

บทที่ 2

ภาษา HTML พื้นฐาน

HTML (Hyper Text Markup Language) คือภาษาคอมพิวเตอร์ที่ใช้ในการสร้างและจัดโครงสร้างเนื้อหาบนเว็บไซต์ ภาษา HTML ทำหน้าที่เป็นโครงร่างพื้นฐานที่ประกอบด้วยแท็ก (Tags) ซึ่งบ่งบอกถึงบทบาทของเนื้อหาต่าง ๆ เช่น ข้อความ ภาพ และลิงก์ บทนี้จะช่วยให้นักศึกษาเข้าใจและสามารถสร้างเว็บไซต์ขั้นพื้นฐาน (W3C, 2024)

ตัวอย่างเริ่มต้นของภาษา HTML

HTML

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

- <!DOCTYPE html>: ใช้บอกเบราว์เซอร์ว่ากำลังใช้ HTML เวอร์ชันไหน ในที่นี้คือ HTML5
- <html>: เปิดหน้า HTML เพื่อแสดงเนื้อหาทั้งหมด
- <head>: บรรจุข้อมูลทางเทคนิค เช่น การตั้งชื่อหน้าเว็บ
- <title>: ระบุชื่อของเว็บที่จะไปแสดงบนแถบเบราว์เซอร์
- <body>: ส่วนที่เก็บเนื้อหาที่จะแสดงผล เช่น ข้อความ รูปภาพ วิดีโอ หรือฟอร์ม
- <meta charset="UTF-8">: กำหนดการเข้ารหัสตัวอักษรเป็น UTF-8 (รูปแบบที่ใช้ได้หลากหลายภาษา รวมถึงภาษาไทย)
- <meta name="viewport" content="width=device-width, initial-scale=1.0">: กำหนดให้หน้าเว็บตอบสนองต่ออุปกรณ์ที่มีขนาดหน้าจอที่แตกต่างกันเพื่อให้หน้าเว็บสามารถแสดงผลได้ดีทั้งบนมือถือและเดสก์ท็อป
- <h1> องค์ประกอบกำหนดหัวข้อความขนาดใหญ่
- <p> องค์ประกอบกำหนดย่อหน้า

2.1 โปรแกรม HTML Editor

โปรแกรมแก้ไข HTML (HTML Editors) เป็นเครื่องมือที่ช่วยให้สามารถเขียนโค้ด HTML ได้สะดวกยิ่งขึ้น โดยสามารถใช้ Text Editor ที่เป็นพื้นฐาน เช่น Notepad (บน Windows) หรือ TextEdit (บน macOS) หรือจะใช้โปรแกรมแก้ไขที่มีฟีเจอร์เสริม เช่น การแสดงผลโค้ดที่สวยงาม (Syntax Highlighting) และการแนะนำโค้ด (Code Autocomplete) ซึ่งช่วยในการเขียนและตรวจสอบโค้ดได้ง่ายขึ้น

ตัวอย่าง HTML Editors ยอดนิยม:

11. Notepad++ (Windows) – เป็นโปรแกรมแก้ไขข้อความที่ฟรีและเบา มาพร้อมกับฟีเจอร์ Syntax Highlighting และการสนับสนุนหลายภาษา
12. Visual Studio Code (VS Code) – Text Editor ที่ได้รับความนิยมมาก มีเครื่องมือเสริมสำหรับการเขียน HTML, CSS, JavaScript และอื่นๆ อีกมากมาย
13. Sublime Text – เป็นโปรแกรมแก้ไขโค้ดที่รวดเร็วและสามารถติดตั้งปลั๊กอินเสริมเพื่อช่วยในการพัฒนาเว็บไซต์
14. Atom – โปรแกรมแก้ไขที่ปรับแต่งได้ง่ายและรองรับการทำงานร่วมกับ GitHub
15. Brackets – Text Editor ที่ออกแบบมาสำหรับนักพัฒนาเว็บ มีฟีเจอร์ Live Preview ที่ช่วยให้เห็นผลลัพธ์ของการเขียน HTML ได้ทันที

2.2 HTML พื้นฐาน

2.2.1 เอกสาร HTML

เอกสาร HTML ทุกฉบับต้องเริ่มต้นด้วยการประกาศประเภทของเอกสาร (Document Type Declaration) โดยใช้ `<!DOCTYPE html>` เพื่อบอกเบราว์เซอร์ว่าเอกสารนี้เป็น HTML5 โครงสร้างของเอกสาร HTML ประกอบด้วย 3 ส่วนหลัก คือ

```
HTML
<!DOCTYPE html>
<html>
  <head>
    <title>My First Webpage</title> <!-- ชื่อที่แสดงบนแท็บเบราว์เซอร์ -->
  </head>
  <body>
    <h1>Welcome to My Website</h1> <!-- ข้อความหัวข้อใหญ่ -->
    <p>This is a paragraph of text.</p> <!-- ข้อความย่อหน้า -->
    <a href="https://www.example.com">Click here to visit Example.com</a> <!-- ลิงก์ -->
  </body>
</html>
```

9. **DOCTYPE Declaration:** `<!DOCTYPE html>` ใช้ในการประกาศประเภทของเอกสาร (Document Type) เพื่อให้เบราว์เซอร์รู้ว่าเป็นเอกสาร HTML5 ต้องปรากฏเพียงครั้งเดียวที่ด้านบนของเอกสาร (ก่อนแท็ก HTML ใดๆ) ไม่แยกแยะตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็ก (DOCTYPE หรือ doctype ก็สามารถใช้ได้)

10. **แท็ก `<html>`:** ส่วนนี้คือจุดเริ่มต้นของเอกสาร HTML และต้องมีการปิดด้วยแท็ก `</html>` เสมอ

11. **แท็ก `<body>`:** ส่วนที่อยู่ระหว่าง `<body>` และ `</body>` คือเนื้อหาที่จะแสดงผลในเบราว์เซอร์ เช่น ข้อความ รูปภาพ ลิงก์ ฯลฯ

2.2.2 หัวข้อ (Heading Tags)

HTML

```
<h1>หัวข้อ 1</h1>
<h2>หัวข้อ 2</h2>
<h3>หัวข้อ 3</h3>
<h4>หัวข้อ 4</h4>
<h5>หัวข้อ 5</h5>
<h6>หัวข้อ 6</h6>
```

แท็ก `<h1>` ถึง `<h6>` ใช้เพื่อระบุระดับการจัดลำดับความสำคัญของข้อความแต่ละส่วนในเอกสาร โดย `<h1>` เป็นหัวข้อที่ใหญ่และมีความสำคัญสูงสุด และ `<h6>` เป็นหัวข้อเล็กที่สุด

2.2.3 ย่อหน้า (Paragraphs)

HTML

```
<p>นี่คือตัวอย่างของย่อหน้า</p>
```

ใช้ `<p>` เพื่อกำหนดข้อความให้เป็นประโยคหรือย่อหน้า มักจะใช้สำหรับการแสดงข้อมูลหลายบรรทัดบนหน้าเว็บ

2.2.4 ลิงก์ (Links)

HTML

```
<a href="https://www.example.com">นี่คือลิงก์</a>
```

แท็ก `<a>` เป็นแท็กที่ใช้ในการสร้างลิงก์ไปยังหน้าอื่นหรือแม้แต่ไปยังส่วนอื่น ๆ ของเพจเดียวกัน โดย `href` เป็นคุณสมบัติที่ใช้ระบุค่า URL ที่จะเชื่อมโยง

2.2.5 รูปภาพ (Images)

HTML

```

```

แท็ก `` ใช้สำหรับการแสดงรูปภาพในเว็บไซต์ ค่า `src` ระบุแหล่งที่มาของรูปภาพ และค่า `alt` (ข้อความทางเลือก) จะถูกแสดงขึ้นหากรูปภาพไม่สามารถโหลดได้ และยังช่วยในการเข้าถึงข้อมูลบนหน้าเว็บในกรณีที่ผู้ใช้ที่ใช้โปรแกรมสำหรับอ่านหน้าจอ

2.2.6 Comments (คอมเมนต์ในโค้ด)

คอมเมนต์ (Comments) เป็นข้อความในไฟล์ HTML ที่ใส่ไว้เพื่อใช้หมายเหตุ ประกาศ หรือให้คำอธิบายโดยจะไม่ปรากฏบนหน้าเว็บจริง คอมเมนต์ช่วยให้นักพัฒนาเข้าใจโค้ดได้มากขึ้น และยังช่วยในการแก้ไขและบำรุงรักษาโดยไม่ทำให้เบราว์เซอร์แสดงผลส่วนที่เราใส่ไว้ในหน้าเว็บ

HTML

```
/* นี่คือ Comment ใน HTML */
```

2.3 องค์ประกอบภาษา HTML (HTML Element)

2.3.1 HTML Elements หรือ Tag

ทุกองค์ประกอบ HTML จะเริ่มต้นจากแท็ก ซึ่งใช้เพื่อกำหนดโครงสร้างและบอกข้อมูลที่อยู่ภายในว่าจะถูกแสดงแสดงออกมาบนหน้าจอเบราว์เซอร์อย่างไร แท็กจะแบ่งเป็น 2 ส่วนหลัก ๆ คือ แท็กเปิด แท็กปิด และแท็กที่ปิดตัวเอง

- แท็กเปิด เช่น `<p>` สำหรับการเริ่มต้นแท็กที่ต้องการใช้
- แท็กปิด เช่น `</p>` สำหรับบอกเบราว์เซอร์ว่าแท็กนั้นสิ้นสุดลงแล้ว

2.3.2 การซ้อน HTML (Nested HTML Elements)

ใน HTML เราสามารถซ้อน (nested) HTML elements ภายในกันและกันได้ ซึ่งหมายความว่าองค์ประกอบหนึ่งสามารถบรรจุอีกองค์ประกอบหนึ่งภายในได้ และเอกสาร HTML ทุกฉบับจะประกอบด้วย HTML elements ที่ถูกซ้อนกัน

2.3.3 องค์ประกอบว่าง (Empty HTML Elements)

Empty HTML Elements คือองค์ประกอบ HTML ที่ไม่มีเนื้อหาใดๆ ภายใน และไม่ต้องการแท็กปิด (Closing Tag) ตัวอย่างที่ใช้บ่อยคือแท็ก `
`, ซึ่งเป็นแท็กที่กำหนดการขึ้นบรรทัดใหม่ (line break) และเป็นองค์ประกอบที่ว่างเปล่า

HTML

```
<p>This is a paragraph with a line break.<br>This text is on a new line.</p>
```

ในตัวอย่างนี้ แท็ก `
` จะทำให้ข้อความที่ตามหลังขึ้นบรรทัดใหม่โดยไม่จำเป็นต้องใช้แท็ก

2.3.4 คุณสมบัติ (HTML Attributes)

ทุก Tag ในภาษา HTML สามารถมีคุณสมบัติได้ (HTML Attributes) คุณสมบัตินี้ คือข้อมูลที่ช่วยควบคุมพฤติกรรมหรือคุณลักษณะของแท็กนั้นเพิ่มเติม คุณสมบัติจะถูกระบุภายในแท็กเปิด (Opening tag) โดยมีการใช้การจับคู่ที่กำหนดไว้ล่วงหน้าก่อนที่จะมีการปิดแท็ก

คุณสมบัติจะถูกระบุอยู่ในแท็กเปิดหลังจากชื่อแท็ก และจะมีรูปแบบ `attribute="value"` คือระบุชื่อคุณสมบัติตามด้วยเครื่องหมายเท่ากับ (=) และค่าที่ต้องการในเครื่องหมายอัญประกาศ (") ตัวแท็กเองจะทำหน้าที่กำหนดบทบาทของเนื้อหา เช่น

```
HTML
<a href="https://www.example.com" target="_blank">เยี่ยมชมเว็บไซต์</a>
```

จากตัวอย่าง

- `href="https://www.example.com"` ระบุที่อยู่ URL ที่ลิงก์จะนำไป
- `target="_blank"` เป็นการระบุให้ลิงก์เปิดหน้าต่างหรือแท็บใหม่

คุณสมบัติที่ใช้บ่อย

- `id`: ใช้กำหนดไอดีที่เป็นเอกลักษณ์ให้กับแท็ก

```
HTML
<div id="content">เนื้อหาหลัก</div>
```

- `class`: ใช้กำหนดกลุ่ม (Class) เพื่อควบคุมลักษณะเฉพาะหลายรายการผ่าน CSS หรือใช้ใน JavaScript เพื่อควบคุมพฤติกรรมหลายแท็กพร้อมกัน

```
HTML
<p class="highlight">ข้อความที่ถูกเน้น</p>
```

- `src`: ระบุแหล่งที่มาของเนื้อหา เช่นรูปภาพหรือวิดีโอ
- `width and height`: ใช้สำหรับแสดงภาพ เพื่อกำหนดความกว้างและความสูงของรูปภาพในหน่วยพิกเซล (pixels) ซึ่งช่วยควบคุมขนาดของภาพที่แสดงบนหน้าเว็บได้อย่างแม่นยำ
- `alt`: ใช้ในการระบุข้อความที่จะแสดงแทนรูปภาพหากไม่สามารถแสดงรูปภาพได้

```
HTML

```

- Style : ใช้ในการเพิ่มสไตล์ให้กับองค์ประกอบ HTML เช่น การกำหนดสี ตัวอักษร ขนาด และอื่นๆ โดยสามารถกำหนดคุณสมบัติ CSS ให้กับองค์ประกอบนั้นได้โดยตรงในโค้ด HTML

HTML

```
<p style="color:red;">This is a red paragraph.</p>
```

2.4 การใช้ลิสต์

Lists (รายการ) ใน HTML มีสองประเภทหลัก คือ

1. รายการแบบลำดับ (Ordered List) ใช้แท็ก
2. รายการแบบไม่ลำดับ (Unordered List) ใช้แท็ก

2.4.1 รายการแบบลำดับ (Ordered List)

รายการแบบลำดับคือรายการที่มีลำดับหมายเลข เช่น 1, 2, 3 เป็นต้น ใช้แท็ก เพื่อกำหนดรายการ ตัวอย่างการสร้าง Ordered List ดังนี้

HTML

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

ผลลัพธ์ที่ได้:

1. Item 1
2. Item 2
3. Item 3

ผู้ใช้สามารถเปลี่ยนสไตล์ของตัวเลขในรายการได้ เช่น เปลี่ยนเป็นตัวอักษรโรมัน หรือตัวอักษรภาษาอังกฤษโดยใช้ attribute type เช่น:

HTML

```
<ol type="I">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

ผลลัพธ์:

- I. Item 1
- II. Item 2
- III. Item 3

2.4.2 รายการแบบไม่ลำดับ (Unordered List)

รายการแบบไม่ลำดับคือรายการที่ใช้เครื่องหมายจุดนำหน้า เช่น bullet points ใช้แท็ก `` ในการสร้าง

HTML

```
<ul>
  <li>Fruit</li>
  <li>Apple</li>
  <li>Banana</li>
</ul>
```

ผลลัพธ์ที่ได้:

- Fruit
- Apple
- Banana

โดยปกติแล้ว จะใช้เครื่องหมายจุดกลม (●) เป็น default แต่คุณสามารถเปลี่ยนเครื่องหมายให้เป็นแบบอื่นได้ เช่น วงกลมกลวง (○) หรือ สีเหลี่ยม (■) โดยใช้ CSS:

HTML

```
<ul style="list-style-type: square;">
  <li>Fruit</li>
  <li>Apple</li>
  <li>Banana</li>
</ul>
```

ผลลัพธ์:

- Fruit
- Apple
- Banana

2.4.3 Nested Lists (รายการซ้อน)

คุณสามารถซ้อนรายการภายในกันได้ โดยมีทั้งแบบลำดับและไม่ลำดับร่วมกันในรายการเดียว

HTML

```
<ul>
  <li>Fruit
    <ul>
      <li>Apple</li>
      <li>Banana</li>
    </ul>
  </li>
  <li>Vegetables
    <ol>
      <li>Carrot</li>
      <li>Broccoli</li>
    </ol>
  </li>
</ul>
```

```

    </ol>
  </li>
</ul>

```

ผลลัพธ์:

- Fruit
 - Apple
 - Banana
- Vegetables
 1. Carrot
 2. Broccoli

2.5 แท็ก Div

แท็ก <div> เป็นหนึ่งในแท็กที่ใช้บ่อยที่สุดใน HTML สำหรับการจัดกลุ่มเนื้อหาหรือองค์ประกอบอื่นๆ บนหน้าเว็บ <div> ย่อมาจาก "division" ซึ่งหมายถึงส่วนของหน้าเว็บ และมักใช้เพื่อจัดเลย์เอาต์ของหน้าโดยการแบ่งหน้าออกเป็นหลายส่วน

ลักษณะการใช้ <div>

HTML

```

<div>
  <h1>Heading inside div</h1>
  <p>This is a paragraph inside a div element.</p>
</div>

```

การใช้งาน <div> ร่วมกับ CSS

HTML

```

<div style="background-color: lightblue; padding: 20px;">
  <h2>This is a styled div</h2>
  <p>Text inside a div with a background color and padding.</p>
</div>

```

การใช้ <div> จัดเลย์เอาต์

HTML

```

<div style="width: 100%; display: flex;">
  <div style="width: 70%; background-color: lightgreen;">
    <h3>Main Content</h3>
    <p>This is the main content area.</p>
  </div>
  <div style="width: 30%; background-color: lightgray;">
    <h3>Sidebar</h3>
    <p>This is the sidebar area.</p>
  </div>
</div>

```


ประโยชน์ของการใช้ <div>

1. การจัดกลุ่มเนื้อหา: ใช้ในการจัดกลุ่มเนื้อหาต่างๆ ที่เกี่ยวข้องกันให้อยู่ภายในบล็อกเดียว
2. การจัดเลย์เอาต์: ใช้เพื่อแบ่งหน้าเว็บออกเป็นส่วนต่างๆ เช่น ส่วนหัว ส่วนเนื้อหา และส่วนท้าย
3. ความยืดหยุ่นในการจัดการ: การใช้ร่วมกับ CSS ทำให้สามารถควบคุมเลย์เอาต์และสไตล์ของหน้าเว็บได้อย่างอิสระ

<div> สามารถปรับเปลี่ยนการใช้งานได้หลากหลาย ไม่ว่าจะเป็นการแสดงผลในขนาดหน้าจอที่ต่างกัน รองรับการใช้งานบนอุปกรณ์ต่าง ๆ อย่างมีประสิทธิภาพ และตอบสนองความต้องการของผู้ใช้งานที่หลากหลาย ซึ่งเราจะกล่าวถึงรายละเอียดเพิ่มเติมในส่วนของเนื้อหาเกี่ยวกับ Responsive Web Design

2.6 CSS

การใช้งาน CSS อย่างง่ายนั้นสามารถเริ่มต้นได้จากการเรียนรู้การกำหนดสไตล์พื้นฐานสำหรับองค์ประกอบของเว็บเพจ เว็บไซต์ที่สร้างโดยใช้ HTML มักจะมีการนำ CSS มาทำให้มีการจัดรูปแบบที่สวยงามและใช้งานได้สะดวก

2.6.1 การกำหนดคุณสมบัติและค่า

หลังจากเลือกองค์ประกอบแล้ว คุณสามารถกำหนดคุณสมบัติที่ต้องการ เช่น สี ขนาดตัวอักษร และระยะห่าง เช่น

css

```
p {
  color: red;
  font-size: 16px;
  margin: 10px;
}
```

2.6.2 การใช้งาน CSS

1. Inline CSS: การใช้งานสไตล์ภายในแท็ก HTML โดยตรง

html

```
<p style="color: green;">This is a paragraph.</p>
```

2. Internal CSS: การใส่ CSS ในส่วน "<head>" ของเอกสาร HTML:

html

```
<head>
  <style>
    body {
      background-color: lightgray;
    }
  </style>
</head>
```

```
</style>
</head>
```

3. External CSS: การเชื่อมต่อไฟล์ CSS ภายนอก:

```
html
```

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

2.6.3 การเลือกองค์ประกอบ (Selectors):

ใน CSS คุณสามารถเลือกองค์ประกอบ HTML โดยใช้ชื่อแท็ก คลาส หรือไอดีได้ เช่น

1. Tag: กำหนดสไตล์ให้กับแท็กทั้งหมด เช่น

```
CSS
```

```
h1 {
  color: blue;
}
```

2. Class: ใช้เครื่องหมาย “.” เพื่อลงทะเบียนคลาส เช่น

```
CSS
```

```
.highlight {
  background-color: yellow;
}
```

3. id: ใช้เครื่องหมาย “#” สำหรับไอดี เช่น

```
CSS
```

```
#main-title {
  font-size: 24px;
}
```

ลองเขียนโค้ด HTML พร้อมกับ CSS อย่างง่าย เพื่อดูตัวอย่าง

```
html
<!DOCTYPE html>
<html lang="th">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>ตัวอย่าง CSS</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
    }
    h1 {
      color: navy;
    }
    p {
      color: darkslategray;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <h1>ยินดีต้อนรับสู่เว็บไซต์ของเรา!</h1>
  <p>นี่คือตัวอย่างการใช้งาน CSS อย่างง่ายเพื่อตกแต่งเว็บเพจของคุณ!</p>
</body>
</html>
```

การใช้งาน CSS อย่างง่ายช่วยให้คุณสามารถปรับแต่งและออกแบบการแสดงผลของเว็บไซต์ได้อย่างมีประสิทธิภาพ เริ่มต้นจากการเลือกองค์ประกอบที่ต้องการและกำหนดคุณสมบัติพื้นฐาน เพื่อสร้างประสบการณ์การใช้งานที่น่าสนใจและสวยงามให้กับผู้ใช้ ทั้งนี้เมื่อ CSS โดยละเอียดจะศึกษาอีกครั้งใน บทที่ 3

2.7 ตาราง

ตารางใน HTML ใช้แท็ก <table> เพื่อแสดงข้อมูลในรูปแบบตาราง โดยข้อมูลจะถูกจัดเรียงในแถว (<tr>) และคอลัมน์ (<td>)

2.7.1 โครงสร้างพื้นฐานของตารางใน HTML

```
html
<table>
  <tr>
    <th>Heading 1</th>
    <th>Heading 2</th>
  </tr>
  <tr>
    <td>Row 1, Column 1</td>
    <td>Row 1, Column 2</td>
  </tr>
  <tr>
    <td>Row 2, Column 1</td>
    <td>Row 2, Column 2</td>
  </tr>
</table>
```

แท็กสำคัญใน HTML Table:

- <table>: ใช้ในการสร้างตาราง
- <tr>: ใช้ในการสร้างแถวของตาราง
- <th>: ใช้ในการสร้างหัวข้อของแต่ละคอลัมน์
- <td>: ใช้ในการใส่ข้อมูลในแต่ละเซลล์

2.7.2 การใส่เส้นขอบให้กับตาราง

การใช้คุณสมบัติ border เพื่อใส่เส้นขอบให้ตาราง

```
html
<table border="1">
  <tr>
    <th>Product</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>$1</td>
  </tr>
  <tr>
    <td>Banana</td>
    <td>$0.5</td>
  </tr>
```

```
</tr>
</table>
```

Product	Price
Apple	\$1
Banana	\$0.5

2.7.3 การรวมเซลล์ด้วย colspan และ rowspan

นักพัฒนาสามารถรวมเซลล์ในแถวหรือคอลัมน์ได้โดยใช้คุณสมบัติ colspan หรือ rowspan

```
html
```

```
<table border="1">
  <tr>
    <th colspan="2">Merged Header</th>
  </tr>
  <tr>
    <td>Row 1, Column 1</td>
    <td>Row 1, Column 2</td>
  </tr>
  <tr>
    <td rowspan="2">Merged Row 2 and 3</td>
    <td>Row 2, Column 2</td>
  </tr>
  <tr>
    <td>Row 3, Column 2</td>
  </tr>
</table>
```

- colspan="2" รวมเซลล์ในแถวให้กินพื้นที่ 2 คอลัมน์
- rowspan="2" รวมเซลล์ในคอลัมน์ให้กินพื้นที่ 2 แถว

Merged Header	
Row 1, Column 1	Row 1, Column 2
Merged Row 2 and 3	Row 2, Column 2
	Row 3, Column 2

2.8 แบบฟอร์ม (HTML Forms)

HTML Forms ใช้สำหรับเก็บข้อมูลจากผู้ใช้และส่งข้อมูลไปยังเซิร์ฟเวอร์ โดยใช้แท็ก <form> เป็นโครงสร้างหลัก ซึ่งภายในแบบฟอร์มจะมีองค์ประกอบต่างๆ เช่น ช่องกรอกข้อความ, ปุ่ม, และตัวเลือก เพื่อให้ผู้ใช้สามารถป้อนข้อมูล

2.8.1 โครงสร้างพื้นฐานของ HTML Form

```
html
<form action="/submit" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name"><br><br>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email"><br><br>

  <input type="submit" value="Submit">
</form>
```

อธิบายโครงสร้าง

- <form>: กำหนดแบบฟอร์ม โดย action กำหนด URL ที่จะส่งข้อมูลไป และ method กำหนดวิธีส่งข้อมูล (GET หรือ POST)

- <label>: ใช้เพื่อระบุชื่อหรือคำอธิบายสำหรับองค์ประกอบที่กรอกข้อมูล เช่น ชื่อ หรือ อีเมล
- <input>: เป็นช่องกรอกข้อมูล ซึ่งมีหลายประเภท เช่น:
 - type="text": ช่องกรอกข้อความ
 - type="email": ช่องกรอกอีเมล
 - type="submit": ปุ่มส่งข้อมูล

2.8.2 Input Types ที่สำคัญ:

1. Text Input (ช่องกรอกข้อความ)

```
HTML
<input type="text" id="name" name="name">
```

2. Email Input (ช่องกรอกอีเมล):

```
HTML
<input type="email" id="email" name="email">
```

3. Password Input (ช่องกรอกรหัสผ่าน):

```
HTML
<input type="password" id="password" name="password">
```

4. Radio Buttons (ปุ่มเลือกตัวเลือก):

```
HTML
<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label>
```

```
<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label>
```

5. Checkboxes (กล่องเลือก):

HTML

```
<input type="checkbox" id="subscribe" name="subscribe" value="yes">
<label for="subscribe">Subscribe to newsletter</label>
```

6. Dropdown List (รายการแบบดรอปดาวน์):

HTML

```
<label for="cars">Choose a car:</label>
<select id="cars" name="car">
  <option value="volvo">Volvo</option>
  <option value="bmw">BMW</option>
  <option value="audi">Audi</option>
</select>
```

7. Text Area (พื้นที่สำหรับพิมพ์ข้อความยาว)

Text Area ใช้สำหรับกรอกข้อความที่มีความยาวมาก เช่น คำอธิบาย หรือความคิดเห็น โดยใช้แท็ก <textarea> ซึ่งสามารถกำหนดจำนวนแถวและคอลัมน์ได้

HTML

```
<label for="message">Message:</label>
<textarea id="message" name="message" rows="4" cols="50">
Write your message here...
</textarea>
```

- <textarea>: ใช้เพื่อสร้างพื้นที่กรอกข้อความยาว
- rows: กำหนดจำนวนแถวของพื้นที่กรอกข้อความ
- cols: กำหนดจำนวนคอลัมน์ของพื้นที่กรอกข้อความ

2.8.3 การส่งข้อมูลในฟอร์ม

- Method: GET - ส่งข้อมูลแบบเข้ารหัสใน URL เหมาะกับการส่งข้อมูลที่ไม่สำคัญ เช่น การค้นหา
- Method: POST - ส่งข้อมูลแบบซ่อนใน body ของ HTTP request เหมาะกับการส่งข้อมูลที่มีความปลอดภัยหรือปริมาณมาก

2.9 Semantic Tags และโครงสร้างหน้าเว็บ

HTML Semantics หมายถึงการใช้แท็ก HTML ที่มีความหมายเฉพาะเจาะจงและชัดเจน เพื่อช่วยให้ทั้งเบราว์เซอร์และผู้พัฒนาเข้าใจโครงสร้างและเนื้อหาของเว็บเพจได้ง่ายขึ้น นอกจากนี้ยังช่วยในเรื่องการเข้าถึง

(accessibility) และการจัดอันดับในเครื่องมือค้นหา (SEO) แท้ก็ Semantics จะบอกถึงวัตถุประสงค์ของเนื้อหาที่อยู่ภายใน ไม่ใช่เพียงแค่การจัดรูปแบบของมัน
ตัวอย่างของแท็ก Semantics ที่นิยมใช้



ภาพที่ 5 องค์ประกอบ Semantic

<header> ส่วนหัวของหน้าเว็บหรือส่วนของหน้า

```
html
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <h1>Welcome to My Website</h1>
</head>
```

<nav> - ใช้สำหรับเมนูนำทาง

```
html
<nav>
  <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#services">Services</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>
```


`<article>` - เนื้อหาหลักที่มีความเป็นอิสระ สามารถแยกจากเนื้อหาอื่นได้ เช่น บทความ หรือโพสต์

```
html
```

```
<article>
  <h2>Article Title</h2>
  <p>This is an article about HTML semantics.</p>
</article>
```

`<section>` - ใช้สำหรับจัดกลุ่มเนื้อหาที่มีความเกี่ยวข้องกัน เช่น ส่วนต่างๆ ของหน้าเว็บ

```
html
```

```
<section>
  <h2>Section Title</h2>
  <p>This is a section of the webpage.</p>
</section>
```

`<footer>` - ส่วนท้ายของหน้าเว็บ หรือส่วนของเอกสาร

```
html
```

```
<footer>
  <p>&copy; 2024 My Website</p>
</footer>
```

`<aside>` - ข้อมูลเสริมหรือเนื้อหาที่เกี่ยวข้องกับเนื้อหาหลัก เช่น คำอธิบายข้างๆ หรือโฆษณา

```
html
```

```
<aside>
  <h3>Related Articles</h3>
  <ul>
    <li><a href="#article1">Article 1</a></li>
    <li><a href="#article2">Article 2</a></li>
  </ul>
</aside>
```

ประโยชน์ของการใช้แท็ก Semantics:

1. การเข้าถึง (Accessibility): ผู้ใช้ที่ใช้เครื่องมืออ่านหน้าจอ (screen readers) สามารถเข้าใจโครงสร้างและความหมายของเนื้อหาได้ดียิ่งขึ้น
2. SEO: เครื่องมือค้นหา เช่น Google สามารถทำความเข้าใจโครงสร้างของเว็บเพจได้ชัดเจน ทำให้จัดอันดับได้แม่นยำขึ้น
3. การบำรุงรักษา (Maintainability): โค้ด HTML ที่เป็น Semantics อ่านง่ายและดูแลรักษาได้สะดวก เนื่องจากโครงสร้างที่ชัดเจน

2.10 Class

ในภาษา HTML class เป็นคุณสมบัติ (Attribute) ที่ใช้ในการกำหนดกลุ่มหรือประเภทขององค์ประกอบ (Elements) บนหน้าเว็บ ซึ่งช่วยในการกำหนดลักษณะการแสดงผลหรือการปรับแต่งฟังก์ชันด้วย CSS หรือ JavaScript อย่างยืดหยุ่น โดยองค์ประกอบที่ถูกกำหนดให้มีค่า class เดียวกันสามารถแชร์การตั้งค่าหรือการควบคุมเฉพาะได้ เช่น การจัดรูปแบบสี, ขนาด, ฟอนต์ หรือพฤติกรรมที่ต้องการให้เหมือนกันหลาย ๆ องค์ประกอบ ตัวอย่างการใช้ Class

html

```
<element class="className">เนื้อหา</element>
```

- element: หมายถึงแท็ก HTML ที่เราต้องการกำหนดคุณสมบัติ class เช่น <div>, <p>, <h1> เป็นต้น
- className: เป็นชื่อของ class ที่คุณตั้งขึ้นเอง (ตั้งชื่อตามหลักการตั้งชื่อ CSS จะทำให้เข้าใจง่ายขึ้น) ชื่อนี้สามารถนำมาใช้ซ้ำในหลาย ๆ องค์ประกอบได้

html

```
<!DOCTYPE html>
<html lang="th">
<head>
  <meta charset="UTF-8">
  <title>เว็บไซต์สาริตการใช้คลาส</title>
  <style>
    .heading {
      color: blue;
      font-size: 24px;
    }

    .paragraph {
      color: green;
      font-size: 16px;
    }

    .highlight {
      background-color: yellow;
    }
  </style>
</head>
<body>
  <h1 class="heading">ข้อความหัวข้อ</h1>      <!-- กำหนด class "heading" -->
  <p class="paragraph">นี่คือย่อหน้าที่ 1</p>    <!-- กำหนด class "paragraph" -->
  <p class="paragraph highlight">นี่คือย่อหน้าที่ 2</p> <!-- กำหนดหลาย class
"paragraph" และ "highlight" -->
</body>
```

```
</html>
```

- <h1>: กำหนด class="heading" ซึ่ง CSS จะทำให้ข้อความหัวข้อ <h1> มีสี blue และขนาดฟอนต์ 24px
- <p>: กำหนด class="paragraph" ซึ่งจะทำให้ข้อความในย่อหน้ามีสี green และฟอนต์ขนาด 16px
- การใช้หลาย class: ในย่อหน้าที่สอง <p> กำหนด class="paragraph highlight" ซึ่งทั้งสองคลาส (paragraph และ highlight) ถูกนำมาใช้พร้อมกัน ทำให้ข้อความมีสีเขียว (จากคลาส paragraph) และพื้นหลังสีเหลือง (จากคลาส highlight)

2.10.1 การเลือกและจัดการ class ใน CSS

เมื่อกำหนด class ให้กับ HTML แล้ว ผู้พัฒนาจะสามารถใช้ CSS เพื่อจัดการการแสดงผลขององค์ประกอบนั้นได้ โดยการอ้างอิง class ใน CSS จะใช้เครื่องหมายจุด (.) นำหน้าชื่อ class

```
html
```

```
.main-title {
    color: blue;
    font-size: 24px;
}

/* การเลือกคลาส.content ใน CSS */
.content {
    color: red;
    margin: 20px;
}

/* การเลือกคลาส.highlight เพื่อกำหนดพื้นหลัง */
.highlight {
    background-color: yellow;
}
```

จากตัวอย่าง ใน CSS

- .main-title จะปรับสีของข้อความให้อยู่ในสี blue และขนาดฟอนต์เป็น 24px
- .content จะกำหนดสีข้อความเป็นสี red และเพิ่มขนาด margin รอบ ๆ 20 พิกเซล
- .highlight จะเปลี่ยนพื้นหลังเป็นสีเหลือง

2.11 ID

id attribute ใน HTML ถูกใช้เพื่อระบุชื่อที่ไม่ซ้ำกันสำหรับองค์ประกอบ (element) บนหน้าเว็บ ซึ่งแต่ละ id ต้องเป็นค่าเฉพาะที่ไม่ซ้ำกับองค์ประกอบอื่นในหน้าเว็บนั้นๆ ทำให้สามารถระบุและเข้าถึงองค์ประกอบได้อย่างเจาะจง ไม่ว่าจะเป็นเพื่อการจัดการสไตล์ด้วย CSS หรือการเข้าถึงด้วย JavaScript มีลักษณะการใช้งานดังนี้

html

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #myHeader {
        color: blue;
        font-size: 24px;
        text-align: center;
      }
      #introduction {
        color: gray;
        font-style: italic;
      }
    </style>
  </head>
  <body>

    <h1 id="myHeader">My Header</h1>
    <p id="introduction">This is an introduction paragraph.</p>

  </body>
</html>
```

การใช้ id ใน CSS สามารถใช้ id เพื่อกำหนดสไตล์ที่เจาะจงสำหรับองค์ประกอบใน CSS ได้ โดยการใช้เครื่องหมาย # นำหน้า id ที่ต้องการ

- แท็ก <h1> มี id="myHeader" ที่ใช้ระบุว่าเป็นหัวข้อหลัก จะทำให้ข้อความหัวข้อมีสีฟ้าและขนาด 24px
- แท็ก <p> มี id="introduction" ที่ใช้ระบุว่าเป็นย่อหน้าแนะนำ จะทำให้ย่อหน้ามีสีเทาและเป็นตัวเอียง

2.11.1 ความแตกต่างระหว่าง Class และ ID

ความแตกต่างระหว่าง class และ id ใน HTML คือ class สามารถนำไปใช้กับหลายองค์ประกอบพร้อมกันภายในหน้าเว็บเดียวเพื่อจัดรูปแบบหรือควบคุมพฤติกรรมที่เหมือนกัน ทำให้มีความยืดหยุ่นในการใช้กับกลุ่มองค์ประกอบ ขณะที่ id ใช้เพื่อระบุองค์ประกอบที่มีเอกลักษณ์และไม่ซ้ำกันในแต่ละหน้า เหมาะสำหรับการควบคุมองค์ประกอบเฉพาะเจาะจง class ถูกอ้างอิงใน CSS โดยใช้เครื่องหมายจุด (.) และสามารถใช้ได้หลายครั้ง

ส่วน id อ้างอิงโดยเครื่องหมาย # และควรใช้เพียงครั้งเดียวในหน้าเดียว การใช้ id มีความสำคัญเฉพาะตัวและมีลำดับความสำคัญสูงกว่า class เมื่อเกิดการชนกันใน CSS หรือ JavaScript

HTML

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #header {
      color: blue;
      text-align: center;
    }
    .title {
      font-size: 24px;
    }
    .content {
      color: gray;
    }
    #footer {
      text-align: right;
      font-size: 12px;
    }
  </style>
</head>
<body>

  <h1 id="header" class="title">Main Header</h1>
  <h2 class="title">Sub Header</h2>

  <p class="content">This is some text content.</p>
  <p id="footer">Footer content here.</p>

</body>
</html>
```

ในตัวอย่างนี้:

- id="header" ใช้เพื่อจัดการกับหัวข้อหลัก <h1> ที่มีสไตล์ไม่ซ้ำกับใคร
- class="title" ใช้สำหรับหัวข้อหลายๆ อัน (<h1> และ <h2>) ที่ต้องการขนาดฟอนต์เดียวกัน
- id="footer" ใช้เพื่อระบุย่อหน้าส่วนท้ายที่จัดการตำแหน่งและขนาดฟอนต์แตกต่างจากเนื้อหาทั่วไป

แบบฝึกหัดท้ายบท

16. สร้างไฟล์ HTML ที่แสดงข้อความต่อไปนี้ในหน้าเว็บ:

- หัวข้อหลัก: "เว็บไซต์แรกของฉัน"
- เนื้อหาในย่อหน้า: "สวัสดี! นี่คือเว็บไซต์ที่ฉันสร้างโดยใช้ HTML"

สร้างลิสต์เรียงลำดับของอาหารที่คุณชอบ 3 อย่าง และลิสต์ไม่เรียงลำดับของกิจกรรมที่คุณชอบทำ 3 อย่าง

สร้างตารางเก็บข้อมูลของเพื่อนอย่างน้อย 2 คน โดยมีคอลัมน์ 'ชื่อ' และ 'อายุ'

สร้างฟอร์มที่มี:

- ช่องให้กรอก "ชื่อ"
- ช่องให้กรอก "อีเมล"
- ปุ่มส่งข้อมูล