

# ICT12367

การใช้กรอบงานสำหรับการพัฒนาเว็บแอปพลิเคชัน  
เพื่อความมั่นคงปลอดภัย

## Chapter 2





# HTML



❑ **HTML (Hypertext Markup Language)** เป็นภาษาที่ใช้สำหรับสร้างเว็บเพจ มีโครงสร้างภาษาโดยใช้ตัวกำกับ (Markup Tag) เพื่อควบคุมการแสดงผลข้อมูล รูปภาพ และวัตถุอื่นๆ ผ่านทาง Web Browser เช่น Google Chrome, Firefox, Safari, Microsoft Edge เป็นต้น ในแต่ละ Tag จะมีส่วนที่เรียกว่า **Attribute** เพื่อควบคุมการทำงานของ Tag แต่ละตัว

❑ **การสร้างไฟล์ HTML** จะต้องอาศัย Text Editor เพื่อใช้สำหรับเขียนคำสั่งต่างๆ ที่ต้องการแสดงผลทางจอภาพ/เว็บเบราว์เซอร์ และเก็บเป็นไฟล์โดยมีนามสกุล .Html

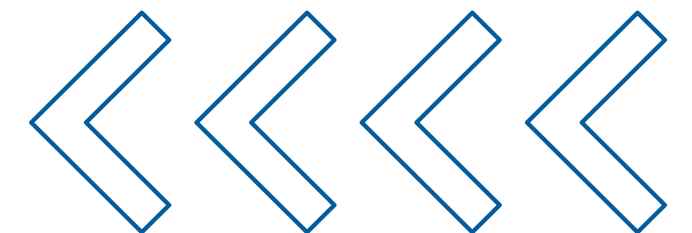


# มาตรฐานของภาษา HTML

❑ มาตรฐานของภาษา HTML (HTML Standards) เป็นสิ่งสำคัญที่ช่วยให้นักพัฒนาเว็บสร้างเว็บไซต์และเว็บแอปพลิเคชันที่ทำงานได้อย่างเหมาะสมในทุกเบราว์เซอร์และอุปกรณ์ มาตรฐานเหล่านี้ถูกกำหนดและดูแลโดย World Wide Web Consortium (W3C) และหน่วยงานอื่นๆ เช่น WHATWG ซึ่งมีหน้าที่พัฒนาและปรับปรุง HTML อย่างต่อเนื่อง

- เวอร์ชันของ HTML และมาตรฐาน

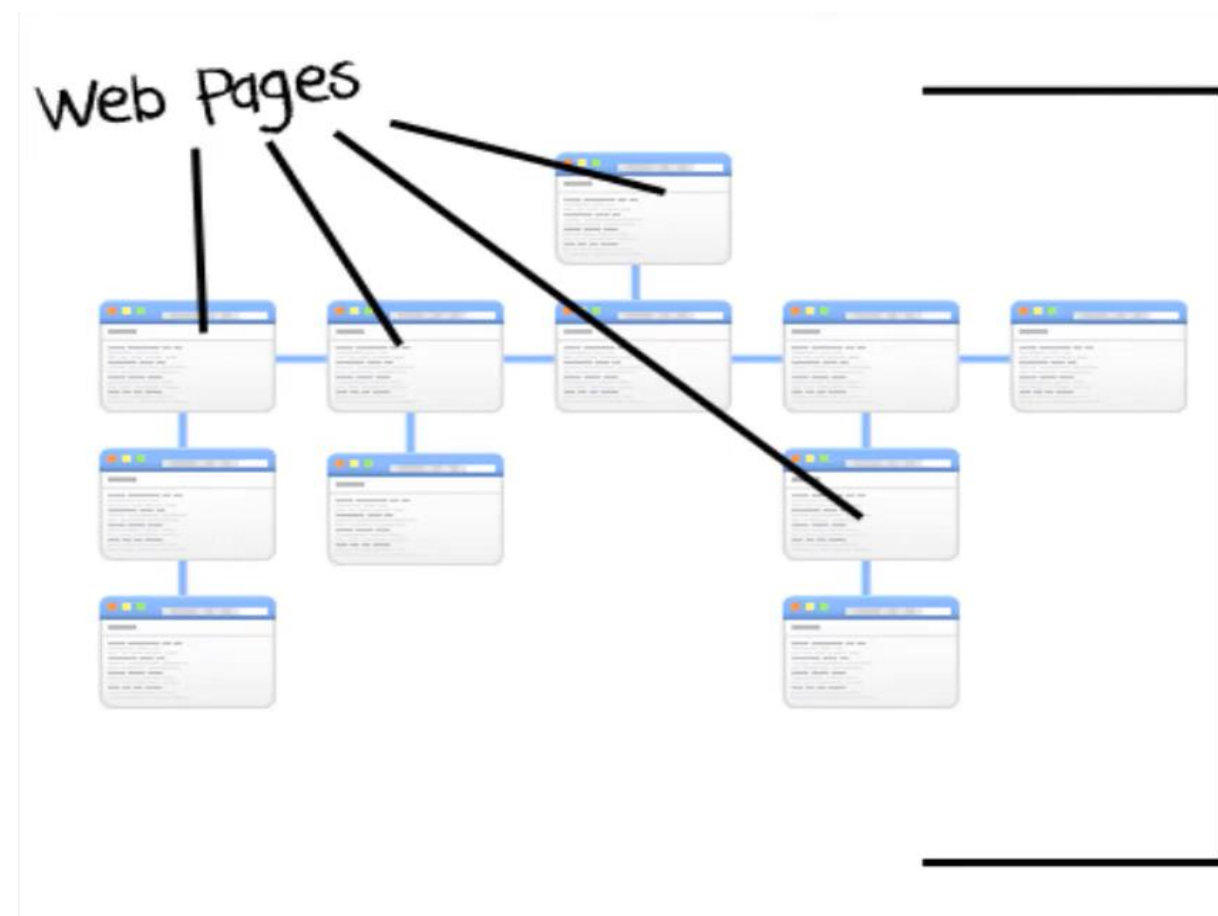
- HTML 1.0 (1993) เป็นเวอร์ชันแรกสุดของ HTML ซึ่งใช้สำหรับโครงสร้างพื้นฐานของเว็บเพจในยุคเริ่มต้น
- HTML 2.0 (1995) เพิ่มคุณสมบัติใหม่ๆ เช่น ฟอรัมและตาราง เพื่อรองรับเว็บไซต์ที่ซับซ้อนมากขึ้น
- HTML 3.2 (1997) เพิ่มการรองรับ CSS และการจัดรูปแบบเว็บเพจด้วยสไตลชีท
- HTML 4.01 (1999) มุ่งเน้นการจัดโครงสร้างเอกสารและรองรับหลายภาษา รวมถึงการใช้งานร่วมกับ XHTML
- HTML5 (2014) ปัจจุบัน HTML5 เป็นมาตรฐานที่ได้รับการใช้งานมากที่สุด โดยเน้น: การสนับสนุนมัลติมีเดีย (เสียง, วิดีโอ) การพัฒนา API ใหม่ (Geolocation, Web Storage) การเพิ่มแท็กโครงสร้าง เช่น `<header>`, `<footer>`, `<section>`, `<article>`





# Web Pages กับ Website ต่างกันอย่างไร

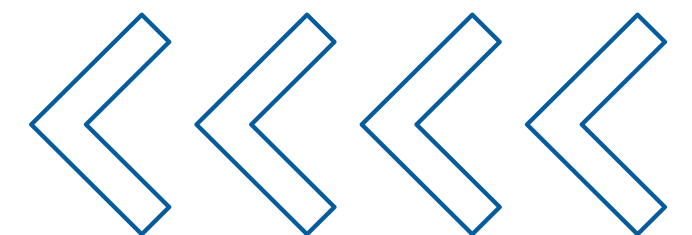
โครงสร้างของเว็บไซต์ ซึ่งประกอบด้วย หน้าเว็บ (Web Pages) หลายหน้าที่เชื่อมโยงกันและทำงานร่วมกันเพื่อสร้าง เว็บไซต์ (Website)



**Web Pages (หน้าเว็บ)** คือหน้าเดี่ยวๆ แต่ละหน้าที่แสดงเนื้อหาเฉพาะเรื่อง เปรียบเหมือนหน้ากระดาษแต่ละหน้าในหนังสือ ประกอบด้วย HTML, รูปภาพ, ข้อความ และองค์ประกอบอื่นๆ เฉพาะหน้านั้น ตัวอย่างเช่น หน้าหลัก, หน้าติดต่อเรา, หน้าสินค้า เป็นต้น

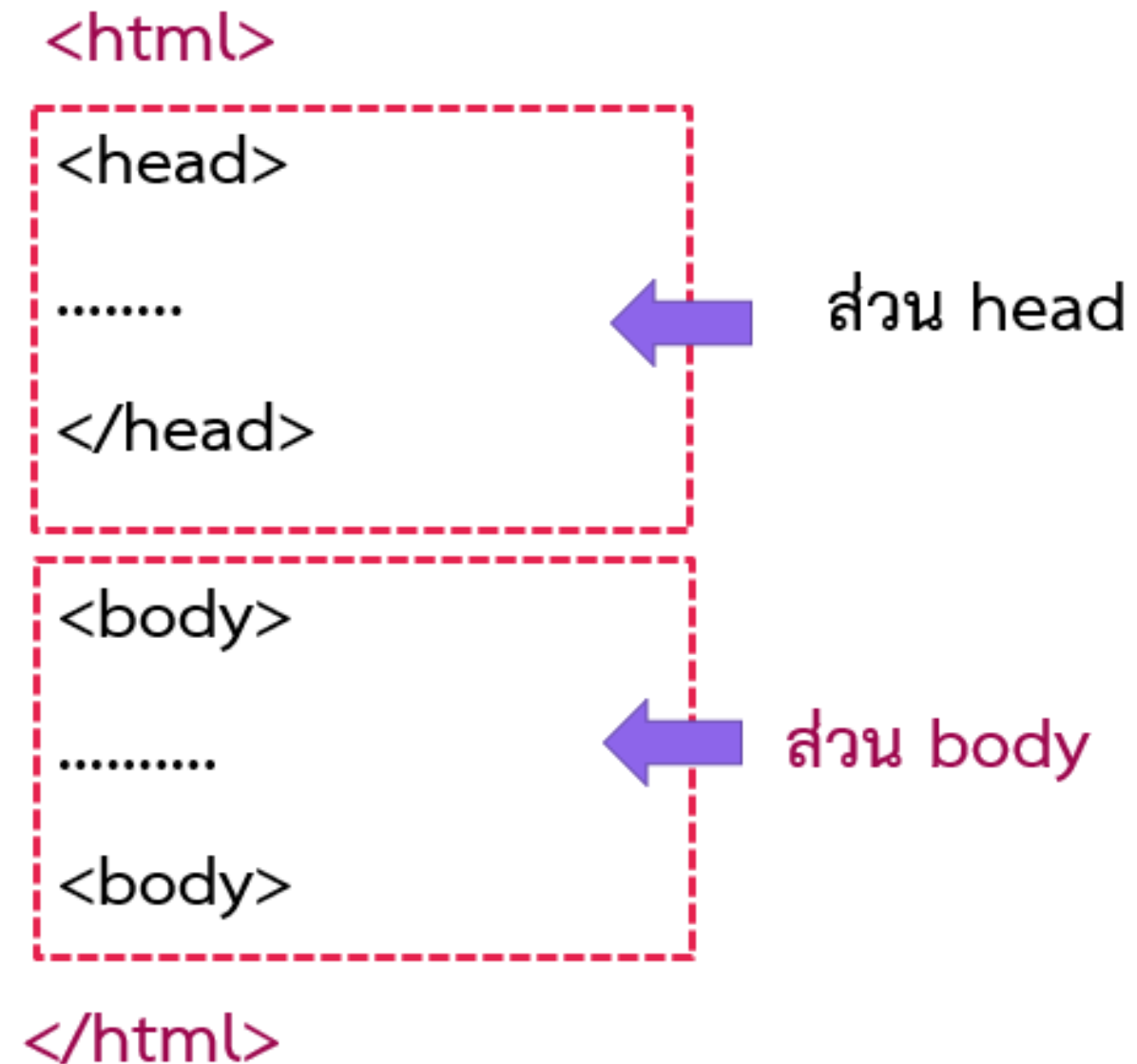
**website**

**Website (เว็บไซต์)** คือการรวมกลุ่มของ Web Pages หลายๆ หน้าเข้าด้วยกัน เปรียบเหมือนหนังสือทั้งเล่มที่มีหลายหน้า มีโดเมนเนม (Domain Name) เป็นที่อยู่หลัก มีระบบนำทาง (Navigation) เชื่อมโยงระหว่างหน้าต่างๆ มีธีม (Theme) และการออกแบบที่เป็นเอกภาพตลอดทั้งเว็บไซต์





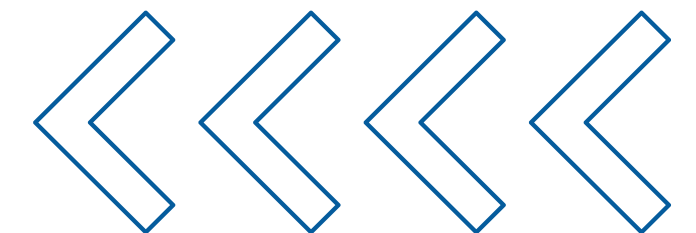
# โครงสร้างภาษา HTML



โครงสร้าง HTML จะแบ่งออกเป็น 2 ส่วน ได้แก่ ส่วน **head** และ ส่วนของ **body**

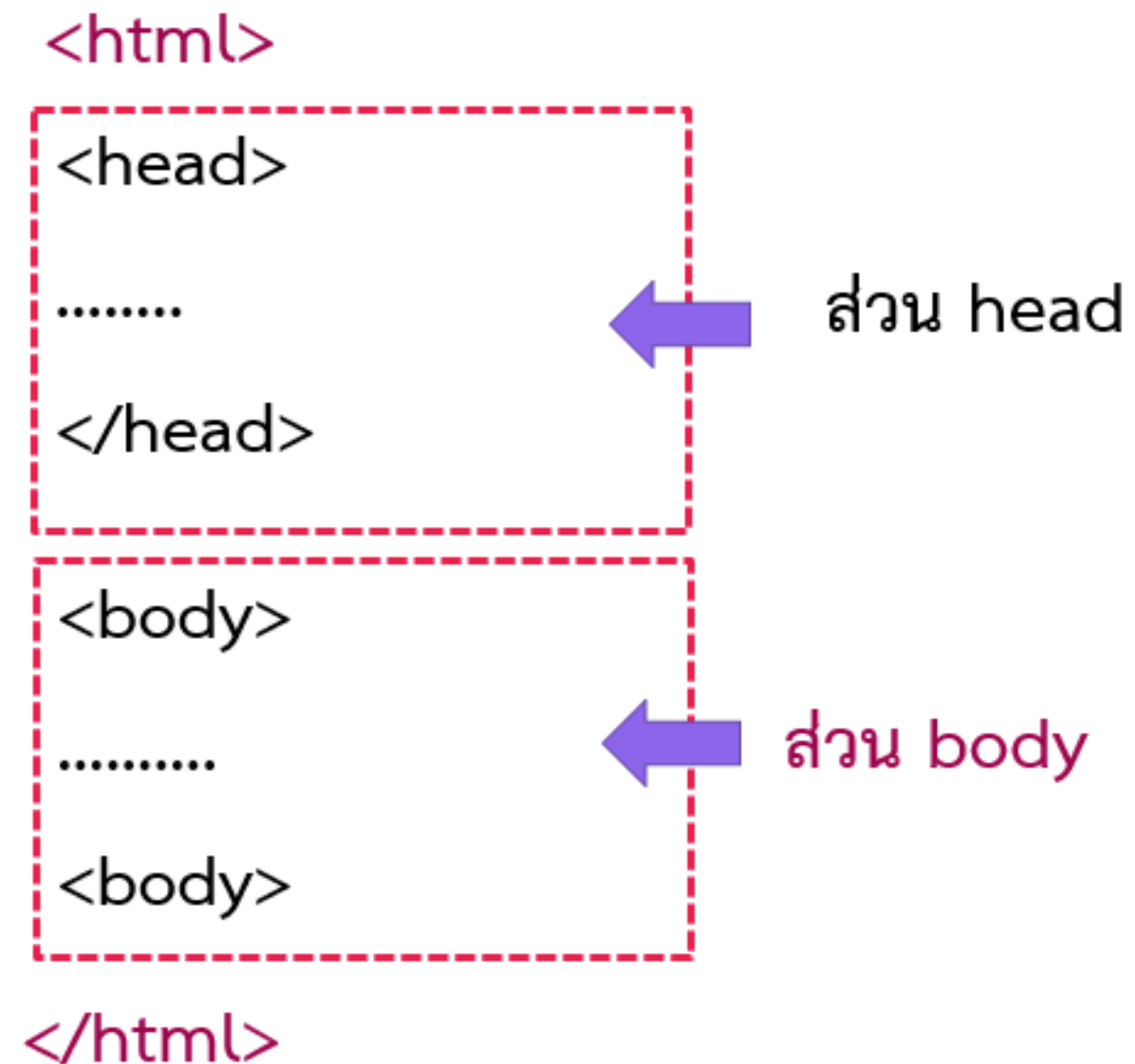
1. ส่วน HEAD เป็นส่วนหัวของเอกสาร HTML ไม่แสดงผลบนหน้าเว็บ โดยตรง ใช้สำหรับกำหนดข้อมูลเกี่ยวกับเอกสาร เช่น:

- <title> - ชื่อเว็บเพจที่แสดงบนแท็บเบราว์เซอร์
- <meta> - ข้อมูลเมตาต่างๆ เช่น การเข้ารหัส, คำอธิบายเว็บไซต์
- <link> - เชื่อมโยงไฟล์ CSS ภายนอก
- <script> - เชื่อมโยงไฟล์ JavaScript
- <style> - กำหนดรูปแบบ CSS ภายใน



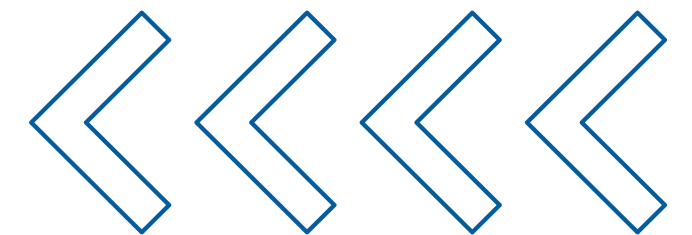
# โครงสร้างภาษา HTML (ต่อ)

โครงสร้าง HTML จะแบ่งออกเป็น 2 ส่วน ได้แก่ ส่วน **head** และ ส่วนของ **body**



1. ส่วน BODY เป็นส่วนที่แสดงเนื้อหาทั้งหมดบนหน้าเว็บ ประกอบด้วยแท็กต่างๆ สำหรับจัดการเนื้อหา เช่น:

- <h1> ถึง <h6> - หัวข้อ
- <p> - ย่อหน้า
- <div> - กล่องเนื้อหา
- <img> - รูปภาพ
- <a> - ลิงก์ <table> - ตาราง
- <form> - แบบฟอร์ม
- <ul>, <ol> - รายการ







# ส่วนประกอบและคำสั่งในเว็บเพจ

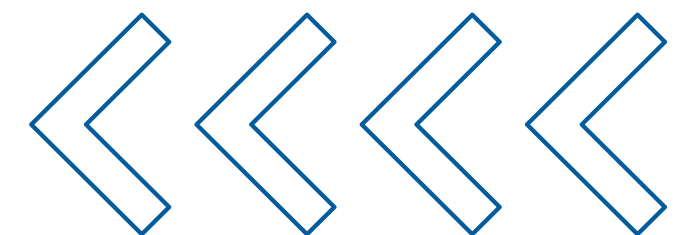
## □ ส่วนโครงสร้างพื้นฐานของเว็บเพจ

ทุกหน้าเว็บเพจที่สร้างด้วย HTML จะต้องมีส่วนโครงสร้างพื้นฐานดังนี้:

```
html
Copy code

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ชื่อหน้าเว็บ</title>
</head>
<body>
  <h1>ยินดีต้อนรับ</h1>
  <p>นี่คือเว็บเพจของคุณ</p>
</body>
</html>
```

- <!DOCTYPE html>: บอกเบราว์เซอร์ว่าเอกสารนี้เป็น HTML5
- <html>: เป็นโครงสร้างหลักของหน้าเว็บ
- <head>: ส่วนที่ใช้สำหรับข้อมูลเมตา เช่น ชื่อเว็บ (<title>), ลิงก์ไปยัง CSS, ไฟล์ JavaScript
- <body>: ส่วนที่แสดงผลเนื้อหาในเว็บเบราว์เซอร์





# แท็กสำคัญใน HTML

## □ แท็กโครงสร้าง

- <header>: ส่วนหัวของเว็บเพจ เช่น โลโก้และเมนูนำทาง
- <nav>: เมนูนำทาง
- <main>: เนื้อหาหลักของหน้าเว็บ
- <footer>: ส่วนท้ายของเว็บ เช่น ลิขสิทธิ์หรือข้อมูลติดต่อ
- <section>: แบ่งส่วนของเนื้อหา
- <article>: เนื้อหาบทความ
- <aside>: เนื้อหาข้างเคียง เช่น แถบโฆษณา

```
html

<header>
  <h1>หัวข้อเว็บ</h1>
  <nav>
    <a href="#home">Home</a>
    <a href="#about">About</a>
  </nav>
</header>
<main>
  <section>
    <h2>บทความ</h2>
    <p>เนื้อหาของบทความ</p>
  </section>
</main>
<footer>
  <p>&copy; 2025 Your Website</p>
</footer>
```





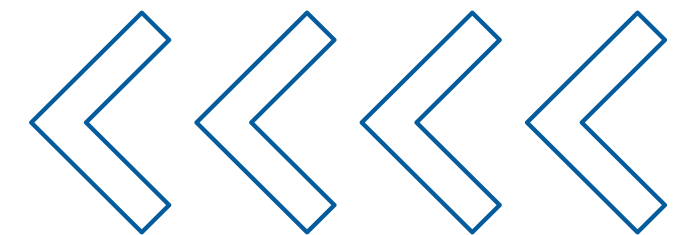
# แท็กสำคัญใน HTML

## □ แท็กข้อความ

- <h1> ถึง <h6>: หัวข้อ (หัวข้อใหญ่ <h1> หัวข้อเล็ก <h6>)
- <p>: ย่อหน้าข้อความ
- <strong>: ข้อความหนา
- <em>: ข้อความตัวเอียง
- <a href="url">: ลิงก์<br>: ขึ้นบรรทัดใหม่

html

```
<h1>หัวข้อใหญ่</h1>
<p>นี่คือย่อหน้า <strong>ตัวหนา</strong> และ <em>ตัวเอียง</em>.</p>
<a href="https://example.com">ไปที่เว็บไซต์</a>
```





# แท็กสำคัญใน HTML

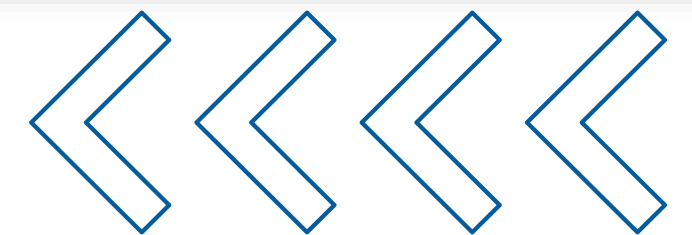
## □ แท็กรูปภาพและมัลติมีเดีย

- : แสดงภาพ
- <audio controls>: ใส่เสียงพร้อมตัวควบคุม
- <video controls>: ใส่วิดีโอพร้อมตัวควบคุม

html

```

<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
</audio>
<video controls>
  <source src="video.mp4" type="video/mp4">
</video>
```





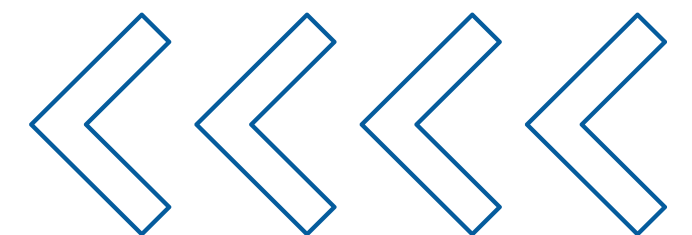
# แท็กสำคัญใน HTML

## □ แบบฟอร์ม

- <form>: ใช้สร้างฟอร์ม
- <input>: ป้อนข้อมูล
- <textarea>: กล่องข้อความ
- <button>: ปุ่ม
- <select>: เมนูแบบเลือก

html

```
<form action="/submit" method="post">  
  <label for="name">ชื่อ:</label>  
  <input type="text" id="name" name="name">  
  <label for="message">ข้อความ:</label>  
  <textarea id="message" name="message"></textarea>  
  <button type="submit">ส่ง</button>  
</form>
```





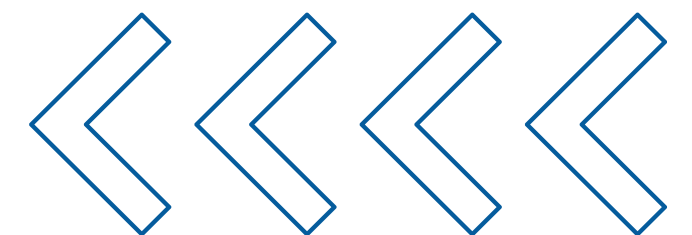
# แท็กสำคัญใน HTML

## □ ตาราง

- <table>: ใช้สร้างตาราง
- <tr>: แถว
- <td>: คอลัมน์
- <th>: หัวคอลัมน์

```
html

<table>
  <tr>
    <th>ชื่อ</th>
    <th>อายุ</th>
  </tr>
  <tr>
    <td>สมชาย</td>
    <td>30</td>
  </tr>
</table>
```

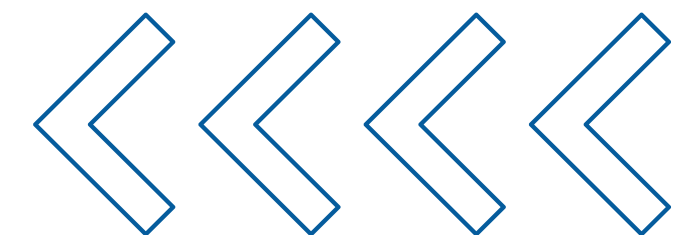
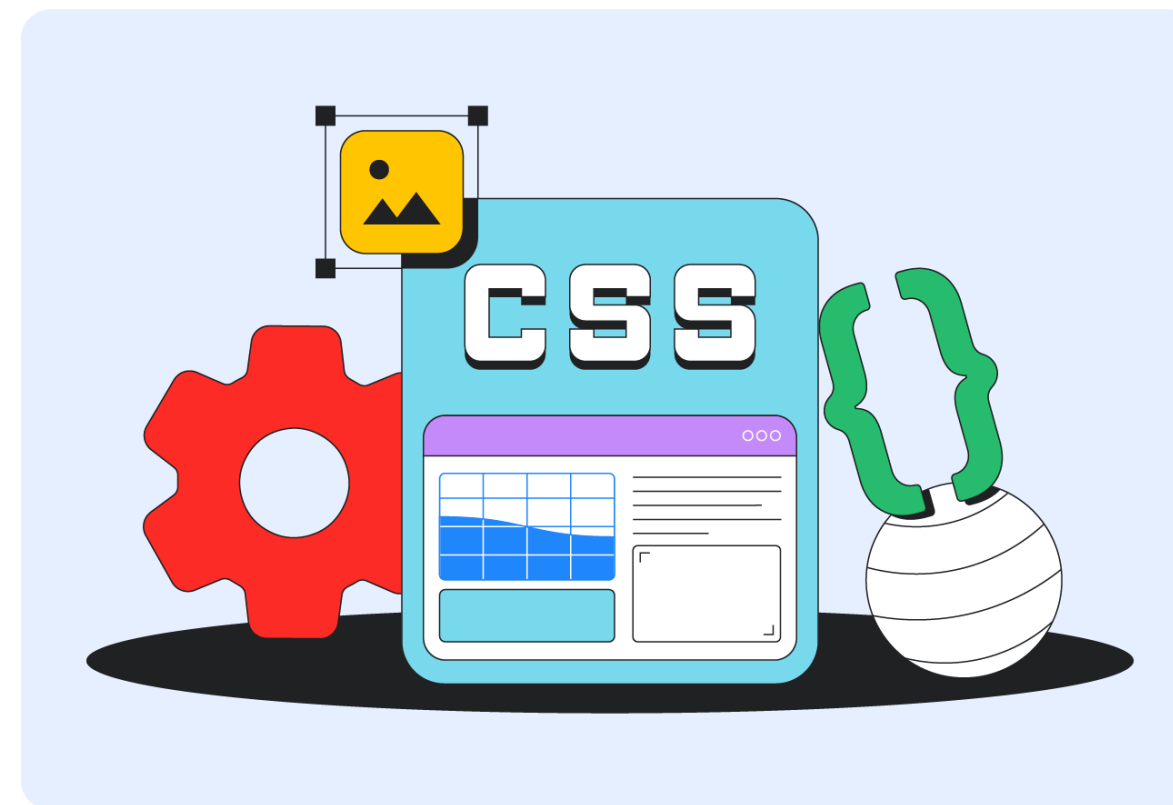




# Cascading Style Sheet คืออะไร



CSS ย่อมาจากคำว่า Cascading Style Sheet เรียกย่อว่า Style sheet เป็นเทคโนโลยีที่นำมาใช้จัดรูปแบบและควบคุมการแสดงผลของเว็บ เช่น การกำหนดขนาด สี แบบอักษร เส้นขอบ พื้นหลัง และตำแหน่งของข้อความ รูปภาพ เป็นต้น





# พื้นฐาน CSS



- โครงสร้างพื้นฐานของ CSS

```
Selector{  
    property: value;  
}
```

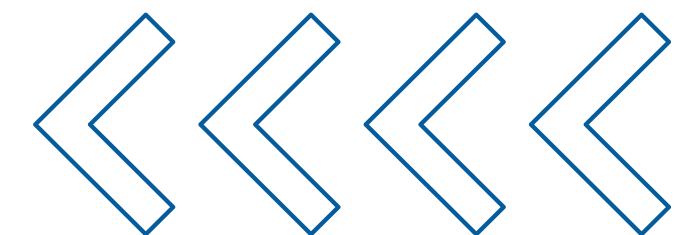
- Selector: ใช้เลือก HTML elements ที่ต้องการตกแต่ง
- Property: คุณสมบัติที่ต้องการกำหนด (เช่น สี, ขนาด, ระยะห่าง)
- Value: ค่าของคุณสมบัติ

CSS

```
h1 {  
    color: blue;  
    font-size: 24px;  
}
```

ในตัวอย่างนี้

- h1 เป็น selector
- color และ font-size เป็น property
- blue และ 24px เป็น value





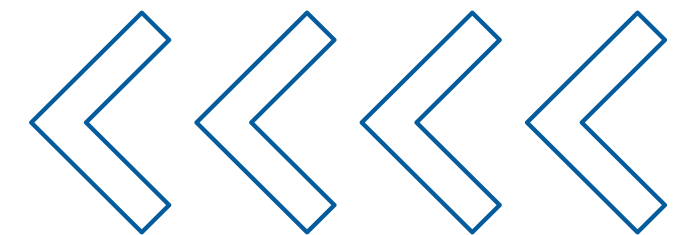


## พื้นฐาน CSS

- การเพิ่ม CSS ในหน้าเว็บ
  - Inline CSS (ในแท็ก HTML)

html

```
<h1 style="color: red;">ข้อความหัวข้อ</h1>
```





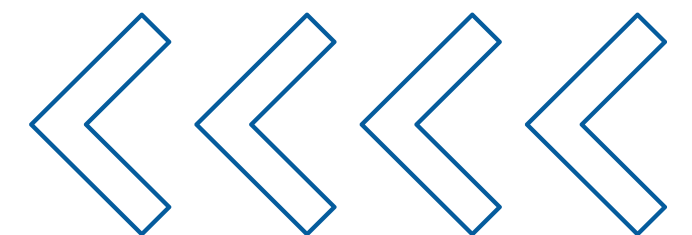
# พื้นฐาน CSS

## □ การเพิ่ม CSS ในหน้าเว็บ

- Internal CSS (ภายใน <style> ในไฟล์ HTML)

```
html

<style>
  p {
    color: green;
  }
</style>
```



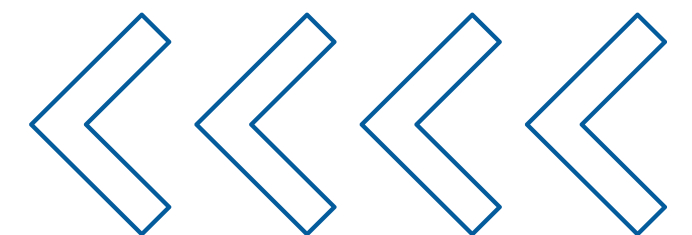


## พื้นฐาน CSS

- การเพิ่ม CSS ในหน้าเว็บ
  - External CSS (ในไฟล์ .css แยกต่างหาก)

html

```
<link rel="stylesheet" href="styles.css">
```





# พื้นฐาน CSS

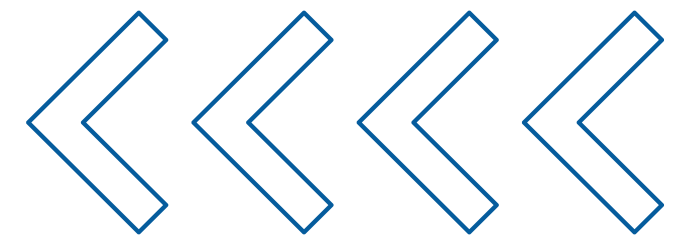
## □ การเลือกเอลิเมนต์ (Selectors) ใน CSS

- Selector ใน CSS ใช้สำหรับกำหนดว่ากฎการตกแต่ง (style rules) จะใช้กับเอลิเมนต์ใดในหน้าเว็บ

1. เลือกโดยชื่อแท็ก (Tag Selector) เลือกเอลิเมนต์ตามชื่อแท็ก HTML:

CSS

```
h1 {  
    color: red;  
}  
p {  
    font-size: 16px;  
}
```



# พื้นฐาน CSS

## ❑ การเลือกเอลิเมนต์ (Selectors) ใน CSS

- Selector ใน CSS ใช้สำหรับกำหนดว่ากฎการตกแต่ง (style rules) จะใช้กับเอลิเมนต์ใดในหน้าเว็บ

### 2. เลือกโดยคลาส (Class Selector) เลือกเอลิเมนต์โดยใช้ชื่อคลาส (Class):

html

```
<p class="highlight">ข้อความสำคัญ</p>  
<p>ข้อความปกติ</p>
```

CSS

```
.highlight {  
  color: orange;  
  font-weight: bold;  
}
```

- ใช้จุด (.) นำหน้าชื่อคลาสใน CSS



# พื้นฐาน CSS

## ❑ การเลือกเอลิเมนต์ (Selectors) ใน CSS

- Selector ใน CSS ใช้สำหรับกำหนดว่ากฎการตกแต่ง (style rules) จะใช้กับเอลิเมนต์ใดในหน้าเว็บ

### 3. เลือกโดยไอดี (ID Selector) เลือกเอลิเมนต์ที่มีไอดี (ID):

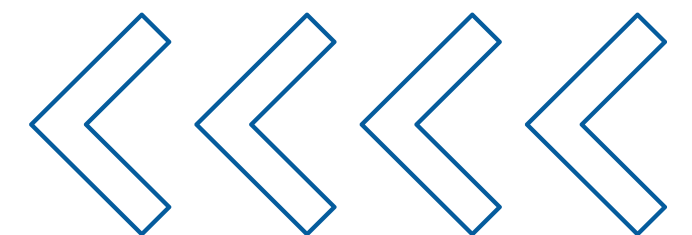
html

```
<p id="unique">ข้อความพิเศษ</p>
```

CSS

```
#unique {  
  color: blue;  
  font-size: 20px;  
}
```

- ใช้เครื่องหมาย (#) นำหน้าไอดีใน CSS







# พื้นฐาน CSS

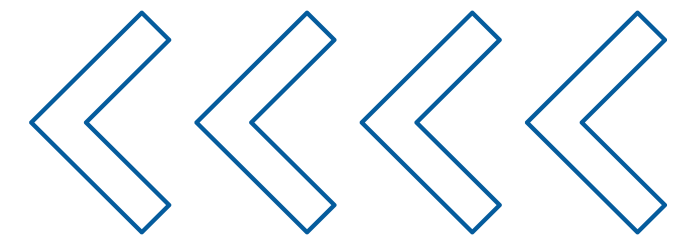
## □ การเลือกเอลิเมนต์ (Selectors) ใน CSS

- Selector ใน CSS ใช้สำหรับกำหนดว่ากฎการตกแต่ง (style rules) จะใช้กับเอลิเมนต์ใดในหน้าเว็บ

4. การเลือกโดยกลุ่มเอลิเมนต์ (Group Selector) เลือกหลายเอลิเมนต์พร้อมกัน:

CSS

```
h1, h2, p {  
    color: green;  
}
```





# พื้นฐาน CSS



## ❑ การเลือกเอลิเมนต์ (Selectors) ใน CSS

- Selector ใน CSS ใช้สำหรับกำหนดว่ากฎการตกแต่ง (style rules) จะใช้กับเอลิเมนต์ใดในหน้าเว็บ

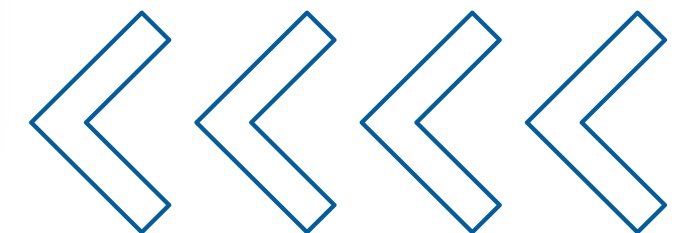
5. การเลือกโดยโครงสร้าง (Descendant Selector) เลือกเอลิเมนต์ที่อยู่ภายในเอลิเมนต์อื่น:

html

```
<div class="container">  
  <p>ข้อความภายใน</p>  
</div>
```

CSS

```
.container p {  
  color: purple;  
}
```



# พื้นฐาน CSS

## ❑ การเลือกเอลิเมนต์ (Selectors) ใน CSS

- Selector ใน CSS ใช้สำหรับกำหนดว่ากฎการตกแต่ง (style rules) จะใช้กับเอลิเมนต์ใดในหน้าเว็บ

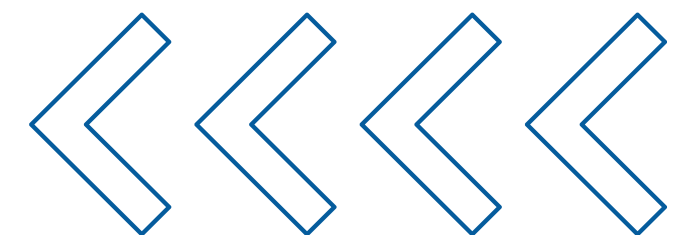
6. การเลือกโดยลูกตรง (Child Selector) เลือกเฉพาะลูกตรงของเอลิเมนต์:

html

```
<div class="parent">
  <p>ลูกตรง</p>
  <div>
    <p>ลูกของลูก</p>
  </div>
</div>
```

CSS

```
.parent > p {
  color: red;
}
```





# พื้นฐาน CSS

## □ การเลือกเอลิเมนต์ (Selectors) ใน CSS

- Selector ใน CSS ใช้สำหรับกำหนดว่ากฎการตกแต่ง (style rules) จะใช้กับเอลิเมนต์ใดในหน้าเว็บ

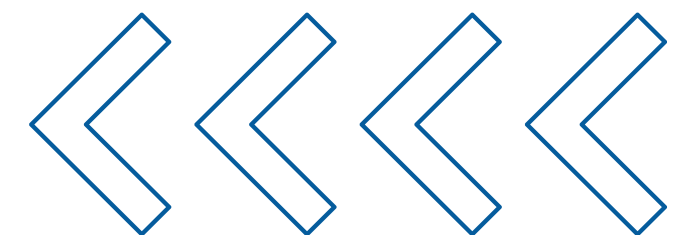
7. การเลือกโดยสถานะ (Pseudo-classes) ใช้เลือกเอลิเมนต์ที่อยู่ในสถานะเฉพาะ:

CSS

```
a:hover {  
    color: blue;  
}  
  
input:focus {  
    border: 2px solid green;  
}
```

ตัวอย่าง:

- **hover** สำหรับเลือกเมื่อผู้ใช้เลื่อนเมาส์
- **focus** เมื่อเอลิเมนต์ได้รับการโฟกัส





## พื้นฐาน CSS

### ❑ การเลือกเอลิเมนต์ (Selectors) ใน CSS

- Selector ใน CSS ใช้สำหรับกำหนดว่ากฎการตกแต่ง (style rules) จะใช้กับเอลิเมนต์ใดในหน้าเว็บ

8. การเลือกโดยแอตทริบิวต์ (Attribute Selector) เลือกเอลิเมนต์ตามแอตทริบิวต์ที่กำหนด:

html

```
<input type="text">  
<input type="submit">
```

CSS

```
input[type="text"] {  
    border: 1px solid gray;  
}
```



# พื้นฐาน CSS

## ❑ ตัวอย่างการใช้ Selectors

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Selectors</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    h1 {
      color: red;
    }
    .highlight {
      background-color: yellow;
      font-weight: bold;
    }
    #unique {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>หัวข้อ</h1>
  <p class="highlight">ข้อความที่เน้น</p>
  <p id="unique">ข้อความพิเศษ</p>
</body>
</html>
```

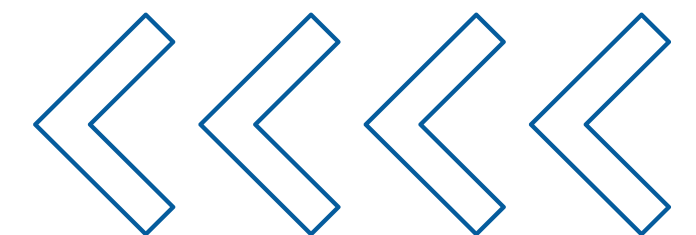






# Bootstrap คืออะไร

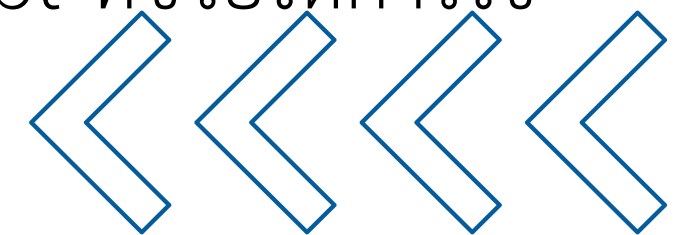
คือเฟรมเวิร์กสำหรับการออกแบบเว็บที่ได้รับความนิยมอย่างมากในปัจจุบัน โดยเป็นโอเพ่นซอร์ส (Open Source) ซึ่งช่วยให้นักพัฒนาสามารถสร้างเว็บเพจและเว็บแอปพลิเคชันที่ตอบสนอง (Responsive) ได้อย่างรวดเร็วและง่ายดาย โดยไม่ต้องเขียนโค้ดตั้งแต่ต้น





# คุณสมบัติเด่นของ Bootstrap

1. **Responsive Design** Bootstrap ถูกออกแบบมาให้รองรับการแสดงผลที่ปรับขนาดได้อัตโนมัติตามอุปกรณ์ที่ใช้งาน เช่น เดสก์ท็อป, แท็บเล็ต และสมาร์ทโฟน
2. **ระบบ Grid Layout** มีระบบ Grid แบบ 12 คอลัมน์ที่ช่วยให้นักพัฒนาสามารถจัดวางองค์ประกอบบนหน้าเว็บได้ง่ายขึ้น
3. **Components** สำเร็จรูปมีองค์ประกอบสำเร็จรูปมากมาย เช่น ปุ่ม, แถบนำทาง (Navbar), การ์ด (Card), Modal และอื่นๆ
4. **Customizable** สามารถปรับแต่งการตั้งค่าต่างๆ ได้ เช่น สี ขนาด และสไตล์ เพื่อให้เหมาะกับความต้องการของโปรเจกต์
5. **Cross-Browser Compatibility** รองรับเบราว์เซอร์หลักทั้งหมด เช่น Chrome, Firefox, Safari, Edge
6. **การผสมผสาน CSS, JavaScript และ HTML** Bootstrap มีทั้งไฟล์ CSS และ JavaScript ที่ช่วยให้การใช้งานง่ายขึ้น เช่น การสร้างเอฟเฟกต์ต่างๆ หรือการจัดการฟังก์ชันในหน้าเว็บ

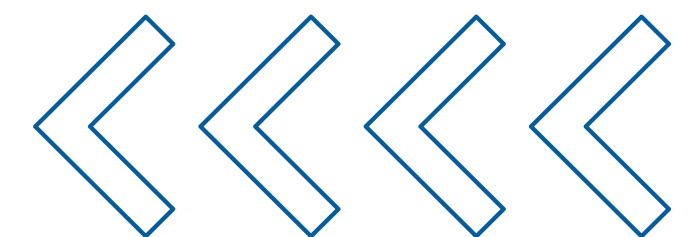




# องค์ประกอบหลักของ Bootstrap



1. **CSS** มีสไตล์ชีทสำเร็จรูปสำหรับองค์ประกอบต่างๆ เช่น ปุ่ม, แบบฟอร์ม และข้อความ
2. **JavaScript Plugins** มีปลั๊กอินที่ช่วยเพิ่มฟังก์ชันการทำงาน เช่น Modal, Carousel, Dropdown
3. **Utility Classes** มีคลาสที่ใช้แก้ไขสไตล์ง่ายๆ เช่น การตั้งค่าระยะห่าง (Margin, Padding), สีพื้นหลัง, หรือการจัดวางข้อความ



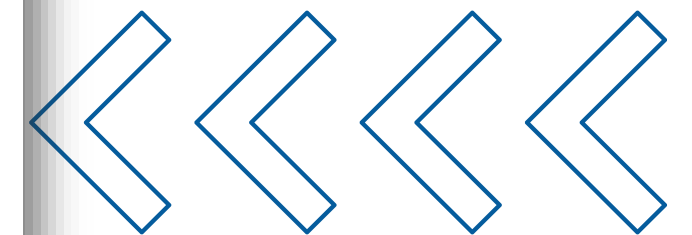


# ตัวอย่างการใช้งาน Bootstrap

การเชื่อม Bootstrap เข้ากับโปรเจกต์สามารถทำได้ง่ายๆ ด้วยการเพิ่มลิงก์ใน  
<head> ของ HTML:

```
html
Copy code

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Example</title>
  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container">
    <h1 class="text-center">Welcome to Bootstrap</h1>
    <button class="btn btn-primary">Click Me</button>
  </div>
  <!-- Bootstrap JavaScript -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```





Q&A