

Week 04-05

Intro to Java Script

6-13 Sep 2025

ผศ.วิญญู นรินทล้าพงศ์ (winyu.nir@dpu.ac.th)

2

การใช้ JavaScript #1

หัวข้อการบรรยาย “การใช้ JavaScript #1”

หน้า 3

- แนวคิดเกี่ยวกับการเขียนโปรแกรมแบบเชิงวัตถุ (Object Oriented Programming)
- ความรู้เบื้องต้นเกี่ยวกับ JavaScript
- พื้นฐานการเขียน JavaScript
- การใช้ Object ใน JavaScript
- การใช้ Include File
- หลักการเขียนโปรแกรม JavaScript

แนวคิดเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ (OOP)

หน้า 4

- หลักการเขียนโปรแกรมเชิงวัตถุ นั้น จะเป็นการมองเห็นและเปรียบเทียบสิ่งต่างๆ ให้เป็นวัตถุ หรือคือการมองเป็น Object นั้นเอง
- เป็นการจำลองมาจากชีวิตจริงของมนุษย์ เช่น Object ในชีวิตจริง อาจจะมีหมายถึง รถยนต์ , บ้าน , คน , สุนัข เป็นต้น และ Object แต่ละตัวก็จะมีสิ่งที่เรียกว่า
 - ▣ Property กับ
 - ▣ Method

แนวคิดเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ (OOP)

หน้า 5

- **Property** ก็คือคุณสมบัติของ Object ตัวนั้นๆ ซึ่ง Object ประเภทเดียวกันก็จะมี Property แบบเดียวกัน แต่จะแตกต่างกันไปในแต่ละ Object เช่น
 - ▣ รถยนต์ จะต้องมียี่ห้อ , ตัวถัง , กระจก แต่ว่ารถแต่ละคันอาจจะมีสีตัวถัง ที่แตกต่างกันได้ เช่น คันหนึ่งสีแดง อีกคันอาจจะสีเขียวก็ได้ รถทั้งสองคันไม่จำเป็นต้องมีคุณสมบัติเหมือนกัน เนื่องจากว่าเป็นรถคนละคัน (คนละ Object นั่นเอง)

แนวคิดเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ (OOP)

- **Method** คือ วิธีการที่เราสามารถจะจัดการได้กับ Object นั้นๆ เช่น ถ้าเราทักทาย นาย ก. หรือ เราทักทาย นาย ข. เป็นต้น หรือรถของเราสีแดงก็สามารถที่จะจัดการเปลี่ยนสีรถเป็นสีเขียวได้
 - ▣ Method (วิธีการ) อาจจะซ้ำกันหรือเหมือนกันได้สำหรับ Object คนละชนิด แต่การตอบสนอง Object นั้นจะไม่เหมือนกัน เช่น เราอาจจะสวัสดีนาย ก. แล้วนาย ก. ตอบรับ แต่ขณะเดียวกันเราสวัสดีนาย ข. นาย ข. อาจจะทำเป็นเฉยๆ เป็นต้น

แนวคิดเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ (OOP)

หน้า 7

- การเขียนโปรแกรมในเชิงวัตถุนั้น เราจะออกแบบส่วนประกอบโปรแกรมของเราให้เป็น “วัตถุ” (Object) โดยที่ object แต่ละตัว จะมีทั้งข้อมูล (ตัวแปร) และ การทำงาน (ฟังก์ชัน) ที่จำเป็นทั้งหมดรวมอยู่ในตัววัตถุนั้นๆ
- ซึ่งถ้าจะมองง่ายๆ “object” ก็คือ structure หรือ record แบบหนึ่ง ที่มีทั้ง variables และ functions (Property และ Method) รวมอยู่ใน structure เดียวกัน

แนวคิดเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ (OOP)

หน้า 8

□ คุณสมบัติเพิ่มเติมของ Object Oriented Programming

- ▣ **Encapsulation** คือการ รวมตัวแปร และ ฟังก์ชัน เข้าไว้ใน object ตัวเดียว และ สามารถปกป้อง ไม่ให้ คนนอก (หมายถึง object อื่นๆ) เข้าถึงได้
- ▣ **Inheritance** หมายถึง สามารถ “สืบทอด” ได้ คือถ้าเราต้องการ object ตัวใหม่ที่ทำางานคล้ายๆ หรือ เก่งกว่า object ตัวเดิม เราไม่จำเป็นต้องแก้ไข source code ของ object แต่ใช้วิธี “ถ่ายทอดกรรมพันธุ์” ซึ่ง object ที่สืบทอดมา จะมีทุกอย่างเหมือนบรรพบุรุษ และสามารถนำมาเขียนโปรแกรมเพิ่ม เฉพาะในส่วนที่ต้องการได้
- ▣ **Polymorphism** หมายถึง การสามารถเขียนโปรแกรมให้ฟังก์ชัน ใดๆ ของ object ที่เราสืบทอดมา ทำงานต่างไปจาก object บรรพบุรุษได้

แนวคิดเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ (OOP)

หน้า 9

□ คุณสมบัติเพิ่มเติมของ Object Oriented Programming

- ▣ **Encapsulation** คือการ รวมตัวแปร และ ฟังก์ชัน เข้าไว้ใน object ตัวเดียว และ สามารถปกป้อง ไม่ให้ คนนอก (หมายถึง object อื่นๆ) เข้าถึงได้
- ▣ **Inheritance** หมายถึง สามารถ “สืบทอด” ได้ คือถ้าเราต้องการ object ตัวใหม่ที่ทำงานคล้ายๆ หรือ เก่งกว่า object ตัวเดิม เราไม่จำเป็นต้องแก้ไข source code ของ object แต่ใช้วิธี “ถ่ายทอดกรรมพันธุ์” ซึ่ง object ที่สืบทอดมา จะมีทุกอย่างเหมือนบรรพบุรุษ และสามารถนำมาเขียนโปรแกรมเพิ่ม เฉพาะในส่วนที่ต้องการได้
- ▣ **Polymorphism** หมายถึง การสามารถเขียนโปรแกรมให้ฟังก์ชัน ใดๆ ของ object ที่เราสืบทอดมา ทำงานต่างไปจาก object บรรพบุรุษได้

JavaScript คืออะไร

หน้า 10

- เป็นภาษาสำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต
- เหมาะสำหรับการพัฒนาโปรแกรมใช้งานในระบบอินเทอร์เน็ตทั้งทางฝั่งไคลเอนต์ และฝั่งเซิร์ฟเวอร์
- สามารถเขียนโปรแกรม JavaScript เพิ่มเข้าไปในเว็บเพจ เพื่อประโยชน์สำหรับงานด้านต่างๆ ทั้งการคำนวณ การแสดงผล การรับ-ส่งข้อมูล
- สามารถตอบโต้กับผู้ใช้ได้อย่างทันทีทันใด
- เว็บเพจที่บรรจุ JavaScript จะมีความน่าสนใจและสมบูรณ์มากกว่าเว็บธรรมดาทั่วไป

JavaScript ใช้ทำอะไรได้บ้าง

หน้า 11

- เพิ่มส่วนของเนื้อหาในเว็บเพจอย่างอัตโนมัติ
- รองรับการเปลี่ยนแปลงของเว็บในการตอบสนองของผู้ใช้
- ช่วยสร้างความน่าสนใจให้เว็บเพจ
- ช่วยในการตรวจสอบข้อมูล (Data Validation) ก่อนส่งไปประมวลผลที่ Server
- ช่วยในการจัดการ frames

ประโยชน์ของการใช้ JavaScript

หน้า 12

- สร้างเมนูที่ตอบสนองต่อการนำเมาส์ไปชี้
- สร้างเครื่องคิดเลข เกมส์ ปฏิทิน
- เปิด ปิด ย้ายตำแหน่ง Windows
- ทำอักษรวิ่ง
- ตรวจสอบความถูกต้องของแบบฟอร์มที่ผู้ใช้กรอก

พื้นฐานการเขียน JavaScript

หน้า 13

- วิธีการเขียน JavaScript มี 2 วิธี คือ
 - ▣ เขียนด้วยชุดคำสั่ง และฟังก์ชันของ JavaScript เอง
 - ▣ เขียนตามเหตุการณ์ที่เกิดขึ้นตามการใช้งานจากชุดคำสั่งของ HTML
- หลักการเขียน JavaScript ใน HTML
 - ▣ เขียนคำสั่งหรือกำหนดฟังก์ชันไว้ภายในส่วนหัวของเว็บเพจ
 - ▣ เรียกใช้ฟังก์ชันจากสคริปต์อื่นๆ ที่เขียนขึ้นมา
 - ▣ เรียกใช้ฟังก์ชันโดยค่าแอตทริบิวต์ (Attribute) ของคำสั่ง HTML

พื้นฐานการเขียน JavaScript

หน้า 14

- TAG ที่ใช้สำหรับเขียน JAVA Script มาทำงานในเว็บเพจ คือ

```
<SCRIPT LANGUAGE="JavaScript">  
.....  
</SCRIPT>
```

- โดยจะแทรกที่ส่วนใดก็ได้ใน HTML Document
- ภายใน TAG จะมีการใส่คำสั่งต่างๆ ของ JAVA Script

พื้นฐานการเขียน JavaScript

หน้า 15

- JavaScript สามารถใส่ไว้ที่ส่วนของ `<head>` หรือ `<body>` ก็ได้
 - ▣ JavaScript *functions* ควรวางที่ `<head>` เพื่อให้มั่นใจว่าจะมีการโหลดคำสั่งดังกล่าวพร้อมใช้งาน
 - ▣ JavaScript ในส่วน `<body>` จะมีการโหลดเมื่อเรียกหน้าเพจนั้นขึ้นมาแสดง
- นอกจากนี้ยังสามารถเรียกใช้ฟังก์ชัน JavaScript ภายในแท็กต่างๆ เช่น `<a ...>` หรือ `` ได้

พื้นฐานการเขียน JavaScript

หน้า 16

- JavaScript สามารถแยกเก็บเป็นไฟล์ **.js** แล้วเรียกใช้

```
<script LANGUAGE="Javascript" src="myJavaScriptFile.js">  
</script>
```

- วาง HTML ด้านบนนี้ไว้ในตำแหน่งที่ต้องการเลย
 - และสามารถวางได้มากกว่า 1 ครั้ง (ในไฟล์ html)
 - หมายเหตุ ในส่วนของไฟล์ **.js** จะต้องไม่มี **<script>** tag ในไฟล์นั้น
- JavaScript สามารถวางใน HTML ในรูปแบบ **Form Object** เช่น ปุ่ม (button) โดยจะมีการแปลคำสั่งส่วนนั้นเมื่อมีการเรียกใช้ Object นั้นๆ

พื้นฐานการเขียน JavaScript

หน้า 17

- Script ควรสิ้นสุดโดยใช้การขึ้นบรรทัดใหม่
- แต่อาจเขียนคำสั่งหลายคำสั่งในบรรทัดเดียวกันก็ได้โดยใช้เครื่องหมาย semi-colons ขึ้นกลาง
- ทางที่ดีที่สุด คือ ควรใช้ semi-colons ขึ้นทุกคำสั่งถึงแม้ว่าจะขึ้นบรรทัดใหม่แล้วก็ตาม
- Start with:
`<SCRIPT LANGUAGE="Javascript">`
- End with:
`</SCRIPT>`

การอ้างอิง Object ใน JavaScript

หน้า 18

- โดยปกติเมื่อมีการใช้งานการแสดงผลต่างๆ บนเว็บเพจจะต้องมีการอ้างอิงตัว Object ด้วย ซึ่งโดยพื้นฐานการอ้างอิง Object ใน JavaScript คือ
 - ▣ Object ตัวที่อยู่ระดับบนสุดคือ **window**
 - ▣ ในอ็อบเจ็ค window สามารถอ้างอิง object ย่อยๆ ลงไปได้อีก เช่น **document , history , location**
 - ▣ และสามารถอ้างอิง Object ต่างๆ ย่อยต่อลงไปได้เรื่อยๆ

การอ้างอิง Object ใน JavaScript

หน้า 19

- โดยปกติเมื่อเราใช้งานเว็บเพจ ก็จะมีการเปิด window ขึ้นมาใช้งานอยู่แล้ว เป็นวิธีการธรรมดาที่สุดที่สร้าง window object
- Object ต่างๆ สามารถที่จะเลือกอ้างอิงชื่อของ Object ที่ใช้งานบนเว็บเพจได้ ตัวอย่างเช่น

`window.document.form1.textbox1.value`

- ซึ่งโดยปกติถ้าเราทำงานใน window อยู่แล้ว Object window จึงไม่จำเป็นต้องอ้างอิงก็ได้ สามารถเรียก Object ย่อยลงไปแทนได้เลย

การใช้ Include file ใน JavaScript

หน้า 20

- เราสามารถใช้ Include file ได้ โดย file ที่ใช้จะมีนามสกุล .js และภายใน file นี้จะประกอบไปด้วย code ของ JAVA Script แบบต่างๆ
- การเรียกใช้งานจะเพิ่ม Attribute SRC=“ชื่อ file ” ใน TAG Script เช่น
`<SCRIPT LANGUAGE=“JAVAScript” SRC=“includefile.js”>`

การใส่ Comment ใน JavaScript

หน้า 21

- ถ้าต้องการใส่ Comment เพื่อเอาไว้เตือนความจำ หรืออธิบาย Code ต่างๆ ก็สามารถใช้ เครื่องหมาย `//` ได้ ซึ่งแสดงว่าข้อความที่อยู่ข้างหลัง `//` จะเป็น Comment จนจบบรรทัด เช่น

```
<SCRIPT LANGUAGE="JavaScript">  
    Document.write("Hello World"); //Comment Here  
</SCRIPT>
```

```
<SCRIPT>  
    Document.write("Hello World"); //Comment Here  
</SCRIPT>
```

การใส่ Comment ใน JavaScript

- ถ้าต้องการใส่ Comment แบบกลุ่ม ก็สามารถใช้ เครื่องหมาย `/*` ตอนเปิด และ `*/` ตอนปิดส่วนท้ายของ Comment ได้ เช่น

```
<SCRIPT LANGUAGE="JavaScript">  
    /*      Comment 1  
           Comment 2  
           Comment 3      */  
</SCRIPT>
```

การใช้ JavaScript ร่วมกับ TAG HTML

หน้า 23

- ในการแสดงผลเราสามารถใส่ TAG HTML ร่วมเข้าไปได้ โดยใส่ลงไปในส่วนข้อความที่ต้องการจะแสดง เช่น

```
<SCRIPT LANGUAGE="JavaScript">  
    document.write("<H1>Hello World</H1>")  
</SCRIPT>
```

หลักการเขียนโปรแกรม JavaScript

หน้า 24

การกำหนดตัวแปร

- ก่อนจะนำตัวแปรไปใช้งาน จะต้องมีการกำหนดตัวแปรก่อน โดยใช้ คำสั่ง VAR เช่น

```
VAR name , address ;  
VAR country = "Thailand" ;  
VAR answer = 20;
```

- ชื่อตัวแปร ตัวเล็ก – ตัวใหญ่ มีผลแตกต่างกัน***
เช่น คำว่า **Address** จะไม่เท่ากับ **address**

หลักการเขียนโปรแกรม JavaScript

หน้า 25

นิพจน์ (Expression)

- เป็นชุดของตัวแปร ค่าคงที่ และ โอเปอเรเตอร์ เครื่องหมายคำนวณทางคณิตศาสตร์ หรือ ตัวนิพจน์เอง เชื่อมต่อกันด้วย โอเปอเรเตอร์ มีผลลัพธ์ให้เกิดข้อมูลใหม่ๆ ขึ้นมา

หลักการเขียนโปรแกรม JavaScript

หน้า 26

นิพจน์ (Expression)

- นิพจน์ใน JAVA Script มี 3 ชนิด คือ
 - ▣ 1. นิพจน์ทางคณิตศาสตร์ เป็นการนำเครื่องหมายทางคณิตศาสตร์มาเชื่อมโยงค่าตัวแปร ค่าคงที่ ผลลัพธ์ที่ได้เป็นตัวเลข
 - ▣ 2. นิพจน์ตรรกะ (logic) จะทำงานเกี่ยวกับทางด้านตรรกศาสตร์ที่ใช้เครื่องหมายเปรียบเทียบ เช่น = , < , > ผลลัพธ์ที่ได้มีค่าเป็นจริง หรือเท็จ
 - ▣ 3. นิพจน์ข้อความ (string) จะทำงานเกี่ยวกับข้อความ ในลักษณะต่างๆ เช่น การรวมคำ แสดงผลข้อความ

หลักการเขียนโปรแกรม JavaScript

หน้า 27

- โอเปอเรเตอร์ใน JAVA Script เป็นเครื่องหมายกำหนดการทำงานทางคณิตศาสตร์ การเปรียบเทียบ ระหว่างข้อมูล 2 ตัว ซึ่งเรียกว่า โอเปอเรแรนด์ (Operand) โดยอาจจะมีค่าเป็นตัวเลข ข้อความ ค่าคงที่ หรือตัวแปรต่างๆ

หลักการเขียนโปรแกรม JavaScript

หน้า 28

□ โอเปอเรเตอร์ทางคณิตศาสตร์

| โอเปอเรเตอร์ | เครื่องหมาย |
|--------------|-------------|
| การบวก | + |
| การลบ | - |
| การคูณ | * |
| การหาร | / |
| โมดูลัส | % |

หลักการเขียนโปรแกรม JavaScript

หน้า 29

- โอเปอเรเตอร์ทางการเปรียบเทียบ ให้ผลลัพธ์เป็นจริงหรือเท็จ

| โอเปอเรเตอร์ | เครื่องหมาย |
|-----------------|-------------|
| เท่ากับ | = |
| ไม่เท่ากับ | != |
| มากกว่า | > |
| มากกว่าเท่ากับ | >= |
| น้อยกว่า | < |
| น้อยกว่าเท่ากับ | <= |

หลักการเขียนโปรแกรม JavaScript

หน้า 30

- โอเปอเรเตอร์ทางตรรกะ ให้ผลลัพธ์เป็นจริงหรือเท็จ

| โอเปอเรเตอร์ | เครื่องหมาย |
|--------------|-------------|
| And | && |
| Or | |
| Not | ! |

หลักการเขียนโปรแกรม JavaScript

หน้า 31

- โอเปอเรเตอร์เชิงข้อความ ใช้สำหรับการเชื่อมต่อตัวแปร หรือข้อมูลชนิดข้อความ เป็นข้อความเดียวกัน ได้แก่เครื่องหมายบวก (+) เช่น

```
Mystring = "I like" + " Bangkok"
```

32

การใช้ JavaScript #2

หัวข้อการบรรยาย “การใช้ JavaScript #2”

33

- โครงสร้างควบคุมการทำงานของภาษา Java Script
- ฟังก์ชัน (Function) ที่ใช้ใน Java Script
- การผ่านค่า Argument
- การคืนค่า function
- Event Handler

โครงสร้างควบคุมการทำงานของภาษา Java Script

34

1. โครงสร้างแบบ if

หมายถึง เมื่อทำการทดสอบเงื่อนไขของนิพจน์ เช่น การเปรียบเทียบ แล้วได้ผลลัพธ์เป็นจริง (True) ก็จะทำงานตามคำสั่งที่อยู่ถัดไป

โครงสร้าง

```
if (expression)
    { statements }
```

ตัวอย่าง

```
var a = 5;
if (a==5)
{ document.write(a); }
```

โครงสร้างควบคุมการทำงานของภาษา Java Script

35

2. โครงสร้างแบบ if...else

หมายถึง เมื่อทำการทดสอบเงื่อนไขของนิพจน์ แล้วได้ผลลัพธ์เป็นจริง (True) ก็จะทำงานตามคำสั่งที่อยู่ถัดจาก if แต่ถ้าหากได้ผลลัพธ์เป็นเท็จ (False) จะทำคำสั่งที่อยู่ถัดจาก else

โครงสร้าง

```
if (expression)
    { statements }
else
    { statements }
```

โครงสร้างควบคุมการทำงานของภาษา Java Script

36

ตัวอย่าง

```
a = 3;  
b = 2;  
if (a==b)  
{ document.write("It is true"); }  
else  
{ document.write("It is false"); }
```

โครงสร้างควบคุมการทำงานของภาษา Java Script

37

3. โครงสร้างแบบ switch

หมายถึง โครงสร้างที่มีทางเลือกของนิพจน์ได้หลายๆ ทาง
โครงสร้าง

```
switch (expression)
{
    case label1 :
        statements;
        break;
    case label2 :
        statements;
        break;
    ...
    default : statements }
```

โครงสร้างควบคุมการทำงานของภาษา Java Script

38

ตัวอย่าง

```
a = 2;
b = 1;
switch (result = a * b) {
    case 1:
        document.write("1");
        break ;
    case 2 :
        document.write("2");
        break ;
    case 3 :
        document.write("3");
        break ;
}
```

โครงสร้างควบคุมการทำงานของภาษา Java Script

39

4. โครงสร้างแบบวนซ้ำ for

หมายถึง โครงสร้างที่ทำงานตามค่าเริ่มต้น เงื่อนไขที่กำหนดให้สิ้นสุด และมีการกำหนดตัวเปลี่ยนแปลงเงื่อนไขด้วย จะมีจำนวนทำงานตามรอบที่แน่นอน

โครงสร้าง

```
for ([initial condition]; [condition]; [update expression])  
{ statements; }
```

ตัวอย่าง

```
for (i=1 ; i < 10 ; i++)  
{ document.write(i+"<BR>"); }
```

โครงสร้างควบคุมการทำงานของภาษา Java Script

40

5. โครงสร้างแบบวนซ้ำ while

หมายถึง โครงสร้างที่ทำงานภายใต้เงื่อนไขที่เป็นจริง และจะทำงานซ้ำไปเรื่อยๆ จนกว่าเงื่อนไขนั้นจะเป็นเท็จ

โครงสร้าง

```
while (condition)
{ statements; }
```

ตัวอย่าง

```
a = 1
while (a < 5)
{ document.write(a+"<BR>");
  a = a + 1; }
```


ฟังก์ชัน (Function) ที่ใช้ใน Java Script

41

รูปแบบ Function ที่ใช้ใน Java Script มี 2 แบบ คือ

1. Standard Function (Built-in functions or method)
2. User-defined Function

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

ฟังก์ชัน (Function) ที่ใช้ใน Java Script

42

1. Standard Function (Built-in functions or method)

- เป็น Function มาตรฐานที่ Java Script กำหนดมาให้โดยสามารถนำไปใช้งานได้ทันที
- Function แบบนี้เป็นได้ทั้ง Method เช่น alert , write และ function สำเร็จรูป เช่น sqrt , parseInt

การเรียกใช้ Function เพียงแค่เรียกชื่อ function ที่ต้องการใช้เท่านั้น ก็สามารถทำงานได้ แต่จะต้อง

- พิมพ์ตัวอักษรเล็ก – ใหญ่ ให้เหมือนกับที่กำหนด
- จะต้องลงท้ายด้วย () ถ้าหาก function ไม่ส่งค่า argument ใดๆ

ฟังก์ชัน (Function) ที่ใช้ใน Java Script

43

ตัวอย่างการใช้ Standard Function

```
<SCRIPT LANGUAGE="JavaScript">
function calc(myform) {
    var result = parseInt(myform.number1.value) +
    parseInt(myform.number2.value);
    alert(result);
}
</SCRIPT>
<FORM>
    <INPUT TYPE="text" NAME="number1">
    <INPUT TYPE="text" NAME="number2">
    <INPUT TYPE="button" onClick="calc(this.form)">
</FORM>
```

ฟังก์ชัน (Function) ที่ใช้ใน Java Script

44

2. User-defined Function

- เป็น Function ที่ผู้ใช้เขียนโปรแกรมกำหนดขึ้นมาเอง

โครงสร้าง

```
function name ([param] , [param] , [...,param])  
{ statements }
```

โดยที่

name เป็น function ซึ่งต้องตั้งชื่อไม่ซ้ำกับ function ที่มีอยู่แล้ว

param เป็นพารามิเตอร์สำหรับ รับ Argument แล้วส่งต่อไปยัง function สามารถมีหรือไม่มีก็ได้ และถ้ามีหลายตัวก็ใส่เครื่องหมาย , คั่น

Statements เป็นคำสั่งต่างๆ ที่อยู่ภายใน function

ฟังก์ชัน (Function) ที่ใช้ใน Java Script

45

ตัวอย่างการใช้ User-Defined Function

```
<SCRIPT LANGUAGE="JavaScript">  
    function RepeatHello() {  
        document.write("Hello World<BR>");  
    }  
    RepeatHello();  
    RepeatHello();  
    RepeatHello();  
</SCRIPT>
```

เมธอด (Method)

46

Method เป็น function ที่ทำงานโดย object ซึ่งเราสามารถกำหนดค่าให้กับ Method ได้ โดยวิธีการเดียวกับ function โดยเขียนรูปได้ดังนี้

```
Object.methodName = function_name
```

ตัวอย่างเช่น

button object มี Method Click ซึ่งเป็นการกระทำจากการคลิก การกำหนดคำสั่งคลิกให้กับ button object ทำให้ปุ่มกดทำงานเหมือนมีผู้ใช้ใช้เมาส์คลิกจริงๆ เช่น `SumButton.Click`

การผ่านค่า Argument

47

- JavaScript สนับสนุนการส่งผ่านค่า Argument ให้กับ Function เพื่อนำไปประมวลผลภายใน function ได้ โดยเป็นลักษณะ Call by value คือ การเปลี่ยนแปลงค่า Argument ไม่มีผลต่อค่าพารามิเตอร์ที่ส่งออกไปจากผู้ใช้แต่อย่างใด
- การส่งผ่านค่า Argument ไปยัง function ถูกกำหนดไว้ด้วยพารามิเตอร์ของ function นั้น เช่น

```
function MyFunction(Word)
{ alert(Word); }
```

```
MyFunction("This is my text.");
```

การคืนค่า function

48

- เราสามารถส่งคืนค่าไปยังโปรแกรมได้ โดยการใช้ return ตามด้วยค่าที่ต้องการส่งกลับ โดยโปรแกรมจะต้องมีตัวแปรมารับค่านี้ด้วย
- ตัวอย่างเช่น

```
<SCRIPT LANGUAGE="JavaScript">  
var result = CutString("Hello World", 4);  
alert(result);  
  
function CutString(text , num)  
{ var textReturn = text.substring(text , num) ;  
  return (textReturn); }  
</SCRIPT>
```


Arrow function

49

□ Arrow function เป็นการเขียนฟังก์ชันของ JavaScript ให้อยู่ในรูปแบบสั้น

▣ With Arrow Function

```
hello = () => {  
    return "Hello  
World!";  
}
```

▣ Arrow Functions Return Value by Default:

```
hello = () => "Hello World!";
```

Arrow function

50

- ▣ Arrow Function With Parameters:

```
hello = (val) => "Hello " + val;
```

```
hello = val => "Hello " + val;
```

- ▣ Without Arrow Function

```
hello = function() {  
    return "Hello World!";  
}
```

Event Handler

51

- เป็นการตอบสนองต่อเหตุการณ์ (event) ที่เกิดขึ้นจากที่ผู้ใช้กระทำ เช่น การคลิกเมาส์บนปุ่ม จะมีตัวจัดการที่เรียกว่า Event Handler เป็นตัวจัดการเหตุการณ์เกิดขึ้น โดยจะไปเรียกใช้งาน Script ให้ทำงานประเภทต่างๆ ได้
- ลักษณะการเขียนโปรแกรมที่ตอบสนองต่อเหตุการณ์ อาจเรียกอีกวิธีได้ว่า “Event-Driven programming”
- สามารถเขียนได้จบในบรรทัดเดียวภายในแท็ก HTML เพื่อเรียกใช้ฟังก์ชันของ JavaScript
- มีรูปแบบการใช้งานดังนี้

```
<TAGNAME eventHandler="JavaScript Code">
```

Event Handler ที่ใช้ใน Java Script

52

| | |
|--------------------------|-------------------------|
| <code>onAbort</code> | <code>onBlur</code> |
| <code>onChange</code> | <code>onClick</code> |
| <code>onError</code> | <code>onFocus</code> |
| <code>onLoad</code> | <code>onMouseOut</code> |
| <code>onMouseOver</code> | <code>onReset</code> |
| <code>onSelect</code> | <code>onSubmit</code> |
| <code>onUnload</code> | |

Event Handler ที่ใช้งานกับ Object ประเภทต่างๆ

53

| Object | Event Handler |
|----------|--|
| Select | onBlur , onChange , onFocus |
| Text | onBlur , onChange , onFocus , onSelect |
| Textarea | onBlur , onChange , onFocus , onSelect |
| Button | onClick |
| Checkbox | onClick |
| Radio | onClick |
| Link | onClick , onMouseOver |
| Document | onLoad , onUnload |
| Window | onLoad , onUnload |
| Form | onSubmit |

เหตุการณ์ onClick

54

□ เป็นการตอบสนองเมื่อผู้ใช้คลิกเมาส์ลงบน Object เช่น ปุ่ม , หน้าเว็บเพจ เป็นต้น

□ ตัวอย่างเช่น

```
<SCRIPT LANGUAGE="JavaScript">
function sayHello() {
    alert("Hello " + myform.yourname.value);
}
</SCRIPT>
<FORM NAME="myform">
    <INPUT TYPE="text" NAME="yourname" >
    <INPUT TYPE="button" VALUE="HELLO"
        onClick="sayHello()">
</FORM>
```

เหตุการณ์ onAbort

55

□ เป็นการตอบสนองเมื่อผู้ใช้ขัดจังหวะการทำงานต่างๆ ของโปรแกรม เช่น กดปุ่ม Stop เพื่อยกเลิกการ load หน้าเว็บเป็นต้น

□ ตัวอย่างเช่น

```
<IMG SRC="http://www.winyou.net/winyou.jpg"  
onAbort="alert('You stopped loading the image') ">
```

เหตุการณ์ onError

56

□ Event Handler ตัวนี้จะทำงานเมื่อมีความผิดพลาดขึ้นในเว็บเพจหน้านั้น เช่น Browser ไม่สามารถจะ load หน้าเว็บขึ้นมาได้

□ ตัวอย่างเช่น

```
<IMG SRC="http://www.winyu.net/winyu.jpg"  
  onError="alert('Error Error Error')">
```


เหตุการณ์ onLoad

57

- จะทำงานทันทีที่มีการ Load window หรือ Frame สิ้นสุด โดยจะต้องเขียนคำสั่งภายใต้ Tag <BODY> หรือ <FRAMESET>

- ตัวอย่างเช่น

```
<SCRIPT LANGUAGE="JavaScript">
function greeting() {
    alert("Welcome to my webpage")
}
</SCRIPT>
<BODY onLoad="greeting()">
...
</BODY>
```

เหตุการณ์ onUnload

58

- จะทำงานตรงข้ามกับ onLoad เมื่อผู้ใช้สิ้นสุดการทำงานของเว็บเพจ หรือปิดเว็บเพจหน้านี้ โดยจะต้องเขียนคำสั่งภายใต้ Tag <BODY> หรือ <FRAMESET> เช่นกัน

- ตัวอย่างเช่น

```
<SCRIPT LANGUAGE="JavaScript">
function Bye() {
    alert("Thank you to visit our website ")
}
</SCRIPT>
<BODY onUnload="Bye()">
...
</BODY>
```

เหตุการณ์ onFocus

59

- จะทำงานเมื่อ Object ที่กำหนดได้ focus พร้อมทั้งจะทำงานได้ เช่น Textbox จะมี Cursor ปรากฏขึ้นให้กรอกข้อความได้
- ตัวอย่างเช่น

```
<BODY>
```

```
<INPUT TYPE="text" NAME="text1"  
  onFocus="alert('You are focus on text1') "><BR>
```

```
<INPUT TYPE="text" NAME="text2"  
  onFocus="alert('You are focus on text2') "><BR>
```

```
</BODY>
```

เหตุการณ์ onBlur

60

- จะทำงานเมื่อ Object อย่างเช่น text , textarea เสีย focus โดยจะนำมาใช้เมื่อต้องการยกเลิก focus ของ object ที่กำหนด

- ตัวอย่างเช่น

```
<SCRIPT LANGUAGE="JavaScript">
    function validForm(form) {
        var input = 0;
        input = document.myform.number1.value;
        if (input < 0 ) {
            alert("Please enter the number that is greater than 0") }
        }
</SCRIPT>

<FORM name="myform">
<INPUT TYPE="text" NAME="number1"
    onBlur="validForm(this.form) "><BR>
</FORM>
```

เหตุการณ์ onChange

61

□ จะทำงานเมื่อ เกิดการเปลี่ยนแปลงข้อมูลภายใน field ของ object ที่กำหนด

□ ตัวอย่างเช่น

```
<SCRIPT LANGUAGE="JavaScript">
function showValue() {
    var show = document.myform.country.value;
    alert("You are going to "+show);
}
</SCRIPT>

<FORM NAME="myform">
<SELECT NAME="country" onChange="showValue()">
    <OPTION VALUE="Thailand">Thailand
    <OPTION VALUE="Indonesia">Indonesia
    <OPTION VALUE="Vietnam">Vietnam
    <OPTION VALUE="Malaysia">Malaysia
</SELECT>
</FORM>
```

เหตุการณ์ onSelect

62

□ จะทำงานเมื่อ ผู้ใช้ทำการเลือกบางส่วนหรือทั้งหมดของข้อความใน Text หรือ Textarea

□ ตัวอย่างเช่น

```
<TEXTAREA ROWS="5" COLS="50" onSelect="alert('You are  
selecting text in this textarea')">
```

Testing data for selection in textarea.

```
</TEXTAREA>
```

เหตุการณ์ onMouseOver / onMouseOut

63

- onMouseOver จะทำงานเมื่อผู้ใช้เลื่อนเมาส์ไปทับพื้นที่ของ Object ที่กำหนด
- onMouseOut จะทำงานต่อเนื่องจาก onMouseOver เมื่อผู้ใช้ได้มีการเลื่อนเมาส์ออกจากพื้นที่ของ Object ที่กำหนด

- ตัวอย่างเช่น

```
<TABLE width=50% border=1 cellspacing=0>  
  <TR onMouseOver="this.bgColor='red'"  
    onMouseOut="this.bgColor=' '">  
    <TD>test data</TD>  
    <TD>test data</TD>  
  </TR>  
</TABLE>
```

เหตุการณ์ onSubmit / onReset

64

- onSubmit จะทำงานเมื่อมีการกดปุ่ม submit form
- onReset จะทำงานเมื่อปุ่ม Reset ถูกกด
- ทั้งสอง Event Handler จะทำงานที่ form ทั้งหมด
- ตัวอย่างเช่น

```
<FORM onReset="alert('The data will be cleared')"  
      onSubmit="alert('Your data will be sent now') ">  
  <INPUT TYPE="text" NAME=""><BR>  
  <INPUT TYPE="submit" VALUE="Send">  
  <INPUT TYPE="reset" VALUE="Clear" >  
</FORM>
```


ตัวอย่าง Event Handler

65

□ Event1.html

```
<html>
  <head>
    <title>in-line JavaScript</title>
  </head>
  <body>
    <form name="form1">
      User Name:
      <input type="text" name="user"><br/>
      <input type="button" name="sendData" value="Send Data"
        onMouseOver = "if (document.form1.user.value.length<1)
          window.alert('Empty Data! Please correct');"
    >
    </form>
  </body>
</html>
```

ตัวอย่าง Event Handler

66

□ Event2.html

```
<html>
<head><title>onClick</title>
  <script>
    function compute(f)
    { var res = parseInt(f.x.value) + parseInt(f.y.value);
      alert(res);
    }
  </script>
</head>
<body>
  <form name = "JavaScript Event">
    <input type = "text" name = "x"> + <input type = "text"
      name = "y">
    <input type = "button" value = "Calculate"
      onClick = "compute(this.form)">
  </form>
</body>
</html>
```

ตัวอย่าง Event Handler

67

□ Event3.html

```
<html>
<head>
  <title>onLoad</title>
  <script>
    function hello() {
      alert("Hello! Welcome to our WebSite.");
    }
  </script>
</head>
<body onLoad = "hello()">
  <h1>Example of onLoad Event Handler:</h1>
</body>
</html>
```

ตัวอย่าง Event Handler

68

□ Event4.html

```
<html><head>
  <script>
    function getKey(e) {
      var code = e.keyCode;
      var character = String.fromCharCode(code);
      var paragraph = document.getElementsByTagName("p")[0];
      paragraph.innerHTML = paragraph.innerHTML + character;
    }
  </script>
</head>
<body onkeypress="getKey(event) ;">
  <h3>Please press any key</h3>
  <p></p>
</body></html>
```

แบบฝึกหัด #1

69

- ให้นักศึกษาแก้ไขโค้ดโปรแกรมในตัวอย่าง event2.html
- เพื่อรับข้อมูลน้ำหนัก (กิโลกรัม) และส่วนสูง (เมตร)
- และนำมาคำนวณค่า BMI ตามสูตรด้านล่าง

- $$BMI = \frac{\text{น้ำหนัก}}{\text{ส่วนสูง} * \text{ส่วนสูง}}$$

- พร้อมทั้งแสดงข้อความภาวะน้ำหนักตัว ดังภาพ



70

การใช้ JavaScript #3

หัวข้อการบรรยาย “การใช้ JavaScript #3”

71

- รู้จักกับ Window Object
- การทำงานกับ Window Object
- รู้จักกับ Document Object
- การทำงานกับ Document Object

การโปรแกรมเชิงวัตถุกับภาษา Java Script

72

- หลังจากที่ได้เรียนรู้ Java Script มาพอสมควรแล้ว จะเห็นได้ว่า Java Script เป็นภาษาที่เขียนในเชิงของวัตถุ โดยจะมองทุกๆ อย่างในโปรแกรมเป็นเสมือน object ต่างๆ มากมาย
- Object จะประกอบไปด้วย Property และ Method
 - ▣ ตัวอย่างเช่น car อาจจะมี Property ที่เป็น brand , version , color , door
- การอ้างถึงคุณสมบัติก็ใช้เครื่องหมาย . (จุด) คั่น เช่น

```
car.brand = TOYOTA
```
- Property บางตัวก็อาจจะเป็น object ก็ได้ เช่น

```
car.door.color = RED
```


การโปรแกรมเชิงวัตถุกับภาษา Java Script

73

- ส่วน Method ของ car ก็อาจจะมี วิธีการที่กำหนดให้ car ทำงาน เช่น
 - ▣ car.go สั่งให้รถวิ่ง
 - ▣ car.stop สั่งให้รถหยุด
- ใน Java Script ก็เช่นกัน มีการใช้โปรแกรมในลักษณะเชิงวัตถุ ซึ่งสามารถแบ่งออกได้เป็น 3 ประเภทใหญ่ๆ ดังนี้
 - ▣ Window Objects เช่น window object , document object , form object เป็นต้น
 - ▣ Built-in Objects เช่น date , string object เป็นต้น
 - ▣ User-define Objects เป็น Object ที่ผู้ใช้สร้างขึ้นเอง

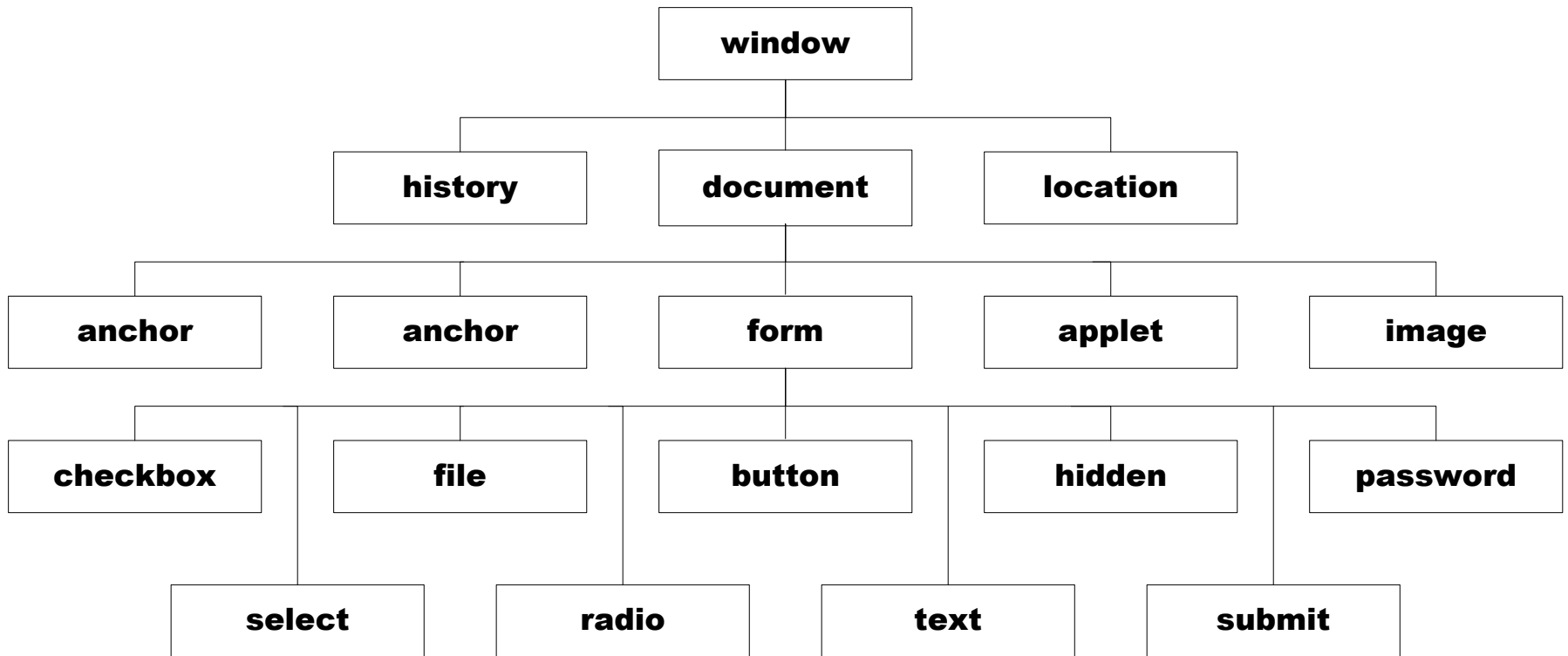
รู้จักกับ Window Objects

74

- Window Object คือ หน้าต่างปัจจุบันของ web browser และเป็นทีเก็บทุกอย่างภายใน window
- เมื่อเปิด web browser ขึ้นมา จะมีการสร้างหน้าต่างขึ้นมา 1 หน้าต่าง (window object) และโปรแกรมจะเริ่มเปิดไปยังเว็บเพจเริ่มต้น (document object) ที่ตั้งค่า default ไว้
- โดยส่วนใหญ่ window จะคงอยู่ตลอดเวลา ถ้าเราไม่ออกจากโปรแกรมไปเสียก่อน ถึงแม้ว่าเราจะเปลี่ยนหน้าเว็บเพจไป ตัว window ก็ยังคงอยู่ แต่หน้าเว็บเพจที่เปลี่ยนไปก็จะทำให้ document object เปลี่ยนไปแทน (ซึ่ง object ตัวนี้จะอยู่ใน window object อีกทีหนึ่ง) หรือถ้ามีการเปิด frame ขึ้นมา frame ก็ถือว่าเป็น อีก object หนึ่ง ซึ่งก็คือ Frame object นั่นเอง

โครงสร้างลำดับชั้นของ object

75



Properties , Method และ Event Handler สำหรับ Window objects

76

| Object | Properties | Method | Event Handler |
|--------|---------------|-----------------|---------------|
| window | defaultstatus | alert | onLoad |
| | frames | blur | onUnload |
| | opener | close | onBlur |
| | parent | confirm | onFocus |
| | scroll | focus | |
| | self | open | |
| | status | prompt | |
| | top | clear / timeout | |
| | window | setTimeout | |

การทำงานกับ Window objects :

(1) การเปิด-ปิด window

77

1. การเปิด – ปิด window

□ เราสามารถสั่งให้ Java Script ทำการเปิด window ขึ้นมาทำงานอีกอันหนึ่ง หรือ ปิด window ที่ไม่ต้องการออกไปก็ได้ โดยจะมีการใช้ method ชื่อ open() และ close() โดย

□ **open()** จะเป็น method สำหรับการเปิด window มีรูปแบบดังนี้

variable = window.open("url", "NAME" , "[OPTION] ")

- variable เป็นชื่อของ window object ที่สร้างขึ้นใหม่
- url เป็นที่อยู่ที่จะใช้ในการอ้างอิง HTML (file ที่ต้องการนำมาแสดง)
- name เป็นชื่อของ window สำหรับอ้างอิงใน HTML

การทำงานกับ Window objects :

(1) การเปิด-ปิด window

78

- [OPTION] เป็นการกำหนดลักษณะของ window โดยนำ option แต่ละชนิดมาเขียนต่อเนื่องกัน ด้วยการใช้เครื่องหมาย , คั่นไว้

| Option | ค่าที่เป็นไปได้ | คำอธิบาย |
|-------------|---------------------------|-------------------------------------|
| toolbar | yes no (ใช้ 1 , 0 แทนได้) | มี toolbar หรือไม่ |
| location | yes no (ใช้ 1 , 0 แทนได้) | แสดง location หรือไม่ |
| directories | yes no (ใช้ 1 , 0 แทนได้) | มี directory button หรือไม่ |
| status | yes no (ใช้ 1 , 0 แทนได้) | มี statusbar หรือไม่ |
| menubar | yes no (ใช้ 1 , 0 แทนได้) | มี menubar หรือไม่ |
| scrollbar | yes no (ใช้ 1 , 0 แทนได้) | มี scrollbar หรือไม่ |
| resizable | yes no (ใช้ 1 , 0 แทนได้) | เพิ่ม ลดขนาด ของ window ได้หรือไม่ |
| width | integer | ความกว้างของ window หน่วยเป็น pixel |
| height | integer | ความสูงของ window หน่วยเป็น pixel |

การทำงานกับ Window objects :

(1) การเปิด-ปิด window

79

□ ตัวอย่าง

```
<SCRIPT LANGUAGE="JavaScript">  
win1 = window.open("http://www.dpu.ac.th","dpu1")  
win2 = window.open("http://www.dpu.ac.th","dpu2","toolbar=0,resizable=0")  
win3 = window.open("http://www.dpu.ac.th","dpu3","toolbar=0,menubar=0,scrollbar=0")  
win4 = window.open("http://www.dpu.ac.th","dpu4","toolbar=0,width=300,height=300")  
</SCRIPT>
```

การทำงานกับ Window objects :

(1) การเปิด-ปิด window

หน้า 80

- **close()** เป็น method สำหรับการปิด window มีรูปแบบดังนี้
 - ▣ `variable.close()` โดยที่ `variable` เป็นชื่อของ window ที่ต้องการปิด

- ตัวอย่าง

```
<SCRIPT LANGUAGE="JavaScript">
function OpenWindow() {
    newwin = window.open("a.html","mywindow1")
}
function CloseWindow() {
    newwin.close();
}
</SCRIPT>
```

```
<INPUT TYPE="button" onClick="OpenWindow()" VALUE="Open new window">
<INPUT TYPE="button" onClick="CloseWindow()" VALUE="Close the window">
```


การทำงานกับ Window objects :

(2) การติดต่อผู้ใช้

หน้า 81

2. การติดต่อกับผู้ใช้

- เราสามารถใช้ Method ต่างของ window object เพื่อทำการติดต่อกับผู้ใช้ได้ โดยจะมี Method ที่ใช้ติดต่อโดยวิธีต่างๆ ดังนี้
- **alert** เป็นข้อความเพื่อเตือนผู้ใช้ขณะทำงาน หรือต้องการสื่อข้อความใดๆ เช่น
`<INPUT TYPE="BUTTON" VALUE="Click" onClick="window.alert('Hello World')"`
- **prompt** เป็นการแสดงข้อความในกล่องโต้ตอบ และมีกล่องข้อความสำหรับรับข้อความ นำเข้า (input) ที่ผู้ใช้ต้องการโต้ตอบกับเอกสารที่ทำงานอยู่ เช่น
`document.write(prompt('Enter your name '));`

การทำงานกับ Window objects :

(2) การติดต่อผู้ใช้

หน้า 82

- **confirm** เป็นกล่องโต้ตอบที่ให้ผู้ใช้นืนยันการกระทำของตน โดยจะมีปุ่ม OK และ Cancel สำหรับถามผู้ใช้ ถ้าผู้ใช้กด OK ก็จะเป็นการคืนค่า true แต่ถ้าผู้ใช้กด Cancel ก็จะเป็นการคืนค่า False ตัวอย่างเช่น

```
<INPUT TYPE="BUTTON" VALUE="Exit" onClick="window.confirm('Do you want to exit ?')"
```

การทำงานกับ Window objects :

(3) Status bar

หน้า 83

3. การแสดงข้อความใน Status bar

- ใน window object มี properties ที่ใช้ควบคุมข้อความใน Status bar 2 ตัว คือ
- `window.defaultStatus` เป็นข้อความเริ่มต้นของ Status bar
- `window.status` เป็นข้อความที่ปรากฏเป็นครั้งคราว เช่น เวลาเอาเมาส์ไปวางบนรูปภาพ หรือ link ซึ่งอาจจะใช้ Event Handler อย่างเช่น `onmouseover` มาช่วยได้

การทำงานกับ Window objects :

(3) Status bar

หน้า 84

□ ตัวอย่างเช่น

```
<SCRIPT LANGUAGE="JavaScript">
    window.defaultStatus="Welcome to my webpage"
</SCRIPT>
<FORM NAME="MyStatusBar">
    <INPUT TYPE="TEXT" NAME="Message">
    <INPUT TYPE="BUTTON" onClick="window.status=MyStatusBar.Message.value">
</FORM>
```

การทำงานกับ Window objects :

(4) Window Synonyms

หน้า 85

- การอ้างอิงถึง window นั้นเราไม่จำเป็นต้องใช้คำว่า window หรือชื่อที่เราตั้งเอาไว้เท่านั้น (ตอนที่เปิด window ใหม่ที่ต้องมีการตั้งชื่อ window) เพราะว่า window จะมี property ที่ใช้แทน window ได้ เช่น
 - `window.self` หมายถึง window ปัจจุบันที่กำลังอ้างอิงอยู่
 - `window.parent` ใช้กับกรณีที่มีหลาย frame โดยหมายถึง node แม่ของมัน
 - `window.opener` หมายถึง window ที่เปิด window ปัจจุบันขึ้นมา
 - `window.top` หมายถึง window ที่อยู่ใน node ระดับบนสุด

การทำงานกับ Window objects :

(4) Window Synonyms

หน้า 86

□ ตัวอย่างเช่น

▣ ไฟล์ Opener.html

```
<FORM NAME="myform">
```

```
  กรุณาใส่รหัสประจำตัวลูกค้า <INPUT TYPE="text" NAME="yourID">
```

```
  <INPUT TYPE="BUTTON" VALUE="ลิ้ม" onClick="window.open('id.html',  
    'win1','menubar=0,width=200,height=200')">
```

```
</FORM>
```

การทำงานกับ Window objects :

(4) Window Synonyms

หน้า 87

□ ไฟล์ id.html

```
<SCRIPT LANGUAGE="JavaScript">
  function FillTextbox() {
    window.opener.document.myform.yourID.value=
    document.idform.id.value;
  }
</SCRIPT>
<FORM NAME="idform">
  <SELECT NAME="id">
    <OPTION VALUE="001">Wholesale
    <OPTION VALUE="002">Retail
    <OPTION VALUE="003">Normal
  </SELECT>
  <INPUT TYPE="button" VALUE="Fill" onClick="FillTextbox()">
</FORM>
```

การทำงานกับ Window objects :

(5) การตั้งเวลาทำงาน

หน้า 88

- Window objects จะมี method ที่ช่วยในการทำงานเกี่ยวกับเรื่องของการตั้งเวลา ดังนี้
- `setTimeout()` จะใช้ตั้งเวลา เมื่อครบกำหนดเวลาจะเรียก function ขึ้นมาทำงาน
- `clearTimeout()` ใช้ยกเลิกการตั้งเวลา
- `setInterval()` จะเป็นการตั้งเวลาเช่นกัน แต่จะทำงานเป็นแบบวนรอบเมื่อครบกำหนดเวลาก็จะเรียก function ขึ้นมาทำงาน แล้วเริ่มจับเวลาต่อทันที
- `clearInterval()` ใช้ยกเลิกการจับเวลาด้วย `setInterval()`

การทำงานกับ Window objects :

(5) การตั้งเวลาทำงาน

หน้า 89

□ ตัวอย่าง

```
<SCRIPT LANGUAGE="JavaScript">
    setInterval("showtime()",1000)
    function showtime() {
        var d = new Date()
        document.myform.text1.value = d.toLocaleTimeString()
    }
    setInterval("reduce()",1000)
    var a = 5;
    function reduce() {
        document.myform.text2.value = a;
        a = a - 1;
    }
</SCRIPT>
<FORM name="myform">
    <INPUT TYPE="text" NAME="text1"> <BR>
    <INPUT TYPE="text" NAME="text2">
</FORM>
```

การทำงานกับ Window objects : ตัวอย่าง

□ ตัวอย่าง (window.html)

```
<html><head><title>Dialog box</title>
<script>
  function Ask1 ()
  { window.alert("Hello!");          }

  function Ask2 ()
  { ans = window.confirm("Are you happy?");
    if (ans == true)
    {   document.write("Good! You are happy."); }
    else
    {   document.write("You are sad."); }
  }

  function Ask3 ()
  { ans = window.prompt("Enter your favorite place:" ,
    "พัทธยา");
    document.write("Your favorite place is " + ans);
  }
</script>
</head>
```

การทำงานกับ Window objects : ตัวอย่าง

□ ตัวอย่าง (window.html)

```
<body>
  <form>
    <input type = "button" value = " Alert " onclick =
      "Ask1()"><p>
    <input type = "button" value = " Confirm " onclick
      = "Ask2()"><p>
    <input type = "button" value = " Prompt " onclick
      = "Ask3()">
  </form>
</body>
</html>
```

การทำงานกับ Document objects

- หมายถึง เอกสาร HTML ซึ่งเป็นเว็บเพจที่กำลังแสดงอยู่ในหน้าต่างหลัก เป็น Property ของ window (window.document)
- ซึ่ง Document เปรียบเสมือนหน้าหนังสือ ที่ต้องมีการบอกเลขหน้า เลขบท หัวเรื่อง ซึ่งต้องมี Property ที่บอกถึงคุณสมบัติเหล่านี้ และนอกจากนี้ในหน้าหนังสือ ก็ต้องมีเนื้อหาสาระประกอบอยู่ภายใน เช่น เนื้อหา หัวข้อ รูปภาพ ตาราง เป็นต้น ซึ่งสิ่งเหล่านี้ล้วนเป็นคุณสมบัติที่สำคัญ ของ Document object
- โดย Document object จะแบ่ง Property ออกเป็น 2 แบบคือ
 - ▣ Property ที่บอกถึงคุณสมบัติของ Document
 - ▣ Property ที่เป็นเนื้อหาภายใน Document

การทำงานกับ Document objects

(1) Property ที่บอกคุณสมบัติของ Document

หน้า 93

| Properties | การทำงาน |
|---------------------|---|
| URL | แสดง URL ของเว็บเพจ |
| title | แสดงชื่อ Title ของเว็บเพจ |
| referrer | แสดง URL ของเว็บเพจที่มีการเชื่อมโยงมาที่เอกสารนี้ |
| lastModified | แสดงวันที่ปรับปรุงเอกสารนี้ครั้งล่าสุด |
| history | เป็น list ของ URL ที่ผู้ใช้เคยเข้าไปเยี่ยมชม โดยใช้ window เดียวกับเพจนี้ |
| domain | Domain ของเว็บเพจ |
| linkColor | กำหนดสีที่ใช้สำหรับส่วนที่เป็นจุด Hyperlink |
| alinkColor | กำหนดสีให้กับ Hyperlink ตอนที่กำลังกดเมาส์ |
| vlinkColor | กำหนดสีให้กับ Hyperlink ที่เคยใช้งานแล้ว |
| bgColor | กำหนดสีพื้นหลังให้กับเว็บเพจ |
| fgColor | กำหนดสีตัวอักษรที่แสดงบนเว็บเพจ |

การทำงานกับ Document objects

(1) Property ที่บอกคุณสมบัติของ Document

หน้า 94

□ ตัวอย่าง

```
<SCRIPT LANGUAGE="JavaScript">  
    document.writeln(document.title);  
    document.writeln(document.lastModified);  
    document.bgColor="#FFFF00";  
    document.fgColor="#0000FF";  
</SCRIPT>
```

การทำงานกับ Document objects

(2) Property ที่เป็นเนื้อหาของ Document

หน้า 95

- ในกรณีของ anchor , applet , form , image และ link นั้นสามารถที่จะเรียกในลักษณะของ Array ได้ เนื่องจาก property เหล่านี้อาจจะมีหลายตัวในเว็บเพจหน้าหนึ่งก็ได้

| Properties | การทำงาน |
|----------------|--|
| *anchor | เป็น anchor object ใน document |
| *applet | เป็น applet ใน document (โปรแกรม JAVA Applet) |
| area | เป็น object ของ image map area ใน document |
| cookie | ให้ค่าข้อความที่บรรจุ cookies ของเว็บเพจที่กำลังใช้งานอยู่ |
| *plugin | เป็นโปรแกรมเพิ่มเติม (plugin) document |
| *form | เป็น object ของ form ใน document |
| *image | เป็น image object ภายใน document |
| *link | เป็นจุดเชื่อมโยงใน document |

การทำงานกับ Document objects

(2) Property ที่เป็นเนื้อหาของ Document

หน้า 96

□ ตัวอย่าง

```
<IMG SRC="car2.jpg" NAME="image1">
```

```
<SCRIPT LANGUAGE="JavaScript">
```

```
    document.write("This image\s width = "+document.image1.width);
```

```
    document.write(" and image\s height = "+document.image1.height);
```

```
</SCRIPT>
```


การทำงานกับ Document objects :

การอ้าง Array of Object ใน Document

หน้า 97

- การอ้างถึง Object ที่มีหลายๆ ตัวในหน้าเว็บนั้น นอกจากจะทำการอ้างอิงชื่อตามแบบที่เคยใช้มา เราอาจจะอ้างได้ในลักษณะของการระบุ **Index ใน Array ของ Object** นั้นได้ด้วย
- ในกรณีของ anchor , applet , form , image และ link นั้นการอ้างในลักษณะของ Array โดยการเติมตัว s เข้าไปและใช้วงเล็บ [n] ต่อท้าย โดยที่ n หมายถึงลำดับที่ปรากฏในเว็บเพจ เช่น

`document.forms[0]`

`document.images[3]`

การทำงานกับ Document objects :

การอ้าง Array of Object ใน Document

หน้า 98

- ตัวอย่างของการใช้ Array of Object

```
<SCRIPT LANGUAGE="JavaScript">
function check1()
    {alert("U R Checking Text from first form :"+ document.forms[0].t.value) }
function check2()
    {alert("U R Checking Text from second form :"+document.forms[1].t.value) }
</SCRIPT>
<FORM>
    <INPUT TYPE="text" NAME="t">
    <INPUT TYPE="button" onClick="check1()">
</FORM>
<FORM>
    <INPUT TYPE="text" NAME="t">
    <INPUT TYPE="button" onClick="check2()">
</FORM>
```

การทำงานกับ Document objects : ตัวอย่าง

□ ตัวอย่าง (document.html)

```
<html><head>
  <script>
    function changeValue ()
    {
      var x = document.getElementById("myHeader") ;
      x.innerHTML = "New Header";
    }
  </script>
</head>
<body>
  <h1 id = "myHeader">This is a header</h1>
  <p>Click the button to change the header's value</p>
  <input type = "button" onclick = "changeValue () "
    value = "Change Header">
</body>
</html>
```

การทำงานกับ Document objects : ตัวอย่าง

หน้า 100

- ตัวอย่าง (dom_css.html)

```
<p id="p2">Hello World!</p>

<script>
  document.getElementById("p2").style.color = "blue";
</script>
```

Method ของ Document

- ใน document เราสามารถใช้ Method ต่างๆ ได้ เช่น

| Method | การทำงาน |
|----------------|---|
| clear | ลบข้อความทั้งใน window ปัจจุบัน (<i>ไม่ใช่แล้วใน JavaScript version ปัจจุบัน</i>) |
| close | ยกเลิกการแสดงผลข้อความแล้วปิด window ปัจจุบัน |
| write | แสดงผลข้อความลงเว็บเพจ |
| writeln | แสดงผลข้อความ โดยมีการขึ้นบรรทัดใหม่ทุกครั้งที่สิ้นสุดข้อความในแต่ละชุด |

102

การใช้ JavaScript #4

หัวข้อการบรรยาย “การใช้ JavaScript #4”

หน้า 103

- รู้จักกับ Location Object
- การทำงานกับ Location Object
- รู้จักกับ Form Object
- การทำงานกับ Form Object

รู้จักกับ Location object

หน้า 104

- **Location** เป็น Object ที่ใช้สำหรับการดูข้อมูลที่อยู่ URL ของเว็บเพจจะให้ค่าต่างๆ ของ Document ที่แสดงบน window และ จะไม่อนุญาตให้กำหนดค่าต่างๆ ของ Location ได้ (เป็น Read-only)

| Properties | คำอธิบาย |
|------------|--|
| hostname | แสดงชื่อ Domain หรือ IP ของ URL ที่เรียกใช้งาน |
| port | แสดงหมายเลข Port ที่เรียกใช้งาน |
| pathname | แสดง path name ของ URL ที่เรียกใช้งาน |
| search | แสดงข้อมูลคำถาม ที่ส่งไปยัง URL ที่เรียกใช้งาน |
| hash | แสดงชื่อ Anchor ที่เชื่อมโยงไปจุดต่างๆ ภายใน URL |

รู้จักกับ Location object

| Properties | คำอธิบาย |
|------------|--|
| href | แสดงชื่อ URL ที่เรียกใช้งาน |
| host | แสดงชื่อ hostname ของ URL ที่เรียกใช้งาน |
| protocol | เป็นการแสดง Protocol ที่เรียกใช้งาน |
| Methods | คำอธิบาย |
| reload() | ใช้ reload ข้อมูลจากเว็บเพจ |
| replace() | ใช้ load ข้อมูลจาก URL ที่กำหนด |

รู้จักกับ Location object

- ตัวอย่าง ให้ทดลองสร้าง file ขึ้นมา และ save ข้อมูล จากนั้นแทรก code ดังนี้

```
<SCRIPT LANGUAGE="JavaScript">  
    document.write(window.location.href+"<BR>")  
    document.write(window.location.protocol+"<BR>")  
</SCRIPT>
```

รู้จักกับ Location object

หน้า 107

- ตัวอย่าง ให้ทดลองสร้าง file ขึ้นมา และ save ข้อมูล จากนั้นแทรก code ดังนี้

```
<SCRIPT LANGUAGE="JavaScript">
    function GoTo() {
        window.location=document.myform.URL.value
    }
</SCRIPT>
<FORM NAME="myform">
    <SELECT NAME="URL" onChange="GoTo()">
        <OPTION VALUE="http://www.yahoo.com">YAHOO
        <OPTION VALUE="http://www.google.com">GOOGLE
        <OPTION VALUE="http://www.pantip.com">PANTIP
        <OPTION VALUE="http://www.fecebook.com">FECEBOOK
    </SELECT>
</FORM>
```

รู้จักกับ Form object

- จากที่ผ่านมามากมาย object นั้น จะเห็นได้ว่าเราสามารถนำ Java Script มาใช้ทำงานร่วมกับ FORM ที่อยู่ใน HTML ได้ โดยจะทำให้ FORM มีความสามารถมากขึ้นด้วย
- หลักสำคัญในการทำงาน อยู่ที่เครื่องมือต่างๆ มาใช้กับ Event Handler ที่เราได้กล่าวถึงในบทก่อน เช่น การใช้ onClick กับ Button หรือ การใช้ onSubmit กับ FORM เป็นต้น

การใช้งาน Form object : Textbox object

- Textbox เป็นช่วงที่ให้กรอกข้อมูลต่างๆ โดย Textbox จะถือเป็น Property หนึ่งของ Form โดยที่ Property ของ Text จะมีดังนี้

| Properties | คำอธิบาย |
|--------------|---|
| name | ชื่อของ Textbox |
| type | ชนิดของ Textbox มีค่าได้ 3 แบบ คือ hidden , password , text |
| defaultValue | ค่าเริ่มต้นของ Textbox |
| value | ข้อความที่ปรากฏอยู่ใน Text |

| Methods | คำอธิบาย |
|----------|-------------------------------|
| focus() | ใช้ระบุจุด Focus ของ Textbox |
| blur() | ใช้ ยกเลิกจุด Focus |
| Select() | เลือกข้อความที่อยู่ใน Textbox |

การใช้งาน Form object : Submit object

- ถูกสร้างขึ้นมาจากคำสั่ง `<INPUT TYPE="submit">` เมื่อผู้ใช้กดปุ่มนี้ จะเป็นการส่ง ข้อมูลในแบบฟอร์มไปยังเครื่อง Server ซึ่งก็จะเกิด Event Handler ก็คือ `onSubmit` นั้นเอง

การใช้งาน Form object : Submit object

□ ตัวอย่าง

```
<SCRIPT LANGUAGE="JavaScript">
  function MailTo(form) {
    var subject=myform.input1.value;
    var cc = myform.input2.value;
    window.location = "mailto:test@it.dpu.ac.th?subject=" + subject + "&cc=" + cc;
  }
</SCRIPT>
<FORM name="myform" onSubmit="return MailTo(this.form)">
  Subject<INPUT TYPE="text" NAME="input1"><BR>
  recipient<INPUT TYPE="text" NAME="input2">
  <INPUT TYPE="submit" value="submit">
</FORM>
```

การใช้งาน Form object : Checkbox object

หน้า 112

- Property และ Method ของ Checkbox โดย Event ที่ใช้ได้กับ Checkbox ก็คือ onBlur , onFocus , onClick

| Properties | คำอธิบาย |
|----------------|--|
| Checked | ค่าบูลีนที่แสดงเป็น T ถ้า Checkbox นี้ ถูก Click |
| defaultChecked | ค่าบูลีนที่แสดงเป็น T ถ้า Checkbox นี้ ถูกเลือกเป็นค่าเริ่มต้น |
| Value | ค่าของ Checkbox เมื่อมีการเลือกใช้ (ข้อมูลที่ได้เมื่อเลือก) |
| Name | ชื่อของ Checkbox object |
| Methods | คำอธิบาย |
| click() | การเลือก Click ที่ Checkbox |

การใช้งาน Form object : Checkbox object

□ ตัวอย่าง

```
<SCRIPT LANGUAGE="JavaScript">
  function checkAll() {
    for (var i = 0 ; i < document.myform.elements.length ; i++) {
      var e = document.myform.elements[i]
      if (e.name != "selectAll") {
        e.checked = !e.checked;
      }
    }
  }
</SCRIPT>
<FORM NAME="myform">
  Select All <INPUT TYPE="checkbox" NAME="selectAll" onClick="checkAll()"><BR>
  <INPUT TYPE="checkbox" NAME="check1" VALUE="check">Italian<BR>
  <INPUT TYPE="checkbox" NAME="check2" VALUE="check">Chinese<BR>
  <INPUT TYPE="checkbox" NAME="check3" VALUE="check">Indian<BR>
  <INPUT TYPE="checkbox" NAME="check4" VALUE="check">Mexican<BR>
</FORM>
```

การใช้งาน Form object : Radio object

- Property , Method และ Event จะเหมือนกับ Checkbox แต่ว่า Radio จะมีข้อแตกต่างตรงที่ แต่ละตัวต้องมีการกำหนดชื่อเป็นชื่อเดียว เพื่อให้เป็น Radio ชุดเดียวกัน
- การติดต่อกับ Radio จะทำได้โดยอ้างถึงหมายเลขลำดับใน Array กับชื่อที่กำหนดไว้ใน HTML โดยรายการแรกของชุดมีหมายเลขลำดับเป็น 0 รายการที่สองมีหมายเลขเป็น 1 เรื่อยไปตามลำดับ

การใช้งาน Form object : Radio object

□ ตัวอย่าง

```
<SCRIPT LANGUAGE="JavaScript">
  function testRadio(form) {
    for (count = 0 ; count < 3 ; count++) {
      if (form.rad[count].checked) {
        break;
      }
    }
    alert("Button " + count + " is selected and value = " + form.rad[count].value)
  }
</SCRIPT>
<FORM NAME="myform">
  <INPUT TYPE="radio" NAME="rad" VALUE="A">Grade A
  <INPUT TYPE="radio" NAME="rad" VALUE="B">Grade B
  <INPUT TYPE="radio" NAME="rad" VALUE="C">Grade C
  <INPUT TYPE="button" onClick="testRadio(this.form)" VALUE="Click">
```

การใช้งาน Form object : Select object

- เป็นรายการแบบเลือกตอบ ที่แสดงตัวเลือกในลักษณะของ Drop Down List Box บรรทัดเดียว หรือ จะเป็นแบบหลายบรรทัด (list box) แล้วแต่กำหนด ซึ่งจะยอมให้ผู้ใช้เลือกได้เพียงตัวเลือกเดียวเท่านั้น

การใช้งาน Form object : Select object

หน้า 117

- Property ของ Select จะมีดังนี้

| Properties | คำอธิบาย |
|-----------------|--|
| name | ชื่อของ Select |
| length | จำนวนตัวเลือกทั้งหมดในรายการ |
| options | ข้อมูลที่แสดงแต่ละรายการ |
| defaultSelected | ค่าบูลีน แสดงถึงตัวเลือกที่กำหนดเป็นค่าเริ่มต้น |
| index | ลำดับของรายการเลือกแต่ละรายการ |
| selected | ค่าบูลีน แสดงหมายถึงตัวเลือกนั้นๆ ถูกเลือกอยู่หรือไม่ |
| selectedIndex | ค่าเลขจำนวนเต็มแสดงหมายเลขที่ถูกเลือก |
| text | ค่า String บรรจุข้อความที่ใช้แสดงในรายการ |
| value | ค่า String ที่ใช้ส่งไปกับฟอร์มถ้าผู้ใช้เลือกตัวเลือกนั้น |

การใช้งาน Form object : Select object

- นอกจากนี้แล้วยังมี Property อีกตัวหนึ่ง ก็คือ Option Array ซึ่งเป็น Array ของตัวเลือกใน Select ที่สร้างโดยคำสั่ง <OPTION>
- สมาชิกทุกตัวของ Option Array คือ ตัวเลือกทุกตัวเลือกใน Selection List นั้นเอง โดยใช้หมายเลขสำหรับการอ้างอิงเรียงตามลำดับกันไป โดยเริ่มต้น Index ที่ 0 และมี Property ดังนี้

| Properties | คำอธิบาย |
|--------------------|---------------------------|
| option(0).index | เป็นหมายเลขของ Options |
| option(0).selected | Option นี้ถูกเลือกหรือไม่ |
| option(0).text | ข้อความของ Option |

การใช้งาน Form object : Select object

□ ตัวอย่าง

```
<SCRIPT LANGUAGE="JavaScript">
  function Show(form) {
    alert(form.device.value + "\nIndex=" + form.device.selectedIndex)
  }
</SCRIPT>
<FORM NAME="myform">
  <SELECT NAME="device" onChange="Show(this.form)">
    <OPTION VALUE="Mainboard">MB
    <OPTION VALUE="Monitor">MN
    <OPTION VALUE="Microprocessor">MP
  </SELECT>
</FORM>
```

120

การใช้ JavaScript #5

หัวข้อการบรรยาย “การใช้ JavaScript #5”

หน้า 121

- ☐ String Object
- ☐ Regular Expression
- ☐ Math Object
- ☐ Date Object
- ☐ Array Object

String Object

หน้า 122

- ข้อความ (String Object) สามารถใช้ฟังก์ชันในการทำงานได้ทั้ง Properties และ Method ดังนี้
- Properties
 - ▣ `length` : Returns the number of characters in a string

String Object

□ Method

- ▣ `charAt(position)` : Returns the character at a specified position
- ▣ `indexOf(pattern)` : Returns the position of the first occurrence of a specified string value in a string
- ▣ `lastIndexOf(pattern)` : Returns the position of the last occurrence of a specified string value, (searching backwards)
- ▣ `Spilt(separator)` : Splits a string into an array of strings
- ▣ `Substr(startIndex,length)` : Extracts a specified number of characters in a string, from a start index
- ▣ `toLowerCase()` : Displays a string in lowercase letters
- ▣ `toUpperCase()` : Displays a string in uppercase letters

Regular Expression

หน้า 124

- นิพจน์ทั่วไป (Regular Expression) คือ ลำดับของอักขระที่สร้างรูปแบบการค้นหา
- นิพจน์ทั่วไปสามารถใช้เพื่อดำเนินการค้นหาข้อความและแทนที่ข้อความทุกประเภท
- Method
 - ▣ `match()` : returns an array with the matches.
 - ▣ `search()` : returns the position of the match.
 - ▣ `replace()` : returns a modified string where the pattern is replaced
 - ▣ `test()` : returns true or false

Regular Expression

□ Regular Expression Patterns

| Modifier | Description |
|----------------|--|
| <code>i</code> | Perform case-insensitive matching |
| <code>g</code> | Perform a global match (find all matches rather than stopping after the first match) |
| <code>m</code> | Perform multiline matching |

Regular Expression

□ Regular Expression Patterns

| Expression | Description |
|------------|---|
| [abc] | Find any of the characters between the brackets |
| [^abc] | Find any character NOT between the brackets |
| [0-9] | Find any of the digits between the brackets |
| [^0-9] | Find any character NOT between the brackets (any non-digit) |
| (x y) | Find any of the alternatives separated with |

Regular Expression

หน้า 127

□ Regular Expression Patterns

| Meta character | Description |
|----------------|--|
| . | Find a single character, except newline or line terminator |
| \w | Find a word character |
| \W | Find a non-word character |
| \d | Find a digit |
| \D | Find a non-digit character |
| \s | Find a whitespace character |
| \S | Find a non-whitespace character |
| \b | Find a match at the beginning/end of a word |
| \B | Find a match, but not at the beginning/end of a word |

Regular Expression

□ Regular Expression Patterns

| Quantifier | Description |
|---------------------|---|
| <code>n+</code> | Matches any string that contains at least one <code>n</code> |
| <code>n*</code> | Matches any string that contains zero or more occurrences of <code>n</code> |
| <code>n?</code> | Matches any string that contains zero or one occurrences of <code>n</code> |
| <code>n{X}</code> | Matches any string that contains a sequence of <code>X</code> <code>n</code> 's |
| <code>n{X,Y}</code> | Matches any string that contains a sequence of <code>X</code> to <code>Y</code> <code>n</code> 's |
| <code>n{X,}</code> | Matches any string that contains a sequence of at least <code>X</code> <code>n</code> 's |
| <code>n\$</code> | Matches any string with <code>n</code> at the end of it |
| <code>^n</code> | Matches any string with <code>n</code> at the beginning of it |

Regular Expression

□ ตัวอย่าง

```
<p id="demo"></p>
<script>
  let text = "Is this all there is?";
  let result = text.match(/is/g);
  document.getElementById("demo").innerHTML = result;
</script>
```

```
<p id="demo"></p>
<script>
  let text = "123456789";
  let pattern = /[1-4]/g;
  result = pattern.test(text);
  document.getElementById("demo").innerHTML = result;
</script>
```

Regular Expression

□ ตัวอย่าง

```
<p id="demo"></p>
<script>
  let text = "re green red green gren gr blue yellow";
  let result = text.match(/(red|green)/g);
  document.getElementById("demo").innerHTML = result;
</script>
```

```
<p id="demo"></p>
<script>
  let text = "Hellooo World! Hello";
  let result = text.match(/lo*/g);
  document.getElementById("demo").innerHTML = result;
</script>
```

Regular Expression

หน้า 131

□ ตัวอย่าง

```
<p id="demo"></p>
<script>
  let text = "100, 1000 or 10000?";
  let pattern = /\d{4,}/g;
  let result = text.match(pattern);
  document.getElementById("demo").innerHTML = result;
</script>
```

```
<script>
  let email = "aurawan.ims@dpu.ac.th";
  console.log(email.toLowerCase().match(
    /^(([^<>()[\]\.\.,;:\s@"]+(\.[^<>()[\]\.\.,;:\s@"]+)*)|("[.+"))@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\]|([a-zA-Z\-\0-9]+\.)+[a-zA-Z]{2,}))$ /
  ));
</script>
```

แบบฝึกหัด #2

หน้า 132

- ให้นักศึกษา เขียนโปรแกรม JavaScript เพื่อตรวจสอบความถูกต้องของข้อมูลที่ใช้กรอกลงในแบบฟอร์ม ตัวอย่างดังภาพ

Welcome to LoveBook.com

Join for Free

User Name: @lovebook.com (max 10 characters) *

Password: (from 4 to 8 characters) *

Confirm Password: *

Email Address: *

Age: (from 1 to 100)

Gender: ☒ Male ☐ Female

Date Object

- Date Object ใช้เพื่อทำงานเกี่ยวกับ วัน (dates) และ เวลา (times)
- การกำหนด Date Object (defining dates)

```
let today = new Date();  
let myDate = new Date('2015-11-01');
```

- Method
 - ▣ **getDate()** : Returns the day of the month from a Date object (1-31)
 - ▣ **getDay()** : Returns the day of the week from a Date object (0-6)
 - ▣ **getMonth()** : Returns the month from a Date object (from 0-11)
 - ▣ **getFullYear()** : Returns the year, as a four-digit number

Date Object

□ Method

- ▣ **getFullYear()**: Returns the year, as a two-digit or a three/four-digit number, depending on the browser. Use **getFullYear()** instead !!
- ▣ **getHours()**: Returns the hour of a Date object (from 0-23)
- ▣ **getMinutes()**: Returns the minutes of a Date object (from 0-59)
- ▣ **getSeconds()**: Sets the seconds in a Date object (from 0-59)

Date Object

หน้า 135

□ ตัวอย่าง (Format_Date.html)

```
<html><head><title>Format_Date.html</title>
<script>
    function formatDate(date, separator){
        let fDate = "";
        let d = date.getDate();
        let m = date.getMonth() + 1;
        let y = date.getFullYear();
        fDate = d + separator + m + separator + y;
        return fDate;
    }
</script>
</head>
<body>
    <script>
        let today = new Date();
        document.write("Today (default format) : " + today + "<br> ");
        document.write("Today (set format) : " + formatDate(today,
"/") );
    </script>
</body></html>
```

แบบฝึกหัด #3

หน้า 136

- ให้นักศึกษาเขียนโปรแกรม html และ javascript เพื่อรับข้อมูลวันเกิดของผู้ใช้ แล้วคำนวณอายุปัจจุบันของผู้ใช้ แสดงออกทางหน้าจอ
 - ▣ หมายเหตุ เมื่อนำวัน 2 วันมาลบกัน จะได้ผลต่างเป็นช่วงเวลาในหน่วย มิลลิวินาที (ms)
 - ▣ (1 second = 1000 ms, 1 hour = 60*60 seconds, 1 day = 24 hours)

Math Object

- Math Object เป็นเมธอดช่วยให้สามารถทำงานทางด้านคณิตศาสตร์ต่างๆ ได้
- การกำหนด Math Object

```
let pi_value = Math.PI;  
let sqrt_value = Math.sqrt(16);
```

- Method
 - ▣ **max(x,y)**: Returns the number with the highest value of x and y
 - ▣ **min(x,y)**: Returns the number with the lowest value of x and y
 - ▣ **pow(x,y)**: Returns the value of x to the power of y
 - ▣ **random()**: Returns a random number between 0 and 1
 - ▣ **round(x)**: Rounds a number to the nearest integer
 - ▣ **ceil(x)**: Returns the value of a number rounded upwards to the nearest integer

Array Object

- **Array Object** ช่วยให้สามารถเก็บค่าต่างๆ ไว้ในอ็อบเจกต์อันเดียว หรือตัวแปรชื่อเดียว (single variable name) กันได้

- การกำหนด Array Object

```
let myCars = new Array();  
myCars[0] = "Saab";  
myCars[1] = "Volvo";  
myCars[2] = "BMW";
```

or **let myCars = new Array("Saab", "Volvo", "BMW");**

or **let myCars=["Saab", "Volvo", "BMW"];**

- การอ้างถึงค่าใน Array Object

```
document.write(myCars[0]);
```

Array Object

□ Properties

- ▣ **length** : Sets or returns the number of elements in an array

□ ตัวอย่าง (Array.html)

```
<script>
  let x, i;
  let fruits = ["Banana", "Orange", "Apple"];
  for (x in fruits)
  { document.write(fruits[x] + "<br />"); }

  for (i = 0; i < fruits.length; i++)
  { document.write(fruits[i] + "<br />"); }
</script>
```

แบบฝึกหัด #4

หน้า 140

- ให้นักศึกษาประยุกต์ใช้ความรู้ในเรื่อง array และ document object ในการแสดง checkbox และจัดการ array ของ checkbox ในแบบฟอร์ม โดยหากผู้ใช้ชอบทุกรายการจะสามารถเลือกที่ช่อง Love All โปรแกรมจะคลิกเลือกผลไม้ทุกรายการให้อัตโนมัติ หรือถ้าคลิกยกเลิกที่ช่อง Love All จะเป็นการยกเลิกการเลือกทุกรายการ

What would you like?

- ☐ Love All
- ☐ Banana
- ☐ Mango
- ☐ Orange
- ☐ Apple