementById(id)` ซึ่งเป็นการอ้างอิงค่าจากแอฟทริบิวท์ `id` ของแท็กคำสั่งเอชทีเอ็มแอล แต่คำสั่ง `innerHTML` เป็นการกำหนดค่าเนื้อหาที่อยู่ในระหว่างแท็กคำสั่งเอชทีเอ็มแอล ตัวอย่างดังภาพที่ 4.3

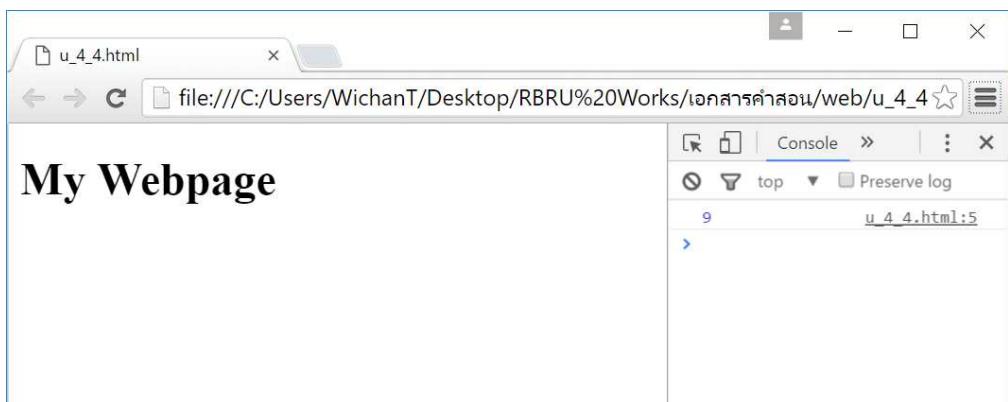
```
<!DOCTYPE html>
<html>
<body>
    <h1>Value of 4 and 5</h1>
    <p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = 4 + 5;
</script>
</body>
</html>
```



ภาพที่ 4.3 การเขียนข้อมูลลงในอัลิเมนต์ของเอกสารเอชทีเอ็มแอล

4. การเขียนข้อมูลลงในส่วนแก้ไขข้อผิดพลาดของเว็บบราวเซอร์ (Debugging) โดยใช้คำสั่ง `console.log()` เพื่อส่งค่าข้อมูลให้กับเว็บเบราว์เซอร์ ซึ่งสามารถเรียกดูค่าโดยการกดปุ่ม F12 จากนั้นเลือกเมนู Console ตัวอย่างดังภาพที่ 4.4

```
<!DOCTYPE html>
<html>
<body>
<h1>My Webpage</h1>
<script>
    console.log(4 + 5);
</script>
</body>
</html>
```



ภาพที่ 4.4 การเขียนข้อมูลลงในส่วนแก้ไขข้อผิดพลาดของเว็บเบราว์เซอร์

รูปแบบคำสั่งในภาษาจาวาสคริปต์

รูปแบบการเขียนคำสั่ง (Syntax) เป็นกฎ ระเบียบ หรือ กติกาในการเขียนคำสั่งภาษาจาวาสคริปต์ เพื่อให้สามารถแสดงผลคำสั่งตามที่ต้องการได้ โดยไม่มีข้อผิดพลาด (Error) ของคำสั่ง ซึ่งแต่ละภาษาจะมีรูปแบบการเขียนคำสั่งที่แตกต่างกัน ซึ่งภาษาสคริปต์เป็นภาษาคริปต์ที่มีรูปแบบการเขียนคือ เมื่อสิ้นสุดการกำหนดค่าหรือเรียกใช้งานคำสั่ง จะต้องจบท้ายคำสั่งด้วยเครื่องหมาย เชมิโคล่อน (;) เสมอ ตัวอย่างเช่น หากต้องการระบุค่าตัวแปร x ให้มีค่าเท่ากับ 3 ตัวแปร y มีค่าเท่ากับ 6 และนำค่าจากตัวแปร x และ y มาบวกกัน สามารถกำหนดค่าได้ดังต่อไปนี้ (w3schools, 2016)

```
var x = 3;
var y = 6;
var z = x + y;
```

ส่วนประกอบของรูปแบบการเขียนคำสั่งภาษาจาวาสคริปต์ ประกอบไปด้วยส่วนหลัก ๆ คือ

1. ส่วนกำหนดค่า (Values) การกำหนดค่าในภาษาจาวาสคริปต์มี 2 รูปแบบ คือ การกำหนดค่าแบบตายตัว (Fixed value) เรียกว่า Literals และการกำหนดค่าให้กับตัวแปร (Variable value) เรียกว่า Variable ซึ่งสามารถระบุค่าได้ทั้งที่เป็นตัวเลข (Number) และตัวอักษร (Text) หากเป็นตัวอักษรการระบุค่าจะต้องมีเครื่องหมาย “-” (Double quote) หรือ ‘-’ (Single quote) สามารถกำหนดค่าได้ดังต่อไปนี้

```
10.50
1001
"Wichan Thumthong"
'Rambhai Barni Rajabhat University'
```

2. ส่วนตัวแปร (Variables) ตัวแปรสำหรับการเขียนโปรแกรมคอมพิวเตอร์จะถูกกำหนดขึ้นมาเพื่อใช้สำหรับการจัดเก็บข้อมูล สำหรับการเรียกไปใช้งานในภายหลัง ซึ่งในภาษาจาวาสคริปต์นี้จะเริ่มต้นการประกาศตัวแปร (Declare variable) โดยใช้คำสั่ง var ตามด้วยคีย์เวิร์ดสำหรับระบุชื่อตัวแปรที่ต้องการ

การระบุค่าตัวแปร (Assign value) จะใช้เครื่องหมายเท่ากับ (Equal sign : =) เพื่อระบุค่าให้กับตัวแปร เช่น ต้องการกำหนดให้ x เป็นตัวแปร จากนั้นระบุค่าตัวแปร x ให้มีค่าเท่ากับ 9 ดังต่อไปนี้

```
var x;
x = 9;
```

3. ตัวดำเนินการ (Operator) เป็นการระบุค่าสำหรับตัวแปรที่ต้องการ ซึ่งจะใช้เครื่องหมายเท่ากับ (=) ดังตัวอย่าง

```
var x = 5;
var y = 'text value';
```

ซึ่งหากมีตัวดำเนินการหารายตัว เช่น ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operator) สามารถใช้เครื่องหมายวงเล็บ เพื่อแยกตัวดำเนินการ ดังตัวอย่าง

```
(4 + 5) * 10 ;
```

4. การแสดงผลลัพธ์ (Expression) เป็นการประมวลผลค่าของตัวแปร กับเครื่องหมายที่เป็นตัวดำเนินการ ซึ่งค่าที่ได้จะเป็นผลลัพธ์ที่ได้จากการคำนวน เช่น $2 * 10$ ผลที่ได้คือ 20 โดยสามารถประมวลผลโดยใช้ชื่อของตัวแปรก็ได้ ดังตัวอย่าง

```
var x = 5 + 6;
var y = x * 10;
```

นอกจากนั้นผู้พัฒนาสามารถเชื่อมค่าตัวแปรหลายประเภทเข้าด้วยกัน โดยใช้เครื่องหมายบวก ยกตัวอย่างเช่น “Wichan” + “ ” + “Thumthong” ผลลัพธ์ที่ได้คือ “Wichan Thumthong”

5. การเพิ่มคำอธิบายคำสั่ง (Comment) เป็นรูปแบบการปิดการแสดงผลคำสั่งที่ต้องการ หรือ กำหนดขอบเขตของคำอธิบายคำสั่ง โปรแกรมจะไม่นำເອາคำสั่งหรือข้อความที่อยู่ในส่วนนี้ไปประมวลผล ซึ่งสามารถเพิ่มคำอธิบายคำสั่งได้แบบบรรทัดเดียวโดยใช้เครื่องหมาย // (Double slash) หรือหากต้องการเพิ่มคำอธิบายคำสั่งแบบหลายบรรทัดสามารถใช้เครื่องหมาย /* และ */ เพื่อกำหนดขอบเขตของคำอธิบายคำสั่ง ดังตัวอย่าง

```
var x = 9; //ตัวแปรถูกนำเอามาประมวลผล
// var x = 6; ตัวแปรจะไม่ถูกนำเอามาประมวลผล
```

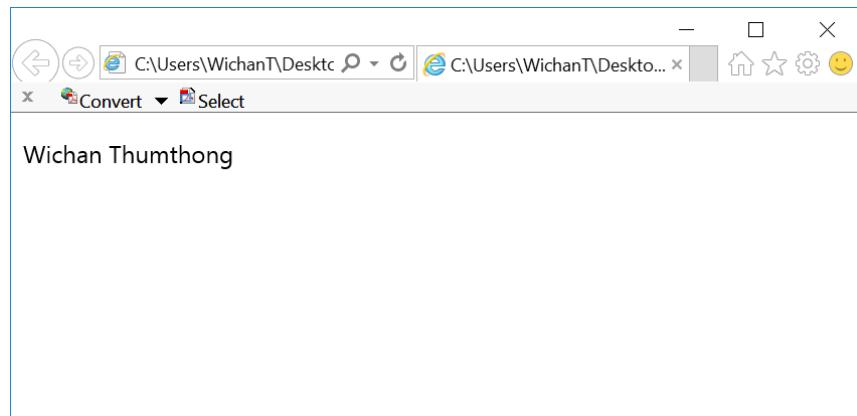
6. การระบุชื่อตัวแปร (Keyword) ในภาษาจาวาสคริปต์ห้ามชื่นต้นด้วยตัวเลข ซึ่งอาจจะชื่นต้นชื่อตัวแปรโดยใช้ตัวอักษร เครื่องหมาย Underscore (_) หรือ สัญลักษณ์ \$ (Dollar sign) ซึ่งตัวอักษรรถด์ไปสามารถระบุตัวเลข ตัวอักษร เครื่องหมาย Underscor (_) เครื่องหมาย Hyphen (-) หรือ สัญลักษณ์ \$ เป็นต้น ยกตัวอย่างเช่น first-name, last-name, first_name, last_name, หรือ FirstName, LastName เป็นต้น เครื่องหมาย Hyphen (-) ไม่นิยมนิมมาใช้ในชื่อของตัวแปร แต่จะถูกจ่อไปใช้สำหรับเป็นเครื่องหมายลบ (Subtractions) และห้ามใช้คำส่วน (Reserved words)

การอ้างอิงชื่อตัวแปร (Identifier) จะต้องคำนึงถึงลักษณะตัวพิมพ์ (Case sensitive) ของชื่อตัวแปร เช่น lastname และ LastName ถือว่าเป็นตัวแปรคนละตัวกัน

การเขียนคำสั่งในภาษาจาวาคริปต์

การเขียนคำสั่ง (Statement) เป็นการแนะนำรูปแบบที่จะดำเนินการตามลำดับของคำสั่งในเอกสารเอชทีเอ็มแอลบนเว็บเบราว์เซอร์ ยกตัวอย่างเช่น การเขียนคำสั่งเพื่อบอกเว็บเบราว์เซอร์ให้เขียนข้อความว่า “Wichan Thumthong” ลงในอลิเมนต์ของเอชทีเอ็มแอล โดยอ้างอิงจากໄอิดีชื่อว่า “demo” ดังตัวอย่างดังภาพที่ 4.5

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "Wichan Thumthong";
</script>
</body>
</html>
```



ภาพที่ 4.5 การเขียนคำสั่งเพื่อเปลี่ยนค่าในอลิเมนต์ของเอชทีเอ็มแอล

ซึ่งสามารถเพิ่มรูปแบบการเขียนโปรแกรม เพื่อดำเนินการตามคำสั่งที่เขียน ซึ่งจะแปลผลคำสั่ง เป็นแบบลำดับขั้น ตามที่ผู้พัฒนาได้เขียนไว้ ยกตัวอย่างเช่น การแสดงผลตัวแปรที่ได้จากการ ประมวลผลทางคณิตศาสตร์ในอลิเมนต์ของเอชทีเอ็มแอล ดังตัวอย่าง

```
var x = 4;
var y = 5;
var z = x + y;
document.getElementById("demo").innerHTML = z;
```

เมื่อสิ้นสุดคำสั่งในแต่ละบรรทัดจะต้องระบุเครื่องหมาย “;” (Semicolon) ซึ่งในการเขียนคำสั่งหากสิ้นสุดการระบุคำสั่งแล้ว สามารถเขียนคำสั่งอื่นต่อท้ายเป็นบรรทัดเดียวกันได้ ดังตัวอย่าง

```
var x = 4; var y = 5;
var z = x + y;
```

ตัวแปรในภาษาจาวาสคริปต์

ตัวแปรในภาษาจาวาสคริปต์ ทำหน้าที่ในการเก็บค่าต่าง ๆ ไว้สำหรับการนำมาใช้ในการอ้างอิงเพื่อการประมวลผล หรือแสดงผลข้อมูล การประกาศตัวแปรในภาษาจาวาสคริปต์จะต้องขึ้นต้นด้วยอักษร var ตามด้วยชื่อตัวแปรที่ต้องการ เช่น

```
var x;
```

หรืออาจจะประกาศตัวแปรจากนั้นระบุค่าตัวแปรไปด้วยเล็กก็ได้ เพื่อความสะดวกในการเขียนคำสั่ง ดังตัวอย่าง เป็นการประกาศตัวแปร x ระบุค่าตัวแปรเป็น 4 ประกาศตัวแปร y ค่าตัวแปรเป็น 5 และประกาศตัวแปร z เก็บผลที่ได้จากการบวกกันของตัวแปร x และ y เป็นต้น

```
var x = 4 ;
var y = 5 ;
var z = x + y ;
```

ค่าของตัวแปรอาจเป็นตัวเลขหรือตัวอักษรก็ได้ ซึ่งหากต้องการระบุค่าตัวแปรเป็นตัวอักษรจะต้องระบุเครื่องหมาย “ ” (Double quote) หรือ ‘ ’ (Single quote) เพื่อรับค่า ดังตัวอย่าง

```
var price = 99.99 ;
var name = "Wichan";
var position = 'programmer';
```

การประกาศตัวแปรสามารถใช้คำสั่ง var ครั้งเดียว แต่ประกาศและระบุค่าตัวแปรได้หลายตัวโดยคั่นตัวแปรโดยใช้เครื่องหมาย , (Comma) ดังตัวอย่าง

```
var price = 99.99, name = "Wichan",
position = 'programmer';
```

หรืออาจเขียนให้อยู่ในบรรทัดเดียวกันก็ได้

```
var price = 99.99 , name = "Wichan", position = 'programmer';
```

หากต้องการแทรกตัวดำเนินการสามารถใช้เครื่องหมาย + เพื่อรวมค่าได้ ดังตัวอย่าง ค่าตัวแปร z จะมีค่าเท่ากับ 29

```
var z = 10 + 4 + 15 ;
```

หากมีเครื่องหมาย “ ” ที่ตัวเลข โปรแกรมจะมองว่าเป็นตัวอักษร ไม่สามารถนำรวมกันได้ ซึ่งผลที่ได้จะแตกต่างกัน ดังตัวอย่าง ค่าตัวแปร z ที่ได้เป็น 10415

```
var z = "10" + 4 + 15 ;
```

หรือหากเป็นการรวมตัวอักษรสามารถใช้เครื่องหมาย + เพื่อรวมค่าตั้งตัวอย่าง

```
var name = "Wichan" + " " + "Thumthong" ;
```

ข้อกำหนดในการตั้งชื่อตัวแปร

การประกาศตัวแปรในภาษาจาวาสคริปต์เพื่อใช้ในการอ้างอิง หรือเรียกไปใช้งาน จะต้องใช้ชื่อตัวแปรที่ไม่ซ้ำกัน ซึ่งอาจจะประกาศตัวแปรโดยใช้ชื่อแบบสั้น ๆ เช่น var x = 1; var y = "wichan"; หรืออาจจะใช้ชื่อตัวแปรที่ตรงตามข้อมูลที่เก็บเพื่อความสะดวกและป้องกันการสับสนจากการเรียกใช้งานตัวแปร เช่น var age = 25 ; var name = "wichan" เป็นต้น ข้อกำหนดในการตั้งชื่อตัวแปรในภาษาจาวาสคริปต์ ดังนี้

1. ชื่อตัวแปรจะต้องประกอบไปด้วย ตัวอักษร, ตัวเลข, เครื่องหมาย Underscores หรือเครื่องหมาย \$ (Dollar sign)
2. การประกาศชื่อตัวแปรจะต้องเริ่มต้นด้วยตัวอักษรเท่านั้น
3. ควรคำนึงถึงเรื่องตัวพิมพ์เล็กตัวพิมพ์ใหญ่ (Case sensitive) เช่น ตัวแปร y และ Y ถือว่าเป็นตัวแปรคละตัวกัน
4. ไม่ควรตั้งชื่อโดยใช้ชื่อที่สงวนไว้ (Reserved words) ซึ่งส่วนมากจะสงวนไว้เพื่อใช้เป็นคำสั่ง หรือ พังก์ชันในภาษาจาวาสคริปต์ เป็นต้น ดังตารางที่ 4.1

ตารางที่ 4.1 คำสงวนในภาษาจาวาสคริปต์

คำสงวน	รายละเอียด
break	หยุดการวนรอบตามคำสั่ง
continue	การออกจากภาระวนรอบการประมวลผลเพื่อเริ่มต้นใหม่
debugger	หยุดการประมวลผลคำสั่งจาวาสคริปต์ เพื่อดำเนินการแก้ไขพังก์ชัน
do while	เป็นการวนรอบการทำงานไปเรื่อย ๆ จนกว่าเงื่อนไขที่ระบุจะเป็นจริง
for	เป็นการดำเนินการของโปรแกรมเพื่อรับເງື່ອນໄຂຈົນກວ່າຈະເກີດເຫຼຸກຮັບສິນ
function	เป็นการประกาศพังก์ชันของภาษาจาวาสคริปต์
if else	เป็นการดำเนินการของโปรแกรมเพื่อรับເງື່ອນໄຂໃຫ້ກັບคำสั่ง
return	การออกจากพังก์ชัน

ตารางที่ 4.1 คำส่วนในภาษาจาวาสคริปต์ (ต่อ)

คำส่วน	รายละเอียด
Switch	การดำเนินการของโปรแกรม กรณีที่เกิดเหตุการที่แตกต่างกัน
try catch	จัดการเกี่ยวกับข้อผิดพลาดของรูปแบบการเขียนโปรแกรม
Var	การประกาศตัวแปร

ที่มา : (w3schools, 2016)

ชนิดของข้อมูลในภาษาจาวาสคริปต์

ค่าของตัวแปรในภาษาจาวาสคริปต์มีหลากหลายรูปแบบขึ้นอยู่กับความเหมาะสมของค่าที่เก็บ เช่น ตัวแปรที่เป็นตัวเลข (Number) ตัวอักษร (String) อาเรย์ (Array) หรือ วัตถุ (Object) เป็นต้น ยกตัวอย่างเช่น

```
var age = 16; // ตัวแปรชนิดตัวเลข (number)
var lastName = "Thumgthong"; // ตัวแปรชนิดตัวอักษร (string)
var cars = ["Toyota", "Honda", "BMW"]; // ตัวแปรชนิดอาเรย์ (array)
var me = {firstName:"Wichan", lastName:"Thumthong"}; // ตัวแปรชนิดวัตถุ (object)
```

ในการเขียนโปรแกรม ชนิดของข้อมูลมีความสำคัญเป็นอย่างมาก เพื่อใช้สำหรับการดำเนินการ หรือ ประมวลผลตัวแปรที่ถูกต้อง ซึ่งการประมวลผลจะเริ่มต้นจากซ้ายไปขวา ซึ่งการวางแผนตัวแปรที่แตกต่างกัน จะทำให้ผลของการดำเนินการที่แตกต่าง เช่น ตัวแปร x โปรแกรมจะดำเนินการนำ 16 รวมกันกับ 4 ก่อนที่จะรวมกันกับตัวอักษร ค่าที่ได้ของตัวแปร x คือ 20WichanT แต่ตัวแปร y ค่าที่ได้คือ WichanT164 เนื่องจากโปรแกรมจะมองว่าเป็นตัวอักษร เป็นต้น

```
var x = 16 + 4 + "WichanT";
var y = "WichanT" + 16 + 4;
```

การประกาศตัวแปรและระบุค่าตัวแปร โปรแกรมจะระบุชนิดของตัวแปรแบบอัตโนมัติ ตามค่าตัวแปรที่กำหนดให้ตัวแปร ดังนี้

1. **ตัวแปรชนิดตัวเลข (Number)** เป็นการระบุค่าตัวแปรเป็นตัวเลข ซึ่งสามารถนำไปใช้ในการประมวลผล หรือ ดำเนินการทางคณิตศาสตร์ได้ ตัวเลขมีหลายรูปแบบ ดังต่อไปนี้

```
var a = 45.97; // ตัวเลขที่มีจุดทศนิยม
var b = 34; // ตัวเลขจำนวนเต็ม
var y = 123e5; // จำนวนเลขคูณ 10 ยกกำลัง 5 มีค่าเท่ากับ 12300000
var z = 123e-5; // จำนวนเลขคูณ 10 ยกกำลัง -5 มีค่าเท่ากับ 0.00123
```

2. ตัวแปรชนิดตัวอักษร (String) เป็นการระบุค่าตัวแปรที่เป็นตัวอักษร ซึ่งจะใช้เครื่องหมาย “ ” หรือ ‘ ’ เพื่อรับค่าตัวแปรได้ ดังตัวอย่าง

```
var answer = "It's alright";           // Single quote ภายใน double quotes
var answer = 'He is called "WichanT"'; // Single quotes ภายใน double quotes
var answer = 'He is called "WichanT"'; // Double quotes ภายใน single quotes
```

3. ตัวแปรชนิดบูลีน (Boolean) ค่าที่ระบุได้มีเพียง 2 สถานะ คือ จริง (True) และ เท็จ (False) โดยทั่วไปจะใช้ดำเนินการเพื่อใช้สำหรับการตรวจสอบเงื่อนไขในการเขียนโปรแกรม ดังตัวอย่าง

```
var x = true ;
var y = false ;
```

4. ตัวแปรชนิดอารเรย์ (Array) เป็นการเก็บข้อมูลเป็นแบบชุดซึ่งจะเขียนโดยใช้เครื่องหมาย “[]” (Square brakcket) และคั่นข้อมูลแต่ละตัวด้วยเครื่องหมาย “,” (Comma) ยกตัวอย่างเช่น การประกาศตัวแปร car เพื่อกีบข้อมูลที่เป็นชื่อชนิดของรถ ดังตัวอย่าง

```
var cars = ["Toyota", "Honda", "BMW"];
```

5. ตัวแปรชนิดวัตถุ (Object) การเขียนจะถูกกำหนดขอบเขตโดยวงเล็บปีกกา { } (Curly braces) โดยสามารถระบุชื่อตัวแปร และค่าของตัวแปร เชื่อมหลายตัวแปรโดยใช้เครื่องหมายคอมม่า ดังตัวอย่าง

```
var me = {firstName:"Wichan", lastName:"Thumthong", age:25, eyeColor:"black"} ;
```

6. ตัวแปรที่ไม่สามารถระบุชนิดของตัวแปรได้ (Undefined) เป็นการประกาศตัวแปรที่ไม่ได้ระบุค่าตัวแปร ดังตัวอย่าง เป็นการประกาศตัวแปร แต่ไม่ได้ระบุค่า หรือ ระบุค่าเป็น Undefined

```
var person ;
person = undefined ;
```

7. ตัวแปรที่มีค่าว่าง (Empty value or null) เป็นการประกาศตัวแปรที่ระบุค่าตัวแปร เป็นค่าว่างเปล่า ดังตัวอย่าง

```
var car = "" ;
```

ค่า null หมายถึง Nothing เป็นการระบุค่าว่างเปล่าให้กับวัตถุ (Object) ดังตัวอย่าง

```
var me = null;
```

หากต้องการทราบชนิดของตัวแปร ในภาษาจาวาสคริปต์สามารถใช้คำสั่ง typeof เพื่อตรวจสอบชนิดของตัวแปรที่ต้องการได้ ตัวอย่างเช่น

```
typeof "WichanT"           // จะได้ค่า string
typeof 3.14                  // จะได้ค่า number
```

ตัวดำเนินการในภาษาจาวาสคริปต์

ตัวดำเนินการ (Operator) เป็นการดำเนินการอย่างโดยย่างหนึ่งกับตัวแปรที่ต้องการ ซึ่งมีหลายรูปแบบ ดังนี้ (w3schools, 2016)

1. ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators) เป็นตัวดำเนินการสำหรับใช้ในการคำนวณค่าของตัวแปร ตัวอักษร หรือ ตัวเลข ดังตารางที่ 4.2

ตารางที่ 4.2 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ค่าที่ได้ (ตัวแปร x)
+	บวก (Addition)	var x = 1 + 2 ;	3
-	ลบ (Subtraction)	var x = 5 - 3 ;	2
*	คูณ (Multiplication)	var x = 3 * 5 ;	15
/	หาร (Division)	var x = 12 / 6 ;	2
%	หารเอาเศษ (Modulus)	var x = 5 % 3 ;	2
++	เพิ่มค่าหนึ่ง (Increment)	var x = 5 ; ++x ;	6
--	ลดค่าหนึ่ง (Decrement)	Var x = 5 ; --x ;	4

2. ตัวดำเนินการสำหรับกำหนดค่า (Assignment operators) เป็นการกำหนดค่าให้กับตัวแปรในภาษาจาวาสคริปต์ ดังตารางที่ 4.3

ตารางที่ 4.3 ตัวดำเนินการสำหรับกำหนดค่า

ตัวดำเนินการ	ตัวอย่างการใช้งาน	เมื่ออนกันกับ	ค่าที่ได้ (ตัวแปร x)
=	var x = 5 ;	var x = 5 ;	5
+=	var x = 2 ; x += 5 ;	var x = 2 ; x = x + 5 ;	7
-=	var x = 5 ; x -= 2 ;	var x = 5 ; x = x - 2 ;	3
*=	var x = 5 ; x *= 2 ;	var x = 5 ; x = x * 2 ;	10
/=	var x = 10 ; x /= 2 ;	var x = 10 ; x = x / 2 ;	5
%=	var x = 5 ; x %= 3 ;	var x = 5 ; x = x % 3 ;	2

3. ตัวดำเนินการสำหรับตัวอักษร (String operators) เป็นการเชื่อมตัวแปรที่เป็นประเภทอักษร โดยสามารถใช้ตัวดำเนินการ “+” ยกตัวอย่างเช่น

```
var name = "Wichan" ; var surname = "Thumthong" ;
var fullname = name + " " + surname ;
```

จากตัวอย่างค่าตัวแปร fullname ที่ได้จะมีค่าเท่ากับ “Wichan Thumthong” นอกจากเครื่องหมาย “+” แล้ว ยังสามารถใช้เครื่องหมาย “+=” ซึ่งเป็นการเชื่อมค่าให้กับตัวแปร กับค่าตัวแปร เดิม ดังตัวอย่าง ค่าตัวแปร uname ที่ได้คือ “Rambhai Barni Rajabhat University”

```
var uname = "Rambhai Barni ";
uname += "Rajabhat University";
```

เครื่องหมาย “+” หากเป็นการเชื่อมระหว่างตัวเลขกับตัวเลข จะหมายถึงการบวก แต่หากค่าตัวแปรเป็นแบบตัวอักษร จะหมายถึงการเชื่อมตัวแปร ดังตัวอย่าง

```
var x = 3 + 6; // ค่าที่ได้คือ 9
var y = "3" + 6; // ค่าที่ได้คือ 36
var z = "Hello" + 6; // ค่าที่ได้คือ Hello6
```

4. ตัวดำเนินการสำหรับการเปรียบเทียบ (Comparison operators) เป็นตัวดำเนินการสำหรับการเปรียบเทียบหาค่าความจริงจาก 2 นิพจน์ ซึ่งค่าที่ได้มีเพียงจริง (True) หรือเท็จ (False) เท่านั้น ตัวอย่างตัวดำเนินการดังตารางที่ 4.4

ตารางที่ 4.4 ตัวดำเนินการสำหรับการเปรียบเทียบ

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ค่าที่ได้ (ตัวแปร x)
==	เท่ากัน	var x = 3 ; x == 3 ;	จริง
==	เหมือนกัน	var x = 3 ; x === "3" ;	เท็จ
!=	ไม่เท่ากัน	var x = 3 ; x != 5 ;	จริง
!=	ไม่เหมือนกัน	var x = 3 ; x !== 3 ;	เท็จ
>	มากกว่า	var x = 3 ; x > 5 ;	เท็จ
<	น้อยกว่า	var x = 3 ; x < 5 ;	จริง
>=	มากกว่าหรือเท่ากัน	var x = 3 ; x >= 3 ;	จริง
<=	น้อยกว่าหรือเท่ากัน	var x = 3 ; x <= 4 ;	จริง

5. ตัวดำเนินการทางตรรกะ (Logical operators) เป็นการเชื่อมความสัมพันธ์ระหว่างค่าตัวแปร หรือ นิพจน์ต่าง ๆ ตัวอย่างตัวดำเนินการดังตารางที่ 4.5

ตารางที่ 4.5 ตัวดำเนินการทางตรรกะ

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ค่าที่ได้ (ตัวแปร x)
&&	และ (And)	var x = 3 ; (x == 3 && x == 5)	เท็จ
	หรือ (Or)	var x = 3 ; (x == 3 x == 5)	จริง
!	ปฏิเสธ (Not)	var x = 3 ; !(x == 3) ;	เท็จ

ฟังก์ชันและการเรียกใช้งาน

ฟังก์ชัน (Function) เป็นชุดคำสั่งในรูปแบบใดแบบหนึ่ง ซึ่งถูกเขียนเอาไว้ ซึ่งจะทำงานเมื่อมีการเรียกใช้งาน ตัวอย่างรูปแบบการเขียนฟังก์ชัน ดังนี้

```
function name (parameter1, parameter2, parameter3) {
    รหัสคำสั่งที่ต้องการประมวลผล
}
```

การกำหนดรูปแบบการเขียนฟังก์ชัน ทำได้โดยการสร้างฟังก์ชันซึ่งสามารถกำหนดชื่อฟังก์ชัน (Name) ที่ต้องการ جانนั้นตามด้วยเครื่องหมายวงเล็บ “()” ซึ่งสามารถส่งตัวแปรเข้าไปในฟังก์ชัน เพื่อประมวลผลตามที่ต้องการ โดยสามารถส่งตัวแปรได้หลายตัวโดยใช้เครื่องหมายคอมม่าเป็นตัวคั่นชื่อ ตัวแปรที่ส่งไป เช่น (parameter1, parameter2, parameter3) เป็นต้น ซึ่งชื่อตัวแปรนี้จะถูกเรียกไปใช้ในการอ้างอิงเพื่อการประมวลผลภายในฟังก์ชัน เรียกว่า อาร์กิวเมนต์ (Argument) ซึ่งอาจจะมีหรือไม่มีก็ได้ขึ้นอยู่กับการทำงานของฟังก์ชัน ซึ่งจะต้องมีเครื่องหมายวงเล็บปีกกาเพื่อรับบุข้อบแทนของฟังก์ชัน ดังตัวอย่าง

```
function myFunction(a, b) {
    return a * b;           // ฟังก์ชันส่งค่าที่ได้จากการ a คูณ b
}
```

การเรียกใช้งานฟังก์ชันสามารถแนบค่าตัวแปรเข้าไปในฟังก์ชันได้ ซึ่งสามารถเรียกใช้งานฟังก์ชันได้หลากหลายรูปแบบ เช่น เรียกใช้งานฟังก์ชันเมื่อผู้ใช้งานคลิกที่ปุ่มที่ต้องการ เรียกใช้งานจากคำสั่ง JavaScipt หรือ เรียกใช้งานฟังก์ชันด้วยตัวของฟังก์ชันเอง เป็นต้น ดังตัวอย่าง หากเรียกใช้งานฟังก์ชันโดยที่แนบค่าตัวแปรไปด้วย ฟังก์ชันจะดำเนินการตามคำสั่งแล้วส่งผลกลับมา ในตัวอย่างนี้ตัวแปร x จะมีค่าเท่ากับ 20

```
var x = myFunction(5, 4);
```

การเรียกใช้ฟังก์ชันโดยให้เรียกใช้ฟังก์ชันเมื่อผู้ใช้งานคลิกที่ปุ่มที่ต้องการ ยกตัวอย่างในเอกสาร เอชทีเอ็มแอล ดังนี้

```
<body>
<script>
    function myFunction(a, b) { x = a * b ; alert("Value of a * b = " + x); }
</script>
<button onclick="myFunction(5,4)">Click Me</button>
</body>
```

องค์ประกอบของวัตถุ

วัตถุ (Object) ในภาษาจาวาสคริปต์ เป็นรูปแบบการมองรหัสคำสั่งให้อยู่ในรูปแบบของวัตถุ ซึ่งจะทำให้ง่ายต่อการพัฒนาโปรแกรม เพื่อให้สามารถแยกโปรแกรมออกเป็นส่วน ๆ เมื่อมีนวัตถุ ซึ่งวัตถุที่พุดถึงสามารถแบ่งออกเป็น 2 องค์ประกอบ คือ

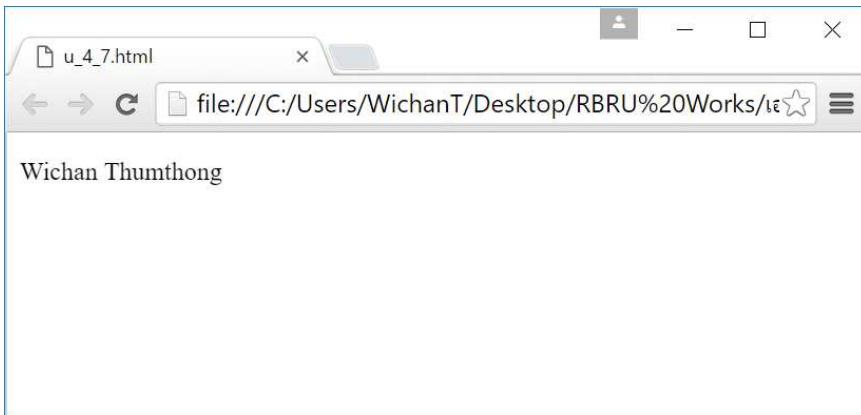
1. **คุณสมบัติของวัตถุ (Properties)** เมื่อมองโปรแกรมออกเป็นวัตถุแล้ว คุณสมบัติของวัตถุจะหมายถึง คุณสมบัติเหมือนกันของวัตถุที่ทุก ๆ วัตถุนั้นจะต้องมี แต่ก็ต่างกันที่ค่าของคุณสมบัติที่กำหนด ยกตัวอย่าง เช่น วัตถุคือรถยนต์ คุณสมบัติของรถยนต์ คือ ยี่ห้อรถ รุ่นของรถ น้ำหนักของรถ แรงม้า รวมไปถึงสีของรถ เป็นต้น หรือ วัตถุคือหนังสือ คุณสมบัติของหนังสือ คือ ชื่อเรื่อง ชื่อผู้แต่ง ราคา จำนวนหน้า หมายเลข ISBN เป็นต้น

2. **วิธีการทำงานของวัตถุ (Methods)** เป็นฟังก์ชันการทำงานที่มีผลต่อวัตถุนั้น ๆ ซึ่งทุก ๆ วัตถุจะต้องมี แต่แตกต่างกันที่เวลาในการเรียกใช้งานฟังก์ชัน ยกตัวอย่าง เช่น รถยนต์ มีวิธีการทำงาน คือ สตาร์ทเครื่องยนต์ ขับรถยนต์ การหยุดรถ เป็นต้น หรือหนังสือ วิธีการทำงานได้แก่ เปิดหนังสือ อ่านหนังสือ เป็นต้น ตัวอย่างดังภาพที่ 4.6

ตัวอย่างคุณสมบัติของวัตถุ

```
<body>
<p id="demo"></p>
<script>
var person = {
    firstName: "Wichan",
    lastName : "Thumthong",
    age      : 25
};

document.getElementById("demo").innerHTML =
person.firstName + " " + person.lastName;
</script>
</body>
```



ภาพที่ 4.6 การใช้งานคุณสมบัติของวัตถุ

การตรวจจับเหตุการณ์

เหตุการณ์ (Event) ที่เกิดขึ้นในเอกสารเช่นที่อีเมลแล้ว จา瓦สคริปต์สามารถตรวจจับเหตุการณ์ที่เกิดขึ้น และสามารถดำเนินการเพื่อตอบกับเหตุการณ์ที่เกิดขึ้นได้ ทำให้เว็บไซต์ที่พัฒนามีความน่าสนใจจากการมีปฏิสัมพันธ์ตอบในหน้าเว็บไซต์

การตรวจจับเหตุการณ์ที่เกิดขึ้นอาจเกิดจากเว็บเบราว์เซอร์ หรือ ผู้ใช้งาน เช่น การโหลดหน้าเว็บเพจ การกรอกข้อมูลผ่านฟอร์ม การโหลดภาพ หรือ การคลิกที่ปุ่ม เป็นต้น เมื่อเกิดเหตุการณ์ขึ้นในหน้าเว็บเพจ จา瓦สคริปต์สามารถดำเนินการบางอย่าง เพื่อสนองตอบเหตุการณ์นั้น โดยวิธีการตรวจจับสามารถเพิ่มเป็นแอ็ทริบิวท์ลงในแท็กเช่นที่อีเมลแล้ว มีวิธีการตรวจจับดังนี้

```
<html-element event="some JavaScript">
```

ตัวอย่างการใช้งาน

```
<button onclick='getElementById("demo").innerHTML=Date()'>The time is?</button>
```

ซึ่งสามารถเขียนให้อยู่ในรูปแบบฟังก์ชันได้ดังนี้

```
<body>
<button onclick="displayDate()">The time is?</button>
<script>
function displayDate() {
    document.getElementById("demo").innerHTML = Date();
}
</script>
<p id="demo"></p>
</body>
```

เหตุการณ์ที่ Java Script สามารถตรวจสอบในเอกสารอิเล็กทรอนิกส์ได้ ดังตารางที่ 4.6

ตารางที่ 4.6 เหตุการณ์ในเอกสารอิเล็กทรอนิกส์ที่อิเมลแอล

เหตุการณ์	ความหมาย
onchange	เหตุการณ์เมื่ออิลิเมนต์ในเอกสารอิเล็กทรอนิกส์เปลี่ยนแปลง
onclick	เหตุการณ์เมื่อผู้ใช้งานคลิกที่อิลิเมนต์ของเอกสารอิเล็กทรอนิกส์
onmouseover	เหตุการณ์เมื่อผู้ใช้งานเลื่อนเมาส์บนอิลิเมนต์ของเอกสารอิเล็กทรอนิกส์
onmouseout	เหตุการณ์เมื่อผู้ใช้งานเลื่อนเมาส์ออกจากอิลิเมนต์ของเอกสารอิเล็กทรอนิกส์
onkeydown	เหตุการณ์เมื่อผู้ใช้งานกดปุ่มบนคีย์บอร์ด
onload	เหตุการณ์เมื่อเบราว์เซอร์โหลดเว็บเพจสำเร็จ

การจัดการเกี่ยวกับวันและเวลา

ภาษา Java Script มีฟังก์ชันสำหรับจัดการเวลาอย่างหลากหลายประเภท ขึ้นอยู่กับการใช้งานและความต้องการของผู้พัฒนา เช่น ฟังก์ชันเกี่ยวกับการคำนวณ ฟังก์ชันเกี่ยวกับคณิตศาสตร์ เป็นต้น การจัดการเกี่ยวกับวันเวลาในโปรแกรมที่พัฒนาถือว่าเป็นส่วนสำคัญ เช่น การแสดงผลเวลา วันที่ ตรวจสอบจำนวนวัน เป็นต้น ซึ่ง Java Script มีฟังก์ชันสำหรับจัดการวันที่ เช่น Date() ; โดยสามารถกำหนดค่าได้ 4 รูปแบบ ดังนี้

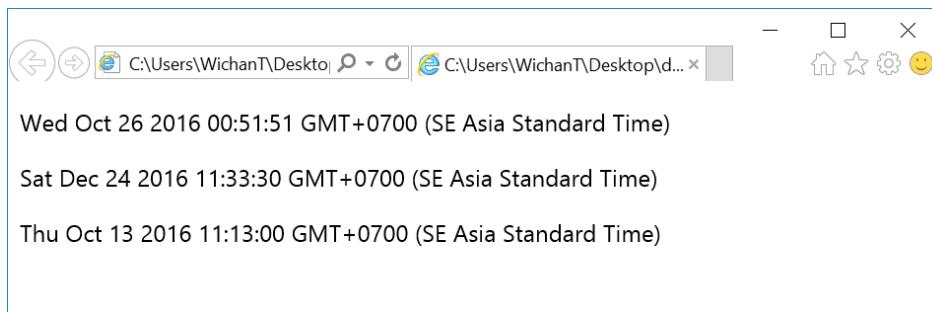
1. new Date() จะแสดงผลเป็นวันและเวลาปัจจุบัน
2. new Date(milliseconds) เป็นการระบุจำนวนเวลาหน่วยเป็นวินาที โดย Java Script จะเริ่มนับเป็นหน่วยมิลลิวินาที จากวันที่ 01 January, 1970 00:00:00 Universal Time (UTC) ซึ่งหนึ่งวันมี 86,400,000 มิลลิวินาที (w3schools, 2016)
3. new Date(dateString) เป็นการระบุค่าให้แสดงผลวันที่และเวลาตามที่กำหนด
4. new Date(year, month, day, hours, minutes, seconds, milliseconds) เป็นการระบุวันเวลาตามที่ต้องการโดยระบุค่าทั้งหมด 7 ค่า

ตัวอย่างดังภาพที่ 4.7

```

<!DOCTYPE html>
<html>
<body>
<p id="demo1">
</p><p id="demo2"></p>
<p id="demo3"></p>
```

```
<script>
document.getElementById("demo1").innerHTML = new Date();
document.getElementById("demo2").innerHTML = new Date(2016,11,24,11,33,30,0);
document.getElementById("demo3").innerHTML = new Date("October 13, 2016
11:13:00");
</script>
</body></html>
```



ภาพที่ 4.7 การใช้งานฟังก์ชัน date

การดึงค่าวันเวลาเพื่อนำไปใช้ในการประมวลผลคำสั่ง หรือแสดงผล สามารถเลือกค่าได้หลากหลาย เช่น ปี เดือน วัน ชั่วโมง นาที วินาที หรือ มิลลิวินาที เป็นต้น โดยใช้คำสั่งต่าง ๆ ดังตารางที่ 4.7 ยกตัวอย่างเช่น `getDay()` ค่าที่ได้เป็นตัวเลข 0-6 ซึ่ง 0 หมายถึงวันอาทิตย์ 1 หมายถึงวันจันทร์ จนถึง 6 หมายถึงวันเสาร์ เป็นต้น ดังตารางที่ 4.7

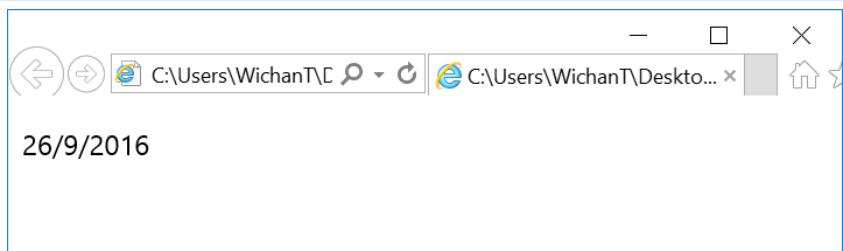
ตารางที่ 4.7 คำสั่งสำหรับดึงค่าวันเวลา

คำสั่ง	ความหมาย
<code>getDate()</code>	ดึงค่าวัน แสดงผลเป็นตัวเลข (1-31)
<code>getDay()</code>	ดึงค่าวันในสัปดาห์ แสดงผลเป็นตัวเลข (0-6)
<code>getFullYear()</code>	ดึงค่าปี ค.ศ. แสดงผลเป็นตัวเลข 4 ตัว (yyyy)
<code>getHours()</code>	ดึงค่าชั่วโมง แสดงผลเป็นตัวเลข (0-23)
<code>getMilliseconds()</code>	ดึงค่ามิลลิวินาที (0-999)
<code>getMinutes()</code>	ดึงค่านาที แสดงผลเป็นตัวเลข (0-59)
<code>getMonth()</code>	ดึงค่าเดือน แสดงผลเป็นตัวเลข (0-11)
<code>getSeconds()</code>	ดึงค่าวินาที แสดงผลเป็นตัวเลข (0-59)
<code>getTime()</code>	ดึงค่าเวลา (หน่วยมิลลิวินาที เริ่มนับจาก January 1, 1970)

ตัวอย่างดังภาพที่ 4.8

```
<body>
<p id="demo"></p>
<script>
var d = new Date();

document.getElementById("demo").innerHTML = d.getDate() + '/' + d.getMonth() + '/' +
d.getFullYear() ;
</script>
</body>
```



ภาพที่ 4.8 การดึงค่าวันที่

การกำหนดและตรวจสอบเงื่อนไข

การกำหนดเงื่อนไขผู้พัฒนาสามารถเลือกกำหนดให้โปรแกรมตรวจสอบ เปรียบเทียบ ค่าตัวแปร ตัวเลข หรือ ตัวอักษรได้ ตามเงื่อนไขที่ต้องการในการออกแบบแบบและพัฒนาโปรแกรม ซึ่งการกำหนดเงื่อนไขสามารถใช้ร่วมกับตัวดำเนินการในการเปรียบเทียบค่า เพื่อทำการเปรียบเทียบและตรวจสอบเงื่อนไขโดยใช้คำสั่งดังนี้

1. **If** เป็นการตรวจสอบเปรียบเทียบค่าแบบหนึ่งเงื่อนไข ซึ่งผลที่ได้จะต้องมีค่าเป็นจริง (True) สามารถเขียนคำสั่งโดยใช้รูปแบบการเขียน (If statement) ดังนี้

```
if (condition) { code to be executed if the condition is true }
```

ยกตัวอย่างเช่น

```
if (hour<18) { txt = 'สวัสดีตอนกลางวัน'; }
```

2. **Else** เป็นการตรวจสอบเงื่อนไขที่เป็นเท็จ (False) จากเงื่อนไข if สามารถเขียนคำสั่งโดยใช้รูปแบบการเขียน (Else statement) ดังนี้

```
if (condition) {
    code to be excecuted if the condition is true
} else {
    code to be excecuted if the condition is false
}
```

ยกตัวอย่างเช่น การแสดงผลข้อความที่ตรวจสอบจากตัวเลขชั่วโมง

```
if (hour<18) {
    txt = 'สวัสดีตอนกลางวัน';
} else {
    txt = 'สวัสดีตอนกลางคืน';
}
```

3. **Else if** เป็นการกำหนดเงื่อนไขใหม่ต่อจากเงื่อนไขแรกที่เป็นเท็จ ซึ่งสามารถระบุเงื่อนไขได้หลายเงื่อนไข โดยมีรูปแบบการเขียนคำสั่ง ดังนี้

```
if (condition1) { code to be excecuted if the condition1 is true }
else if (condition2) { code to be excecuted if the condition2 is true }
else { code to be excecuted if the condition2 is false }
```

ยกตัวอย่างเช่น การแสดงผลข้อความโดยตรวจสอบจากตัวเลขชั่วโมง

```
if (hour<10) {
    txt = 'สวัสดีตอนเช้า';
} else if (hour<13) {
    txt = 'สวัสดีตอนเที่ยง';
} else if (hour<18) {
    txt = 'สวัสดีตอนบ่าย';
} else {
    txt = 'สวัสดีตอนกลางคืน';
}
```

4. **Switch** เป็นการเลือกเงื่อนไขที่ถูกกำหนดค่าไว้ชัดเจนแล้ว รูปแบบการเขียนคำสั่ง (Switch statement) ดังนี้

```
switch (expression) {
    case n :
        code block
        break;
    case n :
        code block
        break;
    default :
        default code block
}
```

ยกตัวอย่างเช่น การตรวจสอบการแสดงผลข้อความโดยตรวจสอบจากวันของสัปดาห์

```
switch (myday) {
    case 6 :
        txt = 'สวัสดีวันหยุด';
        break ;
    case 0 :
        txt = 'สวัสดีวันหยุด';
        break;
    default :
        txt = 'สวัสดีวันทำงาน';
}
```

การวนรอบตามเงื่อนไข

การวนรอบการแสดงผลตามเงื่อนไข เป็นการแสดงผลข้อความหรือตัวแปรในรูปแบบเดิมหลายครั้ง หลายรอบ แต่ที่เปลี่ยนไปคือค่าของตัวแปรในแต่ละรอบ ซึ่งคำสั่งสำหรับการวนรอบตามเงื่อนไข เพื่อแสดงผลดังนี้

1. **For** เป็นการวนรอบการแสดงผล หรือ ประมวลผลคำสั่ง ตามจำนวนหรือครั้งที่ต้องการซึ่งจำนวนรอบผู้พัฒนาสามารถออกแบบได้โดยการกำหนดเงื่อนไขตามรูปแบบการเขียนดังนี้

```
for (statement1, statement2, statement3) {
    code block to be executed
}
```

โดย statement1 เป็นการกำหนดค่าตัวแปรเริ่มต้นของรอบ statement2 เป็นการระบุเงื่อนไขสำหรับการวนรอบโดยจะวนรอบไปเรื่อย ๆ จนกว่าจะเจอเงื่อนไขที่เป็นเท็จ และ statement3 เป็นผลที่เกิดขึ้นหลังจากที่วนรอบเสร็จแล้วรอบ ยกตัวอย่างเช่น วนรอบแสดงผลรอบที่ 0-5

```
for (i = 0; i < 5; i++) {
    txt += "วนรอบครั้งที่ " + i + "<br>";
}
```

2. **While** เป็นการวนรอบการแสดงผล หรือ ประมวลผลคำสั่งเหมือนกันกับ for ซึ่งเป็นการกำหนดค่าการวนรอบการแสดงผลจนกว่าจะเจอเงื่อนไขที่เป็นเท็จ ถึงจะออกจาก การวนรอบ มีรูปแบบการเขียนคำสั่งดังนี้ (w3schools, 2016)

```
while (condition) {
    code block to be executed
}
```

ยกตัวอย่างเช่น การวนรอบการแสดงผลตัวเลขตั้งแต่ 0-10

```
while (i < 10) {
    text += "The number is " + i;
    i++;
}
```

3. **Do while** เป็นการวนรอบเหมือนกันกับ while แต่จะแตกต่างตรงที่ do while จะกระทำตามคำสั่งก่อนเข้าสู่การตรวจสอบเงื่อนไขของการวนรอบ ซึ่งจะวนรอบจนกว่าเงื่อนไขจะเป็นเท็จ ซึ่งมีรูปแบบการเขียนคำสั่งและตัวอย่างการวนรอบการแสดงผลตัวเลขตั้งแต่ 0-10 ดังนี้

```
do {
    code block to be executed
}
while (condition);
```

```
do {
    text += "The number is " + i;
    i++;
}
while (i < 10);
```

สรุป

เทคโนโลยีสารสนเทศสามารถประยุกต์ใช้งานได้หลากหลาย ดังตัวอย่างงานวิจัยและโครงการ ดังต่อไปนี้

(ปฏิคุม ทองจริง และวิชาญ ทุมทอง, 2553) ได้ดำเนินการวิจัยโดยมีวัตถุประสงค์เพื่อพัฒนาระบบสนับสนุนการจัดการฐานข้อมูลของระบบบริหารจัดการการเรียนรู้และเนื้อหาบนเว็บ กรณีศึกษา มหาวิทยาลัยราชภัฏรำไพพรรณี โดยผ่านการประเมินจากผู้เชี่ยวชาญด้านคอมพิวเตอร์ งานวิจัยนี้ใช้ Data Flow Diagram (DFD) ในการวิเคราะห์ระบบการทำงาน และใช้ Entity Relationship Diagram (ER Diagram) ในการออกแบบโครงสร้าง (Schema) เพื่อความสะดวกในการค้นหา เปลี่ยนแปลงแก้ไข การแสดงผลในรูปแบบต่าง ๆ ที่มีประสิทธิภาพ และมีความสะดวกรวดเร็วในการจัดการการเรียนการสอนบนเว็บที่ดีขึ้นกว่าระบบเดิม ทำให้สามารถใช้งานทรัพยากร่มีอยู่ใหม่ประสิทธิภาพมากขึ้น

วัตถุประสงค์ของระบบที่พัฒนาขึ้นเพื่อช่วยให้การบริหารและการจัดการทำได้ดียิ่งขึ้น ทำให้สามารถใช้งานคอมพิวเตอร์ได้อย่างมีประสิทธิภาพและง่ายมากขึ้นกว่าระบบเดิม เพราะใช้ระบบคอมพิวเตอร์และเทคโนโลยีใหม่ ระบบพัฒนาโดยใช้ภาษาพีเอชพี (PHP) ภาษาเอ็กซ์เอชทีเอ็มแอล (XHTML) และจา逼สคริปต์ (JavaScript) โดยใช้เทคนิคการพัฒนาเว็บแบบ AJAX และใช้ระบบการจัดการฐานข้อมูล MySQL บนระบบปฏิบัติการ Microsoft Windows XP จากการทดสอบประสิทธิภาพของระบบใช้วิธีการทดสอบแบบล็อกบ็อกซ์ (Black-box testing) โดยแบ่งผู้ใช้งานออกเป็น 2 กลุ่มได้แก่ ผู้เชี่ยวชาญ และผู้ใช้ทั่วไป พบร่วมจากการประเมินประสิทธิภาพของระบบโดยผู้เชี่ยวชาญได้ค่าเฉลี่ยเท่ากับ 4.39 และส่วนเบี่ยงเบนมาตรฐาน 0.32 และผู้ใช้งานทั่วไปได้ค่าเฉลี่ยเท่ากับ 4.38 และค่าเบี่ยงเบนมาตรฐานเท่ากับ 0.37 ดังนั้นระบบที่พัฒนาขึ้นมีประสิทธิภาพโดยรวมอยู่ในระดับดี และสามารถนำไปประยุกต์ใช้งานได้อย่างมีประสิทธิภาพ

เมื่อปี พ.ศ. 2555 ผู้สอนได้ประยุกต์ใช้งานเทคโนโลยีสารสนเทศในการนำเสนอสื่อมัลติมีเดีย บนระบบเครือข่ายอินเทอร์เน็ต โดยการประยุกต์ใช้ในการนำเสนอข้อมูลห้องเรียนหลายห้อง ทั้งห้องเรียน แก้ว โดยได้นำเสนอแก่ลօรีภาพของห้องเรียนแก้ว วีดีทัศน์เฉลิมพระเกียรติพระนางเจ้ารำไพพรรณี พระราชกรณียกิจสมเด็จพระนางเจ้ารำไพพรรณี รวมไปถึงพระราชประวัติ โดยได้นำเสนอในรูปแบบของภาพ เสียงดนตรีประกอบ และวีดีทัศน์นำเสนอ เพื่อให้เกิดความน่าสนใจ ซึ่งสามารถเข้าเรียนกใช้งานเพื่อดูตัวอย่างได้จากเว็บไซต์ <https://www.rbru.ac.th/sbk/> ตัวอย่างหน้าจอการแสดงผลหน้าหลักของระบบห้องเรียนหลายห้อง ทั้งห้องเรียนแก้ว ดังภาพที่ 4.9



ภาพที่ 4.9 หอจดหมายเหตุวังสวนบ้านแก้ว

จะเห็นได้ว่าภาษาคริปต์สามารถช่วยให้ผู้พัฒนาสามารถสร้างเว็บไซต์ หรือ พัฒนาซอฟต์แวร์ เพื่อให้แสดงผลข้อมูลตามความต้องการ โดยไม่จำเป็นต้องเปิดหน้าใหม่ เพื่อร้องขอข้อมูลจากเครื่องแม่ข่ายใหม่ ซึ่งจะช่วยให้การประมวลผลรวดเร็วยิ่งขึ้น ภาษาคริปต์เป็นภาษาสคริปต์ที่อยู่ในฝั่งของผู้ร้องขอ บริการ จึงทำให้สามารถประมวลผล และแสดงผลได้ทันที ช่วยลดการทำงานของเครื่องแม่ข่าย

ภาษาคริปต์มีความสามารถที่หลากหลาย เช่น สามารถปรับเปลี่ยนรูปแบบการแสดงผลของเอกสารเอชทีเอ็มแอล เปลี่ยนอิเล็กทรอนิกส์ในหน้าเว็บเพจ เปลี่ยนรูปแบบค่าแอฟฟิลิวท์ของแท็กในเอกสารเอชทีเอ็มแอลได้ รวมไปถึงสามารถกำหนดรูปแบบการแสดงผลของโดยใช้ภาษาซีเอสเอสได้อย่างมีประสิทธิภาพ ซึ่งทำให้สามารถออกแบบและพัฒนาโปรแกรมเล็ก ๆ เพื่อใช้งานบนหน้าเว็บเพจได้ เช่น โปรแกรมหาคำนวนค่าอัตราดอกเบี้ย เงินกู้ธนาคาร การกู้ซื้อรถยนต์ การอนุมัติบัตรเครดิตเป็นต้น เป็นต้น

แบบฝึกหัดบทที่ 4

จงตอบคำถามต่อไปนี้

1. จาวасคริปต์ คืออะไร
2. จาวасคริปต์ มีข้อดีหรือมีประโยชน์อย่างไร
3. การใช้งานจาวасคริปต์มีรูปแบบอะไรบ้าง
4. คำสั่งจาวасคริปต์สำหรับแสดงผลมีรูปแบบอะไรบ้าง
5. รูปแบบการเขียนคำสั่งจาวасคริปต์มีส่วนประกอบอะไรบ้าง
6. ข้อกำหนดในการตั้งชื่อตัวแปรมีอะไรบ้าง
7. ชนิดของข้อมูลในภาษาจาวасคริปต์มีประเภทอะไรบ้าง จงยกตัวอย่าง
8. ตัวดำเนินการในภาษาจาวасคริปต์มีอะไรบ้าง จงยกตัวอย่าง
9. การตรวจจับเหตุการณ์ในเอกสารเช่นที่เอ็มแอลมีอะไรบ้าง
10. คำสั่งสำหรับการวนรอบและคำสั่งสำหรับการตรวจสอบเงื่อนไขมีอะไรบ้าง

เอกสารอ้างอิง

- ปฏิคุม ทองจริง และวิชาญ ทุมทอง. (2553). ระบบบริหารจัดการการเรียนรู้และเนื้อหาบนเว็บ
กรณีศึกษา มหาวิทยาลัยราชภัฏรำไพพรรณี. งานวิชาการรำไพพรรณี มหาวิทยาลัยราช
ภัฏรำไพพรรณี.
- hellomyweb. (2553). ความรู้พื้นฐานเกี่ยวกับ JavaScript. (ออนไลน์). แหล่งที่มา :
<http://www.hellomyweb.com/index.php/main/content/131>. 4 มิถุนายน
2559.
- w3schools. (2016). **JavaScript While Loop**. (online). Available :
http://www.w3schools.com/js/js_loop_while.asp. 26 December 2016.
- _____ . (2016). **JavaScript Date**. (online). Available :
http://www.w3schools.com/js/js_dates.asp. 26 October 2016.
- _____ . (2016). **JavaScript Operators**. (online). Available :
http://www.w3schools.com/js/js_operators.asp. 6 December 2016.
- _____ . (2016). **JavaScript Output**. (online). Available :
<http://www.w3schools.com/js/default.asp>. 4 June 2016
- _____ . (2016). **JavaScript Statements**. (online). Available :
http://www.w3schools.com/js/js_statements.asp. 6 December 2016.
- _____ . (2016). **JavaScript Syntax**. (online). Available :
http://www.w3schools.com/js/js_syntax.asp. 5 June 2016.

