



# ICT12367

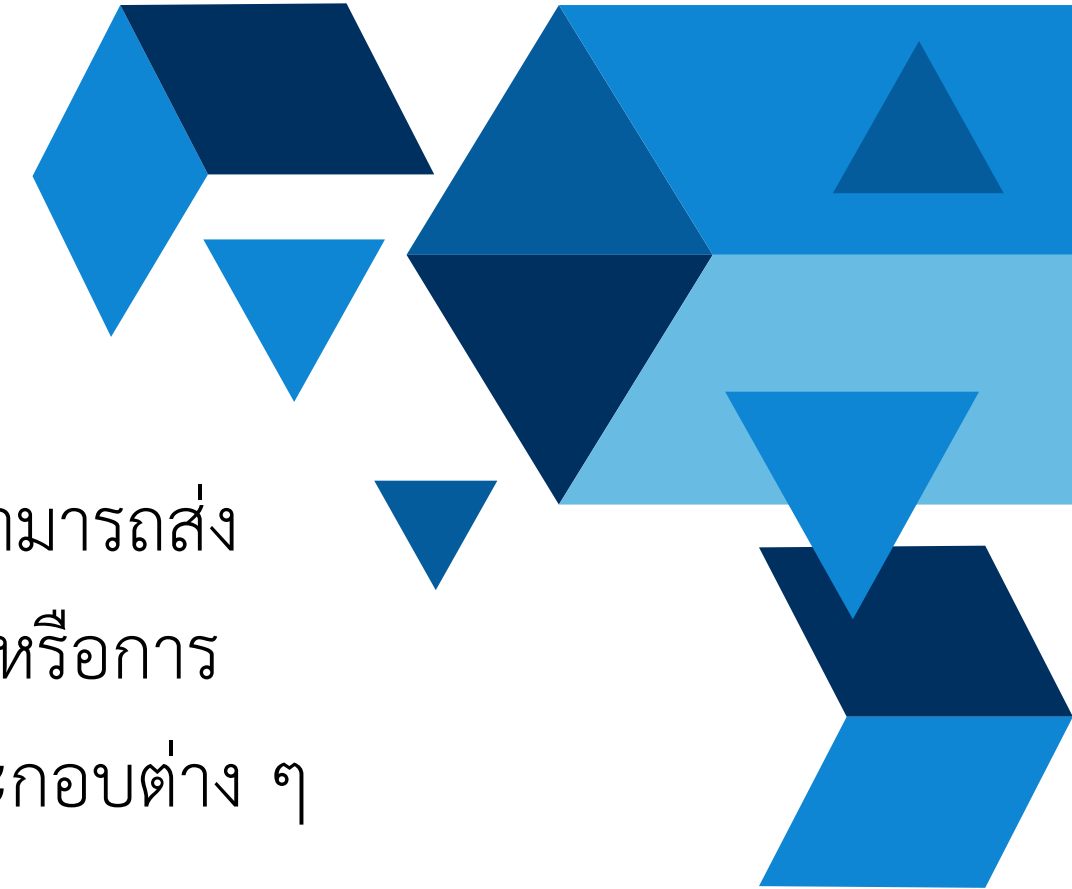
การใช้กรอบงานสำหรับการพัฒนาเว็บแอปพลิเคชัน  
เพื่อความมั่นคงปลอดภัย

## Chapter 3





# HTML FORM



HTML Form คือ องค์ประกอบ (element) ใน HTML ที่ใช้สำหรับรวบรวมข้อมูลจากผู้ใช้ และสามารถส่งข้อมูลนั้นไปยังเซิร์ฟเวอร์เพื่อการประมวลผล เช่น การกรอกแบบฟอร์มลงทะเบียน, การล็อกอิน, หรือการค้นหาในเว็บไซต์ฟอร์มใน HTML ถูกสร้างขึ้นด้วยแท็ก <form> ซึ่งทำหน้าที่เป็นตัวครอบองค์ประกอบต่าง ๆ เช่น <input>, <textarea>, <button> และ <select> เพื่อให้ผู้ใช้สามารถป้อนข้อมูลได้

**สมัคร** X  
ง่ายและเร็ว

ชื่อ  นามสกุล

หมายเลขโทรศัพท์มือถือหรืออีเมล

รหัสผ่านใหม่

วันเกิด

เพศ ☐ หญิง ☐ ชาย ☐ กำหนดเอง

คนที่ให้บริการของเราอาจใช้ข้อมูลติดต่อของคุณไปยัง Facebook เรียนรู้เพิ่มเติม

เมื่อคลิกสมัคร แสดงว่าคุณยินยอมตามข้อกำหนด นโยบายความเป็นส่วนตัว และนโยบายคุกกี้ของเรา คุณอาจได้รับการแจ้งเตือนทาง SMS จากเราและสามารถเลือกไม่รับได้ทุกเมื่อ

**สมัคร**

Google  
สร้างบัญชี Google  
ไปยัง Gmail

ชื่อ  นามสกุล

ชื่อผู้ใช้อีเมล   
คุณใช้ตัวอักษร ตัวเลข และจุดได้

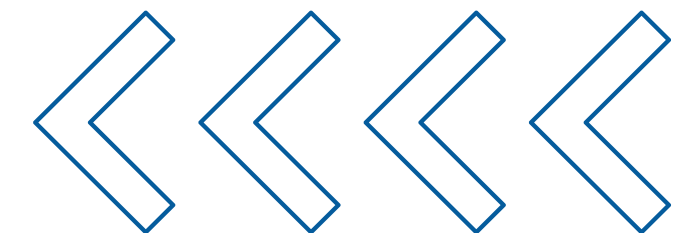
รหัสผ่าน  ยืนยัน

ใช้อักษร 8 ตัวขึ้นไปที่มีทั้งตัวอักษร ตัวเลข และสัญลักษณ์ผสมกัน

☐ แสดงรหัสผ่าน

มีเพียงบัญชีเดียวที่เข้าถึงทุกผลิตภัณฑ์และบริการของ Google ได้

ลงชื่อเข้าใช้แทน

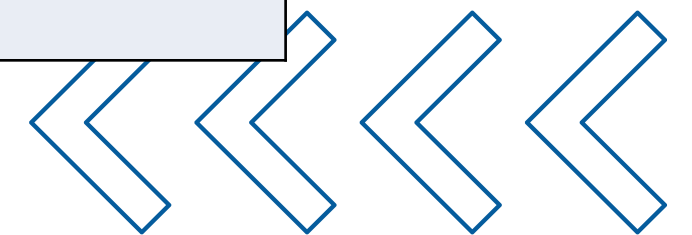




# อีลีเมนต์ในแบบ Form

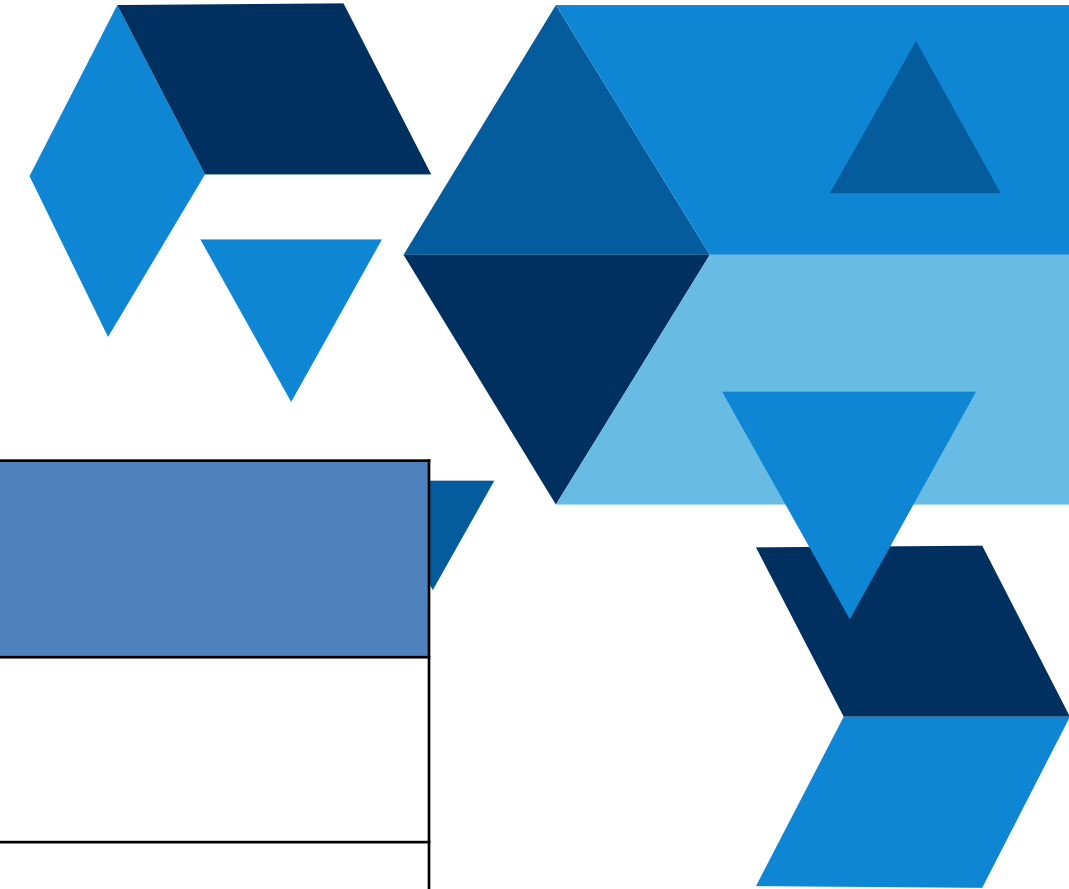


ประเภท	ทำหน้าที่
Input type="text"	รับข้อความทั่วไป
Input type="password"	รับรหัสผ่าน
Input type="number"	รับตัวเลข
Input type="tel"	เบอร์โทรศัพท์
Input type="search"	ใส่เงื่อนไขค้นหาข้อมูล

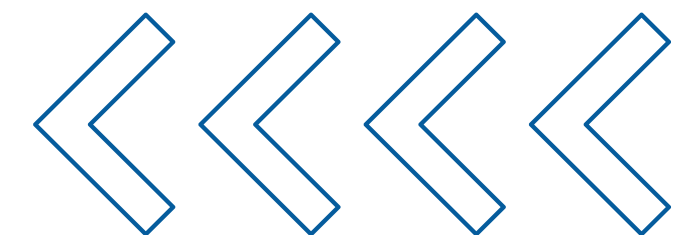




# อีลีเมนต์ในแบบ Form

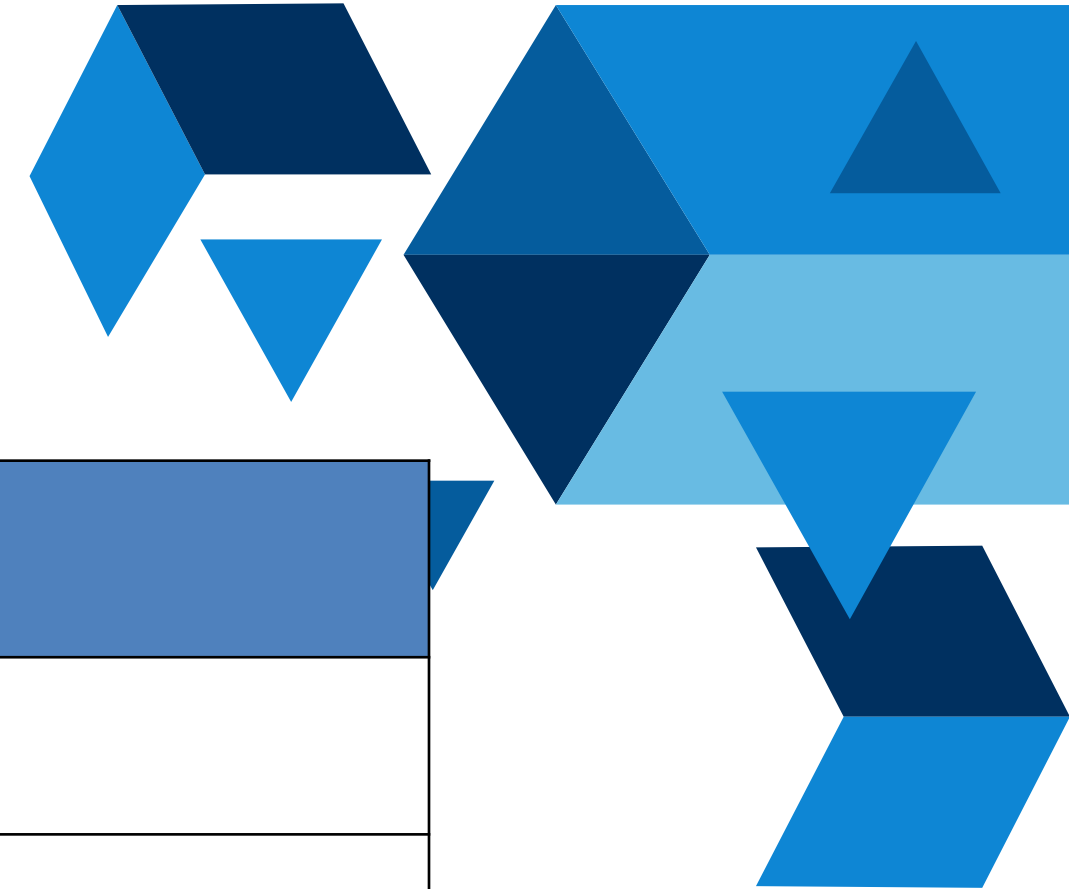


ประเภท	ทำหน้าที่
Input type="url"	รับข้อมูลที่เป็นพาร URL
Input type="email"	รับข้อมูลที่เป็น E-mail
Input type="file"	รับข้อมูลที่เป็นไฟล์จากเครื่องผู้ใช้งาน
Input type="radio"	ตัวเลือกประเภทเลือกได้อย่างเดียว
Input type="checkbox"	ตัวเลือกประเภทเลือกได้หลายตัวเลือก
Input type="range"	ตัวเลื่อนเลือกช่วงข้อมูล

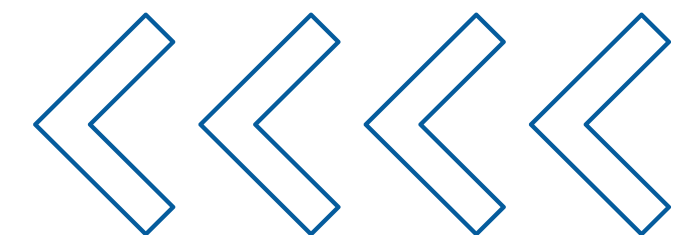




# อีลีเมนต์ในแบบ Form



ประเภท	ทำหน้าที่
Input type="url"	รับข้อมูลที่เป็นพาร URL
Input type="email"	รับข้อมูลที่เป็น E-mail
Input type="file"	รับข้อมูลที่เป็นไฟล์จากเครื่องผู้ใช้งาน
Input type="radio"	ตัวเลือกประเภทเลือกได้อย่างเดียว
Input type="checkbox"	ตัวเลือกประเภทเลือกได้หลายตัวเลือก
Input type="range"	ตัวเลื่อนเลือกช่วงข้อมูล





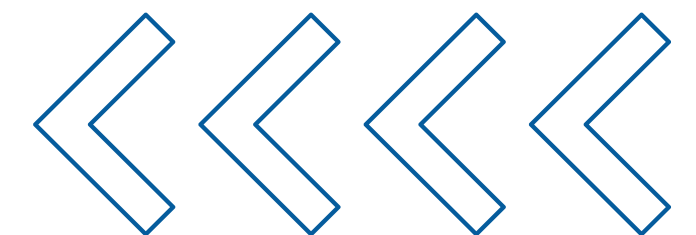
# การเขียน comment (CSS)



- ส่วนที่ใช้ในการอธิบาย Code จะช่วยให้สามารถเข้าใจและ สามารถแก้ไข code ได้ในภายหลัง

- รูปแบบ

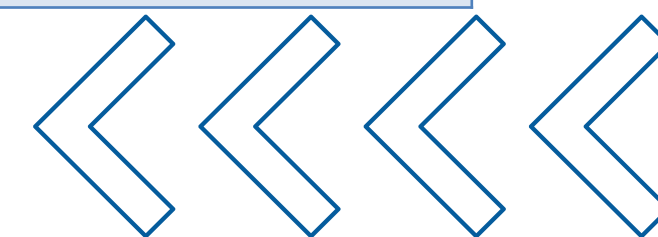
```
/* ข้อความอธิบายโค้ด */
```

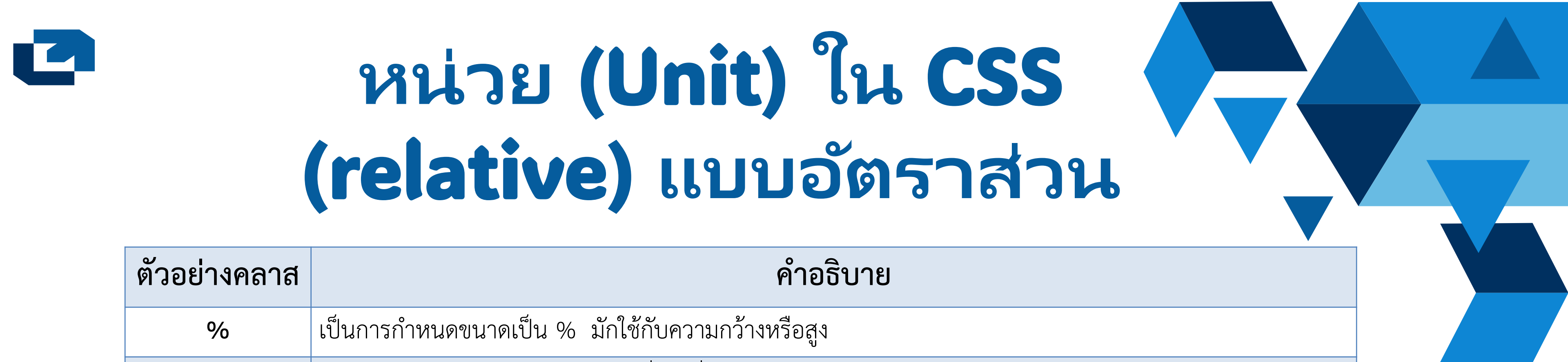


# หน่วย (Unit) ใน CSS (Absolute)

## แบบตายตัว

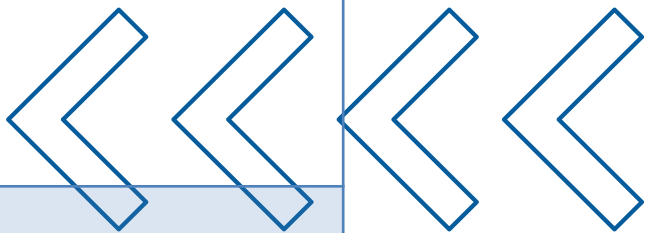
ตัวอย่างคลาส	คำอธิบาย
px	Pixel เป็นหน่วยที่นิยมใช้มากที่สุด โดยที่ $1\text{px} = 0.75\text{ pt}$ สัมพันธ์กับรายละเอียดหน้าจอ
pt	Point โดยที่ $1\text{ pt} = 1/72\text{ inches}$ ใช้ในงานสิ่งพิมพ์
cm	เซนติเมตร
mm	มิลลิเมตร
in	Inches ( $1\text{ in} = 96\text{px} = 2.54\text{ cm}$ )
pc	Picas ( $1\text{ px} = 12\text{ pt}$ )





# หน่วย (Unit) ใน CSS (relative) แบบอัตราส่วน

ตัวอย่างคลาส	คำอธิบาย
%	เป็นการกำหนดขนาดเป็น % มักใช้กับความกว้างหรือสูง
em	อ้างอิงขนาดกับ element parent ที่ใกล้ที่สุด ใช้ในการกำหนดขนาดจำนวนเท่าของขนาดปัจจุบัน เช่น หากขนาดที่ใช้ตอนนี้คือ 10px <ul style="list-style-type: none"><li>- ถ้ากำหนดขนาดเป็น 2em จะหมายถึง 2 เท่าของขนาดปัจจุบัน คือ 20px</li><li>- ถ้ากำหนดขนาดเป็น 1.4em จะหมายถึง 1.4 เท่าของของปัจจุบัน คือ14px</li><li>- ถ้าเรากำหนดเป็นขนาดเดิมก็กำหนดเป็น 1em = 10px</li></ul>
rem	กำหนดขนาดโดยอ้างอิงกับ root element ปกติ font-size จะอยู่ที่ 16px
vm	1% หรือ 1/100 ของ viewport width <ul style="list-style-type: none"><li>- width ของ browser viewpont เท่ากับ 750px ค่า 1vw = 7.5px</li></ul>
vh	1% หรือ 1/100 ของ viewport height <ul style="list-style-type: none"><li>- height ของ browser viewpont เท่ากับ 900px ค่า 1vw = 9px</li></ul>
vmin, vmax	กำหนดค่าต่ำสุด และ สูงสุดของ viewport <b>8</b>



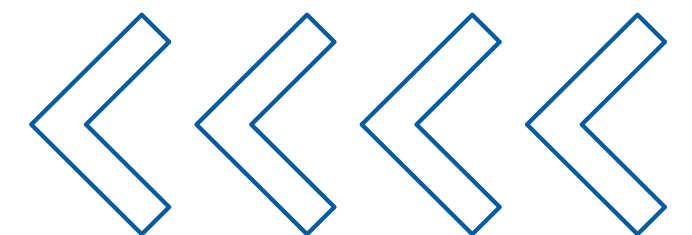




# รูปแบบตัวอักษร

- **Font Family** เป็นการกำหนดค่ารูปแบบของตัวอักษร [font family: ..... ;] และถ้า font family มีมากกว่าหนึ่งค่า ให้ใช้เครื่องหมาย quotation marks (,)

ตัวอย่างคลาส	คำอธิบาย
font-family	กำหนดรูปแบบตัวอักษร
font-size	กำหนดขนาดตัวอักษร
font-style	กำหนดลักษณะตัวอักษร
font-variant	กำหนดตัวแรกจะมีขนาดใหญ่
font-weight	กำหนดน้ำหนักตัวอักษร

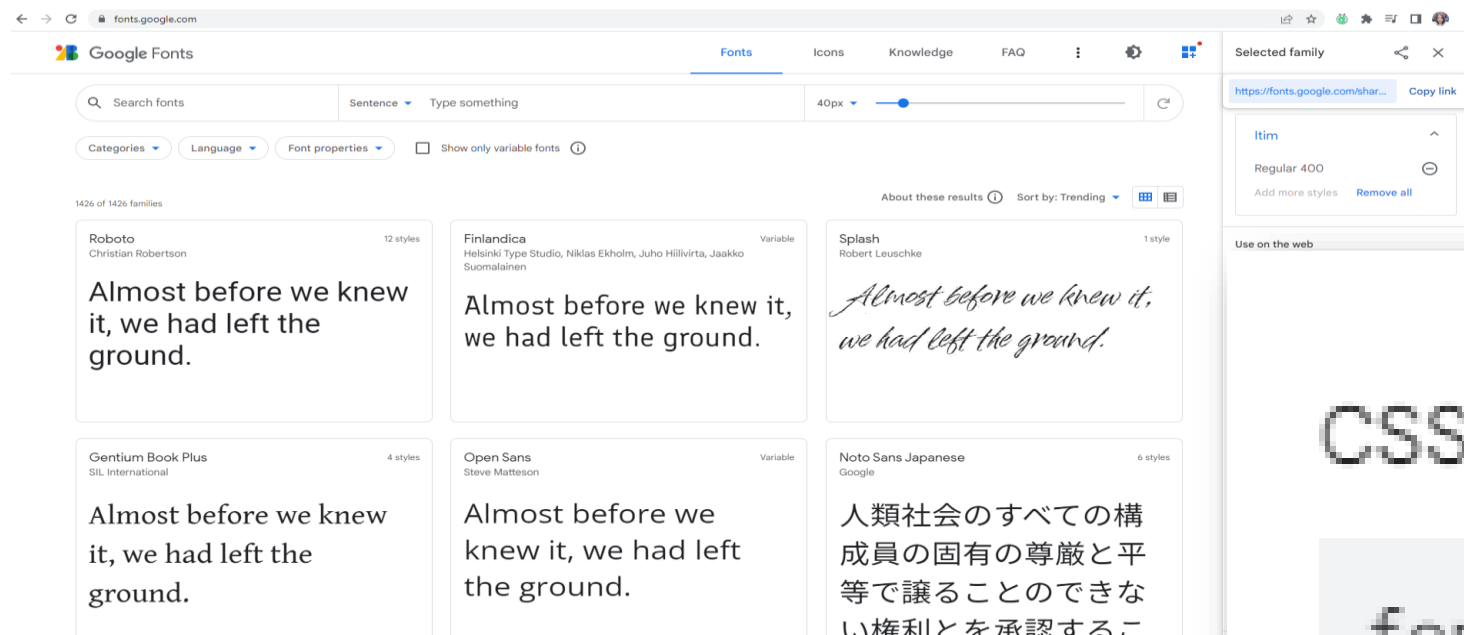




# การใช้งาน google font แบบ link



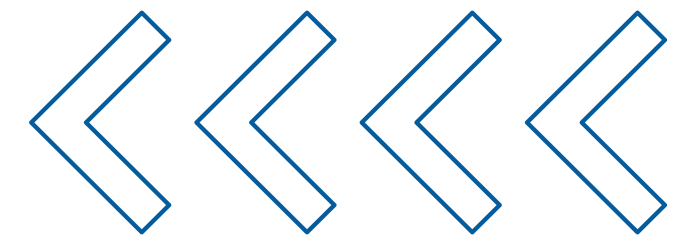
- พิมพ์ [fonts.google.com](https://fonts.google.com)



CSS rules to specify families

```
font-family: 'Itim', cursive;
```

```
<style type="text/css">
*{
    font-family: 'Itim', cursive;
}
</style>
```

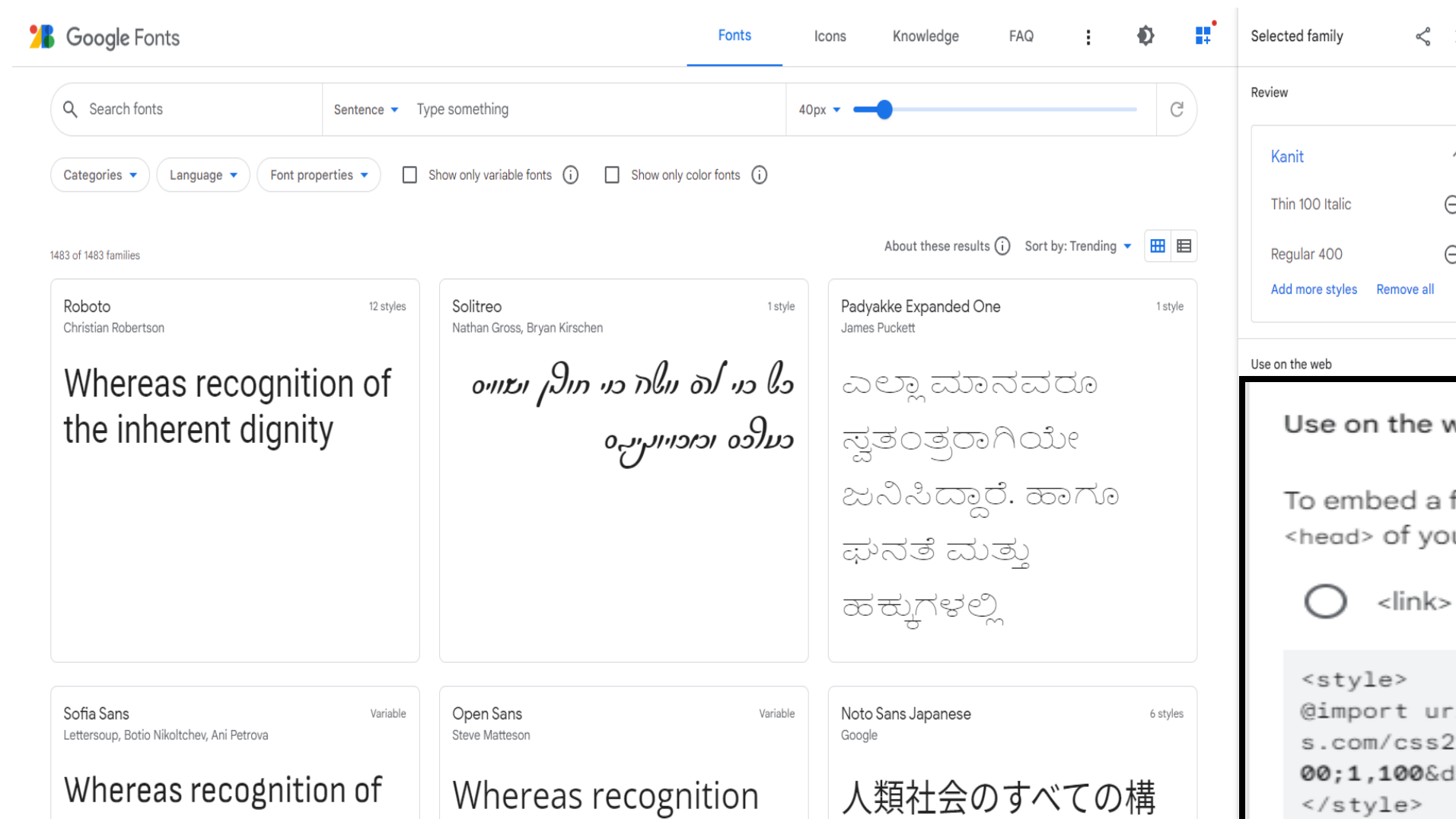




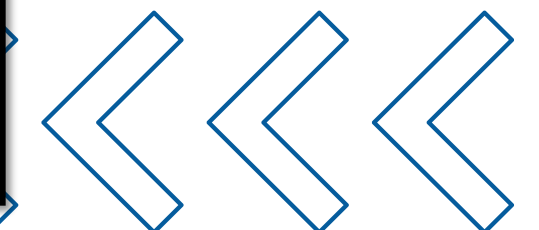
# การใช้งาน google font แบบ @import



- พิมพ์ fonts.google.com



```
<style>
@import url('https://fonts.googleapis.com/css2?family=Kanit:ital,wght@0,400;1,100&display=swap');
</style>
```





# รูปแบบการใช้งาน Font Family

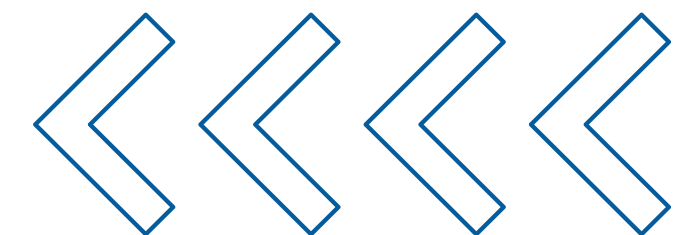


CSS rules to specify families

```
font-family: 'Kanit', sans-serif;
```



```
* {  
    font-family: 'Kanit', sans-serif;  
}
```



# พื้นฐาน JavaScript สำหรับเว็บ

- JavaScript (JS) เป็นภาษาโปรแกรมที่ใช้สำหรับเพิ่มความสามารถแบบโต้ตอบ (interactivity) บนเว็บไซต์ เช่น การตรวจสอบฟอร์ม การเปลี่ยนแปลง DOM แบบไดนามิก หรือการสร้างเอฟเฟกต์ต่าง ๆ เพื่อพัฒนาเว็บให้มีประสิทธิภาพและน่าสนใจมากขึ้น

## 1. การแทรก JavaScript ในเว็บเพจ

JavaScript สามารถเพิ่มเข้าไปใน HTML ได้ 3 รูปแบบ:

### 1.1 ในแท็ก `<script>`

- แทรก JavaScript ไว้ในไฟล์ HTML โดยตรง

```
<script>  
  console.log("Hello, JavaScript!");  
</script>
```





# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)



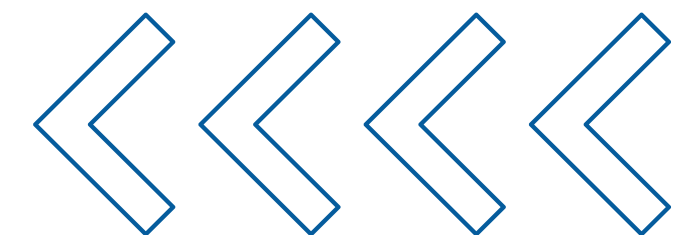
## 1. การแทรก JavaScript ในเว็บเพจ

JavaScript สามารถเพิ่มเข้าไปใน HTML ได้ 3 รูปแบบ:

### 1.2 ในไฟล์ภายนอก

- สร้างไฟล์ .js และลิงก์เข้ากับ HTML

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





# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)



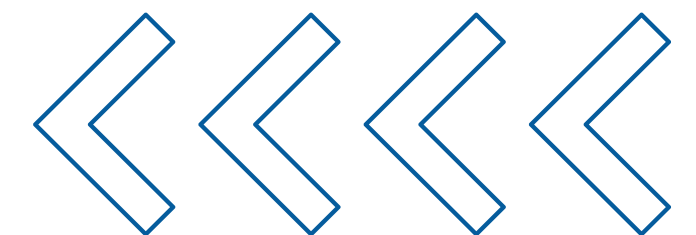
## 1. การแทรก JavaScript ในเว็บเพจ

JavaScript สามารถเพิ่มเข้าไปใน HTML ได้ 3 รูปแบบ:

### 1.3 แทรกในแท็ก HTML โดยตรง (ไม่แนะนำ)

- ใช้ในคุณสมบัติ onclick, onmouseover เป็นต้นhtmlCopyEdit

```
<button onclick="alert('Button clicked!')">Click me</button>
```

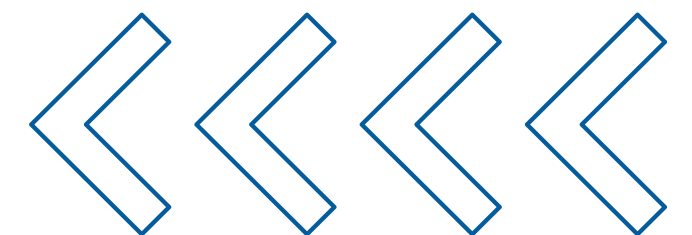


# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)

## 2. การใช้งานพื้นฐานใน JavaScript

### 2.1 การประกาศตัวแปร

```
let name = "John"; // ตัวแปรที่เปลี่ยนค่าได้  
const age = 25;    // ค่าคงที่ เปลี่ยนไม่ได้  
// แบบเก่า  
var country = "Thailand"; // ไม่แนะนำ
```



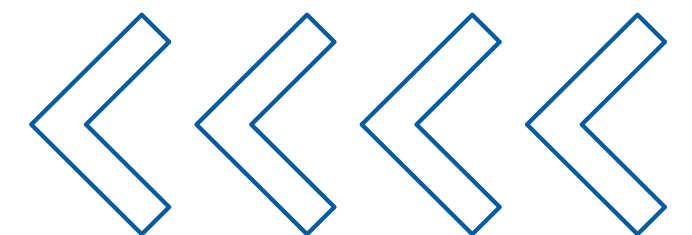


# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)

## 2. การใช้งานพื้นฐานใน JavaScript

### 2.2 การแสดงผล

```
console.log("Hello, Console!"); // แสดงข้อความใน console  
alert("Welcome!");              // แสดงกล่องข้อความ  
document.write("Hello, World!"); // แสดงในหน้าเว็บ
```





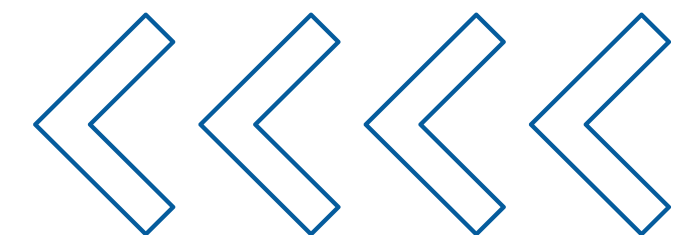
# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)



## 2. การใช้งานพื้นฐานใน JavaScript

### 2.3 เงื่อนไข

```
if (age > 18) {  
    console.log("You are an adult.");  
} else {  
    console.log("You are underage.");  
}
```





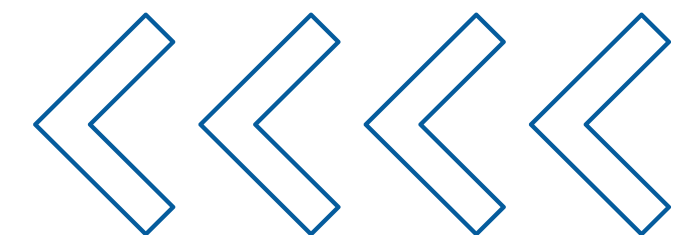
# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)



## 2. การใช้งานพื้นฐานใน JavaScript

### 2.4 การวนลูป

```
for (let i = 0; i < 5; i++) {  
  console.log("Number:", i);  
}
```

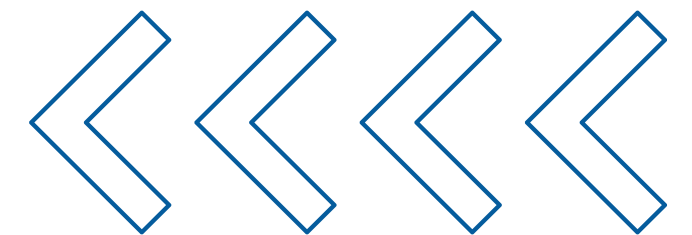


# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)

3. การจัดการ DOM (Document Object Model) JavaScript ใช้จัดการเนื้อหาและโครงสร้างของ HTML ได้โดยตรง

## 3.1 การเลือกองค์ประกอบ

```
let element = document.getElementById("myElement"); // เลือกด้วย id
let items = document.getElementsByClassName("myClass"); // เลือกด้วย class
let buttons = document.querySelectorAll("button"); // เลือกด้วย CSS Selector
```

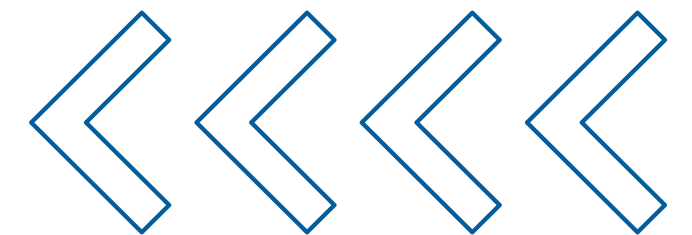


# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)

3. การจัดการ DOM (Document Object Model) JavaScript ใช้จัดการเนื้อหาและโครงสร้างของ HTML ได้โดยตรง

## 3.2 การเปลี่ยนแปลงเนื้อหา

```
document.getElementById("title").innerText = "Hello, JavaScript!";
```





# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)

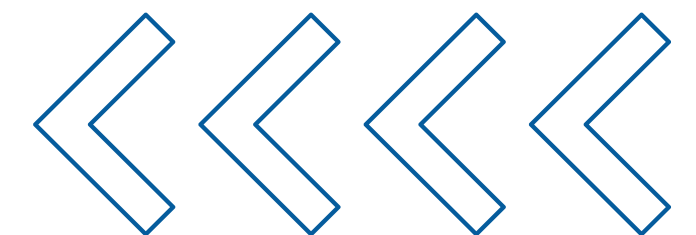


## 4. การจัดการ Event

JavaScript ใช้ Event เพื่อจัดการการกระทำของผู้ใช้ เช่น การคลิก หรือการพิมพ์

```
<button id="myButton">Click me</button>

<script>
  document.getElementById("myButton").addEventListener("click", function() {
    alert("Button clicked!");
  });
</script>
```





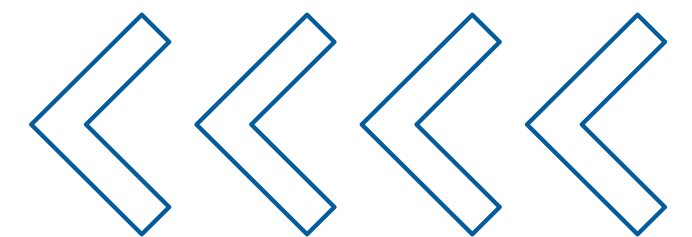
# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)



## 5. การใช้งานฟังก์ชัน

ฟังก์ชันคือกลุ่มของคำสั่งที่สามารถเรียกใช้งานได้

```
function greet(name) {  
    return "Hello, " + name;  
}  
  
let message = greet("John");  
console.log(message); // Hello, John
```



# พื้นฐาน JavaScript สำหรับเว็บ (ต่อ)

## ตัวอย่างการใช้งานพื้นฐาน

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScript Basics</title>
  <style>
    #box { width: 100px; height: 100px; background: red; }
  </style>
</head>
<body>
  <h1 id="title">Hello, World!</h1>
  <button id="changeText">Change Text</button>
  <div id="box"></div>
```

```
<script>
  // การเปลี่ยนข้อความ
  document.getElementById("changeText").addEventListener("click", function() {
    document.getElementById("title").innerText = "Hello, JavaScript!";
  });

  // การเปลี่ยนสี
  document.getElementById("box").addEventListener("mouseover", function() {
    this.style.backgroundColor = "blue";
  });
  document.getElementById("box").addEventListener("mouseout", function() {
    this.style.backgroundColor = "red";
  });
</script>
</body>
</html>
```





# เหตุผลที่ต้องใช้ JavaScript



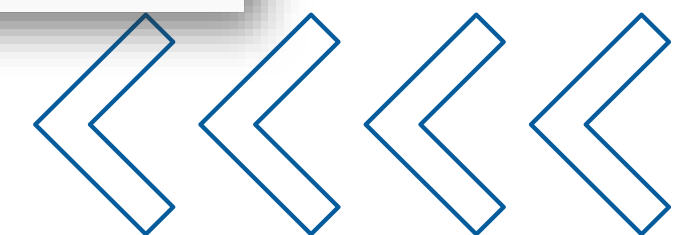
เหตุผลที่ต้องใช้ JavaScript เป็นหนึ่งในภาษาที่สำคัญที่สุดสำหรับการพัฒนาเว็บ เนื่องจากความสามารถที่หลากหลายและเป็นที่ยอมรับอย่างกว้างขวางในหมู่นักพัฒนาเว็บ

## 1. การเพิ่มความสามารถแบบโต้ตอบ (Interactivity)

JavaScript ช่วยเพิ่มการโต้ตอบกับผู้ใช้ในเว็บไซต์ เช่น:

- การตรวจสอบฟอร์มก่อนส่งข้อมูล
- การคลิกปุ่มเพื่อแสดง/ซ่อนเนื้อหา
- การสร้างแอนิเมชันหรือเอฟเฟกต์ที่สวยงาม

```
document.getElementById("btn").addEventListener("click", function() {  
    alert("Button clicked!");  
});
```





# เหตุผลที่ต้องใช้ JavaScript (ต่อ)

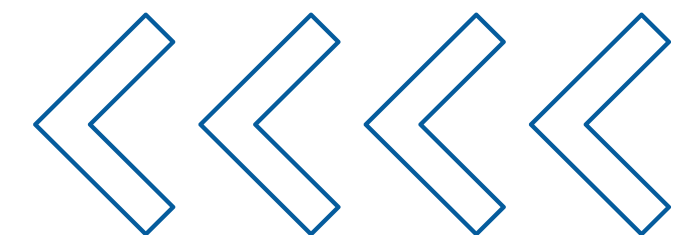


## 2. การจัดการ DOM (Document Object Model)

JavaScript สามารถปรับเปลี่ยนโครงสร้าง HTML และ CSS แบบไดนามิก:

- การเพิ่ม/ลบ/แก้ไของค์ประกอบในหน้าเว็บ
- การเปลี่ยนสไตล์ขององค์ประกอบ

```
document.getElementById("title").innerText = "New Title!";
```





# เหตุผลที่ต้องใช้ JavaScript (ต่อ)

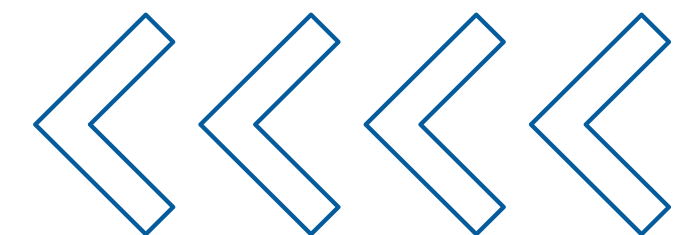


## 3. รองรับการทำงานฝั่งไคลเอนต์ (Client-Side)

JavaScript ทำงานในเบราว์เซอร์ของผู้ใช้:

- ลดภาระของเซิร์ฟเวอร์ เพราะสามารถตรวจสอบข้อมูลหรือแสดงผลเบื้องต้นได้ในฝั่งไคลเอนต์
- เพิ่มความเร็วในการตอบสนอง (Response Time)

```
if (document.getElementById("email").value === "") {  
    alert("Please fill in your email.");  
}
```





# เหตุผลที่ต้องใช้ JavaScript (ต่อ)

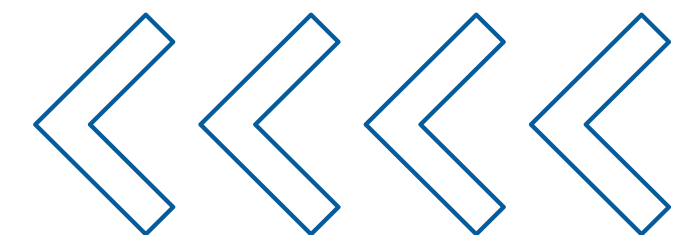
## 5. รองรับเทคโนโลยีสมัยใหม่

JavaScript เป็นพื้นฐานสำหรับการพัฒนาเทคโนโลยีเว็บยุคใหม่ เช่น::

- Frontend Frameworks: React, Angular, Vue.js
- Backend Frameworks: Node.js (รัน JavaScript ในฝั่งเซิร์ฟเวอร์)
- Progressive Web Apps (PWA): สร้างแอปพลิเคชันเว็บที่ให้ประสบการณ์เหมือนแอปบนมือถือ

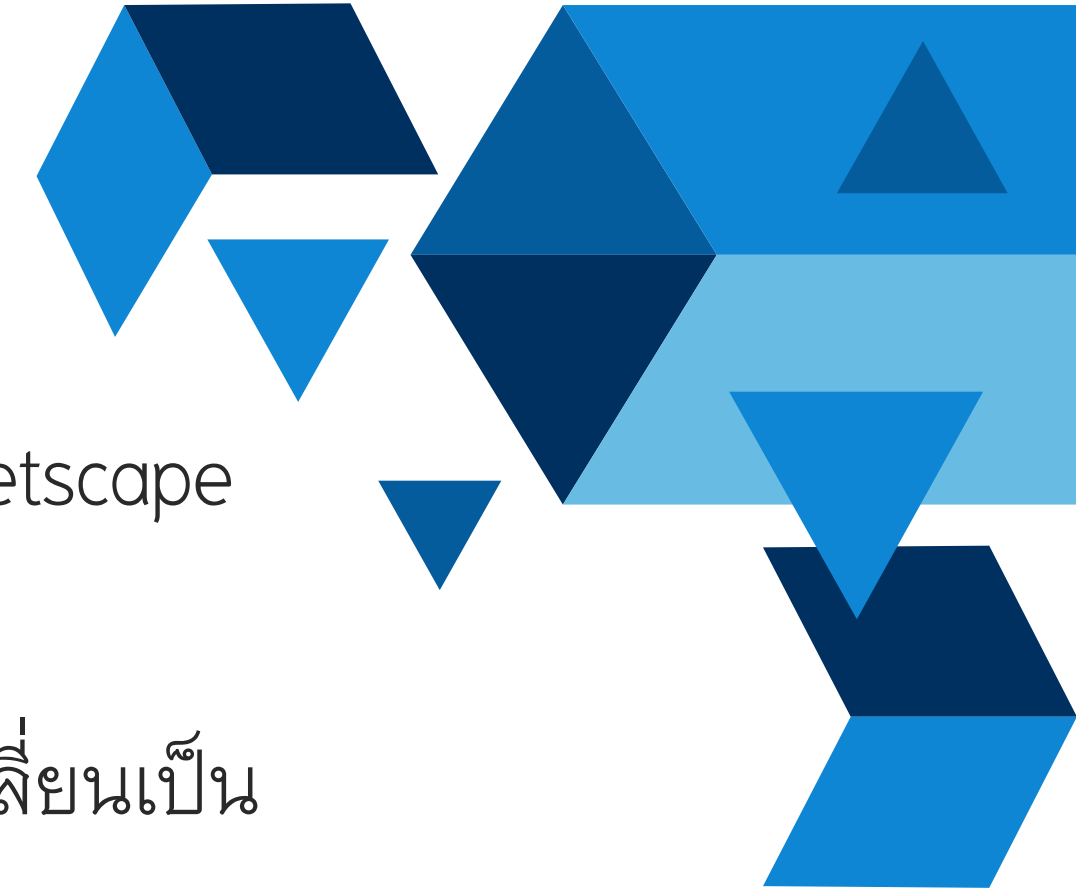
## 6. ใช้ได้กับทุกแพลตฟอร์ม

JavaScript ทำงานได้กับทุกเบราว์เซอร์ เช่น Chrome, Firefox, Safari และ Edge โดยไม่ต้องติดตั้งโปรแกรมเพิ่มเติม

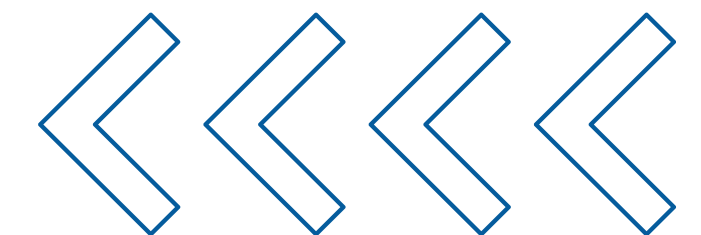




# ที่มาของ JavaScript



- ❑ **ผู้สร้าง:** JavaScript ถูกสร้างโดย Brendan Eich ในปี 1995 ขณะทำงานที่ Netscape Communications Corporation
- ❑ **จุดเริ่มต้น:** เดิมถูกเรียกว่า Mocha และเปลี่ยนชื่อเป็น LiveScript ก่อนจะเปลี่ยนเป็น JavaScript
- ❑ **แรงบันดาลใจ:** ออกแบบมาเพื่อเป็นภาษาที่ง่ายสำหรับนักพัฒนาเว็บที่ต้องการเพิ่มความสามารถแบบไดนามิกให้เว็บไซต์
- ❑ **การพัฒนาและมาตรฐาน:**
  - JavaScript กลายเป็นมาตรฐานขององค์กร ECMA International โดยรู้จักในชื่อ ECMAScript
  - รุ่นแรก (ECMAScript 1) เปิดตัวในปี 1997 และพัฒนาต่อเนื่องจนถึงเวอร์ชันปัจจุบัน

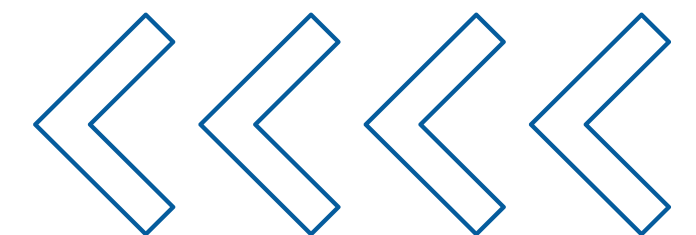
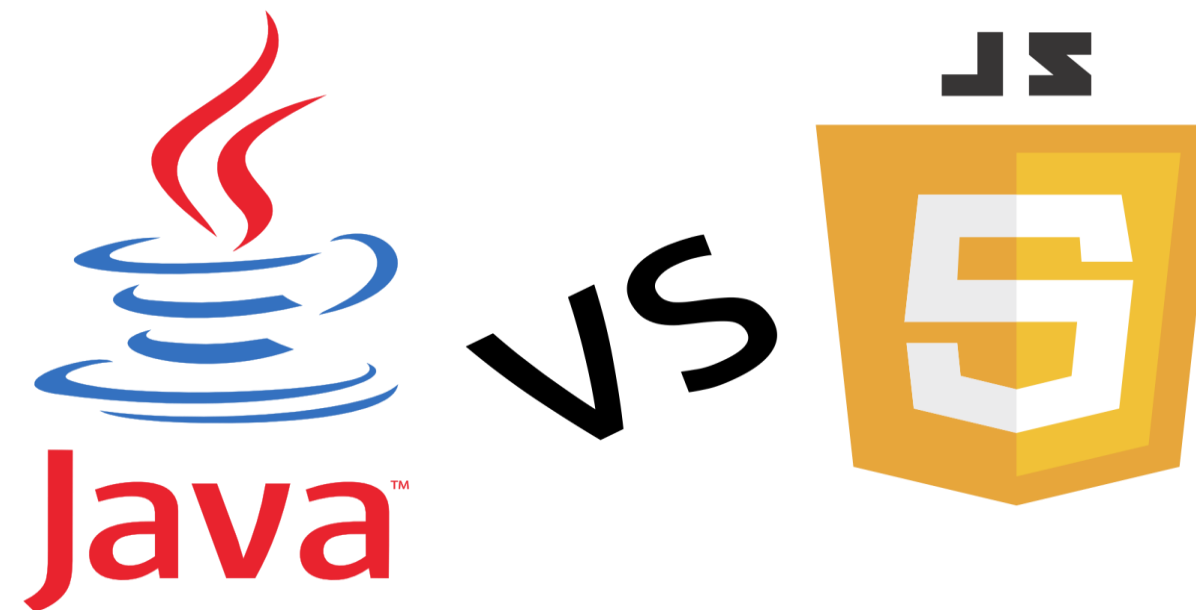




# ความเข้าใจผิดเกี่ยวกับชื่อ JavaScript

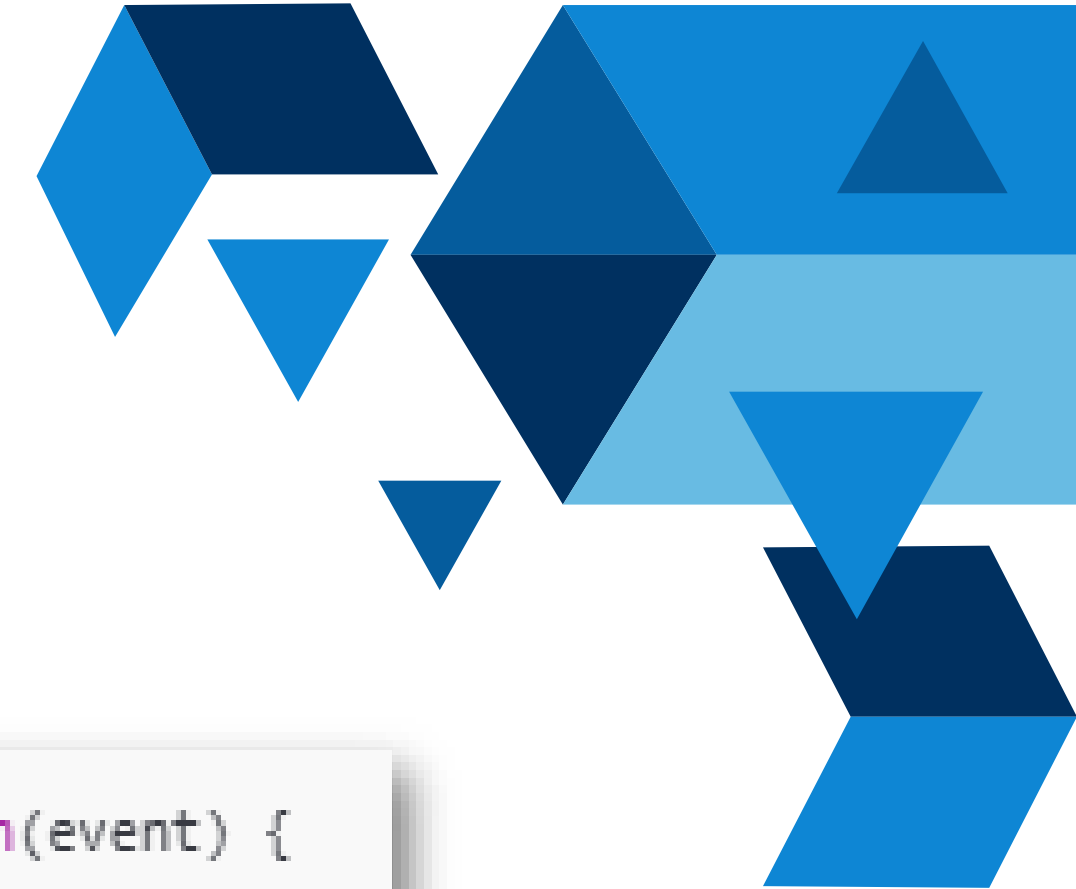


- แม้ชื่อ JavaScript จะมีคำว่า "Java" แต่ไม่มีความเกี่ยวข้องกับภาษา Java โดยตรง การตั้งชื่อนี้เป็นกลยุทธ์การตลาดในยุคที่ Java กำลังเป็นที่นิยม



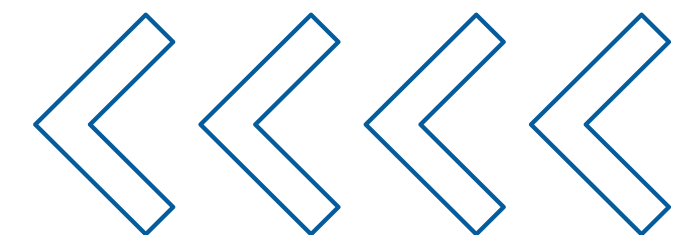


# ตัวอย่างการใช้งานจริง



## 1. ฟอรั่มตรวจสอบข้อมูล

```
document.getElementById("form").addEventListener("submit", function(event) {  
    let email = document.getElementById("email").value;  
    if (!email.includes("@")) {  
        event.preventDefault();  
        alert("Please enter a valid email.");  
    }  
});
```



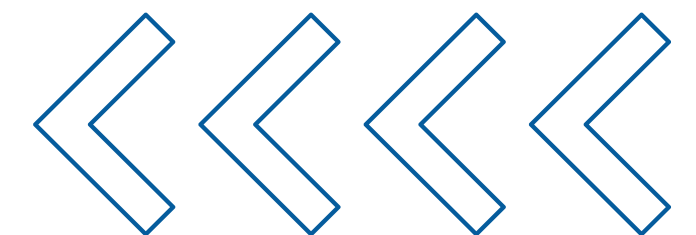


# ตัวอย่างการใช้งานจริง (ต่อ)



## 2. เพิ่มการโต้ตอบในหน้าเว็บ

```
document.getElementById("button").addEventListener("mouseover", function() {  
    this.style.backgroundColor = "blue";  
});
```

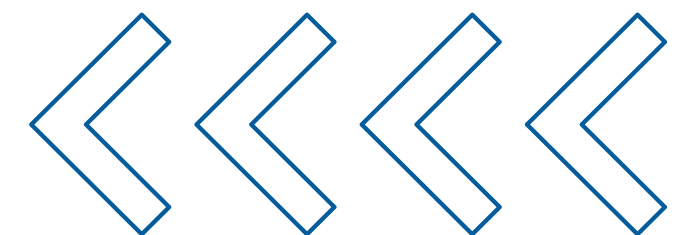






# เหตุผลสำคัญที่ต้องใช้ JavaScript

- เพิ่มความโต้ตอบในเว็บไซต์
- ลดการพึ่งพาเซิร์ฟเวอร์และเพิ่มความเร็ว
- รองรับเทคโนโลยีเว็บยุคใหม่และการพัฒนาเรียลไทม์
- มีความยืดหยุ่นและใช้งานได้หลากหลาย
- ช่วยสร้างเว็บไซต์ที่ทันสมัยและมีประสิทธิภาพ





Q&A