**ใบงานที่ 3**

**เรื่อง ภาษา HTML**

**วัตถุประสงค์เชิงพฤติกรรม**

1. ใช้แท็กพื้นฐาน HTML5 ได้อย่างเหมาะสม

2. สร้างหน้าเว็บไซต์ตามที่กำหนดได้อย่างถูกต้อง

3. บันทึก ไฟล์ HTML และเปิดดูผลลัพธ์บนเว็บเบราว์เซอร์ได้

**เครื่องมือที่ใช้**

1. โปรแกรม Text Editor (Visual Studio Code)

2. Web Browser (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari)

**ทฤษฎีที่เกี่ยวข้อง**

ภาษา HTML (Hypertext Markup Language) เป็นภาษามาตรฐานที่ใช้ในการสร้างและออกแบบเว็บเพจบนอินเทอร์เน็ต ไม่ใช่ภาษาโปรแกรมมิ่ง แต่เป็นภาษา Markup ที่ใช้ "แท็ก (Tag)" ในการกำหนดโครงสร้างและเนื้อหาของเว็บเพจ เพื่อให้เว็บเบราว์เซอร์สามารถเข้าใจและแสดงผลได้อย่างถูกต้อง

**1. ความเป็นมาของ HTML5:**

HTML มีการพัฒนามาอย่างต่อเนื่องตั้งแต่ทศวรรษที่ 90 โดยมีเวอร์ชันสำคัญเช่น HTML2.0 (ค.ศ. 1995), HTML3.0 (ค.ศ. 1997), HTML4.0 (ค.ศ. 1997) และ HTML4.01 (ค.ศ. 1999) โดยองค์กร W3C (World Wide Web Consortium) เป็นผู้ดูแลมาตรฐาน นำโดย นายทิม เบอร์เนอร์ส ลี (Tim Berners-Lee) ในปี ค.ศ. 2014 W3C ได้ประกาศใช้งาน HTML5 อย่างเป็นทางการ ซึ่งได้รับการสนับสนุนอย่างดีจากเบราว์เซอร์ค่ายต่างๆ (Safari, Chrome, Internet Explorer, Firefox และ Opera)

**2. คุณลักษณะที่น่าสนใจของ HTML5:**

- Audio/Video Support: สามารถเล่นไฟล์เสียงและวิดีโอได้โดยตรงโดยไม่ต้องติดตั้ง Plug-in เพิ่มเติม

- New Format Input: เพิ่มชนิดของ Input Form แบบใหม่ เช่น E-mail, Number, URL, DateTime, Range เพื่อรับข้อมูลที่เฉพาะเจาะจง

- Semantic Markup Page Layout: มีแท็กใหม่ที่ช่วยจัดโครงสร้างเว็บเพจให้มีความหมายมากขึ้น เช่น <nav>, <header>, <aside>, <footer>, <section>, <article>

- Canvas Graphics: ช่วยให้สามารถวาดกราฟิกแบบ 2 มิติ หรือสร้างเกมบนเว็บเพจได้โดยใช้ JavaScript ควบคุม

- Drag & Drop: รองรับการลากและวางออบเจกต์บนหน้าเพจ

- Web Storage: กลไกการจัดเก็บข้อมูลทางฝั่งเบราว์เซอร์ที่ซับซ้อนและดีกว่า Cookie

- Geolocation: ระบุตำแหน่งทางภูมิศาสตร์ของผู้ใช้

**3. อีลีเมนต์ (Element) และการใช้แท็ก (Tag):**

- อีลีเมนต์ (Element): คือส่วนประกอบที่เป็นหน่วยย่อยของโครงสร้าง HTML แต่ละอีลีเมนต์จะใช้แท็กในการกำหนดขอบเขตและลักษณะ

- แท็ก (Tag): คือเครื่องหมายเฉพาะของภาษา HTML ที่ใช้ระบุรูปแบบคำสั่ง HTML ภายในเครื่องหมาย < และ > (less than bracket และ greater than bracket)

- แท็กเปิด (start tag): <html> เพื่อบอกจุดเริ่มต้นของอีลีเมนต์

- แท็กปิด (end tag): </html> เพื่อบอกจุดสิ้นสุดของอีลีเมนต์

**4. ประเภทของอีลีเมนต์:**

- Void Elements: มีเพียงแท็กเปิดก็สามารถใช้งานได้เลย ไม่ต้องมีแท็กปิด เช่น <br>, <hr> (บางครั้งเขียนเป็น <hr /> เพื่อบอกว่าเป็น Void Element)

- Raw Text Elements: ประกอบด้วยแท็กเปิด, ข้อความ และแท็กปิด เนื้อหาภายในเป็นข้อความ ไม่สามารถบรรจุอีลีเมนต์อื่นได้ เช่น <script>, <style>

- Nested Elements: ประกอบด้วยแท็กเปิด, ข้อความ หรือ อีลีเมนต์อื่น ๆ และแท็กปิด เป็นการซ้อนกันของอีลีเมนต์ เช่น <ul> ที่มี <li> อยู่ภายใน โดย <ul> เป็น parent element และ <li> เป็น child element

**5. โครงสร้างพื้นฐานของ HTML5:**

เอกสาร HTML5 มีโครงสร้างหลัก 3 ส่วน:

- ส่วนประกาศ: <!DOCTYPE html>

- ส่วนหัวของเว็บเพจ: <head>...</head> (ข้อมูลที่ไม่แสดงผลบนหน้าเว็บโดยตรง เช่น <title>, <meta charset="UTF-8">)

- ส่วนของเนื้อหาของเว็บเพจ: <body>...</body> (เนื้อหาที่แสดงผลบนหน้าเว็บ)

**6. คำสั่งพื้นฐานของโครงสร้าง HTML**

- <!DOCTYPE html>: คำสั่งประกาศชนิดเอกสารว่าเป็น HTML5

- <html>...</html>: แท็กหลักที่ครอบคลุมเนื้อหาทั้งหมดของเอกสาร HTML

- <head>...</head>: ส่วนหัวของเอกสาร ใช้กำหนดข้อมูลที่ไม่แสดงบนหน้าเว็บโดยตรง

- <title>...</title>: กำหนดชื่อเรื่องของเว็บเพจที่แสดงบนแถบชื่อเรื่องของเบราว์เซอร์

- <meta charset="UTF-8">: กำหนดชุดตัวอักษรเพื่อให้แสดงผลภาษาไทยได้อย่างถูกต้อง

- <body>...</body>: ส่วนเนื้อหาหลักของเอกสาร ที่จะแสดงผลบนหน้าเว็บจริง

**ตัวอย่างโครงสร้าง HTML5:**

<!DOCTYPE html>

<html>

<head>

<title>ชื่อเว็บเพจ</title>

<meta charset="UTF-8">

</head>

<body>

<!-- เนื้อหาของเว็บเพจ -->

</body>

</html>

**6. การแสดงภาษาไทยใน HTML:**

การใส่แท็ก <meta charset="UTF-8"> ภายในแท็ก <head> เป็นสิ่งสำคัญเพื่อให้เบราว์เซอร์ทราบว่าเว็บเพจใช้ชุดตัวอักษร UTF-8 ซึ่งรองรับภาษาไทยและภาษาอื่นๆ ได้อย่างถูกต้อง นอกจากนี้ยังสามารถใช้ <html lang="th"> เพื่อระบุภาษาของเนื้อหาในเอกสารว่าเป็นภาษาไทย

**7. แท็กข้อความพื้นฐาน :**

- <h1>...</h1> (ถึง <h6>...</h6>): แท็กสำหรับหัวข้อ (Heading) โดย <h1> คือหัวข้อที่ใหญ่ที่สุดตัวอย่าง: <h1>ยินดีต้อนรับ</h1>

- <p>...</p>: แท็กสำหรับย่อหน้า (Paragraph) ตัวอย่าง: <p>นี่คือข้อความในย่อหน้าแรก</p>

- <br>: แท็กขึ้นบรรทัดใหม่ (Line Break) เป็น Void Element (ไม่มีแท็กปิด) ตัวอย่าง: ข้อความบรรทัดแรก<br>ข้อความบรรทัดที่สอง

- <hr>: แท็กสำหรับสร้างเส้นคั่นแนวนอน (Horizontal Rule) เป็น Void Element (ไม่มีแท็กปิด) ตัวอย่าง: <hr>

- <b>...</b> หรือ <strong>...</strong>: แท็กทำให้ข้อความเป็นตัวหนา (Bold / Strong Importance) ตัวอย่าง: <b>ข้อความตัวหนา</b>

- <i>...</i> หรือ <em>...</em>: แท็กทำให้ข้อความเป็นตัวเอียง (Italic / Emphasis) ตัวอย่าง: <i>ข้อความตัวเอียง</i>

- <u>...</u>: แท็กทำให้ข้อความมีขีดเส้นใต้ (Underline) ตัวอย่าง: <u>ข้อความขีดเส้นใต้</u>

**8.สร้างรายการ (List):**

โดยใช้แท็กรายการ :

- รายการแบบไม่มีลำดับ (Unordered List): ใช้แท็ก <ul> ครอบแท็ก <li>

**ตัวอย่างการใช้งาน**</ul>**:**

<h3>อาหารที่ฉันชอบ:</h3>

<ul>

<li>ผัดไทย</li>

<li>ส้มตำ</li>

<li>ข้าวผัด</li>

</ul>

- รายการแบบมีลำดับ (Ordered List): ใช้แท็ก <ol> ครอบแท็ก <li>

**ตัวอย่างการใช้งาน**</ol>**:**

<h3>ขั้นตอนการเรียน HTML เบื้องต้น:</h3>

<ol>

<li>เปิด Text Editor</li>

<li>เขียนโค้ด HTML</li>

<li>บันทึกไฟล์ .html</li>

<li>เปิดด้วยเว็บเบราว์เซอร์</li>

</ol>

**9. การสร้างตารางด้วย HTML**

การสร้างตารางใน HTML ใช้สำหรับแสดงข้อมูลในรูปแบบแถวและคอลัมน์ ทำให้ข้อมูลมีความเป็นระเบียบและอ่านง่าย นี่คือแท็กพื้นฐานและแท็กเพิ่มเติมที่ใช้ในการกำหนดโครงสร้างตาราง พร้อมความหมายและ**ตัวอย่างการใช้งาน : แท็กพื้นฐานสำหรับการสร้างตาราง**

- <table>...</table> เป็นแท็กหลักที่ใช้สำหรับกำหนดจุดเริ่มต้นและจุดสิ้นสุดของตารางทั้งหมด เนื้อหาทั้งหมดที่เกี่ยวข้องกับตารางจะต้องอยู่ภายในแท็กนี้

ตัวอย่าง:

<table>

<!-- เนื้อหาตารางจะอยู่ภายในนี้ -->

</table>

- <tr>...</tr> (Table Row) เป็นแท็กที่ใช้สำหรับกำหนด แต่ละแถว ในตาราง แต่ละแถวจะประกอบไปด้วยเซลล์ข้อมูล (หรือเซลล์หัวข้อ)

ตัวอย่าง:

<table>

<tr>

<!-- เซลล์ในแถวนี้ -->

</tr>

<tr>

<!-- เซลล์ในแถวถัดไป -->

</tr>

</table>

- <th>...</th> (Table Header) เป็นแท็กที่ใช้สำหรับกำหนด เซลล์หัวข้อ ของตาราง โดยทั่วไปจะใช้ในแถวแรกของตารางเพื่อระบุชื่อคอลัมน์ ข้อความภายในแท็ก <th> มักจะแสดงผลเป็น ตัวหนา และอยู่ กึ่งกลาง โดยอัตโนมัติโดยเบราว์เซอร์

ตัวอย่าง:

<table>

<tr>

<th>ชื่อสินค้า</th>

<th>ราคา</th>

</tr>

</table>

- <td>...</td> (Table Data) เป็นแท็กที่ใช้สำหรับกำหนด เซลล์ข้อมูล ทั่วไปในตาราง แต่ละ <td> จะเป็นข้อมูลหนึ่งช่องในแต่ละแถว ซึ่งเป็นส่วนของข้อมูลจริงที่ปรากฏในตาราง

ตัวอย่าง:

<table>

<tr>

<th>ชื่อสินค้า</th>

<th>ราคา</th>

</tr>

<tr>

<td>ปากกา</td>

<td>15 บาท</td>

</tr>

</table>

**แท็กโครงสร้างตารางเพิ่มเติม (Semantic Table Structure)**

แท็กเหล่านี้ช่วยเพิ่มความหมายและโครงสร้างให้กับตาราง ทำให้เข้าใจง่ายขึ้นสำหรับเบราว์เซอร์และเครื่องมือช่วยการเข้าถึง (Accessibility tools)

- <thead>...</thead> (Table Head) ใช้สำหรับจัดกลุ่ม แถวหัวข้อ ของตาราง ควรวาง <tr> ที่มี <th> อยู่ภายในแท็กนี้ เพื่อแยกส่วนหัวของตารางออกจากส่วนเนื้อหา

ตัวอย่าง:

<table>

<thead>

<tr>

<th>สินค้า</th>

<th>จำนวน</th>

</tr>

</thead>

<!-- ส่วนเนื้อหาตาราง -->

</table>

- <tbody>...</tbody> (Table Body) ใช้สำหรับจัดกลุ่ม แถวข้อมูลหลัก ของตาราง ควรวาง <tr> ที่มี <td> อยู่ภายในแท็กนี้ เพื่อแยกส่วนเนื้อหาของตารางออกจากส่วนหัวและส่วนท้าย

ตัวอย่าง:

<table>

<thead>...</thead>

<tbody>

<tr>

<td>นม</td>

<td>2 กล่อง</td>

</tr>

<tr>

<td>ขนมปัง</td>

<td>1 แถว</td>

</tr>

</tbody>

<!-- ส่วนท้ายตาราง -->

</table>

- <tfoot>...</tfoot> (Table Foot) ใช้สำหรับจัดกลุ่ม แถวส่วนท้าย ของตาราง (เช่น ผลรวม, ค่าเฉลี่ย, หมายเหตุ) มักจะปรากฏอยู่ด้านล่างสุดของตาราง แม้จะเขียนโค้ดไว้ก่อน <tbody> ก็ตาม (เบราว์เซอร์จะจัดเรียงให้เอง)

ตัวอย่าง:

<table>

<thead>...</thead>

<tbody>...</tbody>

<tfoot>

<tr>

<td colspan="2">รวมสินค้า: 2 รายการ</td>

</tr>

</tfoot>

</table>

- <caption>...</caption> (Table Caption) ใช้สำหรับกำหนด คำอธิบายหรือชื่อของตาราง ที่ให้ข้อมูลสรุปเกี่ยวกับเนื้อหาของตารางนั้นๆ ต้องวางไว้เป็นลูกแรกของแท็ก <table> โดยปกติจะแสดงอยู่เหนือตาราง

ตัวอย่าง:

<table>

<caption>ตารางแสดงยอดขายประจำเดือน</caption>

<thead>...</thead>

<tbody>...</tbody>

</table>

**แอตทริบิวต์สำคัญสำหรับการรวมเซลล์ (Merging Cells)**

แอตทริบิวต์เหล่านี้ใช้กับแท็ก <th> หรือ <td> เพื่อรวมเซลล์เข้าด้วยกัน ทำให้ตารางมีความยืดหยุ่นในการจัดวางข้อมูลที่ซับซ้อน

- colspan (Column Span) ใช้เพื่อระบุว่าเซลล์นั้นจะ ขยายครอบคลุมไปกี่คอลัมน์ (รวมเซลล์ในแนวนอน) ค่าของ colspan คือจำนวนคอลัมน์ที่เซลล์นั้นจะครอบคลุม

ตัวอย่าง:

<table>

<tr>

<td colspan="2">ข้อมูลนี้ครอบคลุม 2 คอลัมน์</td>

<td>คอลัมน์ที่ 3</td>

</tr>

<tr>

<td>คอลัมน์ 1</td>

<td>คอลัมน์ 2</td>

<td>คอลัมน์ 3</td>

</tr>

</table>

- rowspan (Row Span) ใช้เพื่อระบุว่าเซลล์นั้นจะ ขยายครอบคลุมไปกี่แถว (รวมเซลล์ในแนวตั้ง) ค่าของ rowspan คือจำนวนแถวที่เซลล์นั้นจะครอบคลุม

ตัวอย่าง:

<table>

<tr>

<td rowspan="2">ข้อมูลนี้ครอบคลุม 2 แถว</td>

<td>แถว 1 คอลัมน์ 2</td>

</tr>

<tr>

<!-- แถวนี้จะไม่มี td แรก เพราะถูก rowspan ของแถวบนครอบคลุมไปแล้ว -->

<td>แถว 2 คอลัมน์ 2</td>

</tr>

</table>

**ขั้นตอนการทดลอง**

**ส่วนที่ 1 :** **เปิดโปรแกรม Text Editor (Visual Studio Code)**

1. ดาวน์โหลดไฟล์ ใบงานที่ 3

2. เปิดโปรแกรมประมวลผลคำ ()

3. คลิก เปิด (Open) -> เลือกไฟล์ที่ดาวน์โหลด

4. เปิดโปรแกรม Visual Studio Code

5. เข้าสู่ระบบ Github ภายในโปรแกรม Visual Studio Code

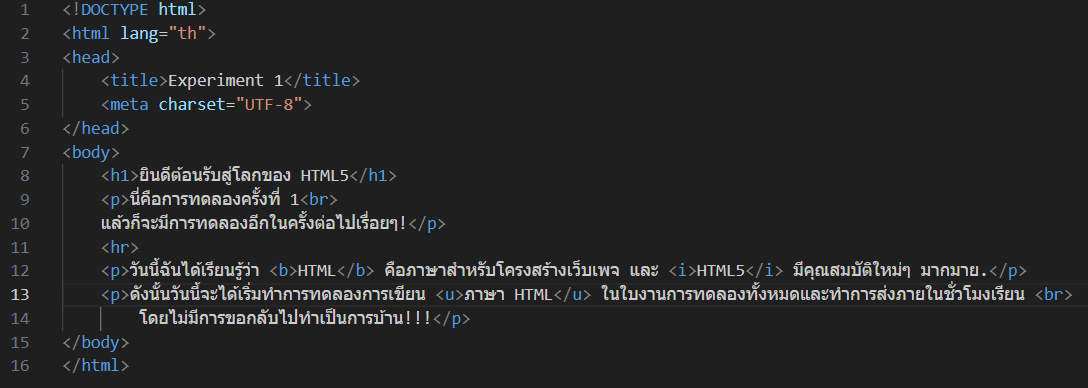
6. สร้างโฟลเดอร์ โดยตั้งชื่อดังนี้ xxx-Name-03 (xxx คือ เลข 3 ตัวท้ายของรหัสประจำตัวนักศึกษา)

**ส่วนที่ 2 : การทดลอง**

**การทดลองครั้งที่ 1 :** การสร้างเว็บเพจแรกด้วยโครงสร้างพื้นฐานและแท็กข้อความ

7. สร้างไฟล์ โดยตั้งชื่อว่า experiment1.html

8. พิมพ์ชุดคำสั่ง HTML ดังรูปที่ 3.1 ลงในหน้าต่างโปรแกรม Visual Studio Code



รูปที่ 3.1 การพิมพ์ชุดคำสั่งด้วยแท็กพื้นฐาน

9. บันทึกไฟล์ โดยสามารถทำได้ดังนี้

- ไปที่เมนู File (ไฟล์)

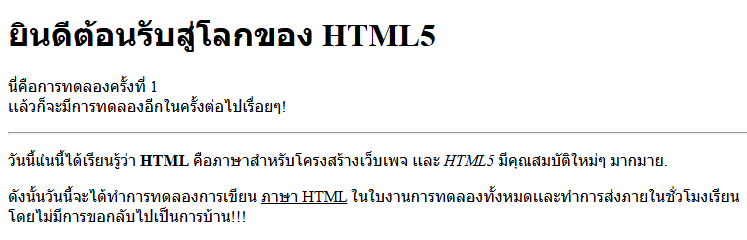
- คลิกปุ่ม Save (บันทึก)

10. เปิดดูผลลัพธ์ในเว็บเบราว์เซอร์ โดยสามารถทำได้ดังนี้

- ไปที่โฟลเดอร์ที่คุณบันทึกไฟล์ experiment1.html ไว้

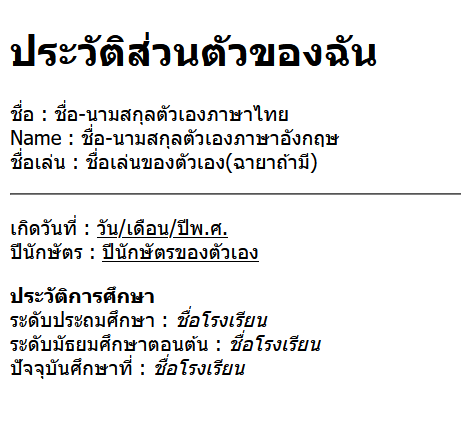
- ดับเบิลคลิกที่ไฟล์ experiment1.html ไฟล์จะถูกเปิดขึ้นมาในเว็บเบราว์เซอร์ที่คุณตั้งค่าไว้เป็นค่าเริ่มต้น

**บันทึกผลการทดลอง**



(ความสำเร็จ 10%)

11. แก้ไขชุดคำสั่ง experiment1.html ให้ได้ผลลัพธ์การแสดงผลดังรูปที่ 3.2



รูปที่ 3.2 แก้ไขชุดคำสั่ง experiment1.html

12. บันทึกไฟล์ (กด Ctrl+S) และรีเฟรช (F5 หรือปุ่มวงกลมลูกศร)

**บันทึกผลการทดลอง** (บันทึกรูปภาพการแสดงผลบนเว็บเบราว์เซอร์)

(ความสำเร็จ 20%)

**การทดลองครั้งที่ 2 :** การใช้งานแอตทริบิวต์ contenteditable และระบุ ID/Class

13. สร้างไฟล์ โดยตั้งชื่อว่า experiment2.html

14. พิมพ์ชุดคำสั่ง HTML ดังรูปที่ 3.3 ลงในหน้าต่างโปรแกรม Visual Studio Code

A screen shot of a computer program

AI-generated content may be incorrect.

รูปที่ 3.3 การใช้งานแอตทริบิวต์ contenteditable และระบุ ID/Class

**บันทึกผลการทดลอง** (บันทึกรูปภาพการแสดงผลบนเว็บเบราว์เซอร์)

15. แก้ไขชุดคำสั่ง experiment2.html โดยสามารถทำได้ดังนี้

- เพิ่มย่อหน้าใหม่ที่สามารถแก้ไขเนื้อหาได้ โดยกำหนด contenteditable="true"

- กำหนดให้ id = “editP”

- กำหนดให้ class = “edit-box”

- บันทึกไฟล์ (กด Ctrl+S) และรีเฟรช (F5 หรือปุ่มวงกลมลูกศร)

**บันทึกผลการทดลอง**

- เปลี่ยนการจัดรูปแบบ ดังนี้

(1) เปลี่ยนสีพื้นหลัง

(2) เปลี่ยนสีเส้นขอบ

(3) เปลี่ยนสไตล์ของเส้นขอบของ .edit-box ให้เป็นแบบ dotted

(4) บันทึกไฟล์ (กด Ctrl+S) และรีเฟรช (F5 หรือปุ่มวงกลมลูกศร)

**บันทึกผลการทดลอง** (บันทึกรูปภาพการแสดงผลบนเว็บเบราว์เซอร์)

- ปิดการแก้ไขเนื้อหาในส่วนของแท็กที่มี id="editP"

- บันทึกไฟล์ (กด Ctrl+S) และรีเฟรช (F5 หรือปุ่มวงกลมลูกศร)

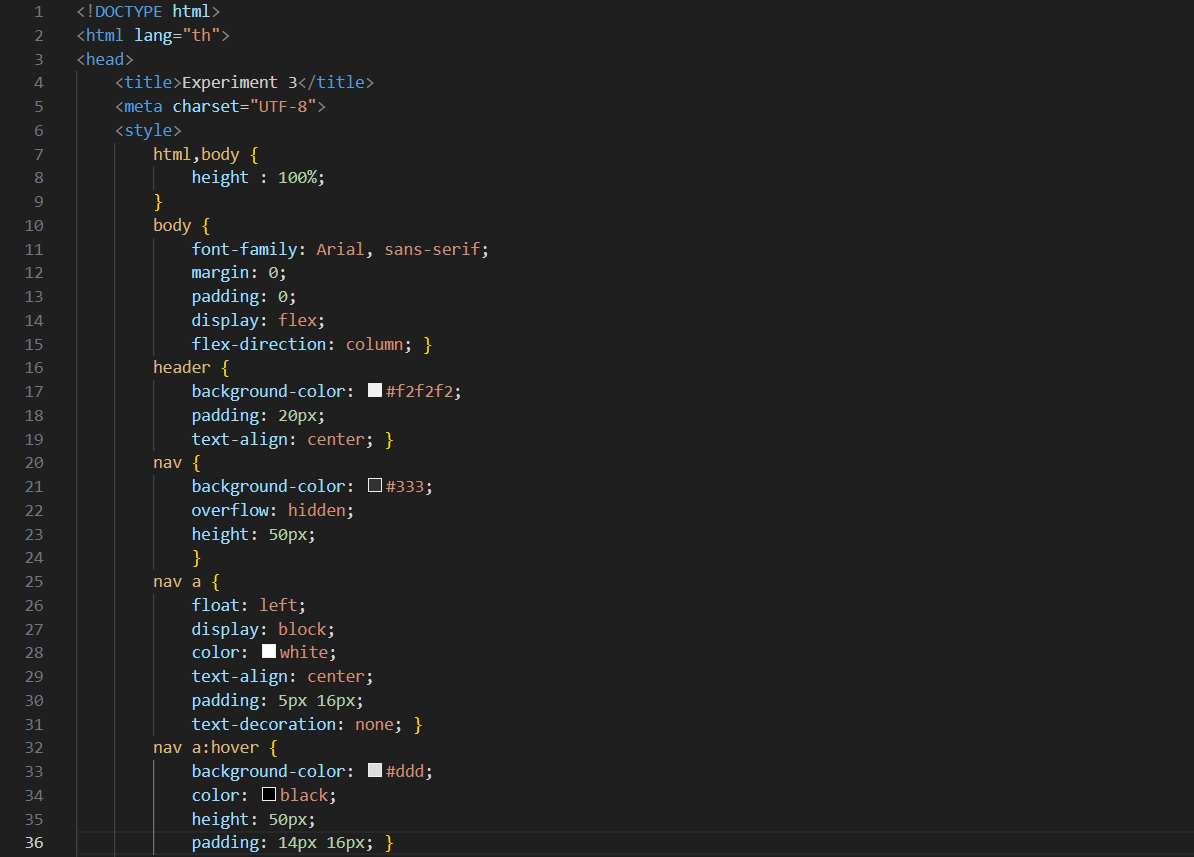
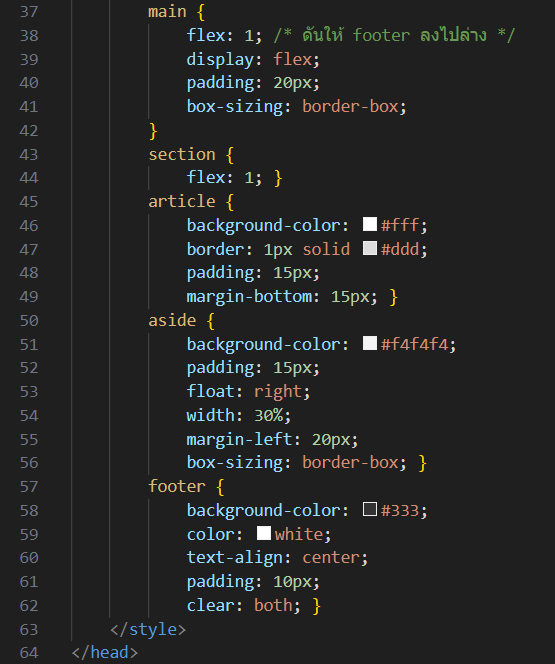
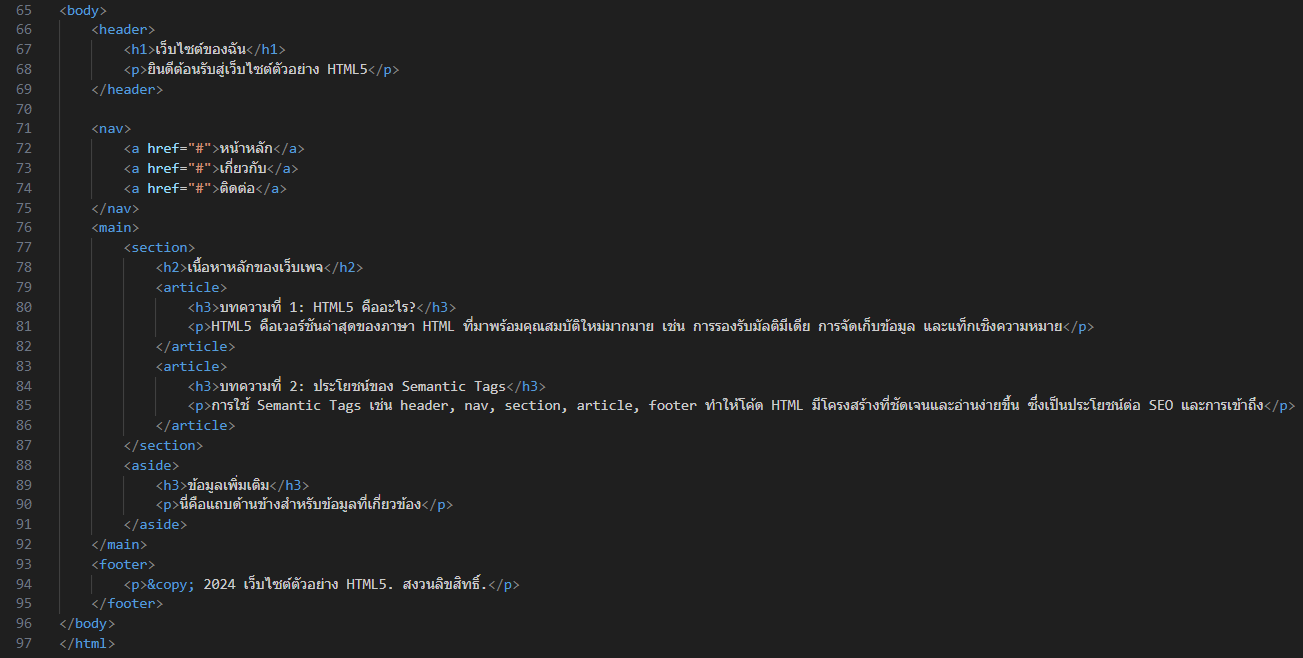
**บันทึกผลการทดลอง**

(ความสำเร็จ 40%)

**การทดลองที่ 3 :** การใช้งาน Semantic Elements ของ HTML

16. สร้างไฟล์ โดยตั้งชื่อว่า experiment3.html

17. พิมพ์ชุดคำสั่ง HTML ดังรูปที่ 3.4 ลงในหน้าต่างโปรแกรม Visual Studio Code



รูปที่ 3.4 การใช้งาน Semantic Elements ของ HTML

**บันทึกผลการทดลอง** (บันทึกรูปภาพการแสดงผลบนเว็บเบราว์เซอร์)

18. แก้ไขชุดคำสั่ง experiment3.html โดยสามารถทำได้ดังนี้

- เพิ่มแท็ก <section> ใหม่ภายใน <body>

- เพิ่มลิงก์ใน Navigation เป็นลิงค์ "แกลเลอรี"

- บันทึกไฟล์ (กด Ctrl+S) และรีเฟรช (F5 หรือปุ่มวงกลมลูกศร)

**บันทึกผลการทดลอง** (บันทึกรูปภาพการแสดงผลบนเว็บเบราว์เซอร์)

(ความสำเร็จ 60%)

**การทดลองที่ 4 :** การใช้งาน Input Type ใน HTML

19. สร้างไฟล์ โดยตั้งชื่อว่า experiment4.html

20. พิมพ์ชุดคำสั่ง HTML ดังรูปที่ 3.5 ลงในหน้าต่างโปรแกรม Visual Studio Code

21. บันทึกไฟล์ (กด Ctrl+S) และรีเฟรช (F5 หรือปุ่มวงกลมลูกศร)

รูปที่ 3.5 การใช้งาน Input Type

**บันทึกผลการทดลอง** (บันทึกรูปภาพการแสดงผลบนเว็บเบราว์เซอร์)

22. แก้ไขชุดคำสั่ง experiment4.html โดยสามารถทำได้ดังนี้

- เพิ่ม Input Type ประเภท email

- บันทึกไฟล์ (กด Ctrl+S) และรีเฟรช (F5 หรือปุ่มวงกลมลูกศร)

**บันทึกผลการทดลอง** (บันทึกรูปภาพการแสดงผลบนเว็บเบราว์เซอร์)

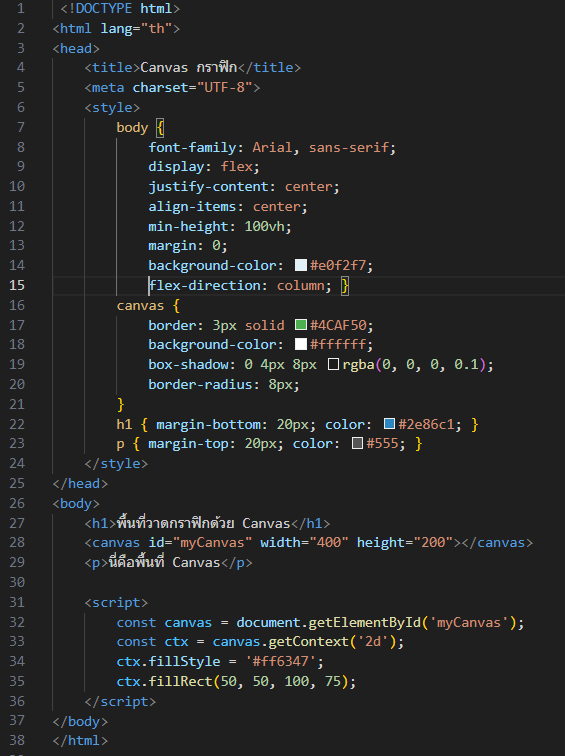
(ความสำเร็จ 80%)

**การทดลองที่ 5 :** การใช้งานแท็ก canvas สำหรับกราฟิก (เบื้องต้น)

23. สร้างไฟล์ โดยตั้งชื่อว่า experiment5.html

24. พิมพ์ชุดคำสั่ง HTML ดังรูปที่ 3.6 ลงในหน้าต่างโปรแกรม Visual Studio Code

25. บันทึกไฟล์ (กด Ctrl+S) และรีเฟรช (F5 หรือปุ่มวงกลมลูกศร)



รูปที่ 3.6 การใช้งานแท็ก canvas

**บันทึกผลการทดลอง** (บันทึกรูปภาพการแสดงผลบนเว็บเบราว์เซอร์)

26. แก้ไขชุดคำสั่ง experiment5.html โดยสามารถทำได้ดังนี้

- กำหนดให้มีการแสดงผลเป็นสัญลักษณ์ “ -\_- ”

- บันทึกไฟล์ (กด Ctrl+S) และรีเฟรช (F5 หรือปุ่มวงกลมลูกศร)

**บันทึกผลการทดลอง** (บันทึกรูปภาพการแสดงผลบนเว็บเบราว์เซอร์)

(ความสำเร็จ 100%)

**สรุปผลการทดลอง**

**คำถามท้ายการทดลอง**

1. จากคุณลักษณะใหม่ของ HTML5 ที่ได้เรียนรู้ คุณสมบัติใดที่คุณคิดว่ามีประโยชน์มากที่สุดในการสร้างเว็บในปัจจุบัน และเพราะเหตุใด? (อธิบายอย่างน้อย 1 คุณสมบัติ)

ตอบ

2. อธิบายความแตกต่างระหว่าง Void Elements, Raw Text Elements และ Nested Elements พร้อมยกตัวอย่างแท็กที่พบในแต่ละประเภท

ตอบ

3. สมมติว่าคุณต้องการสร้างบล็อกโพสต์บนเว็บไซต์ส่วนตัว หากใช้ Semantic Tags คุณจะใช้แท็กใดบ้างในการจัดโครงสร้างหัวข้อบทความ เนื้อหา และส่วนท้ายของบทความ

ตอบ

4. นอกจากการแสดงวันที่แล้ว คุณสมบัติ New Format Input ของ HTML5 มี type แบบใดอีกบ้างที่คุณคิดว่ามีประโยชน์ในการเก็บข้อมูลจากผู้ใช้ จงยกตัวอย่างและบอกประโยชน์

ตอบ

**เกณฑ์การประเมิน (คะแนนเต็ม 10)**

|  |  |  |
| --- | --- | --- |
| เกณฑ์การประเมิน | รายละเอียด | คะแนนเต็ม |
| 1. การสร้างโครงสร้าง HTML5 และแท็กพื้นฐาน (การทดลองที่ 1) | 2 = สร้างได้ถูกต้องครบถ้วน  1.5 = สร้างได้ถูกต้องเกือบครบถ้วน  1 = สร้างได้บางส่วน แต่มีข้อผิดพลาดเล็กน้อย แสดงผลได้บางส่วน  0.5 = สร้างได้ไม่ถูกต้อง หรือแสดงผลไม่ได้  0 = ไม่ได้ทำ | 2 |
| 2. การประยุกต์ใช้แอตทริบิวต์ (contenteditable, id, class) (การทดลองที่ 2) | 2 = ใช้งานแอตทริบิวต์ได้ถูกต้องและเข้าใจหลักการทำงาน  1.5 = ใช้งานแอตทริบิวต์ได้ถูกต้องแต่ยังไม่เข้าใจหลักการทั้งหมด  1 = ใช้งานแอตทริบิวต์ได้บางส่วน มีข้อผิดพลาด  0.5 = ใช้งานแอตทริบิวต์ไม่ได้  0 = ไม่ได้ทำ | 2 |
| 3. การใช้งาน Semantic Elements (การทดลองที่ 3) | 2 = สร้างโครงสร้างด้วย Semantic Tags ได้ถูกต้องและเหมาะสม  1.5 = สร้างได้แต่การจัดวางอาจยังไม่สมบูรณ์  1 = สร้างได้เพียงบางส่วน  0.5 = สร้างไม่ได้หรือไม่เข้าใจแนวคิด  0 = ไม่ได้ทำ | 2 |
| 4. การใช้งาน Input Type (การทดลองที่ 4) | 2 = สร้าง input type="date" ได้ถูกต้อง และเข้าใจการทำงาน  1.5 = สร้างได้ถูกต้อง แต่ยังไม่เข้าใจการทำงานในเบราว์เซอร์ที่ต่างกัน  1 = สร้างได้แต่มีข้อผิดพลาด หรือไม่แสดงผล  0.5 = สร้างไม่ได้  0 = ไม่ได้ทำ | 2 |
| 5. การสรุปผลการทดลอง | 2 = สรุปได้ครบถ้วน ชัดเจน และเชื่อมโยงกับสิ่งที่เรียนรู้  1.5 = สรุปได้แต่ยังไม่ครบถ้วน หรือไม่เชื่อมโยงกับเนื้อหาทั้งหมด  1 = สรุปได้น้อย หรือมีความเข้าใจผิดบางส่วน  0.5 = สรุปไม่ได้  0 = ไม่ได้ทำ | 2 |
| รวม | | 10 |