

HTML5

Web Programming

Web Programming Introduction - Basic HTML - Semantic Tags
Text Format - Table - Form - Multimedia - Graphic
Drag and Drop - Local Storage - Geolocation API

คำนำ

ในยุคดิจิทัลที่เทคโนโลยีสารสนเทศเข้ามามีบทบาทสำคัญในทุกมิติของชีวิตประจำวัน ไม่ว่าจะเป็นด้านการศึกษา การสื่อสาร การทำงาน หรือแม้แต่การดำเนินธุรกิจ การเข้าถึงข้อมูลและบริการผ่านระบบเครือข่ายอินเทอร์เน็ตได้กลายเป็นกิจวัตรที่ไม่อาจหลีกเลี่ยง ทักษะด้านการพัฒนาเว็บไซต์จึงกลายเป็นองค์ความรู้พื้นฐานที่สำคัญ โดยเฉพาะอย่างยิ่งในกลุ่มนักเรียน นักศึกษา ผู้ประกอบวิชาชีพ และบุคคลทั่วไปที่ต้องการสื่อสารข้อมูลหรือบริการในรูปแบบออนไลน์

HTML5 ถือเป็นภาษาหลักที่ใช้สำหรับสร้างและจัดการเนื้อหาในหน้าเว็บ ซึ่งมีความสามารถในการรองรับเทคโนโลยีเว็บสมัยใหม่ เช่น การจัดการมัลติมีเดีย การโต้ตอบกับผู้ใช้ผ่านฟอร์ม การวาดกราฟิกด้วย Canvas หรือ SVG รวมถึงการจัดเก็บข้อมูลภายในเบราว์เซอร์โดยไม่ต้องเชื่อมต่อกับเซิร์ฟเวอร์ หนังสือเล่มนี้จึงถูกจัดทำขึ้นเพื่อเป็นแนวทางในการศึกษาและฝึกปฏิบัติการใช้งาน HTML5 อย่างเป็นระบบและถูกต้อง

เนื้อหาในหนังสือประกอบด้วยบทเรียนตั้งแต่ระดับพื้นฐาน เช่น ความรู้เกี่ยวกับอินเทอร์เน็ต โครงสร้างของเว็บแอปพลิเคชัน ไปจนถึงการเขียนคำสั่ง HTML5 ที่ใช้บ่อยในการสร้างเว็บไซต์ในชีวิตจริง ได้แก่ คำสั่งสำหรับการจัดรูปแบบข้อความ การสร้างตาราง การใช้งานฟอร์ม การจัดการมัลติมีเดีย การวาดกราฟิก การใช้งาน API ต่าง ๆ เช่น Local Storage และ Geolocation API ซึ่งเป็นหัวข้อที่มีการนำไปใช้อย่างแพร่หลายในวงการพัฒนาเว็บในปัจจุบัน

รูปแบบการนำเสนอในหนังสือเน้นความเข้าใจง่าย เหมาะสำหรับผู้เริ่มต้นโดยไม่จำเป็นต้องมีพื้นฐานด้านการเขียนโปรแกรมมาก่อน มีการยกตัวอย่างประกอบในแต่ละบท พร้อมแบบฝึกหัดที่ช่วยเสริมสร้างความเข้าใจให้แก่ผู้อ่าน สามารถนำไปทดลองทำตามได้ทันที ซึ่งจะช่วยให้การเรียนรู้เป็นไปอย่างราบรื่นและเห็นผลลัพธ์ที่ชัดเจน

นอกจากนี้ หนังสือยังเน้นการเชื่อมโยงความรู้กับการใช้งานจริงในโลกยุคใหม่ โดยเฉพาะในการพัฒนาเว็บไซต์ที่ตอบสนองต่อผู้ใช้งานผ่านอุปกรณ์หลากหลาย การจัดวางเนื้อหาที่สอดคล้องกับมาตรฐานสากลจะช่วยให้ผู้อ่านสามารถต่อยอดไปสู่การเรียนรู้เฟรมเวิร์กหรือเทคโนโลยีเว็บอื่น ๆ ได้อย่างมั่นใจ

ผู้เขียนมีความตั้งใจที่จะถ่ายทอดเนื้อหาให้เป็นลำดับขั้นตอนอย่างชัดเจน และอ้างอิงแหล่งข้อมูลจากเว็บไซต์วิชาการที่เชื่อถือได้ เช่น MDN Web Docs, W3Schools, W3C, HTML5 Rocks และเว็บไซต์อื่น ๆ ที่เกี่ยวข้อง เพื่อให้ผู้อ่านสามารถค้นคว้าเพิ่มเติมและเรียนรู้ได้อย่างต่อเนื่อง

สุดท้ายนี้ ผู้เขียนหวังเป็นอย่างยิ่งว่าหนังสือเล่มนี้จะมีส่วนช่วยพัฒนาทักษะด้านการเขียนเว็บไซต์ด้วย HTML5 ให้แก่ผู้อ่าน ไม่ว่าจะเป็นเพื่อการศึกษ การทำงาน หรือการต่อยอดในสายอาชีพ หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้เขียนยินดีน้อมรับคำแนะนำอันเป็นประโยชน์เพื่อปรับปรุงเนื้อหาให้ดียิ่งขึ้นในโอกาสต่อไป

ด้วยความปรารถนาดี

ศูนย์หนังสือราคานักเรียน

สารบัญ

หน้า

บทที่ 1 ความรู้เบื้องต้นเกี่ยวกับอินเทอร์เน็ตและการเขียนภาษาเอชทีเอ็มแอล	1
•วิวัฒนาการของเทคโนโลยีเว็บ	
•ประวัติการพัฒนาเวิลด์ไวด์เว็บ	
•ยุคของเว็บ	
•สถาปัตยกรรมของเว็บแอปพลิเคชัน	
•ขั้นตอนการพัฒนาเว็บแอปพลิเคชัน	
•เอชทีเอ็มแอล	
•โครงสร้างพื้นฐานของหน้าเว็บและวิวัฒนาการของ HTML	
•วิวัฒนาการของ HTML	
•การเขียนภาษา HTML	
บทที่ 2 คำสั่งพื้นฐาน.....	15
•พื้นฐานความรู้เอชทีเอ็มแอล	
•ชนิดของ HTML Tag	
•คำสั่งย่อหน้า	
•แท็กจัดการรูปภาพ	
•การสร้างเส้นคั่นในหน้าเว็บ	
•การสร้างหัวข้อรายการ	
•กำหนดรูปแบบ	
บทที่ 3 คำสั่งเชิงความหมาย	50
•องค์ประกอบความหมายใน HTML	
•แท็ก <section>	
•แท็ก <article>	
•การบูรณาของ section และ article	
•แท็ก <header>	
•แท็ก <footer>	

- แท็ก <nav>
- แท็ก <aside>
- แท็ก <figure> และ <figcaption>
- แท็ก <main>
- แท็ก <mark>
- แท็ก <time>

บทที่ 4 คำสั่งจัดรูปแบบข้อความ 99

- คำสั่งจัดรูปแบบข้อความใน HTML5
- แท็กที่ใช้จัดรูปแบบโครงสร้างข้อความ
- การใช้งานแท็กต่าง ๆ
- แท็กที่ใช้กับข้อความแบบมีโครงสร้างความหมาย (Semantic Tags)

บทที่ 5 คำสั่งใช้งานตาราง..... 134

- โครงสร้างพื้นฐานของ <table>
- คำสั่งและแท็กย่อยที่ใช้ภายใน <table>
- คุณสมบัติการจัดรูปแบบเพิ่มเติม
- คำสั่ง <table> พร้อมการใช้ CSS
- ไจทท์ที่ใช้งาน <table> พร้อม เฉลย
- การขยายความเกี่ยวกับ <table>

บทที่ 6 คำสั่งฟอร์ม..... 168

- การใช้งานฟอร์ม
- แท็ก <form>
- ประเภทของฟอร์ม
- Input Controls
- semantic input tags
- กลุ่มของ input fields
- Form Submission Controls

บทที่ 7 คำสั่งมัลติมีเดีย 209

- multimedia tags

● โจอทย์และเจลยสำหรับคำสั่ง Multimedia	
● โจอทย์แบบบูรณาการสำหรับ Multimedia Tags ใน HTML5	
บทที่ 8 คำสั่งกราฟิกและแคนวาส.....	227
● Canvas API ใน HTML5 คืออะไร?	
● ฟังก์ชันหลักของ Canvas API	
● การนำ Canvas ไปใช้ในโลจจริง	
● SVG ใน HTML5 คืออะไร?	
● การทำให้ SVG โต้ตอบได้ด้วย JavaScript	
บทที่ 9 คำสั่งแดร็กแอนด์ดร็อป	261
● การใช้งาน Drag and Drop ใน HTML5	
● หลักการทำงานของ Drag and Drop	
● การใช้ Drag and Drop	
บทที่ 10 คำสั่งโลคอลสโตร์จ	293
● Local Storage ใน HTML5	
● คุณสมบัติหลักของ Local Storage	
● วิธีใช้งาน Local Storage	
● ข้อแตกต่างระหว่าง Local Storage และ Session Storage	
● ตัวอย่าง โจอทย์เกี่ยวกับ Local Storage	
บทที่ 11 คำสั่งจีโอโลเคชันเอพีไอ	315
● Geolocation API ใน HTML5	
● การใช้งาน Geolocation API	
● ขั้นตอนการใช้งาน GeoLocation API	
● ตัวอย่างการใช้งาน Geolocation API เพิ่มเติม	
● ตัวอย่างการใช้งาน Geolocation API ที่สามารถนำไปปรับใช้ในแอปพลิเคชัน	
● การทำงานของ Geolocation API โดย JavaScript	
● การใช้งาน Geolocation API กับฟังก์ชันอื่น ๆ	
บรรณานุกรม	393

บทที่ 1

ความรู้เบื้องต้นเกี่ยวกับอินเทอร์เน็ตและการเขียนภาษาเอชทีเอ็มแอล

เนื้อหา

- วิวัฒนาการของเทคโนโลยีเว็บ
- ประวัติการพัฒนาเวปไซต์
- ยุคของเว็บ
- สถาปัตยกรรมของเว็บแอปพลิเคชัน
- ขั้นตอนการพัฒนาเว็บแอปพลิเคชัน
- เอชทีเอ็มแอล
- โครงสร้างพื้นฐานของหน้าเว็บและวิวัฒนาการของ HTML
- วิวัฒนาการของ HTML
- การเขียนภาษา HTML

อินเทอร์เน็ตเป็นปัจจัยสำคัญในชีวิตประจำวัน ปัจจุบันมีทั้งผู้ใช้งานทั่วไปและนักพัฒนาโปรแกรมที่ใช้ประโยชน์จากเครือข่ายอินเทอร์เน็ต สำหรับผู้ใช้งานทั่วไป การเข้าถึงข้อมูลทำได้ผ่านเว็บเบราว์เซอร์ (Browser) ซึ่งมีอินเทอร์เน็ตเฟสแบบกราฟิกที่ช่วยให้ใช้งานได้ง่ายโดยไม่ต้องมีความรู้ด้านเทคนิคเชิงลึกในทางกลับกัน นักพัฒนาโปรแกรมจำเป็นต้องมีความเข้าใจเชิงลึกเกี่ยวกับเทคโนโลยีเว็บ โดยเริ่มจากภาษา HTML ซึ่งเป็นภาษาหลักสำหรับการแสดงผลบนเว็บเบราว์เซอร์และทำงานผ่านโปรโตคอล HTTP นอกจากนี้ การพัฒนาเว็บแอปพลิเคชันยังเกี่ยวข้องกับการติดต่อฐานข้อมูลและการส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตเพื่อแสดงผลบนเบราว์เซอร์ นักพัฒนาต้องเรียนรู้การใช้ภาษาสคริปต์ที่สามารถผสมผสานเข้ากับ HTML เพื่อดึงและจัดการข้อมูลจากฐานข้อมูล รวมถึงการออกแบบการแสดงผลให้เหมาะสมกับผู้ใช้งาน

บทนี้จะนำเสนอความรู้พื้นฐานเกี่ยวกับการพัฒนาเว็บ ก่อนที่ผู้อ่านจะได้ศึกษาเกี่ยวกับ HTML โดยเน้นถึงองค์ประกอบสำคัญและสภาพแวดล้อมที่จำเป็นสำหรับการสร้างเว็บเพจ

วิวัฒนาการของเทคโนโลยีเว็บ

พัฒนาการของเทคโนโลยีเว็บไซต์และบทบาทของเซอร์ทิโมที จอห์น เบร์เนิร์ส-ลี การพัฒนาเทคโนโลยีเว็บไซต์เริ่มต้นจากผลงานของเซอร์ทิโมที จอห์น เบร์เนิร์ส-ลี (Sir Timothy John Berners-Lee) ผู้คิดค้นและพัฒนาเวิลด์ไวด์เว็บ (World Wide Web) ซึ่งส่งผลให้เทคโนโลยีดังกล่าวได้รับการพัฒนาอย่างต่อเนื่องบนระบบอินเทอร์เน็ตและกลายเป็นองค์ประกอบสำคัญในโลกดิจิทัลปัจจุบัน ความก้าวหน้าของเวิลด์ไวด์เว็บส่งเสริมให้เกิดการพัฒนาแอปพลิเคชันบนอินเทอร์เน็ตอย่างแพร่หลาย อันเป็นรากฐานสำคัญของสังคมสารสนเทศ

ประวัติการพัฒนาเวิลด์ไวด์เว็บ

จากข้อมูลที่ระบุใน Wikipedia (http://th.wikipedia.org/wiki/ทิม_เบร์เนิร์ส-ลี) การพัฒนาเวิลด์ไวด์เว็บของเบร์เนิร์ส-ลีสามารถสรุปได้ดังนี้

1. แนวคิดเบื้องต้นและโครงการ ENQUIRE

ในปี พ.ศ. 2523 ขณะที่เบร์เนิร์ส-ลีทำงานอยู่ที่องค์การวิจัยนิวเคลียร์ยุโรป (CERN) เขาได้เสนอแนวคิดเกี่ยวกับ "ข้อความหลายมิติ" (Hypertext) เพื่อนำมาใช้ในการแลกเปลี่ยนและปรับปรุงข้อมูลระหว่างนักวิจัย แนวคิดนี้ถูกนำไปใช้เป็นต้นแบบในโครงการ **ENQUIRE** ซึ่งเป็นระบบที่ช่วยให้สามารถสร้างและเชื่อมโยงเอกสารระหว่างกันได้

2. การบูรณาการแนวคิด Hypertext กับอินเทอร์เน็ต

ในปี พ.ศ. 2532 เมื่อ CERN กลายเป็นศูนย์กลางของเครือข่ายอินเทอร์เน็ตที่ใหญ่ที่สุดในยุโรป เบร์เนิร์ส-ลีเล็งเห็นโอกาสในการนำแนวคิด **Hypertext** มาผนวกเข้ากับระบบอินเทอร์เน็ต เขาได้ร่างข้อเสนอแรกในเดือนมีนาคม พ.ศ. 2532 และได้รับการสนับสนุนจากโรเบิร์ต ไคลิเย (Robert Cailliau) ซึ่งช่วยปรับปรุงข้อเสนอก่อนจะส่งให้ไมค์ เซนดอลล์ (Mike Sendall) ผู้จัดการของเขาพิจารณา

3. การพัฒนาเวิลด์ไวด์เว็บ

ข้อเสนอดังกล่าวได้กลายเป็นพื้นฐานของเวิลด์ไวด์เว็บ ซึ่งเบร์เนิร์ส-ลีได้พัฒนา **เว็บเบราว์เซอร์** และตัวแก้ไข **HTML** ตัวแรกของโลก ภายใต้ชื่อ **WorldWideWeb** โดยทำงานบนระบบปฏิบัติการ NEXTSTEP ของสตีฟ จอบส์ นอกจากนี้ เขายังพัฒนาเว็บเซิร์ฟเวอร์ตัวแรกของโลกที่ชื่อว่า **httpd** (HyperText Transfer Protocol Daemon)

4. เว็บไซต์แรกของโลก

เว็บไซต์แรกถูกสร้างขึ้นที่ CERN และเผยแพร่สู่สาธารณะเมื่อวันที่ 6 สิงหาคม พ.ศ. 2534 เนื้อหาของเว็บไซต์ให้ข้อมูลเกี่ยวกับเวิลด์ไวด์เว็บ วิธีการติดตั้งเบราว์เซอร์ และการใช้งานเว็บเซิร์ฟเวอร์ นอกจากนี้ เว็บไซต์ดังกล่าวยังทำหน้าที่เป็น เว็บไคเร็กทอรีแรกของโลก โดยเบอร์เนิร์ส-ลีเป็นผู้ดูแลรายชื่อเว็บไซต์อื่นทั้งหมดในช่วงแรก

5. การจัดตั้งกลุ่ม World Wide Web Consortium (W3C)

ในปี พ.ศ. 2537 เบอร์เนิร์ส-ลีได้ก่อตั้ง **World Wide Web Consortium (W3C)** ที่สถาบันเทคโนโลยีแมสซาชูเซตส์ (MIT) โดยมีเป้าหมายเพื่อพัฒนามาตรฐานและแนวทางปฏิบัติสำหรับเวิลด์ไวด์เว็บ ต่อมาในเดือนธันวาคม พ.ศ. 2547 เขาได้รับตำแหน่งประธานสาขาวิทยาศาสตร์คอมพิวเตอร์ที่มหาวิทยาลัยเซาท์แธมป์ตัน สหราชอาณาจักร เพื่อดำเนินโครงการ "**ซีแมนติกเว็บ**" (**Semantic Web**) ซึ่งเป็นแนวคิดที่ช่วยให้ข้อมูลบนอินเทอร์เน็ตสามารถประมวลผลและเชื่อมโยงกันได้อย่างมีประสิทธิภาพมากขึ้น

อิทธิพลของเบอร์เนิร์ส-ลีต่อวงการเทคโนโลยี

เบอร์เนิร์ส-ลีมีแนวคิดที่ว่าเทคโนโลยีเวิลด์ไวด์เว็บควรเปิดให้ทุกคนสามารถใช้งานได้โดยไม่มีค่าใช้จ่าย เขาจึงไม่จดลิขสิทธิ์หรือเรียกเก็บค่าตอบแทนสำหรับการคิดค้นของเขา องค์กร W3C จึงมีนโยบาย ไม่คิดค่าธรรมเนียม จากการนำมาตรฐานไปใช้ เพื่อให้ทุกฝ่ายสามารถพัฒนาเทคโนโลยีบนพื้นฐานของมาตรฐานเดียวกัน โดยไม่ถูกจำกัดด้วยต้นทุนลิขสิทธิ์

ยุคของเว็บ

เทคโนโลยีเว็บมีการพัฒนาเป็นลำดับขั้นจาก Web 1.0 ถึง Web 5.0 โดยแต่ละยุคมีคุณลักษณะเฉพาะที่สะท้อนถึงวิวัฒนาการของอินเทอร์เน็ตและพฤติกรรมของผู้ใช้

Web 1.0: Static Web (ยุคเว็บคงที่)

ช่วงเวลา: 1990 - ต้น 2000

ลักษณะสำคัญ:

- เป็นเว็บแบบ อ่านได้อย่างเดียว (Read-Only Web)
- เนื้อหาเป็น **Static Web Pages** ไม่มีการโต้ตอบระหว่างผู้ใช้
- ไม่มี Social Media หรือระบบการแชร์ข้อมูล
- การออกแบบหน้าเว็บใช้ **HTML** และ **CSS** เท่านั้น
- ผู้ดูแลระบบเป็นเพียงฝ่ายเดียวที่สามารถอัปเดตข้อมูล

เทคโนโลยีที่ใช้: HTML, CSS, HTTP, CGI

ตัวอย่างเว็บไซต์: Yahoo! (ยุคแรก), Netscape, Britannica Online

Web 2.0: Social Web (ยุคเว็บเชิงโต้ตอบ)

ช่วงเวลา: กลาง 2000 - ปัจจุบัน

ลักษณะสำคัญ:

- เปลี่ยนจากเว็บแบบคงที่เป็น **เว็บแบบโต้ตอบ (Interactive Web)**
- ผู้ใช้สามารถสร้างและแบ่งปันเนื้อหาได้ (**User-Generated Content**)
- เกิด Social Media เช่น **Facebook, YouTube, Twitter**
- รองรับ AJAX และ API ทำให้เว็บตอบสนองเร็วขึ้น
- มีระบบ CMS เช่น WordPress สำหรับสร้างเว็บไซต์ง่ายขึ้น

เทคโนโลยีที่ใช้: AJAX, JavaScript, PHP, REST API, HTML5, CSS3

ตัวอย่างเว็บไซต์: Facebook, YouTube, Wikipedia, Twitter

Web 3.0: Semantic & Decentralized Web (ยุคเว็บอัจฉริยะ)

ช่วงเวลา: ปลาย 2010 - ปัจจุบัน

ลักษณะสำคัญ:

- ใช้ **AI และ Machine Learning** เพื่อวิเคราะห์ข้อมูลและให้คำตอบที่แม่นยำ
- **Semantic Web** ช่วยให้ข้อมูลมีโครงสร้างและความหมายมากขึ้น
- มีการใช้ **Blockchain และ Decentralization** เพื่อลดการพึ่งพาตัวกลาง
- การพัฒนา **DApps (Decentralized Applications)** ที่ไม่ต้องมีเซิร์ฟเวอร์กลาง
- มุ่งเน้น **ความเป็นส่วนตัวของข้อมูล** และความปลอดภัย

เทคโนโลยีที่ใช้: AI, Blockchain, Smart Contracts, Big Data, RDF

ตัวอย่างเว็บไซต์/แพลตฟอร์ม: OpenSea (NFT), Uniswap (DeFi), Brave Browser

Web 4.0: Intelligent & Ubiquitous Web (ยุคเว็บอัจฉริยะเต็มรูปแบบ)

ช่วงเวลา: กำลังพัฒนา (อนาคตอันใกล้)

ลักษณะสำคัญ:

- เว็บที่ผสานกับ **Internet of Things (IoT)** และอุปกรณ์อัจฉริยะ
- การประมวลผลข้อมูลแบบเรียลไทม์ (**Real-Time Processing**)
- ระบบ **AI ขั้นสูง** ที่สามารถสื่อสารกับผู้ใช้ได้แบบมนุษย์
- การควบคุมผ่าน เสียง, ท่าทาง, และอุปกรณ์สวมใส่ (**Wearable Technology**)

เทคโนโลยีที่ใช้: IoT, 5G, AI, Cloud Computing, Edge Computing

ตัวอย่างแพลตฟอร์ม: Siri, Google Assistant, Alexa, Smart Homes

Web 5.0: Emotional Web (ยุคเว็บที่เข้าใจอารมณ์)

ช่วงเวลา: ยังกู้ในช่วงแนวคิด (อนาคต)

ลักษณะสำคัญ:

- เว็บที่สามารถ **ตรวจจับและตอบสนองต่ออารมณ์**ของผู้ใช้
- การพัฒนา **AI ที่มีอารมณ์ (Emotional AI)**
- รองรับ **VR (Virtual Reality)** และ **AR (Augmented Reality)**
- เว็บสามารถเรียนรู้ **พฤติกรรมและความรู้สึก**ของผู้ใช้ เพื่อให้ประสบการณ์ส่วนบุคคล

เทคโนโลยีที่ใช้: Emotional AI, Brain-Computer Interface (BCI), Neurotechnology, Metaverse

ตัวอย่างแพลตฟอร์ม: Metaverse, AI Chatbots ที่เข้าใจอารมณ์

สรุปตารางที่ 1

ตารางที่ 1 สรุปเกี่ยวกับเนื้อหาเว็บจาก 1.0-5.0

ยุคของเว็บ	ลักษณะเด่น	เทคโนโลยีหลัก	ตัวอย่างแพลตฟอร์ม
Web 1.0 (Static Web)	เว็บคงที่ (อ่านได้อย่างเดียว)	HTML, CSS, HTTP	Yahoo! (ยุคแรก), Britannica Online
Web 2.0 (Social Web)	เว็บโต้ตอบ ผู้ใช้สร้างเนื้อหาได้	AJAX, JavaScript, PHP	Facebook, YouTube, Wikipedia
Web 3.0 (Semantic Web)	AI และ Blockchain เน้นความเป็นส่วนตัว	AI, Blockchain, Smart Contracts	OpenSea, Brave Browser, Uniswap
Web 4.0 (Intelligent Web)	IoT และ AI ที่เชื่อมโยงกับทุกอุปกรณ์	IoT, 5G, Cloud Computing	Siri, Alexa, Smart Homes
Web 5.0 (Emotional Web)	AI ที่เข้าใจอารมณ์และ VR/AR	Emotional AI, Metaverse, BCI	Metaverse, AI Chatbots

แนวโน้มของอนาคต

- **Web 4.0 และ Web 5.0** กำลังเป็นแนวทางใหม่ของอินเทอร์เน็ต
- การรวม **AI, IoT, Blockchain และ VR/AR** จะสร้างประสบการณ์เว็บที่สมจริงและปลอดภัยยิ่งขึ้น

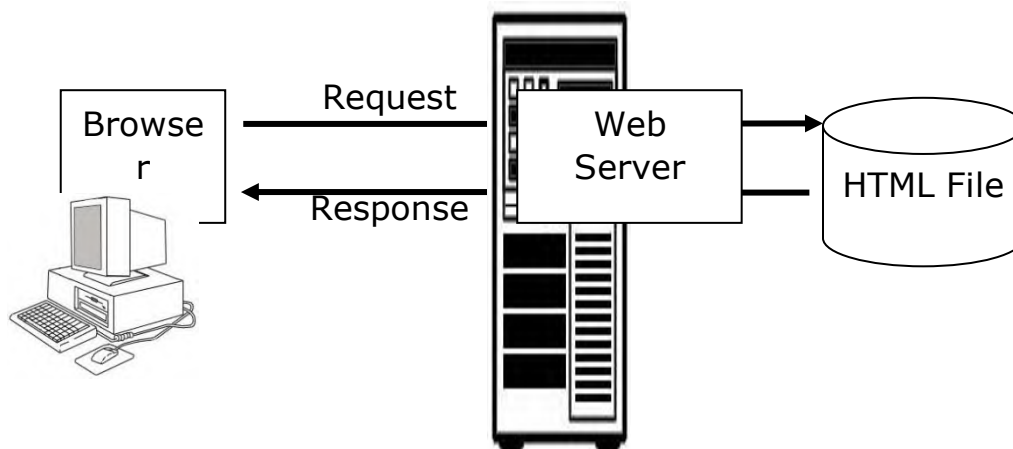
- การพัฒนา **Metaverse** และ **Brain-Computer Interface (BCI)** อาจเปลี่ยนแปลงการใช้งานเว็บในอนาคต

สถาปัตยกรรมของเว็บแอปพลิเคชัน

การพัฒนาโปรแกรมผ่านเครือข่ายอินเทอร์เน็ตนั้นผู้ใช้งานเรียกว่าไคลเอนต์ (Client) ที่เรียกดูข้อมูลผ่านเว็บเพจ (Webpage) ซึ่งก็คือหน้าของเว็บแต่ละหน้าที่แสดงทางเครื่องคอมพิวเตอร์ฝั่งไคลเอนต์ด้วยเว็บเบราว์เซอร์ ส่วนอีกฝั่งหนึ่งให้บริการข้อมูลเรียกว่าเว็บเซิร์ฟเวอร์ (Web server) ซึ่งก็คือส่วนเก็บข้อมูลหลักของหน้าเว็บนั้น ๆ และในการสร้างเว็บเพจในยุคแรก ๆ จะสร้างโดยเป็นหน้าเว็บที่ตายตัวไม่ว่าจะเข้ามาต่างเวลา ต่างโอกาสอย่างไร ก็จะมีหน้าเว็บแบบนั้นเสมอ เรียกสแตติกเว็บเพจ (Static Webpage) ซึ่งผู้พัฒนาจะต้องออกแบบและเขียนทุกอย่างเก็บไว้ที่ สถาปัตยกรรมของเว็บดังนี้

สแตติกเว็บ

การทำงานโดยไคลเอนต์จะร้องขอโดยใช้โปรแกรมเว็บเบราว์เซอร์ ส่งการร้องขอผ่านเว็บเพจที่ต้องการผ่านเครือข่ายอินเทอร์เน็ต ไปยังเว็บเซิร์ฟเวอร์ ที่เก็บ เว็บเพจนั้น ๆ อยู่ซึ่งจะได้รับการตอบสนองจากเว็บเซิร์ฟเวอร์ แล้วส่งผลตามที่ร้องขอกลับมายังไคลเอนต์ผ่านเบราว์เซอร์ เช่นกัน ทำให้ผู้ใช้ได้ข้อมูลในรูปแบบที่ต้องการสำหรับส่วนของการจัดการผ่านกราฟิกของเว็บเบราว์เซอร์เป็นตัวจัดการให้ไคลเอนต์ได้ข้อมูลตรงตามข้อมูลที่เซิร์ฟเวอร์ส่งมา แนวคิดในการพัฒนาจะได้ดังภาพ



ภาพที่ 1.1 แสดงขั้นตอนการทำงานของสแตติกเว็บ

ขั้นตอนการพัฒนาและประมวลผล

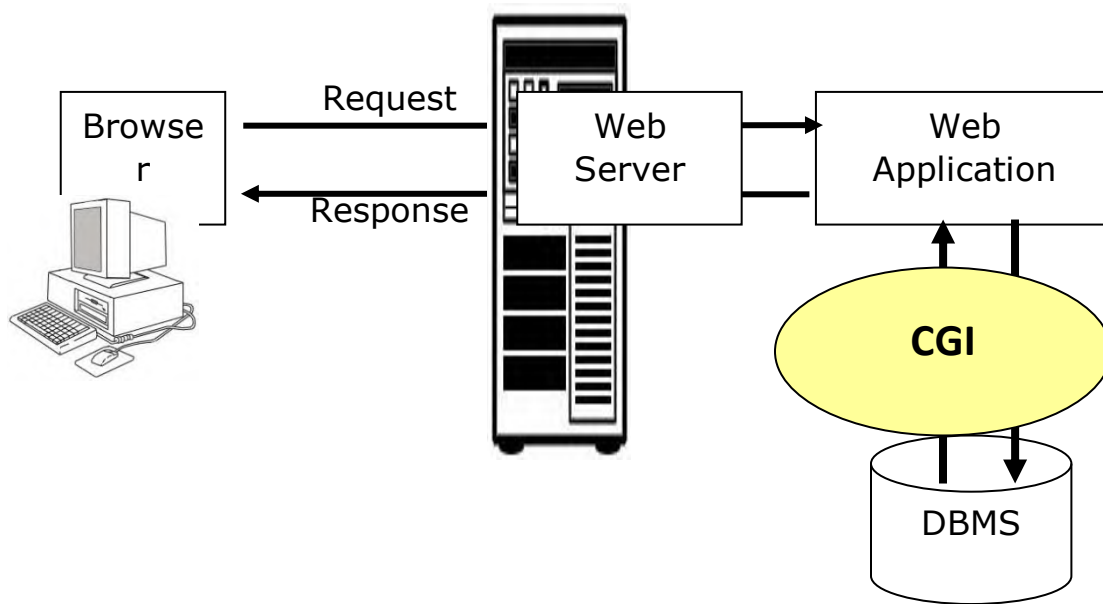
1. ผู้พัฒนาโปรแกรมจะต้องสร้างไฟล์ html และจัดการทุกอย่างตามที่ภาษา HTML ทำได้ แล้วเก็บไฟล์ไว้ในเว็บเซิร์ฟเวอร์ตามที่เซิร์ฟเวอร์แต่ละตัวกำหนด
2. ไคลเอนต์ (Client) เมื่อมีการร้องขอเว็บเพจจากไคลเอนต์จะต้องผ่านโปรแกรมเว็บเบราว์เซอร์
3. เว็บเซิร์ฟเวอร์จะทำการหาไฟล์ html ที่ไคลเอนต์ต้องการในเว็บเซิร์ฟเวอร์ของตนเอง
4. เมื่อหาพบก็จะส่งเป็นส่วนหนึ่งของสตรีม (stream) ของ html ผ่านระบบเครือข่ายอินเทอร์เน็ตไปยังไคลเอนต์
5. เมื่อไคลเอนต์ได้รับสตรีม html มาแล้ว ก็จะนำมาประมวลผลเพื่อแสดงผลโดยขั้นตอนนี้ โปรแกรมเว็บเบราว์เซอร์จะดำเนินการประมวลผล ที่ได้มาและแสดงผลให้ไคลเอนต์ได้รับผล

ไดนามิกเว็บ

สำหรับไดนามิก (Dynamic web page) จะมีความยืดหยุ่นในการแสดงผลหรือประมวลผลให้ได้ผลลัพธ์ในเวลาต่างกัน ต่างไคลเอนต์จะสามารถเข้าถึงข้อมูลเว็บเพจที่แตกต่างกันไม่เหมือนกัน สำหรับโครงสร้างการทำงานคล้าย ๆ กันกับสแตติก ความแตกต่างที่เห็นได้ชัดก็คือการใช้งานภาษาสคริปต์ (Script Language) ที่จะเข้ามาช่วยในการทำงานให้เว็บเพจมีความยืดหยุ่นตามแนวทางนี้ได้ ลักษณะของโครงสร้างภาษาสคริปต์นั้นจะเขียนแท็ก (Tag) ร่วมกับภาษา HTML ในไฟล์เดียวกัน เช่น java script, VBscript, PHP, ASP, JSP และสำหรับภาษาสคริปต์นี้ สามารถใช้งานได้กับเบราว์เซอร์ไม่ทั้งหมดทุกโปรแกรม ดังนั้นในการเลือกใช้จะต้องพิจารณาด้วยว่าโปรแกรมเบราว์เซอร์โปรแกรมใดสนับสนุนภาษาสคริปต์ใดบ้าง

เนื่องจากความต้องการข้อมูลของผู้ใช้แตกต่างกัน ดังนั้นส่วนของเว็บเซิร์ฟเวอร์ จึงต้องให้ตัวเองสามารถยืดหยุ่นได้ดังนี้ แนวคิดของการพัฒนาให้ได้ดังกล่าวโดยใช้ซีจีไอ (CGI) มาใช้สำหรับอ่าน เขียน และส่งข้อมูลจากฐานข้อมูลไปยังเว็บเซิร์ฟเวอร์ให้ส่งต่อไปให้กับผู้ใช้งาน ซึ่งสามารถเขียนด้วยภาษาคอมพิวเตอร์ที่แตกต่างกัน เช่น C , C++, Perl เป็นต้น และภาษาจาวาก็อยู่ในรูปแบบนี้ด้วย

แนวคิดพื้นฐานของการพัฒนาเว็บแอปพลิเคชัน ตามแนวทางของภาษาสคริปต์แสดง ดังภาพต่อไป



ภาพที่ 1.2 แสดงสถาปัตยกรรมในเครือข่ายแบบไดนามิก

ขั้นตอนการพัฒนาเว็บแอปพลิเคชัน

ขั้นตอนละเอียดของการติดตั้งและพัฒนาในหนังสือเล่มนี้จะไม่กล่าวในเชิงลึกมากนัก แต่จะเป็นแนวทางและให้ผู้อ่านได้เข้าใจลำดับขั้นตอนต่าง ๆ ของการพัฒนาและความสามารถของภาษาสคริปต์ที่เลือกใช้ในการพัฒนาขั้นตอนต่าง ๆ มีดังนี้

1. ผู้พัฒนาจะต้องติดตั้งเว็บเซิร์ฟเวอร์ที่รองรับภาษาสคริปต์แต่ละชนิดที่ผู้จะใช้ คือ เจเอสพี (JSP) จะต้องติดตั้ง Apache Tomcat หรือ GlassFish เป็นต้น

2. ติดตั้งโปรแกรมระบบจัดการฐานข้อมูล (Database Server) ซึ่งอาจจะติดตั้งไว้ที่เครื่องเดียวกันหรือต่างเครื่องก็ได้ สำหรับระบบจัดการฐานข้อมูลมีหลากหลายประเภททั้งใหญ่เล็กแตกต่างกันไป เช่น Oracle Database Server, MySQL, MS-SQL, Sybase Informix, MS-ACCESS เป็นต้น

สิ่งที่สำคัญต้องพิจารณาให้เหมาะว่าควรจะใช้ระบบจัดการฐานข้อมูลใด เช่น ถ้าเป็น เจเอสพี ควรจะใช้มายเอสคิวแอล เนื่องจากเป็นโปรแกรมในแนวทางของโอเพ่นซอร์สที่สามารถดาวน์โหลดมาใช้ได้โดยไม่มีค่าใช้จ่าย แต่ทั้งนี้หากต้องการที่จะข้ามแพลตฟอร์มของฐานข้อมูล ภาษาสคริปต์ก็สามารถทำได้ ซึ่งต้องมีไดรเวอร์เพิ่มเติมเพื่อให้สามารถเข้าถึงข้อมูลได้

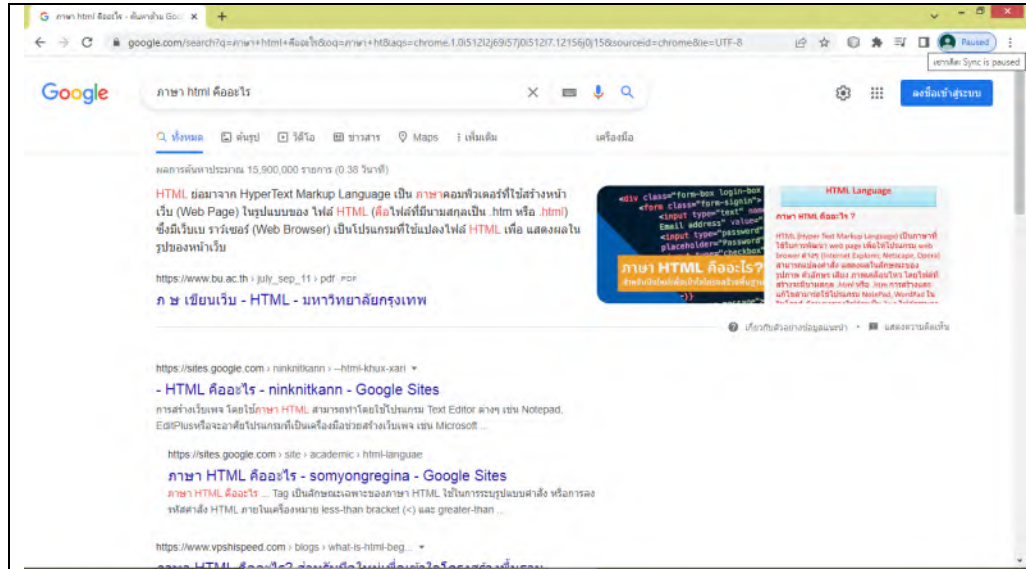
3. จัดหาไดรเวอร์สำหรับติดต่อกับฐานข้อมูลที่ลงได้ที่ได้ติดตั้งไว้ เช่น โอราเคิลจะมี ไดรเวอร์ให้พร้อม หรือหากเป็นมายเอสคิวแอลจะต้องหาไดรเวอร์มาใช้เองซึ่งสามารถดาวน์โหลดมาใช้งานได้

4. หาเครื่องมือที่ใช้ในการพัฒนาภาษาสคริปต์ ซึ่งแบ่งเครื่องมือในการพัฒนาได้ดังนี้
 - เครื่องมือที่เป็นพื้นฐานในวินโดวส์ คือ โน้ตแพด ซึ่งเป็นโปรแกรมที่เป็นเท็กซ์อีดิเตอร์ธรรมดา ซึ่งผู้เขียนโปรแกรมจะต้องเขียนคำสั่งภาษาสคริปต์ต่าง ๆ รวมถึง HTML ทั้งหมดด้วยตนเอง
 - เครื่องมือที่เป็นกราฟิกอินเทอร์เฟซ เช่นโปรแกรม Macromedia Dream weaver, MS FrontPage หรือเครื่องมือที่สามารถพัฒนาได้ทั้งแอปพลิเคชันและเว็บไซต์ เช่น เน็ตบีนเป็นต้น ซึ่งโปรแกรมเหล่านี้จะทำให้ผู้พัฒนา ทำได้เร็วขึ้น และช่วยในการออกแบบในรูปแบบของกราฟิกอินเทอร์เฟซ และบางโปรแกรมยังจัดหาไดรเวอร์หรือเตรียมเครื่องมือ ในการติดต่อฐานข้อมูลให้พร้อมอีกด้วย ตัวอย่างเช่น MS.NET, ASP.NET หรือเน็ตบีนเป็นต้น
5. พัฒนาโปรแกรมโดยวิธีใดวิธีหนึ่ง ซึ่งอาจเป็นเขียนแบบใช้อัตโนมัติหรือโปรแกรมช่วยก็ได้ โดยพัฒนาและบันทึกเป็นภาษาสคริปต์นั้น ๆ
 - เมื่อเขียนเสร็จแล้วก็นำโค้ดเหล่านั้นไปไว้ ณ ตำแหน่งที่เว็บเซิร์ฟเวอร์แต่ละประเภทกำหนดไว้ เช่น ASP ใช้ Webserver IIS อยู่ที่ ไดรเกทอรี www JSP ใช้ Tomcat ไว้ที่ webapps หรือกลาสฟิช สามารถสร้างการติดต่อได้โดยตรงกับส่วนเก็บข้อมูล หรือการติดตั้งโดยใช้ .war เป็นต้น

เอชทีเอ็มแอล

HTML ย่อมาจาก Hyper Text Markup Language ซึ่งถือว่าเป็นมาตรฐานสำหรับการสร้างเว็บเพจ ภาษานี้เป็นภาษาที่เขียนด้วยเท็กซ์ (Text) โดยกำหนดนามสกุลเป็น .htm หรือ .html และรันโปรแกรมด้วยโปรแกรมเว็บเบราว์เซอร์ รูปแบบการเขียนประกอบด้วยแท็ก (Tag) ประกอบด้วยเครื่องหมายเปิดแท็ก < (Left angle bracket) ตามด้วยคำสั่งภายในหรือแท็ก และปิดด้วย > (Right angle bracket) เช่น แท็กเริ่มส่วนหัว <HEAD> แท็กชื่อหน้าเพจ <TITLE> หรือแท็กส่วนเนื้อหาหลัก <BODY> เป็นต้น โดยที่คำสั่งในเครื่องหมาย <> จะเขียนด้วยอักษรพิมพ์เล็กหรือพิมพ์ใหญ่ก็ได้

เบื้องต้น ภาษาที่ใช้สร้างเท็กซ์หลายมิติในหน้าเว็บนั้นคือภาษา HTML ในกรณีนี้ ผู้เขียนขอแสดงหน้าเว็บจำนวนหนึ่งหน้าเพจหนึ่งหน้าที่อยู่บนหน้าเว็บส่วนมาก สิ่งประกอบกันนั้นแสดงดังภาพ


Server side Script
Java Script
CSS
HTML

ภาพที่ 1.3 หน้าเพจและส่วนที่เป็นกลไกของโค้ดภายใน

โครงสร้างพื้นฐานของหน้าเว็บและวิวัฒนาการของ HTML

การพัฒนาและแสดงผลของหน้าเว็บในปัจจุบันอาศัยองค์ประกอบสำคัญที่ทำงานร่วมกันเพื่อสร้างประสบการณ์ที่สมบูรณ์แบบสำหรับผู้ใช้ โดยโครงสร้างพื้นฐานของโค้ดที่เป็นกลไกสำคัญของหน้าเว็บสามารถแบ่งออกเป็น 4 ส่วนหลัก ได้แก่

1. **HTML (HyperText Markup Language)** – เป็นภาษาหลักที่ใช้ในการสร้างโครงสร้างของเว็บเพจ และจัดการการแสดงผลของข้อความและเนื้อหาหลายมิติ (Hypertext) บนอินเทอร์เน็ต
2. **CSS (Cascading Style Sheets)** – ใช้สำหรับกำหนดรูปแบบ (Styling) และความสวยงามของหน้าเว็บ เช่น สี ขนาดตัวอักษร และการจัดวางองค์ประกอบต่าง ๆ
3. **JavaScript** – เป็นภาษาสคริปต์ที่ช่วยเพิ่มความสามารถแบบไดนามิกให้กับหน้าเว็บ ทำให้สามารถโต้ตอบกับผู้ใช้ได้แบบเรียลไทม์โดยทำงานฝั่งไคลเอนต์ (Client-Side)
4. **Server-Side Scripting** – ใช้ในการประมวลผลข้อมูลที่เกี่ยวข้องกับเซิร์ฟเวอร์ เช่น การเข้าถึงฐานข้อมูล การยืนยันตัวตนของผู้ใช้ และการประมวลผลข้อมูลก่อนส่งกลับไปยังเว็บเบราว์เซอร์

ดังนั้น การเรียนรู้พื้นฐานของการพัฒนาเว็บเพจจำเป็นต้องเข้าใจองค์ประกอบทั้ง 4 ส่วนนี้ เพื่อให้สามารถสร้างเว็บที่ครบถ้วนและมีประสิทธิภาพ

วิวัฒนาการของ HTML

ตั้งแต่การคิดค้นเวิลด์ไวด์เว็บโดย **Tim Berners-Lee** ในปี 1989 HTML ได้รับการพัฒนาและปรับปรุงอย่างต่อเนื่องเพื่อตอบสนองต่อเทคโนโลยีอินเทอร์เน็ตที่ก้าวหน้า โดยมีการเปลี่ยนแปลงที่สำคัญในแต่ละช่วงเวลาดังนี้

ตารางที่ 1.2 แสดงการพัฒนาปรับปรุง HTML

ปี	เวอร์ชัน HTML	การเปลี่ยนแปลงสำคัญ
1989	Tim Berners-Lee คิดค้นเวิลด์ไวด์เว็บ	จุดเริ่มต้นของอินเทอร์เน็ตเชิงโต้ตอบ
1991	HTML เวอร์ชันแรก	โครงสร้างพื้นฐานของเว็บ
1993	HTML+ (ร่างโดย Dave Raggett)	เพิ่มฟีเจอร์เพิ่มเติมจากเวอร์ชันแรก
1995	HTML 2.0	มาตรฐานแรกของ HTML อย่างเป็นทางการ
1997	HTML 3.2 (W3C Recommendation)	สนับสนุนจาวาสคริปต์และสไตร์ชีต
1999	HTML 4.01 (W3C Recommendation)	ปรับปรุงโครงสร้างและเพิ่มมาตรฐานใหม่
2000	XHTML 1.0 (W3C Recommendation)	ปรับรูปแบบให้เข้ากับมาตรฐาน XML
2008	HTML5 (WHATWG First Public Draft)	รองรับมัลติมีเดียโดยไม่ต้องใช้ปลั๊กอิน
2012	HTML5 (Living Standard)	ปรับปรุงต่อเนื่องโดย WHATWG
2014	HTML5 (W3C Recommendation)	มาตรฐานเว็บสมัยใหม่อย่างเป็นทางการ
2016	HTML 5.1 (W3C Candidate Recommendation)	ปรับปรุงการรองรับฟีเจอร์ใหม่
2017	HTML 5.1 (2nd Edition) และ HTML5.2	เพิ่มความปลอดภัยและรองรับเทคโนโลยีใหม่ ๆ

ปัจจุบัน HTML อยู่ภายใต้การดูแลของ WHATWG (Web Hypertext Application Technology Working Group) ซึ่งเน้นการพัฒนา HTML ในรูปแบบ Living Standard หมายความว่ามาตรฐาน HTML จะได้รับการปรับปรุงอย่างต่อเนื่องตามแนวโน้มเทคโนโลยีและความต้องการของผู้ใช้งาน

วิวัฒนาการของ HTML ส่งผลให้การพัฒนาเว็บสามารถทำได้ง่ายขึ้น มีประสิทธิภาพมากขึ้น และรองรับการใช้งานบนอุปกรณ์ที่หลากหลาย ซึ่งเป็นรากฐานสำคัญของการพัฒนาเว็บสมัยใหม่

โครงสร้างไค้ดภาษา HTML

สามารถแบ่งไค้ดออกเป็นส่วหัว และส่วตัวไค้ด โดยแต่ละคำสั้จะแบ่งเป็นแท็กซึ่งจะมีทั้งส่วเริ่มต้นและจบแท็กกรณีเป็นแท็กคู่ แต่หากเป็นแท็กเดี่ยวจะมีแค่เพียงแท็กเปิดเท่านั้น ตัวอย่างไค้ดของภาษาดังนี้

```
<!DOCTYPE html>
```

← ส่วนหัว

```
<html>
```

```
<body>
```

← ส่วนตัวโค้ดหลัก

```
<h1>สวัสดี</h1>
```

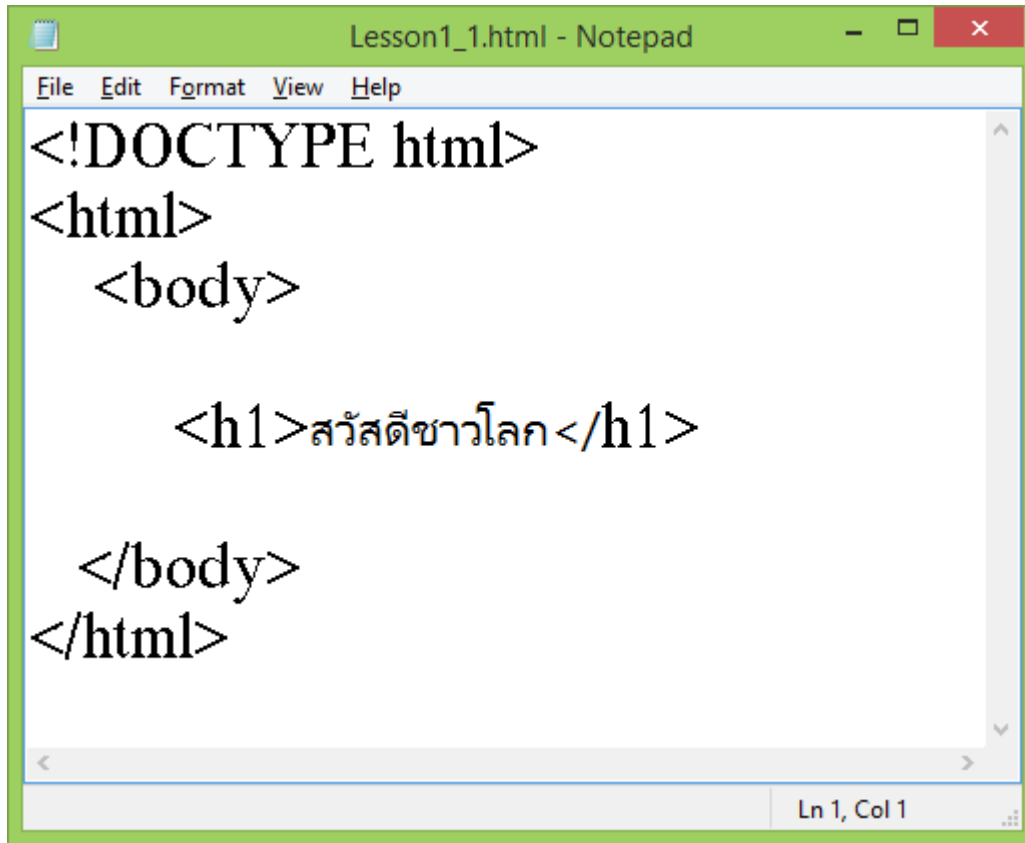
```
</body>
```

```
</html>
```

การเขียนภาษา HTML

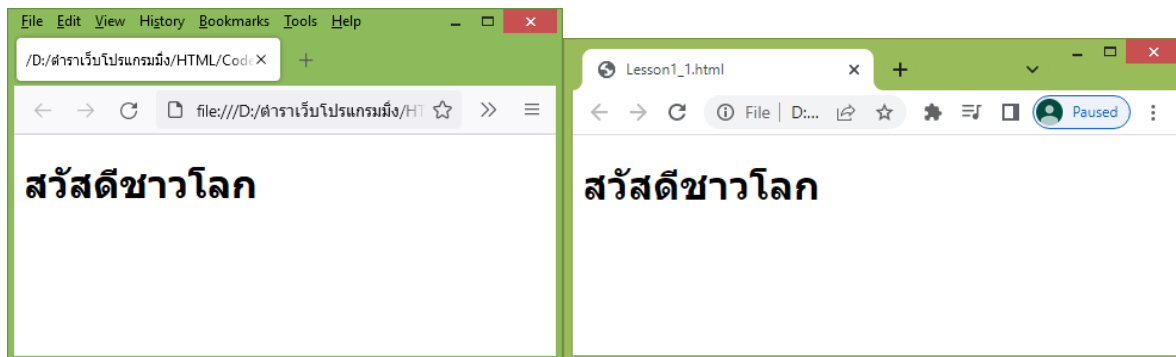
การเขียนภาษาเอชทีเอ็มแอลสามารถเขียนจากโปรแกรมอิดิตเตอร์ (Editor) ต่าง เช่น Notepad Wordpad หรือโปรแกรมที่สร้างเท็กซ์ต่างๆ ได้ นอกจากนั้นยังสามารถเขียนด้วยแอปพลิเคชันเพื่อสร้างหน้าเว็บต่างๆ ได้ ซึ่งสามารถหาซื้อหรือโหลดได้ทั่วไป หรือแม้แต่ในอินเทอร์เน็ตก็มีเว็บไซต์สำหรับช่วยสร้างโค้ดภาษาเอชทีเอ็มแอลได้เช่นกัน

ตัวอย่างต่อไปนี้เป็นการสร้างภาษาเอชทีเอ็มแอลผ่านโปรแกรม Notepad



ภาพที่ 1.4 ตัวอย่างโปรแกรมเขียนโค้ดด้วย Notepad

เมื่อเขียนโค้ดเรียบร้อยแล้วบันทึกเป็นแฟ้มนามสกุล .htm หรือ .html เพื่อให้โปรแกรมบราวเซอร์สามารถนำไปคอมไพล์และรันโปรแกรมได้ การรันโปรแกรมสามารถเปิดจากโปรแกรมบราวเซอร์ใดก็ได้ ตัวอย่างเปิดโปรแกรมในตัวอย่างดังกล่าวดังนี้



ภาพที่ 1.5 ตัวอย่างการรันโปรแกรมด้วยบราวเซอร์ Mozilla Firefox และ Google Chrome

สรุป

ภาษาที่ใช้ในการสร้างเท็กซ์หลายมิติหรือไฮเปอร์เท็กซ์ที่ใช้สื่อสารส่งข้อมูลกันในอินเทอร์เน็ตคือภาษา HTML ซึ่งเป็นภาษาที่สามารถสร้างหน้าเว็บไซต์ต่างๆ เขียนโดยใช้เท็กซ์อีดิเตอร์หรือโปรแกรมสร้างโดยเฉพาะก็ได้ เมื่อเขียนเสร็จเรียบร้อยแล้วจะต้องบันทึกเป็นแฟ้มนามสกุล .htm หรือ .html และสามารถรันโปรแกรมได้ผ่านโปรแกรมเว็บเบราว์เซอร์ต่างๆ

บทที่ 2

คำสั่งพื้นฐาน

เนื้อหา

- พื้นฐานความรู้เอชทีเอ็มแอล
- ชนิดของ HTML Tag
- คำสั่งย่อหน้า
- แท็กจัดการรูปภาพ
- การสร้างเส้นคั่นในหน้าเว็บ
- การสร้างหัวข้อรายการ
- กำหนดรูปแบบ

จากบทที่ 1 นำเสนอความรู้เบื้องต้นเกี่ยวกับภาษาเอชทีเอ็มแอล โครงสร้างโค้ด เครื่องมือในการเขียนโค้ด การรันโปรแกรมมาแล้ว ในบทนี้ขอแนะนำคำสั่งต่างๆ ที่เป็นคำสั่งมาตรฐานที่มีการใช้งานพื้นฐานตั้งแต่เวอร์ชันแรกๆ ของการพัฒนาภาษานี้ ซึ่งยังคงใช้งานอยู่ในปัจจุบัน และยังมีคำสั่งอื่นๆ ที่น่าสนใจอีกด้วย รวมถึงแนวทางการประยุกต์ใช้สำหรับบูรณาการเข้าสู่หน้าเว็บไซต์อีกด้วย

พื้นฐานความรู้เอชทีเอ็มแอล

ดังที่กล่าวแล้วในบทที่ 1 ว่า HTML ย่อมาจาก Hyper Text Markup Language เป็นภาษามาร์กอัปมาตรฐานในการสร้างเว็บเพจ องค์ประกอบภายในคำสั่งประกอบด้วยแท็กต่างๆ โดยที่แท็กจะมีอยู่สองแบบคือ แท็กแบบคู่และและแท็กแบบเดี่ยว โดยที่แท็กคู่ จะมีลักษณะดังนี้ เริ่มต้นเขียนด้วย `< tagname >` เนื้อหาที่จะใช้แท็กคำสั่งนี้บังคับ `< / tagname >` ซึ่งจะจบแท็กด้วย `</tagname>` เช่น `<title> หัวข้อ </title>` เป็นต้น ส่วน แท็กเดี่ยวจะใช้เพียง `< tagname >` เนื้อหาที่จะใช้แท็กนี้บังคับ ขอยกตัวอย่างจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.1 แสดงการใช้งานแท็กเดี่ยวและแท็กคู่ สำหรับแสดงการต้อนรับ

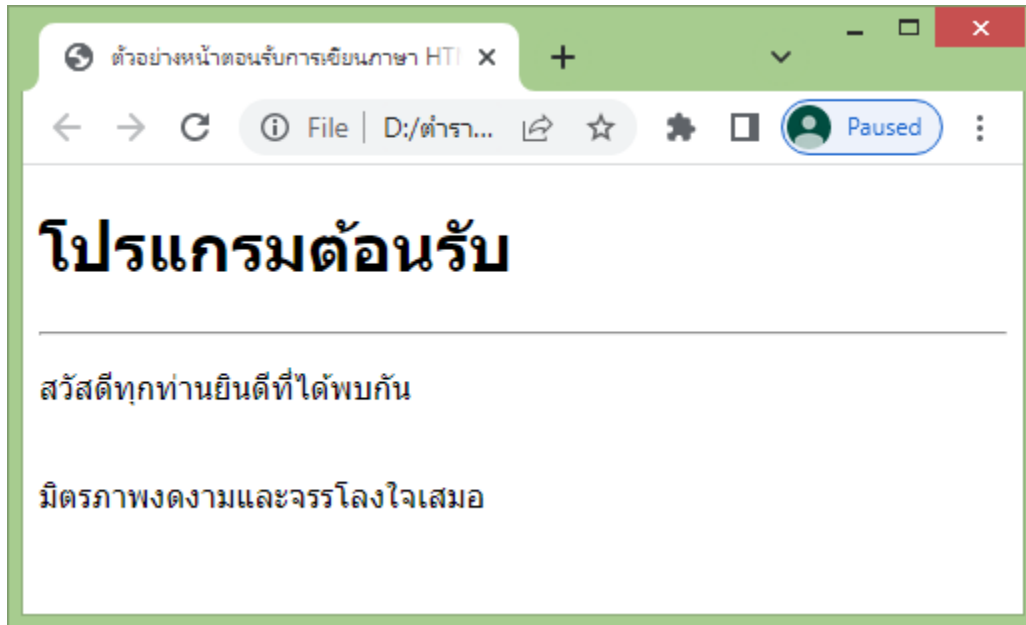
```
<!DOCTYPE HTML>
<HTML>
<HEAD>
```

```
<TITLE>ตัวอย่างหน้าตอนรับการเขียนภาษา HTML</TITLE>
</HEAD>
<BODY>
  <H1>โปรแกรมต้อนรับ</H1>
  <HR>
  <P>สวัสดีทุกท่านยินดีที่ได้พบกัน</P>
  <BR>
  มิตรภาพงดงามและจรรโลงใจเสมอ
</BODY>
</HTML>
```

อธิบายคำสั่ง

- <!DOCTYPE HTML> เป็นแท็กคำสั่งเพื่อระบุว่าเอกสาร(โปรแกรม)นี้เป็นเอกสารแบบ HTML5
- <HTML>,,,</HTML> เป็นแท็กคู่เริ่มต้นของโปรแกรม
- <HEAD>...</HEAD> เป็นแท็กคู่ระบุส่วนหัวของโปรแกรม
- <TITLE>...</TITLE> เป็นแท็กคู่ใช้ระบุข้อความที่จะแสดงบนส่วนใดเดิ้ลบาร์ของเว็บเบราว์เซอร์ตอนรันโปรแกรม
- <body>องค์ประกอบกำหนดร่างกายของเอกสารและเป็นภาษาสำหรับทุกเนื้อหาที่มองเห็นได้เช่น ห่วยหน้าภาพเชื่อมโยงหลายมิติตาราง, รายการ, ฯลฯ
- <h1>องค์ประกอบกำหนดหัวข้อที่มีขนาดใหญ่
- <p>องค์ประกอบกำหนดย่อหน้า

ผลการรันโปรแกรม



ชนิดของ HTML Tag

ชนิดของแท็กมีอยู่ 2 แบบ คือ

- 1) แท็กเดี่ยว (Single tag) เป็นแท็กที่ระบุเฉพาะการเปิดแท็กใช้งานเท่านั้น คำสั่งหรือข้อความต่างๆ ที่ตามมาจะเกิดการดำเนินการตามคำสั่งของแท็ก เช่น

<hr> เป็นแท็กสำหรับสร้างเส้น

<p> เป็นแท็กของการกำหนดย่อหน้า

 เป็นแท็กสำหรับขึ้นย่อบรรทัดใหม่

 เป็นแท็กสำหรับการแสดงรูปภาพ

ตัวอย่างที่ 2 แสดงการใช้คำสั่งแท็กเดี่ยว

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ตัวอย่างการใช้คำสั่งแท็กเดี่ยว HTML</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
    แสดงตัวอย่างคำสั่งแท็กเดี่ยว
```

```
    <HR> คำสั่งแสดงเส้น
```

```

สวัสดีทุกท่าน
<BR> ขึ้นบรรทัดใหม่

<HR>

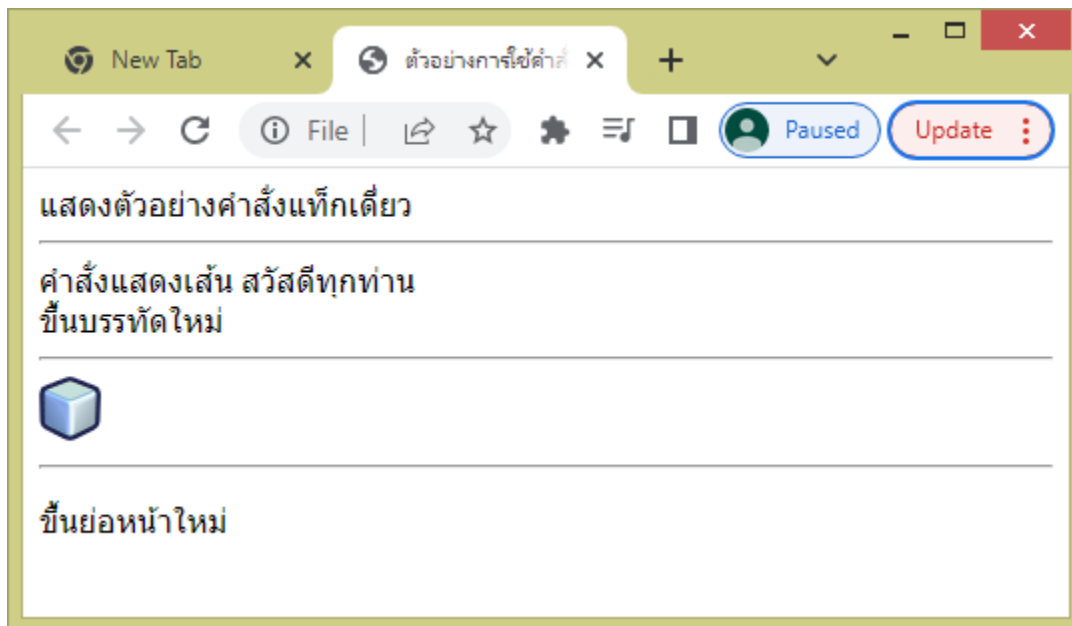
<IMG SRC="icon.png">

<HR>

<P>ขึ้นย่อหน้าใหม่
</BODY>
</HTML>

```

ผลการรันโปรแกรม



- 2) แท็กคู่ (Paired tag) คือแท็กคำสั่งที่ต้องมีสองส่วน คือ ส่วนคำสั่ง เปิดแท็ก วางไว้หน้าข้อความ หรือวัตถุต่างๆ ที่เป็นการกำหนดของคำสั่งนั้นๆ และส่วนคำสั่งปิดแท็กโดยใช้ / นำหน้า ตามด้วยคำสั่งที่เปิดไว้ในตอนแรก การวางจะวางไว้หลังข้อความหรือวัตถุที่ต้องการจัดการด้วยคำสั่งนั้นๆ ตัวอย่างเช่น

```

<body> กำหนดส่วนการเขียนโค้ดหลักของ HTML </body>
<title> คำสั่งกำหนดไตเติ้ล </title>
<h1> คำสั่งกำหนดหัวข้อ </h1>

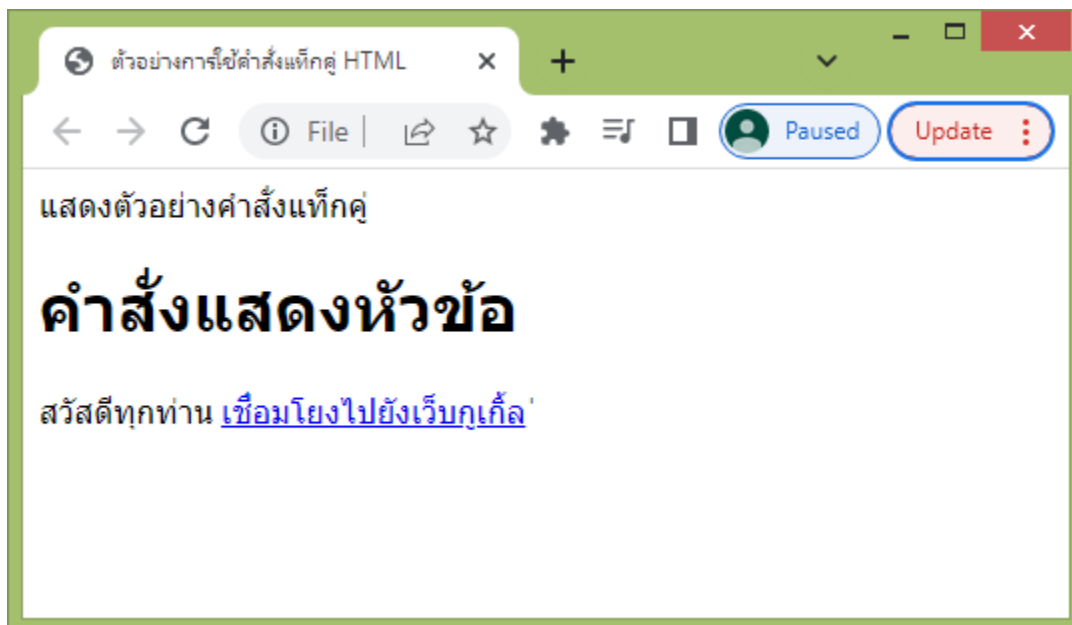
```

```
<a href="https://www.w3schools.com">This is a link</a>
```

ตัวอย่างที่ 3 แสดงตัวอย่างคำสั่งแท็กคู่

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>ตัวอย่างการใช้คำสั่งแท็กคู่ HTML</TITLE>
</HEAD>
<BODY>
  แสดงตัวอย่างคำสั่งแท็กคู่
  <H1> คำสั่งแสดงหัวข้อ </H1>
  สวัสดีทุกท่าน
  <a href="http://www.google.com">เชื่อมโยงไปยังเว็บกูเกิ้ล</a> ่
</BODY>
</HTML>
```

ผลการรันโปรแกรม



กำหนดสีพื้นหลังตัวอักษร

สามารถกำหนดสีพื้นหลังหน้าเว็บได้จาก แท็ก <BODY>
โดยกำหนดในให้เป็น

รูปแบบ <BODY BGCOLOR="สีที่ต้องการ">

เช่น <BODY BGCOLOR="RED"> เป็นการกำหนดสีแดงให้กับพื้นของหน้าเว็บ

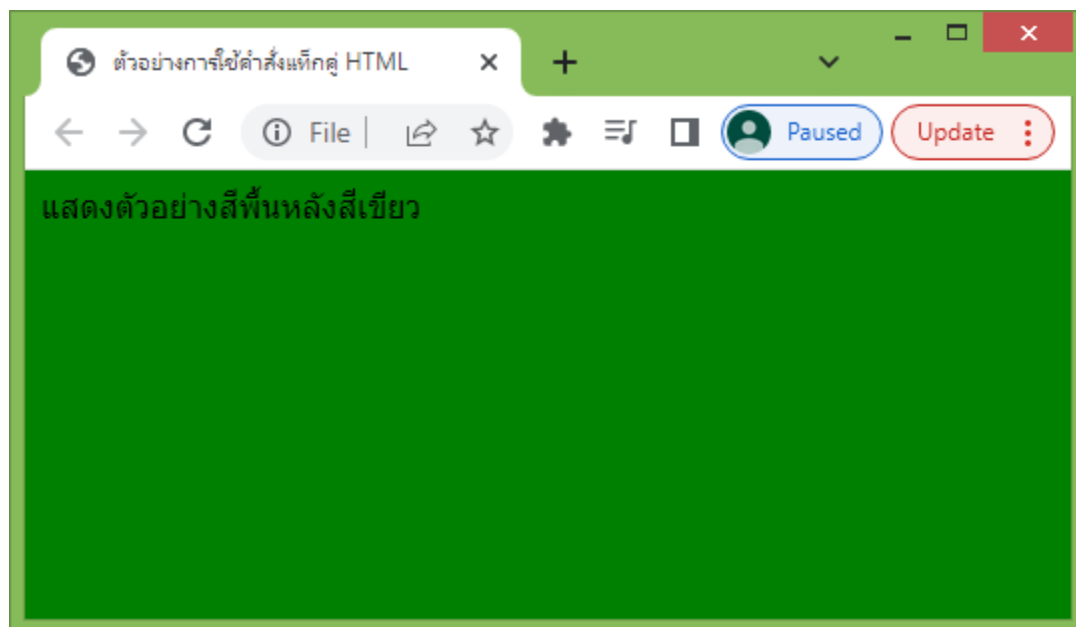
การกำหนดสีสามารถกำหนดได้ 2 แบบ คือ

- 1) กำหนดเป็นข้อความชื่อของสีที่ต้องการ เช่น RED, GREEN, BLUE, YELLOW เป็นต้น

ตัวอย่างที่ 4 การกำหนดสีพื้นตัวอักษรสำหรับหน้าเว็บด้วยข้อความ

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>ตัวอย่างการใช้คำสั่งแท็ก HTML</TITLE>
</HEAD>
<BODY BGCOLOR="GREEN">
  แสดงตัวอย่างสีพื้นหลังสีเขียว
</BODY>
</HTML>
```

ผลการรันโปรแกรม



- 2) กำหนดสีแบบระบุตัวเลขฐานสิบหก เพื่อกำหนดค่าสี แบบ การผสมสี ด้วยแม่สี แดง เขียว น้ำเงิน R G B ระบุเป็นตัวเลขสามชุด สองหลักแรกคือค่าสีแดง สองหลักตรงกลางคือค่าสีเขียว และ สองหลักสุดท้ายคือค่าสีน้ำเงิน รูปแบบ คือ

<BODY BGCOLOR="#RRGGBB">

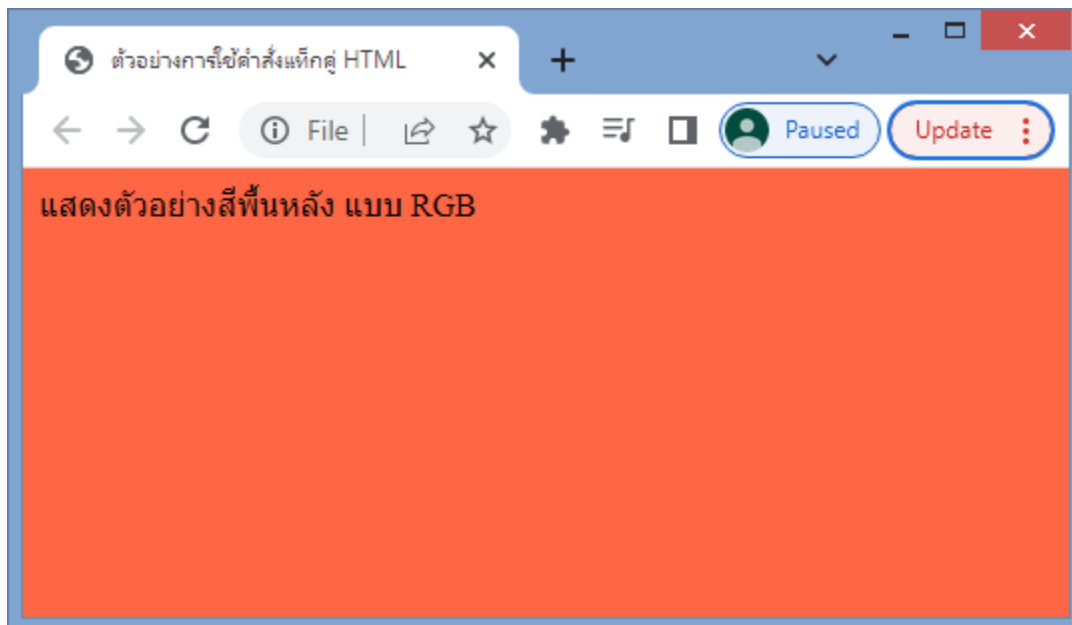
ตัวอย่าง เช่น <BODY BGCOLOR="#FF6644">

โดยที่ FF คือค่าเลขฐานสิบหกแสดงค่าของสีแดง FF เลข 66 คือค่าสีเขียว และ 44 คือค่าสีน้ำเงิน

ตัวอย่างที่ 5 แสดงการใช้ค่าสีแบบ RGB

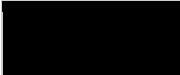
```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>ตัวอย่างการใช้คำสั่งแท็ก HTML</TITLE>
</HEAD>
<BODY BGCOLOR="#FF6644">
  แสดงตัวอย่างสีพื้นหลัง แบบ RGB
</BODY>
</HTML>
```

ผลการรันโปรแกรม



การกำหนดสี สามารถแสดงชื่อของสีต่างๆ ได้ดังนี้

ตารางที่ 2.1 แสดงค่าสีพื้นฐาน

สี	ชื่อของสี	เลขฐานสิบหก #RRGGBB
	Black	#000000
	White	#FFFFFF
	Red	#FF0000
	Lime	#00FF00
	Blue	#0000FF
	Yellow	#FFFF00
	Cyan / Aqua	#00FFFF
	Magenta / Fuchsia	#FF00FF
	Silver	#C0C0C0
	Gray	#808080
	Maroon	#800000
	Olive	#808000
	Green	#008000
	Purple	#800080
	Teal	#008080
	Navy	#000080

ที่มา : https://www.rapidtables.com/web/color/RGB_Color.html

สามารถดูเพิ่มเติมได้จากเว็บไซต์ดังกล่าว https://www.rapidtables.com/web/color/RGB_Color.html

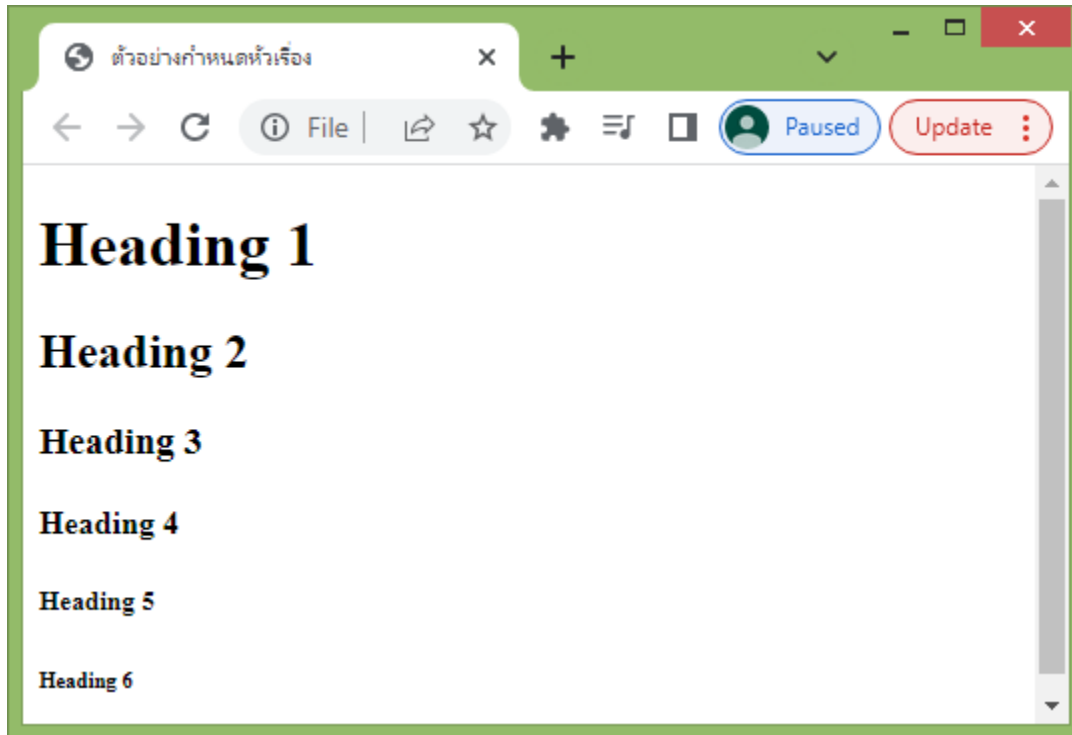
คำสั่งกำหนดหัวเรื่อง

คำสั่งกำหนดหัวเรื่อง คือแท็ก <h1> ถึง <h6> โดยที่เป็นแท็กคู่ที่ต้องเปิดและปิดแท็ก เช่น <h1> กำหนดหัวข้อ </h1> เป็นต้น หากไม่ปิดแท็กจะเกิดการกำหนดหัวข้อให้กับข้อความทั้งเว็บเพจได้ หัวข้อมีขนาดใหญ่จาก <H1> ไล่ลำดับไปหาขนาดเล็กที่สุดคือ <H6> เป็นต้น

ตัวอย่างที่ 6 แสดงตัวอย่างการกำหนดหัวเรื่อง

```
<!doctype html>
<html>
<head>
<title>ตัวอย่างกำหนดหัวเรื่อง</title>
</head>
<body>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
</body>
</html>
```

ผลการรันโปรแกรม



คำสั่งย่อหน้า

คำสั่งจัดย่อหน้าของเอกสารในภาษา HTML คือ แท็ก <p> ซึ่งมีการใช้งานอยู่สองลักษณะคือ ใช้งานแบบแท็กเดี่ยว และแท็กคู่ก็ได้ดังนี้

แท็ก <p> แบบเดี่ยว จะใช้จัดย่อหน้าใหม่ ขึ้นบรรทัดใหม่คล้ายแท็ก
 แต่จะมีช่องว่างระหว่างบรรทัดเพิ่มขึ้น รูปแบบการใช้งาน จะสั่งก่อนข้อความที่กำหนดให้ขึ้นย่อหน้าใหม่ ดังนี้

<p> ข้อความที่ต้องการจัดย่อหน้า

การใช้งานอาจใช้แท็ก <dd> เพิ่มการเยื้องย่อหน้าได้อีกด้วย

ตัวอย่างที่ 7 การใช้งานคำสั่ง <p> แบบแท็กเดี่ยว

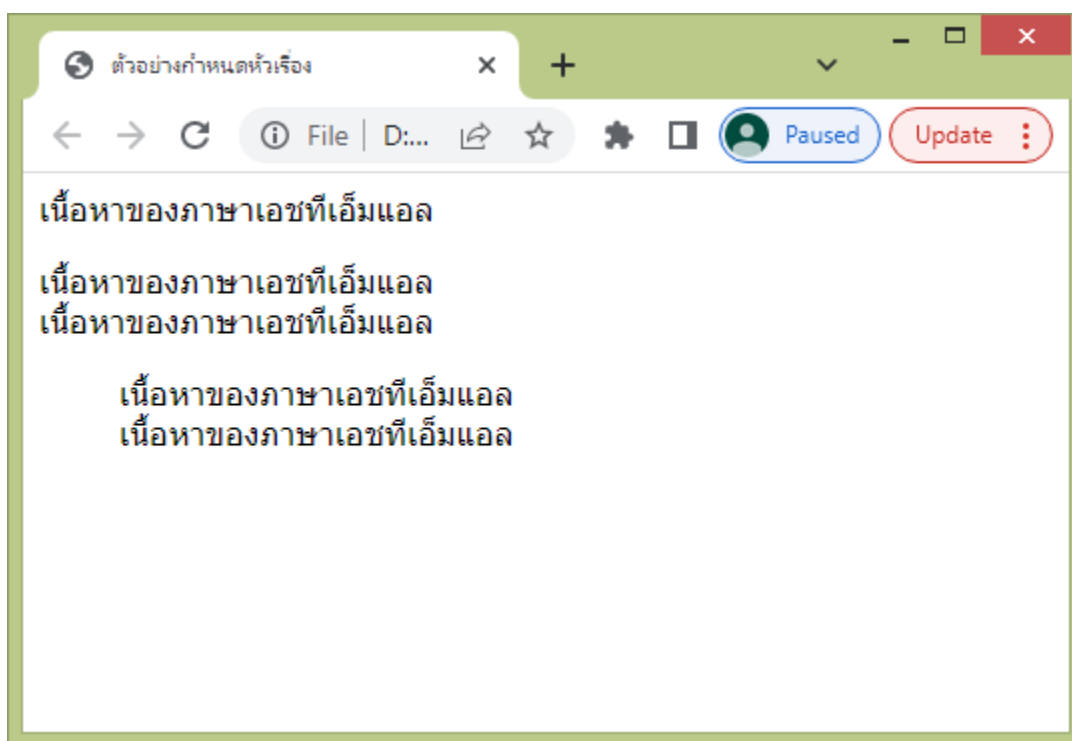
```
<!doctype html>
<html>
<head>
<title>ตัวอย่างกำหนดหัวเรื่อง</title>
</head>
```

```

<body>
  เนื้อหาของภาษาเอชทีเอ็มแอล
  <p>เนื้อหาของภาษาเอชทีเอ็มแอล
  <br>เนื้อหาของภาษาเอชทีเอ็มแอล
  <p><dd>เนื้อหาของภาษาเอชทีเอ็มแอล
  <br>เนื้อหาของภาษาเอชทีเอ็มแอล
</body>
</html>

```

ผลการรันโปรแกรม



การใช้งาน `<p>` แบบแท็กคู่ ใช้สำหรับจัดข้อความเป็นบล็อกเพื่อให้เป็นระเบียบโดยไม่ต้องขีดซ้ายตลอดเวลา เพราะสามารถสร้างคุณสมบัติให้ไปอยู่ตรงกลาง หรือด้านขวาของหน้าเพจได้ รูปแบบการใช้งานดังนี้

```
<p>ข้อความหรืออ็อบเจกต์อื่นๆ</p>
```

การกำหนดคุณสมบัติ จะเรียกใช้ดังนี้

```
<p align="left/right/center"> ข้อความ/อ็อบเจกต์</p>
```

หากต้องการให้ขีดซ้ายกำหนด align = "left"

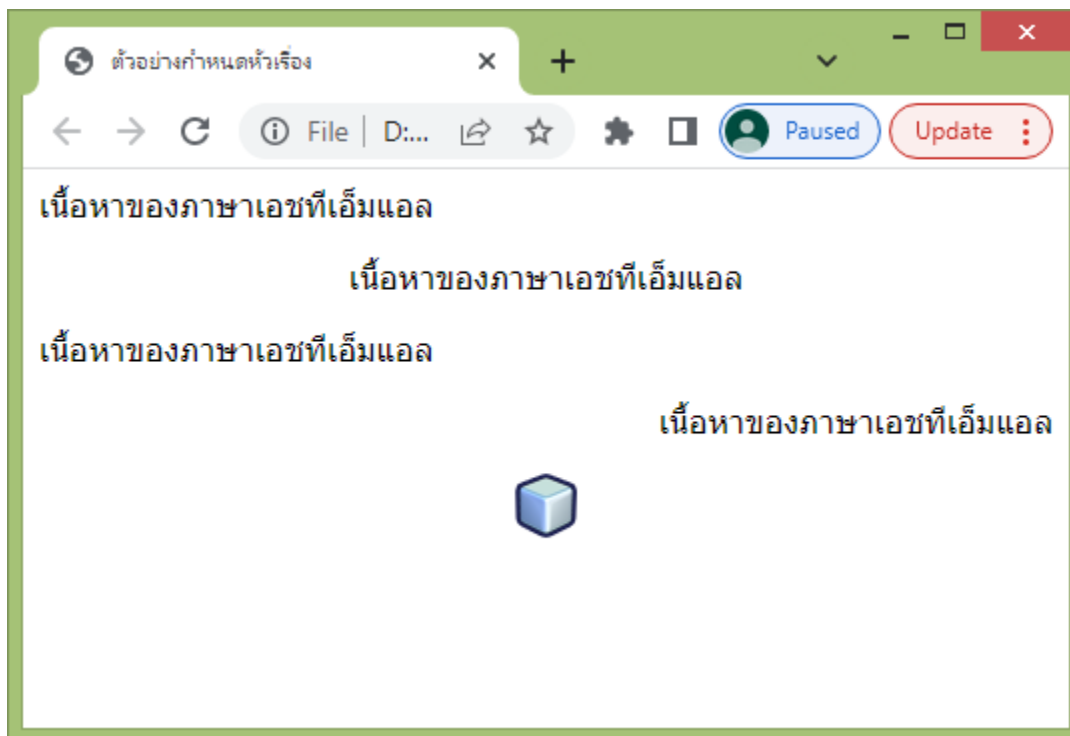
หากต้องการให้อยู่กึ่งกลางเพจกำหนด align = "center"

หากต้องการให้ชิดขวากำหนด align = "right"

ตัวอย่างที่ 8 การใช้งาน <p> แบบแท็กคู่

```
<!doctype html>
<html>
<head>
<title>ตัวอย่างกำหนดหัวเรื่อง</title>
</head>
<body>
เนื้อหาของภาษาเอชทีเอ็มแอล
<p align="center">เนื้อหาของภาษาเอชทีเอ็มแอล</p>
<p align="left">เนื้อหาของภาษาเอชทีเอ็มแอล</p>
<p align="right">เนื้อหาของภาษาเอชทีเอ็มแอล</p>
<p align="center"><IMG SRC="icon.png"></p>
</body>
</html>
```

ผลการรันโปรแกรม



การกำหนดการเชื่อมโยง

คำสั่งคือ แท็ก <a> ซึ่งเป็นแท็กคู่ รูปแบบคือ

```
<a href="เป้าหมาย">สิ่งที่จะสื่อถึงการเชื่อมโยง</a>
```

เมื่อ href= “เป้าหมายของการเชื่อมโยง”

เป้าหมายของการเชื่อมโยง อาจเป็น URL หรือ File หรือ ตำแหน่งในเอกสาร HTML ก็ได้

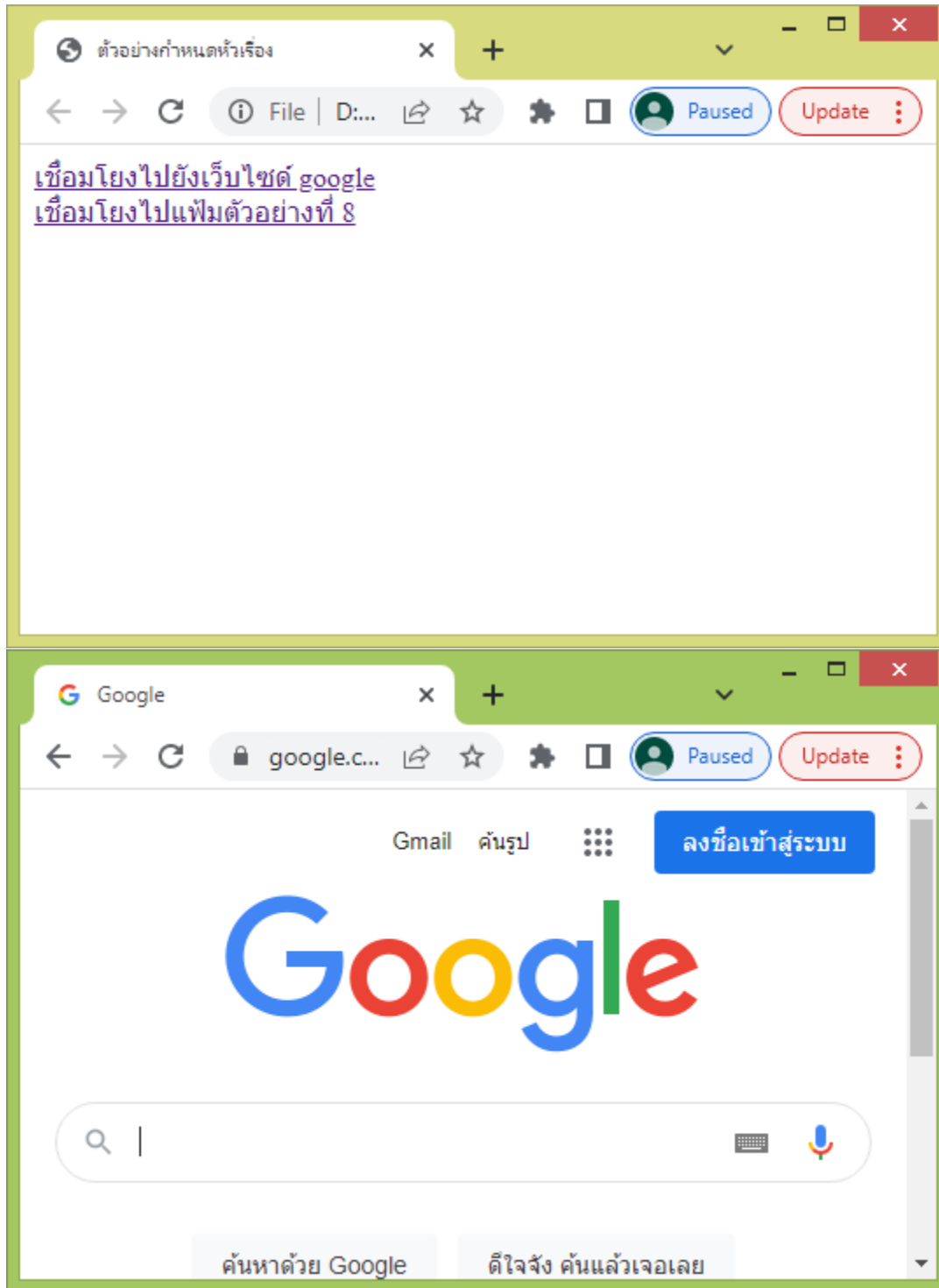
สิ่งที่ใช้สื่อถึงการเชื่อมโยงหมายถึง ข้อความ หรือรูปภาพ หรือวัตถุที่จะให้ผู้ใช้งานเว็บไซต์ได้เห็นตอน

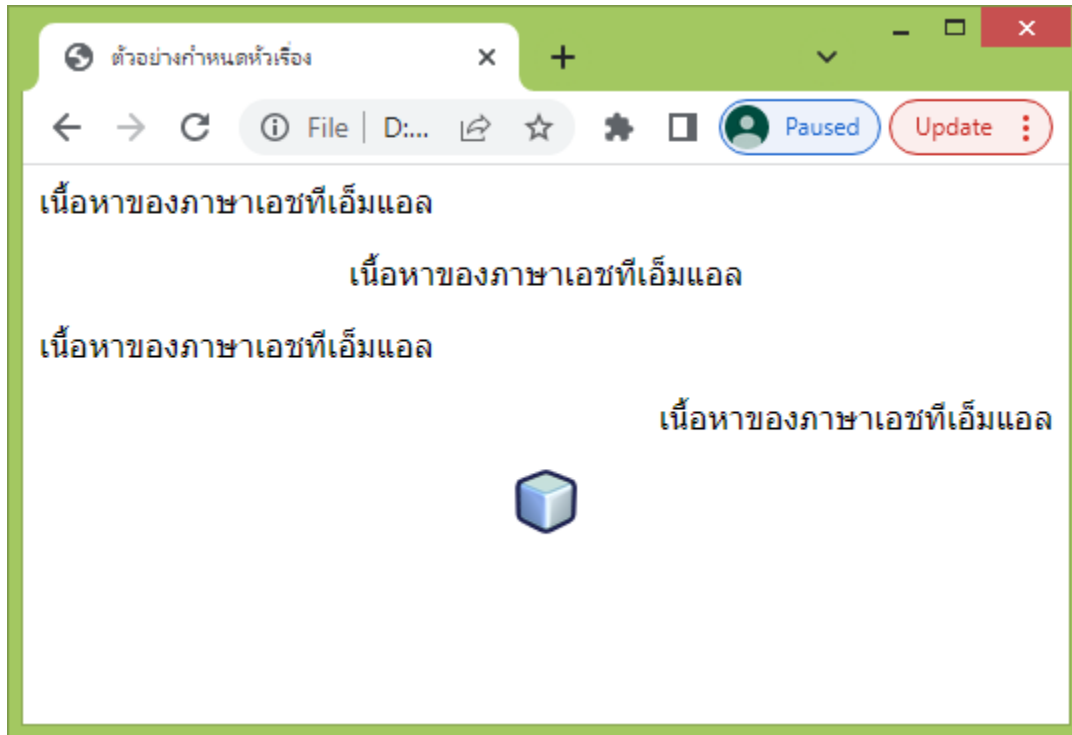
จะคลิกไปสู่เป้าหมาย

ตัวอย่างที่ 9 เชื่อมโยงไปยัง URL หรือแฟ้มอื่น

```
<!doctype html>
<html>
<head>
<title>ตัวอย่างกำหนดหัวเรื่อง</title>
</head>
<body>
  <a href="http://www.google.com">เชื่อมโยงไปยังเว็บไซต์ google</a>
  <br>
  <a href="example_8.html">เชื่อมโยงไปแฟ้มตัวอย่างที่ 8</a>
</body>
</html>
```

ผลการรันโปรแกรม





การเชื่อมโยงในเพจเดียวกัน คือการกำหนดจุดเชื่อมโยงเอาไว้ให้สามารถกระโดดไปยังจุดที่สร้างเอาไว้โดยกำหนดจุดคือ แท็ก `<a name= "ชื่อจุดที่ต้องการกระโดดมาจากการเชื่อม"> ข้อความแสดง </a>`

ตัวอย่างเช่น `<a name= "top"> หัวเพจ </a>`

`<a name= "bottom"> ท้ายเพจ </a>` เป็นต้น

สำหรับการเชื่อมโยงไปยังจุดที่กำหนด ต้องระบุ `<a href="#ชื่อจุดเชื่อมโยง">`

ตัวอย่างที่ 10 เชื่อมโยงไปยังตำแหน่งในหน้าเพจเดียวกัน

```
<!doctype html>
<html>
<head>
<title>ตัวอย่างกำหนดหัวเรื่อง</title>
</head>
<body>
  <a name="ส่วนหัว">เนื้อหาของภาษาเอกซ์ทีเอ็มแอล</a>
  <a href="#ส่วนท้าย">เชื่อมโยงไปท้ายเพจ</a>
```