ICT12367

การใช้กรอบงานสำหรับการพัฒนาเว็บแอปพลิเคชัน เพื่อความมั่นคงปลอดภัย

Chapter 4

สร้างเว็บแอพพลิเคชั่นด้วย JavaScript

รูปแบบการเขียน JavaScript

1.แบบ Internal คือ กำหนด JavaScript ไว้ในส่วนของ <head></head> หรือ <body></body>

```
<script type="text/javascript">
        ..... Statement.....
 </script>
```

```
<script type="text/javascript">
    document.write("Kong Ruksiam");
</script>
```

รูปแบบการเขียน JavaScript

2. แบบ External คือ กำหนด JavaScript ไว้เป็นไฟล์ด้านนอกที่มีนามสกุล .js จากนั้นก็นำเข้ามาทำงานในหน้าเว็บ หรือ HTML ไฟล์

```
<script src="ชื่อไฟล์.js"></script>
```

```
document.write("KongRuksiam");
document.write("<br>");
document.write("JavaScript เบื้องต้น");
```

การแสดงผลข้อมูล

- **document.write** (“ข้อความที่ต้องการแสดง”) แสดงเป็นข้อความตัวเลข ตัวแปร หรือ แท็ก HTML ก็ได้ในหน้าเว็บ
- **alert**(“ข้อความแจ้งเตือน”) สำหรับแจ้งเตือนผู้ใช้ในหน้าเว็บ
- **Console.log**(“ข้อความ หรือ ตัวแปร”) สำหรับ debug ค่าต่างๆ แต่จะไม่แสดงผลในหน้าเว็บ

วิธีการแสดงผลใน JavaScript

- การใช้ document.write
  - ใช้เพื่อเขียนข้อความหรือ HTML ลงในเอกสาร
  - มักใช้ในช่วงการโหลดเอกสาร แต่ควรหลีกเลี่ยงในโค้ดที่ซับซ้อน เนื่องจากอาจลบเนื้อหาเดิมทั้งหมดหากใช้หลังจากหน้าโหลดแล้ว

```javascript
document.write("Hello, World!");
```

```html
<!DOCTYPE html>
<html>
<body>
    <script>
        document.write("<h1>Welcome to JavaScript!</h1>");
    </script>
</body>
</html>
```

วิธีการแสดงผลใน JavaScript

- การใช้ console.log()
  - ใช้สำหรับแสดงผลในคอนโซล (Console) ของเบราว์เซอร์
  - เหมาะสำหรับการดีบักหรือแสดงข้อมูลสำหรับนักพัฒนา

```javascript
console.log("Debugging information here.");
```

```html
<!DOCTYPE html>
<html>
<body>
    <script>
        console.log("This message is displayed in the console.");
    </script>
</body>
</html>
```

วิธีการแสดงผลใน JavaScript

- การใช้ innerHTML
  - ใช้เพื่อปรับแต่งหรือแสดงข้อความในองค์ประกอบ HTML ที่ระบุ
  - นิยมใช้ในการสร้างหน้าแบบไดนามิก

```javascript
document.getElementById("myDiv").innerHTML = "This is dynamic content.";
```

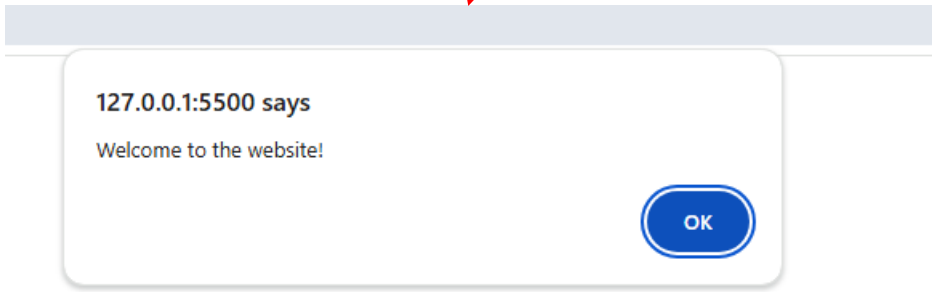
```html
<!DOCTYPE html>
<html>
<body>
    <div id="myDiv"></div>
    <script>
<div id="myDiv"></div>
<script>
    document.getElementById("myDiv").innerHTML = "<p>This is inserted via JavaScript!</p>";
</script>
</body>
</html>
```

วิธีการแสดงผลใน JavaScript

- การใช้ alert()
  - ใช้เพื่อแสดงข้อความในกล่องแจ้งเตือน (Alert Box)
  - มักใช้เพื่อแสดงข้อมูลหรือแจ้งเตือนผู้ใช้

```javascript
alert("Hello, this is an alert box!");
```

```html
<!DOCTYPE html>
<html>
<body>
    <script>
        alert("Welcome to the website!");
    </script>
</body>
</html>
```



ตารางสรุปคำสั่งแสดงผลใน JavaScript

คำสั่ง	ใช้สำหรับ	ตัวอย่างการใช้งาน
document.write	แสดงผลหรือเขียนเนื้อหาในเอกสาร	เขียนข้อความ HTML
innerHTML	ปรับแต่งข้อความในองค์ประกอบ HTML	แก้ไขหรือเพิ่มเนื้อหาใน
console.log	แสดงผลในคอนโซลสำหรับดีบัก	ข้อความดีบัก
alert	แสดงข้อความแจ้งเตือน	กล่องข้อความแจ้งเตือน
prompt	รับอินพุตจากผู้ใช้ผ่านกล่องข้อความ	กล่องข้อความที่ผู้ใช้กรอกได้
Confirm	ขอคำยืนยันจากผู้ใช้	แสดงตัวเลือก OK/Cancel
window.print	เปิดหน้าต่างพิมพ์	เปิด Print Dialog

สรุป

แต่ละคำสั่งใน JavaScript มีบทบาทเฉพาะตัวในการแสดงผลข้อมูลหรือโต้ตอบกับผู้ใช้ คุณสามารถเลือกใช้คำสั่งให้เหมาะสมกับความต้องการ เช่น การดีบักใช้ console.log การโต้ตอบใช้ alert, prompt, หรือ confirm และการสร้างหน้าแบบไดนามิกใช้ innerHTML หรือ document.write

การเขียนคำอธิบาย (Comment)

- วิธีที่ 1 โดยใช้เครื่องหมาย Slash(/) ใช้ในการอธิบาย

  คำสั่งสั้นๆ ในรูปแบบบรรทัดเดียว เช่น

```
1    const count = document.querySelector(".count") //แสดงข้อความในหน้าเว็บ
2    const input = document.querySelector("input") //แสดงข้อความในหน้าเว็บ
```

- วิธีที่ 2 เขียนคำอธิบายไว้ในเครื่องหมาย /*/ ใช้ในการอธิบาย

  คำสั่งยาวๆ หรือแบบหลายบรรทัด

```
7  /*
8     แสดงข้อความในแท็บ console ของเว็บ browser
9     */
```

ตัวแปรและชนิดข้อมูล

- ตัวแปร คือ **ชื่อที่ถูกนิยามขึ้นมา** เพื่อใช้เก็บค่าข้อมูล สำหรับนำไปใช้งานในโปรแกรม โดยข้อมูลอาจประกอบด้วย ข้อความ ตัวเลข ตัวอักษรหรือผลลัพธ์จากการประมวลผลข้อมูล

รูปแบบการตั้งชื่อ

1. ขึ้นต้นด้วยตัวอักษรในภาษาอังกฤษตามด้วยตัวอักษรหรือตัวเลข
2. ห้ามขึ้นต้นด้วยตัวเลขหรือสัญลักษณ์พิเศษ
3. ขึ้นต้นด้วย $ (dollar sign) และ _(underscore) ได้
4. มีลักษณะเป็น case sensitive คือ ตัวพิมพ์เล็กใหญ่จะมีความหมายที่แตกต่างกัน
5. ไม่ซ้ำกับคำสงวน (keyword)

การนิยามตัวแปร

var (เปลี่ยนแปลงค่าในตัวแปรได้)

var ชื่อตัวแปร;

var ชื่อตัวแปร = ค่าเริ่มต้น;

var ชื่อตัวแปร = ค่าเริ่มต้น, ชื่อตัวแปร = ค่าเริ่มต้น

```
var money;
var money=100;
money=200;


var a,b,c,d;
var x = 10, y=20, z=30;
```

*** ตัวแปรที่ประกาศไว้แต่ยังไม่ได้กำหนดค่า จะมีค่าเป็น undefined โดยอัตโนมัติ

การนิยามตัวแปร

const (ค่าคงที่)

const ชื่อตัวแปร = ค่าของตัวแปร;

เช่น

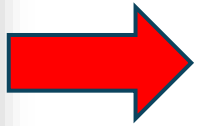
```
const money=100;
money=200;// เปลี่ยนแปลงค่าเดิมไม่ได้
```

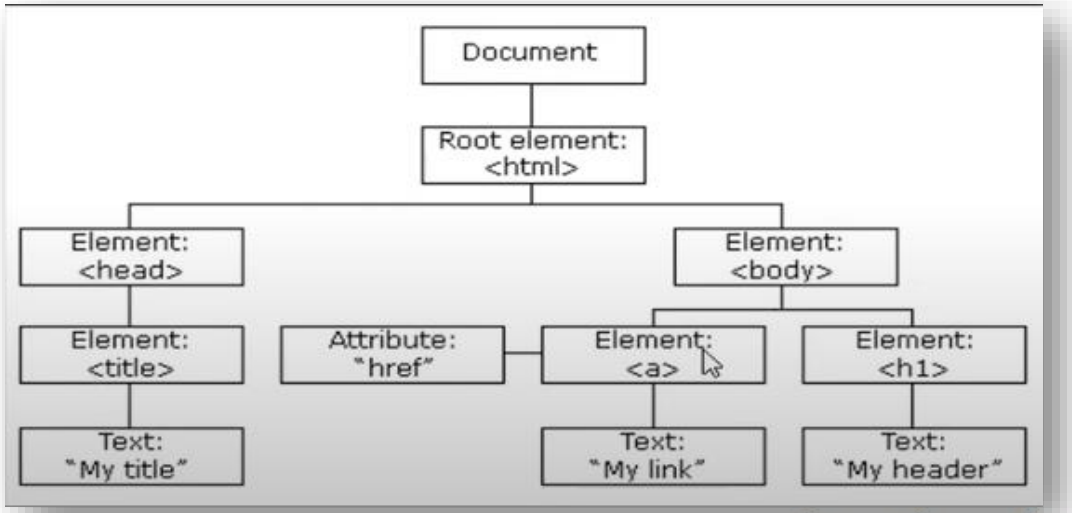
Data Type	คำอธิบาย	รูปแบบข้อมูล
boolean	ค่าทางตรรกศาสตร์	True /False
number	ตัวเลขที่ไม่มีจุดทศนิยม	20
	ตัวเลขที่มีจุดทศนิยม	30.15
string	ข้อความ	“kongruksiam”
object	ข้อมูลเชิงวัตถุ	{firstName:"kong", lastName:"ruksiam", age:20};
array	ชุดข้อมูล	["มะม่วง", "มะละกอ", "ส้ม"]

HTML DOM (Document Object Model)

- เมื่อหน้าเว็บโหลดเสร็จเรียบร้อย Web browser มันจะสร้าง DOM ของหน้านั้นขึ้นมา โดยมอง Html เป็นโครงสร้างต้นไม้ (ก้อน Object) หรือ เรียกว่า DOM

```
<html>
<head>
        <title>My title</title>
</head>
 <body>
<a href="#">My link</a>
<h1>My header</h1>
</body>
</html>
```





** Tag ต่างๆ ใน html จะเรียกว่า Element

คุณสมบัติของ Html Dom

- เข้าถึงและเปลี่ยนคุณสมบัติทั้งหมดของ Element ในหน้าเว็บได้
- ควบคุมและเปลี่ยนรูปแบบ CSS ได้
- สามารถตอบสนองกับทุกเหตุการณ์ที่เกิดขึ้นหน้าเว็บได้

การเข้าถึง element ผ่าน id, tag, class

- **document.getElementByld(“ชื่อไอดี”)** ส่วนนี้จะใส่ชื่อ id ของ element แล้วจะได้ object ของ element นั้นออกมา
- **document.getElementByTagName(“ชื่อแท็ก”)** ส่วนนี้จะใส่ชื่อ tag แล้วจะได้ array ของ object ทุกตัวตามชื่อ tag ที่เราส่งไป
- **document.getElementByClassName(“ชื่อคลาส”)** ส่วนนี้จะใส่ชื่อ class ของ element แล้วจะได้ array ของ object ที่มี class ตามที่เราเรียกไปออกมา

DOM Document

- เปลี่ยนเนื้อหา Html : **element.innerHTML**
- เปลี่ยนเนื้อหา Text: **element.innerText**
- เปลี่ยน style Element: **element.style.properties = value**

ดำเนินการผ่าน Method

- **element.setAttribute(attribute, value)**

DOM Nodes

- Document.createElement(element) // สร้าง element ใหม่
- Document.removeChild(element) //ลบ node ลูก
- Document.appendChild(element) // นำ element ไปต่อใน node แม่
- Document.replaceChild(new, old) // แทนที่ element

DOM CSS Ass & Remove Class

- Element.classList.add(“class”) // เพิ่ม class style
- Element.classList.remove(“class”) //ลบ class style
- Element.classList.toggle(“class”) // สลับ class style
- Element.classList.contains(“class”) // เปรียบเทียบ class style

DOM Event

- คือ เหตุการณ์หรือการกระทำบางอย่างที่เกิดขึ้นกับอิลิเมนต์ เช่น การคลิกเมาส์ การเคลื่อนย้ายเม้าส์ การกดปุ่มคีย์บอร์ด เป็นต้น
- โดยผู้พัฒนาสามารถใช้อีเว้นต์ที่เกิดขึ้นเป็นกำหนดให้ตอบสนองหรือกระทำบางอย่างได้ เช่น การคลิกแล้วแจ้งเตือน เป็นต้น

ชื่อ Event	ความหมาย	ทำงานร่วมกับแท็ก
onfocus=“ ”	เมื่ออิลิเมนต์นั้นได้รับการโฟกัส	select, text, textarea
onblur=“ ”	เมื่ออิลิเมนต์นั้นสูญเสียการโฟกัส หรือถูกย้ายโฟกัสไปยังอิลิเมนต์อื่น	select, text, textarea
onchange=“ ”	เมื่อผู้ใช้เปลี่ยนแปลงค่าในฟอร์มรับข้อมูล	select, text, textarea
onselect=“ ”	เมื่อผู้ใช้เลือกข้อความ (ใช้เมาส์ลาก) ในช่องข้อความ	text, textarea
onsubmit=“ ”	เมื่อผู้ใช้คลิกปุ่ม submit	form

ชื่อ Event	ความหมาย	ทำงานร่วมกับแท็ก
onMouseover=“ ”	เกิดเมื่อออบเจกต์นั้นถูกเลื่อน mouse pointer ไปทับ	a,div
onMouseout=“ ”	เกิดเมื่อออบเจกต์นั้นถูกเลื่อน mouse pointer ที่ทับอยู่ออกไป	a,div
onclick=“ ”	เกิดเมื่อออบเจกต์นั้นถูกคลิก	a, button, checkbox, radio, reset, submit
onload=“ ”	เกิดเมื่อโหลดเอกสารเสร็จ	body
onunload=“ ”	เกิดเมื่อยกเลิกการโหลด เช่น คลิกปุ่ม Stop	body

EventListener

- คือเหตุการณ์หรือการกระทำบางอย่างที่เกิดขึ้นกับอิลิเมนต์ แต่รูปแบบการเขียน จะเขียนในฝั่ง JavaScript ทั้งหมด

โครงสร้าง :

```
element.addEventListener(event,callback)
```

ตัวอย่างภาษา JavaScript

```
<!DOCTYPE html>
<html>
<body>

<h1>My First JavaScript</h1>

<button type="button"
onClick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.</button>

<p id="demo"></p>

</body>
</html>
```

การประยุกต์ใช้งาน JavaScript

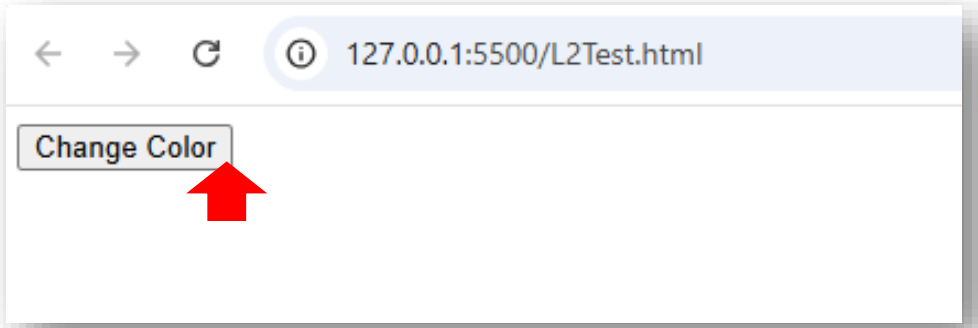
1. **Front-End Development**
   - ใช้ร่วมกับ HTML และ CSS เพื่อสร้างเว็บไซต์แบบไดนามิก
   - Framework ที่นิยม: React, Angular, Vue.js
2. **Back-End Development**
   - ใช้กับ Node.js เพื่อพัฒนา API และเซิร์ฟเวอร์
   - Framework เช่น Express.js
3. Mobile Application
   - พัฒนาแอปพลิเคชันมือถือผ่าน Framework เช่น React Native
4. Game Development
   - ใช้สร้างเกมเบราว์เซอร์ผ่าน Framework เช่น Phaser.js

ตัวอย่างภาษา JavaScript

- สร้างปุ่มที่เปลี่ยนสีเมื่อคลิก

```html
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Example</title>
</head>
<body>
    <button id="colorButton">Change Color</button>

    <script>
        const button = document.getElementById("colorButton");
        button.addEventListener("click", function() {
            document.body.style.backgroundColor = "lightblue";
        });
    </script>
</body>
</html>
```



การเปลี่ยนแปลง CSS ด้วย JavaScript

- หลักการ:

1. เข้าถึงองค์ประกอบที่ต้องการโดยใช้เมธอด เช่น getElementById, getElementsByClassName, หรือ querySelector
2. ใช้ style.propertyName เพื่อเปลี่ยนแปลงค่า CSS

ตัวอย่างคำสั่งที่ใช้เปลี่ยน CSS

javascript Copy Edit

```javascript
document.getElementById("myElement").style.color = "red"; // เปลี่ยนสีข้อความเป็นสีแดง
document.getElementById("myElement").style.backgroundColor = "yellow"; // เปลี่ยนพื้นหลังเป็นสีเ
document.getElementById("myElement").style.fontSize = "20px"; // เปลี่ยนขนาดตัวอักษร
document.getElementById("myElement").style.display = "none"; // ซ่อนองค์ประกอบ
```

ตัวอย่างโค้ด HTML + JavaScript: เปลี่ยน CSS เมื่อคลิกปุ่ม

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Change CSS with JavaScript</title>
</head>
```

```html
<body>
    <h1 id="myHeading">Welcome to JavaScript DOM</h1>
    <button id="changeStyleButton">Change Style</button>

    <script>
        // เข้าถึงปุ่มและตั้งค่าเหตุการณ์เมื่อคลิก
        document.getElementById("changeStyleButton").addEventListener("click", function()
            // เข้าถึง <h1> และเปลี่ยน CSS
            const heading = document.getElementById("myHeading");
            heading.style.color = "white";              // เปลี่ยนสีตัวอักษร
            heading.style.backgroundColor = "blue";     // เปลี่ยนพื้นหลัง
            heading.style.padding = "10px";             // เพิ่ม Padding
            heading.style.borderRadius = "5px";         // ทำมุมโค้ง
        });
    </script>
</body>
</html>
```

- document.getElementById("myHeading")
  - ใช้เพื่อเข้าถึง <h1> ที่มี id="myHeading“
- style.propertyName
  - เช่น style.color, style.backgroundColor เป็นการระบุว่าต้องการเปลี่ยนค่า CSS อะไร
- Event Handling (addEventListener)
  - ตั้งเหตุการณ์ click เพื่อให้การเปลี่ยนแปลง CSS เกิดขึ้นเมื่อผู้ใช้คลิกปุ่ม


Q&A