

สร้างเว็บแบบไม่รู้อะไรเลย

ด้วย

AI Coding

+ พื้นฐาน HTML, CSS, JavaScript, React และ AI



เหมาะสำหรับ

นักเรียน นักศึกษาและ
บุคคลทั่วไปที่ต้องการ
เป็นนักพัฒนาเว็บ
ในยุค AI

- พื้นฐาน HTML, CSS และ JavaScript สำหรับนักพัฒนาเว็บ
- การใช้ React และ React Router สร้างเว็บแอปพลิเคชัน
- ตกแต่งสไตล์โดยใช้ CSS Module และยูทิลิตี้คลาสจาก Tailwind CSS
- ตัวอย่างการสร้างแอปพลิเคชันติดต่อกับ API
- การใช้ AI สร้างแอปพลิเคชันทั้งระบบ Frontend และ Backend
- วิธีเขียนโค้ดและแก้ไขโค้ดโดยใช้ AI Code Editor
- วิธีใช้งาน GitHub Copilot พัฒนาเว็บโดยใช้พลังจาก AI
- เทคนิคการสั่งให้ AI เรียนรู้คู่มือหรืออ่านเอกสารก่อนเขียนโค้ด

พิเศษ !

ดาวน์โหลดโค้ดตัวอย่าง
เพื่อใช้ประกอบการเรียน

จิราวุธ วารินทร์

สารบัญ

Part 1 HTML		
01	พื้นฐานสำหรับการพัฒนาเว็บ	8
02	ติดตั้งเครื่องมือ และทดสอบโค้ด HTML	18
03	พื้นฐาน HTML	29
04	ข้อความและรูปภาพ	42
05	จัดโครงสร้างและแบ่งกลุ่มเอลิเมนต์	53
06	HTML5	62
07	ตาราง	67
08	แบบฟอร์มและอินพุต	73
Part 2 CSS		
09	พื้นฐาน CSS	94
10	การกำหนดสี	103
11	การปรับแต่งข้อความ	112
12	การใช้ Selector เลือกเอลิเมนต์	125
13	CSS Box Model	142
14	การจัดตำแหน่งเอลิเมนต์ (Positioning)	157
15	จัดเรียงเอลิเมนต์ด้วย Flexbox	170
16	การใช้ Transition	197
17	การออกแบบเว็บเพจสไตล์ Responsive Design	206
18	การใช้งาน Tailwind CSS	212
Part 3 Javascript		
19	แนะนำ JavaScript	232
20	ชนิดข้อมูล (Data Type)	246
21	สตริง	257
22	ชนิดข้อมูลแบบ number	269
23	ฟังก์ชันและสโคป	282
24	การตัดสินใจ	300
25	การวนซ้ำ	315
26	ออบเจกต์และ JSON	323
27	อาร์เรย์	337
28	คลาสและการสืบทอดคลาส	354
29	ใช้ JavaScript จัดการ DOM	366

Part 4 React		
30	พื้นฐานการใช้งาน React	384
31	คอมโพเนนต์	392
32	การตกแต่งสไตล์ใน React	403
33	การใช้งานพรอปส์และวิธีกำหนดอีเวนต์	413
34	พื้นฐานการใช้งาน state ใน React	426
35	การใช้ฮุคในฟังก์ชันคอมโพเนนต์	444
36	การใช้ Promise และการติดต่อกับ API	462
37	พื้นฐานการใช้งาน Context	481
Part 5 React Router		
38	การใช้ React Router	490
39	การกำหนดเมนูและลิงก์	498
40	การใช้งาน Loader	506
41	การกำหนดรูล่อย	516
42	การส่งข้อมูลจากแบบฟอร์ม	523
43	การใช้งาน Git และการ Deploy แอปพลิเคชัน	535
Part 6 AI Coding		
44	เทคนิคการใช้ Prompt สั่งงาน AI	548
45	การนำ AI มาใช้พัฒนาโปรแกรม	554
46	รู้จักและใช้งาน AI Code Editor	562

The book cover features a dark gray background with a subtle, repeating geometric pattern of interlocking hexagons. A vertical stripe on the left side consists of diagonal lines in varying shades of gray. A dark gray horizontal band at the bottom contains the title text in white.

Basic Web Development + AI Coding

03

พื้นฐาน HTML

HTML คือ โครงสร้างของเว็บเพจ สามารถเปรียบ HTML ได้กับเสาและคานที่เป็นโครงสร้างหลักของอาคารบ้านเรือน หากไม่มี HTML ก็ไม่สามารถแสดงเนื้อหาที่อยู่ในเว็บเพจได้

โครงสร้างของ HTML

HTML ใช้เพื่อกำหนดโครงสร้างของเว็บเพจ บอกให้ทราบว่าเว็บเพจจะประกอบด้วยอะไรบ้าง และมีการจัดเรียงอย่างไร โดย HTML จะถูกตีความโดยเบราว์เซอร์และแสดงผลออกมาเป็นหน้าเว็บเพจ

สามารถแบ่ง HTML ได้เป็น 2 ส่วนหลัก ได้แก่ ส่วน head และส่วน body

- head คือ ส่วนที่ใช้เก็บข้อมูลของเว็บเพจ ซึ่งข้อมูลในส่วนนี้จะไม่แสดงให้ผู้ใช้เห็นโดยตรง เช่น คำอธิบายเว็บเพจ การเข้ารหัสภาษา โค้ด CSS (ใช้ปรับแต่งสไตล์เว็บเพจ) หรือโค้ด JavaScript (ใช้ทำให้เว็บเพจโต้ตอบได้)
- body คือ ส่วนที่ใช้เก็บเนื้อหาซึ่งจะถูกนำมาแสดงบนเว็บเพจ เช่น ข้อความ รูปภาพ ปุ่ม ตาราง ฯลฯ

โครงสร้างพื้นฐานของ HTML จะมีรูปแบบดังนี้

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!-- ข้อมูลของเว็บเพจ -->
5   </head>
6   <body>
7     <!-- เนื้อหาของเว็บเพจ -->
8   </body>
9 </html>
```

- บรรทัดที่ 1 คำสั่ง <!DOCTYPE html> คือ คำประกาศที่บอกว่า โค้ด HTML ที่ใช้ในเว็บเพจนี้เป็น HTML5 ซึ่งเป็น HTML เวอร์ชันล่าสุด
- บรรทัดที่ 2-9 แท็ก <html></html> คือ เอลิเมนต์หลักของเว็บเพจ (Root Element) ที่ใช้ครอบเนื้อหาทั้งหมดของเว็บเพจ
- บรรทัดที่ 3-5 แท็ก <head></head> คือ ส่วนที่ใช้กำหนดข้อมูลของเว็บเพจ ซึ่งข้อมูลในส่วนนี้จะไม่ถูกนำมาแสดงให้ผู้ใช้เห็นโดยตรง
- บรรทัดที่ 6-8 แท็ก <body></body> คือ ส่วนที่ใช้กำหนดเนื้อหาของเว็บเพจ

กำหนดข้อมูลเว็บเพจลงในแท็ก <head></head>

HTML จะมีแท็ก <head></head> จำนวน 1 แท็กเท่านั้น ใช้เพื่อเก็บข้อมูลเกี่ยวกับเว็บเพจที่ไม่ต้องการแสดงผลบนหน้าเว็บเพจโดยตรง สามารถกำหนดข้อมูลภายใน <head></head> ได้หลายรูปแบบ เช่น

```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <title>Document</title>
5   <link rel="stylesheet" href="style.css">
6   <script src="script.js"></script>
7 </head>
```

- บรรทัดที่ 2 <meta charset="UTF-8"> กำหนดรูปแบบการเข้ารหัสตัวอักษร เพื่อให้เว็บเพจรองรับภาษาไทยหรือภาษาอื่น ๆ ได้อย่างถูกต้อง
- บรรทัดที่ 3 <meta name="viewport" content="width=device-width, initial-scale=1.0"> กำหนดให้หน้าเว็บเพจรองรับการแสดงผลในอุปกรณ์ต่าง ๆ โดยสามารถปรับขนาดให้เหมาะสมกับความกว้างของหน้าจอของอุปกรณ์
- บรรทัดที่ 4 <title>Document</title> กำหนดชื่อเว็บเพจ ซึ่งจะแสดงผลบนแท็บของเบราว์เซอร์
- บรรทัดที่ 5 <link rel="stylesheet" href="style.css"> เชื่อมโยงกับไฟล์ CSS ภายนอก สำหรับปรับแต่งสไตล์เว็บเพจ
- บรรทัดที่ 6 <script src="script.js"></script> ใช้ไฟล์ JavaScript จากภายนอก เพื่อให้เว็บเพจสามารถโต้ตอบกับผู้ใช้ได้

กำหนดเนื้อหาภายในแท็ก <body></body>

HTML จะมีแท็ก <body></body> จำนวน 1 แท็ก ใช้เพื่อกำหนดเนื้อหาที่จะนำมาแสดงผลบนเว็บ เช่น ข้อความ รูปภาพ ปุ่ม ตาราง แบบฟอร์ม ฯลฯ

```
1 <body>
2   <h1>My Header</h1>
3   <p>My Paragraph</p>
4   <div>My Section</div>
5 </body>
```

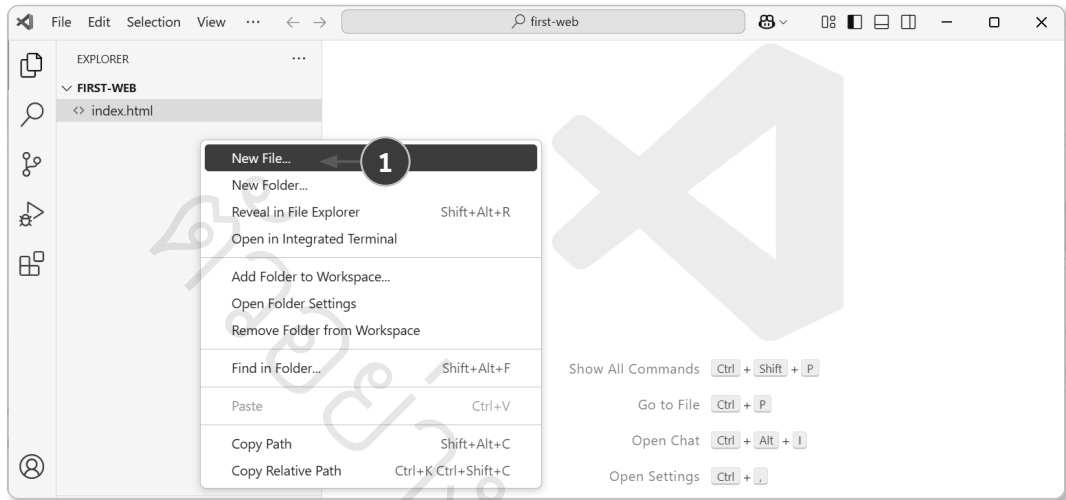
- บรรทัดที่ 2 <h1>My Header</h1> กำหนดหัวข้อ ซึ่งใช้แสดงข้อความสำคัญ (heading) บอกถึงจุดประสงค์ของเว็บเพจ
- บรรทัดที่ 3 <p>My Paragraph</p> กำหนดย่อหน้า (paragraph) เพื่อแสดงข้อความบนเว็บเพจ
- บรรทัดที่ 4 ใช้ <div></div> เพื่อจัดกลุ่มเนื้อหา เช่น ใช้แยกระหว่างส่วนเมนูกับส่วนที่เป็นเนื้อหาหลัก เป็นต้น การแบ่งกลุ่มยังช่วยให้เราสามารถเลือกปรับแต่งสไตล์ หรือกำหนดส่วนที่ต้องติดต่อกับผู้ใช้ได้ง่ายขึ้นด้วย

ตัวอย่าง 3.1

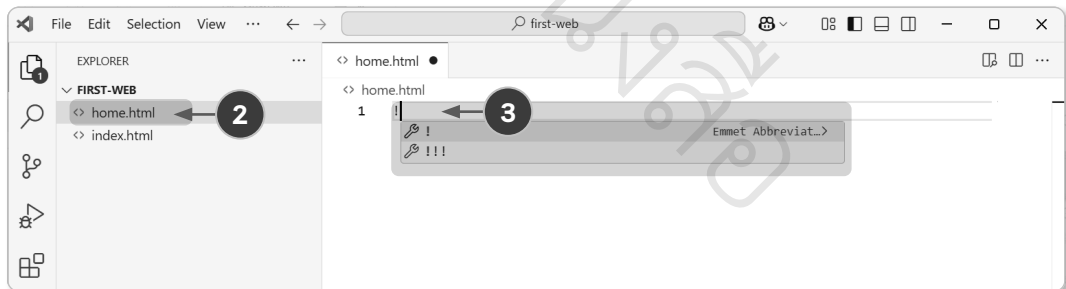
สร้างไฟล์ HTML อย่างรวดเร็ว

การเขียนโค้ด HTML ลงในโปรเจกต์ สามารถพิมพ์โค้ด HTML ทั้งหมดด้วยตนเอง หรือใช้วิธีลัด ดังต่อไปนี้

1. ที่หน้าต่างซ้ายมือของ VS Code (เรียกว่า Explorer) จะแสดงโครงสร้างของไฟล์โฟลเดอร์ที่อยู่ภายในโปรเจกต์ สร้างไฟล์ใหม่ ด้วยวิธีคลิกขวาบริเวณที่ว่างภายใน Explorer และเลือกคำสั่ง New File (หรือคลิกปุ่ม )



2. ตั้งชื่อไฟล์ โดยใช้นามสกุลไฟล์เป็น .html เช่น สร้างไฟล์ home.html
3. แทรกโค้ด HTML โดยพิมพ์ ! จากนั้นกดปุ่ม <Enter> หรือพิมพ์ html:5 แล้วกดปุ่ม <Enter>



4. โค้ด HTML ที่เป็นโครงสร้างพื้นฐานของ HTML จะถูกแทรกลงในไฟล์โดยอัตโนมัติ

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9
10 </body>
11 </html>

```

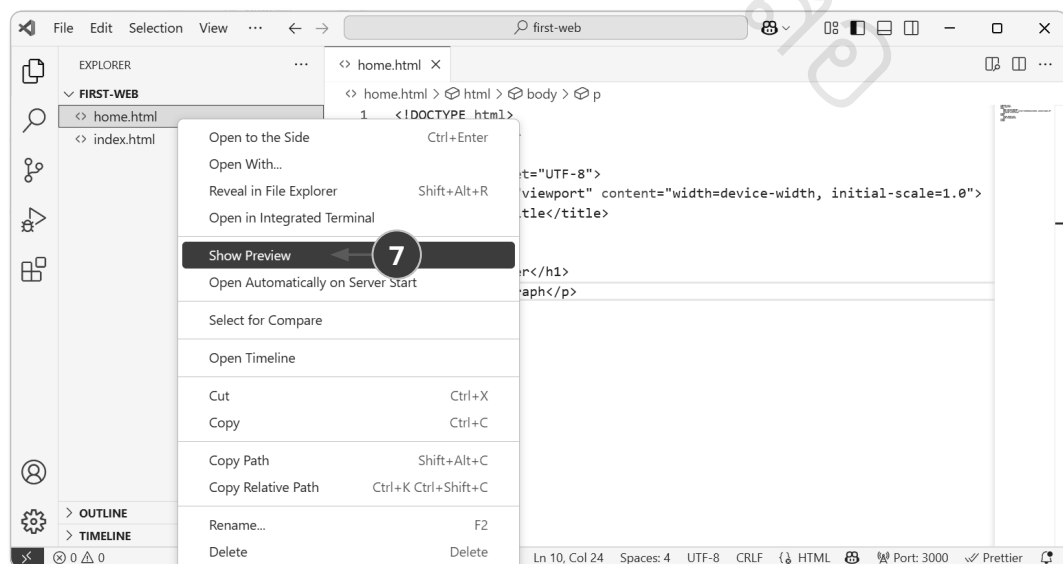
- บรรทัดที่ 1 คำสั่ง <!DOCTYPE html> บอกให้ทราบว่าเว็บเพจนี้เป็น HTML5
 - บรรทัดที่ 2-11 แท็ก <html></html> ใช้กำหนดขอบเขตของโค้ด HTML
 - บรรทัดที่ 3-7 แท็ก <head></head> ใช้กำหนดข้อมูลให้กับเว็บเพจ เป็นข้อมูลที่ไม่ต้องการนำมาแสดงผลบนเว็บเพจโดยตรง
 - บรรทัดที่ 6 แท็ก <title></title> ใช้แสดงชื่อเว็บเพจ ซึ่งควรใส่ในทุก ๆ เพจ จุดประสงค์ คือ เพื่อให้ Google หรือ Search Engine อื่น ๆ ทราบว่า เพจนี้เกี่ยวข้องกับอะไร
 - บรรทัดที่ 8-10 แท็ก <body></body> ใช้เพื่อกำหนดเนื้อหาที่จะนำมาแสดงผลบนเว็บเพจ
5. เปลี่ยนชื่อเว็บเพจ โดยแก้ไขโค้ดเป็น <title>My Title</title>
 6. ภายใน <body></body> ให้แทรก <h1>My Header</h1> ไว้ เพื่อแสดงหัวข้อที่มีความสำคัญ และแทรก <p>My Paragraph</p> เพื่อกำหนดย่อหน้าบนเว็บเพจ

```

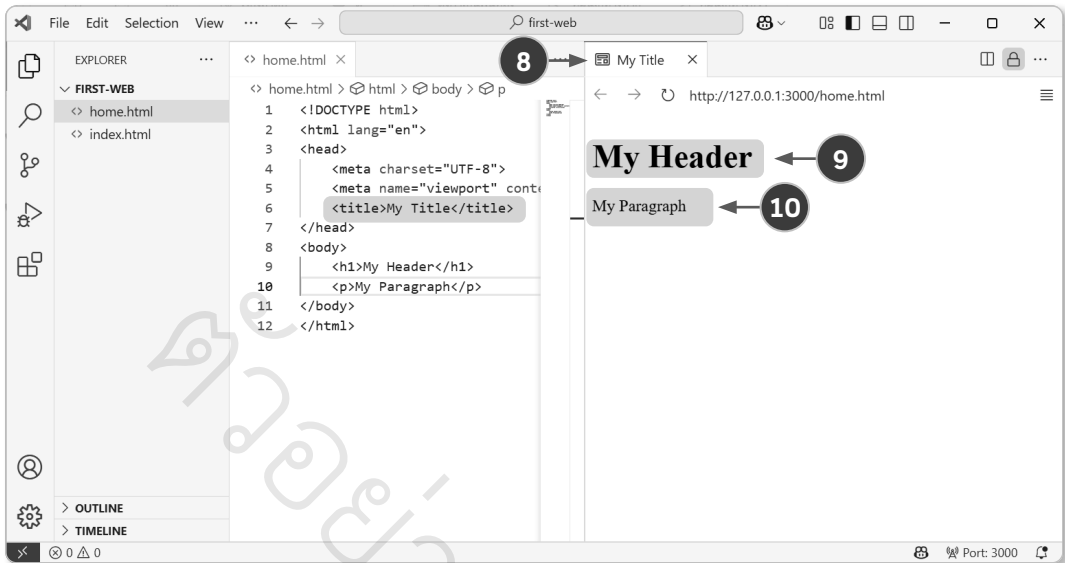
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>My Title</title>
7 </head>
8 <body>
9   <h1>My Header</h1>
10  <p>My Paragraph</p>
11 </body>
12 </html>

```

7. ดูผลลัพธ์โดยใช้ Live Preview โดยคลิกขวาที่ไฟล์ home.html แล้วเลือกคำสั่ง Show Preview



8. ข้อความ My Title แสดงที่แท็บด้านบนของเว็บเพจ (ข้อความนี้กำหนดไว้ที่แท็ก <title>My Title</title>)
9. สังเกตผลลัพธ์บนเว็บเพจ จะพบข้อความ My Header ที่มีขนาดใหญ่ที่บอกถึงหัวข้อหลักของเว็บเพจ
10. พบข้อความ My Paragraph ที่มีตัวอักษรขนาดเล็ก แสดงถึงเนื้อหาของเว็บเพจทั่วไป



Note

ใน VS Code จะมีเครื่องมือที่เรียกว่า Emmet เป็นตัวช่วยซึ่งทำให้การเขียนโค้ด HTML และ CSS ทำได้เร็วขึ้น เมื่อเรารอกรอคำสั่งย่อ (Abbreviations) โค้ดเต็มก็จะถูกนำมาแทรกโดยอัตโนมัติ เช่น กรอก ! แล้วกดปุ่ม <Enter> ก็จะแทรกโค้ด HTML แบบเต็มให้อัตโนมัติ การใช้ Emmet จึงช่วยประหยัดเวลาในการเขียนโค้ดได้มาก

แท็ก (Tag) และเอลิเมนต์ (Element)

ก่อนเริ่มเขียนโค้ด HTML สิ่งที่ต้องเข้าใจเป็นลำดับแรก คือ แท็ก (Tag) และเอลิเมนต์ (Element)

แท็ก (Tag)

แท็ก (Tag) คือ เครื่องหมายที่ใช้กำหนดประเภทของเนื้อหาใน HTML เพื่อบอกให้เบราว์เซอร์ทราบว่าการแสดงผลในรูปแบบใด เช่น หัวข้อ ย่อหน้า ปุ่ม รูปภาพ ฯลฯ

- แท็ก <title></title> กำหนดชื่อเว็บเพจ
- แท็ก <body></body> กำหนดเนื้อหาที่ต้องการนำมาแสดงผลบนเว็บเพจ
- แท็ก <h1></h1> แสดงหัวข้อหลักของเว็บเพจ เพื่ออธิบายจุดประสงค์ของเว็บเพจ
- แท็ก <p></p> แสดงข้อความในย่อหน้า
- แท็ก แสดงรูปภาพบนเว็บเพจ
- แท็ก
 กำหนดให้ขึ้นบรรทัดใหม่

แท็กใน HTML จะมีอยู่ 2 ประเภท ได้แก่ แท็กคู่ (Open/Close Tag) และแท็กเดี่ยว (Self-Closing Tag)

1. แท็กคู่ (Open/Close Tag) จะประกอบด้วยแท็กจำนวน 2 ชุด ได้แก่ แท็กเปิด (Open Tag) และแท็กปิด (Close Tag) ครอบเนื้อหาที่อยู่ภายใน

```
1 <body>
2   <h1>หัวข้อหลัก</h1>
3   <p>ข้อความในย่อหน้า</p>
4 </body>
```

- บรรทัดที่ 2 ใช้แท็ก <h1></h1> เพื่อแสดงหัวข้อหลัก ประกอบด้วยแท็กเปิด <h1> และแท็กปิด </h1>
 - บรรทัดที่ 3 ใช้แท็ก <p></p> เพื่อแสดงข้อความในย่อหน้า ประกอบด้วยแท็กเปิด <p> และแท็กปิด </p>
2. แท็กเดี่ยว (Self-Closing Tag) คือ การกำหนดแท็กที่มีแต่แท็กเปิดไม่ต้องมีแท็กปิด

```
1 <body>
2   
3   <hr>
4   <br>
5 </body>
```

- บรรทัดที่ 2 ใช้แท็ก แสดงรูปภาพ (ใช้เพียงแท็กเปิด ไม่ต้องใส่แท็กปิด)
- บรรทัดที่ 3 ใช้แท็ก <hr> แสดงเส้นคั่น (ใช้เพียงแท็กเปิด <hr> ไม่ต้องใส่แท็กปิด </hr>)
- บรรทัดที่ 4 ใช้แท็ก
 สั่งให้ข้อความขึ้นบรรทัดใหม่ (ใช้เพียงแท็กเปิด
 ไม่ต้องใส่แท็กปิด </br>)

Note

สามารถเขียนแท็กเดี่ยว (Self-Closing Tag) ในสไตล์ XML หรือ XHTML โดยเริ่มด้วยเครื่องหมาย < และปิดท้ายแท็กด้วย /> เช่น

- ใช้แท็ก แสดงรูปภาพ
- ใช้แท็ก <hr /> แสดงเส้นคั่น
- ใช้แท็ก
 เพื่อสั่งให้ขึ้นบรรทัดใหม่

ปัจจุบันไม่นิยมเขียน HTML ในสไตล์ XML หรือ XHTML แล้ว นิยมเขียนในแบบ HTML5 มากกว่า เช่น , <hr> และ
 เป็นต้น

เอลิเมนต์ (Element)

เอลิเมนต์ (Element) คือ ส่วนย่อยที่อยู่ภายในโครงสร้าง HTML ซึ่งสามารถแบ่งได้เป็น 3 แบบ ได้แก่ Void Element, Raw Text Element และ Nested Elements

09

พื้นฐาน CSS

CSS คือ ภาษาที่ใช้กำหนดรูปแบบการแสดงผลในเว็บเพจ ใช้เพื่อตกแต่งหน้าตา สี สัน สไตล์ หรือกำหนดรูปแบบการจัดเรียงส่วนประกอบต่าง ๆ ในเว็บเพจ การกำหนดสไตล์โดยใช้ CSS สามารถทำได้หลายวิธี เช่น inline style, internal style และ external style

โครงสร้างของ CSS

โครงสร้างพื้นฐานของคำสั่ง CSS จะเป็นดังนี้

```
1 selector {  
2     property1: value1;  
3     property2: value2;  
4     ...  
5 }
```

- selector เป็นส่วนที่ใช้เลือกที่จะปรับแต่งส่วนใดของเว็บเพจ เช่น เลือกปรับแต่งแท็ก <div></div> ทั้งหมดที่อยู่ในเว็บเพจ หรือเลือกปรับแต่งเฉพาะแท็ก <div></div> แรก เป็นต้น
- property คือ ชื่อของคุณสมบัติ
- value คือ ค่าที่จะกำหนดให้กับคุณสมบัตินั้น ๆ

จากโครงสร้างพื้นฐานของ CSS ประกอบด้วยส่วนสำคัญ 2 ส่วน ได้แก่ ส่วนที่ใช้เลือกเอลิเมนต์จาก DOM (เรียกว่า selector) และส่วนที่ใช้ปรับแต่งสไตล์ให้กับเอลิเมนต์

1. ใช้ selector เลือกเอลิเมนต์ที่ต้องการจาก DOM
 2. กำหนดค่าไปยังพรอปเพอร์ตี้เพื่อปรับแต่งสไตล์ให้กับเอลิเมนต์ ซึ่งจะมีผลเฉพาะเอลิเมนต์ที่ได้เลือกมา
- ตัวอย่างการเขียน CSS เพื่อแก้ไขสไตล์ให้กับแท็ก <h1></h1> โดยกำหนดให้ข้อความเป็นสีขาว และสีของพื้นหลังเป็นสีแดง

```
1 h1 {  
2     color: white;  
3     background-color: red;  
4 }
```

- บรรทัดที่ 1 กำหนด selector เลือกแท็ก <h1></h1> ทั้งหมด
- บรรทัดที่ 2 คำสั่ง color: white; เปลี่ยนสีข้อความเป็นสีขาว
- บรรทัดที่ 3 คำสั่ง background-color: red; เปลี่ยนสีพื้นหลังเป็นสีแดง

 Note

ชื่อพรอปเพอร์ตี้ที่ใช้ใน CSS จะเป็นคำศัพท์ภาษาอังกฤษที่มีความหมายที่ค่อนข้างชัดเจน เช่น พรอปเพอร์ตี้ background หมายถึง พื้นหลังของเอลิเมนต์, พรอปเพอร์ตี้ color จะหมายถึง สีของเอลิเมนต์ และ border หมายถึง ขอบของเอลิเมนต์ ฯลฯ

สำหรับชื่อพรอปเพอร์ตียาว ๆ ที่ประกอบด้วยหลายคำ จะใช้ขีดกลางคั่นระหว่างคำ เช่น background-color, border-width, border-radius, margin-right, margin-bottom เป็นต้น

วิธีนำ CSS มาใช้กับเว็บเพจ

วิธีนำโค้ด CSS มาปรับแต่งสไตลให้กับเว็บเพจ สามารถทำได้ 3 วิธี

- วิธีที่ 1 กำหนดคำสั่ง CSS ไปยังแอตทริบิวต์ style ของ HTML แท็ก
- วิธีที่ 2 เขียนโค้ด CSS ภายในแท็ก <style></style> ของ HTML
- วิธีที่ 3 ลิงก์ไปยังไฟล์ CSS (ไฟล์นามสกุล .css)

วิธีที่ 1 กำหนด CSS ไปยังแอตทริบิวต์ style ของ HTML แท็ก

สามารถเขียนคำสั่ง CSS ลงในแอตทริบิวต์ style ของแท็ก เพื่อปรับแต่งเอลิเมนต์นั้น ๆ โดยตรง เรียกวิธีกำหนดสไตล์แบบนี้ว่า inline style

```
1 <h1 style="text-align: center;">หัวข้อใหญ่</h1>
2 <p style="color: gray;">เนื้อหาในย่อหน้า</p>
```

- บรรทัดที่ 1 กำหนดคำสั่ง text-align: center; ไปยังแอตทริบิวต์ style เพื่อจัดให้ข้อความอยู่ตรงกลาง
- บรรทัดที่ 2 กำหนดคำสั่ง color: gray; ไปยังแอตทริบิวต์ style เพื่อใช้ตัวอักษรสีเทา

คำสั่ง CSS ที่กำหนดลงไปแอตทริบิวต์ style ของ HTML จะประกอบด้วยคู่ของ property: value; สามารถใช้คำสั่ง CSS หลายคำสั่ง โดยแต่ละคำสั่งจะต้องปิดท้ายด้วยเครื่องหมาย ; เพื่อแยกแต่ละคำสั่งออกจากกัน

```
1 <h1 style="color: blue; text-align: center;">หัวข้อใหญ่</h1>
2 <p style="color: gray; font-size: 16px;">เนื้อหาในย่อหน้า</p>
```

- บรรทัดที่ 2 คำสั่ง color: gray; กำหนดข้อความเป็นสีเทา คำสั่ง font-size: 16px; กำหนดขนาดตัวอักษรเป็น 16 พิกเซล

 Note

การเขียนคำสั่ง CSS ไปยังแอตทริบิวต์ style ไม่ต้องใส่ส่วนที่เป็น selector เนื่องจากเป็นการกำหนดโค้ด CSS ให้กับเอลิเมนต์โดยตรง วิธีนี้เหมาะสำหรับการทดสอบการแก้ไขสไตล์ เนื่องจากคำสั่ง CSS จะมีผลเฉพาะเอลิเมนต์ที่สนใจเพียงเอลิเมนต์เดียว

วิธีที่ 2 เขียนโค้ด CSS ภายในแท็ก <style></style>

สามารถเขียนคำสั่ง CSS ลงในแท็ก <style></style> เพื่อปรับแต่งสไตล์ให้กับเอลิเมนต์ต่าง ๆ ที่อยู่ในเพจปัจจุบัน เรียกวิธีกำหนดสไตล์แบบนี้ว่า internal style

ให้กำหนดแท็ก <style></style> ไว้ภายในแท็ก <head></head> ที่เป็นส่วนหัวของ HTML

```
1 <!DOCTYPE html>
2 <html lang="th">
3 <head>
4   <style>
5     /* แทรกคำสั่ง CSS ตรงนี้ */
6   </style>
7 </head>
8 <body>
9
10
11 </body>
12 </html>
```

โค้ด CSS ที่อยู่ในแท็ก <style></style> จะเริ่มจากการใช้ selector เลือกเอลิเมนต์ที่ต้องการจากเว็บเพจปัจจุบัน จากนั้นจึงกำหนดค่าไปยังพรอปเพอร์ตี้ของเอลิเมนต์ เพื่อปรับแต่งเอลิเมนต์ตามต้องการ

```
1 <head>
2   <style>
3     h1 {
4       color: blue;
5       text-align: center;
6     }
7     p {
8       color: gray;
9       font-size: 16px;
10    }
11  </style>
12 </head>
13 <body>
14   <h1>หัวข้อใหญ่</h1>
15   <p>เนื้อหาในย่อหน้า</p>
16 </body>
```

- บรรทัดที่ 2-11 กำหนดแท็ก <style></style> ไว้ภายใต้แท็ก <head></head>
- บรรทัดที่ 3-6 ใช้คำสั่ง CSS เพื่อปรับแต่งแท็ก <h1></h1>
- บรรทัดที่ 7-10 ใช้คำสั่ง CSS เพื่อปรับแต่งแท็ก <p></p>

Note

การเขียนโค้ด CSS ไว้ภายในแท็ก <style></style> เป็นการปรับแต่งสไตล์ให้กับเว็บเพจปัจจุบัน ไม่สามารถแชร์สไตล์ไปใช้กับเว็บเพจอื่น ๆ ผ่านทางลิงก์ได้ หากต้องการสไตล์แบบเดียวกันจะต้องคัดลอกโค้ดทั้งหมดไปใช้งาน

วิธีที่ 3 ลิงก์ไปยังไฟล์ CSS

เราสามารถเขียนโค้ด CSS ลงในไฟล์ .css หากเว็บเพจใดต้องการใช้งานโค้ด CSS จากไฟล์ ก็ให้กำหนดแท็ก <link> เพื่อลิงก์ไปยังไฟล์ CSS ที่ต้องการ เรียกวิธีการใช้โค้ด CSS จากไฟล์ภายนอกเช่นนี้ว่า external style

การใช้งาน CSS จากไฟล์ภายนอก มีขั้นตอน ดังนี้

1. สร้างไฟล์นามสกุล .css (เช่น ไฟล์ style.css) เพื่อเก็บโค้ด CSS จากนั้นใส่โค้ด CSS ลงในไฟล์

```
1 h1 {
2   color: blue;
3   text-align: center;
4 }
5 p {
6   color: gray;
7   font-size: 16px;
8 }
```

2. เว็บเพจที่ต้องการใช้งาน CSS จากไฟล์ภายนอก (เช่น ไฟล์ index.html) จะกำหนดแท็ก <link> และระบุพาธแสดงชื่อและที่อยู่ของไฟล์ CSS ในแอตทริบิวต์ href

```
1 <head>
2   <link rel="stylesheet" href="style.css">
3 </head>
```

- บรรทัดที่ 1-3 กำหนดแท็ก <head></head> ซึ่งเก็บข้อมูลเกี่ยวกับส่วน header ของ HTML
- บรรทัดที่ 2 กำหนดแท็ก <link> เพื่อลิงก์ไปยังโค้ด CSS ซึ่งอยู่ที่ไฟล์ style.css (อยู่ในโฟลเดอร์เดียวกับโฟลเดอร์ปัจจุบัน)

การใช้ CSS จากไฟล์ภายนอกถือเป็นวิธีที่นิยม เนื่องจากเป็นวิธีปรับแต่งสไตล์จากศูนย์กลาง แก้ไขครั้งเดียวก็จะมีผลต่อทุก ๆ เว็บเพจได้ทันที ไม่ต้องตามแก้ไขทีละไฟล์

- แยกโค้ด CSS ออกจาก HTML ทำให้โค้ดเป็นระเบียบ เข้าใจง่าย
- สามารถกำหนดไฟล์ .css เพียงไฟล์เดียว ตกแต่งหลาย ๆ เว็บเพจได้พร้อมกัน

Note

แอตทริบิวต์ href ของแท็ก <link> คือ พาธซึ่งบอกให้ทราบว่าไฟล์ CSS ที่จะใช้งานเก็บอยู่ที่ไหนและมีชื่อว่าอะไร ดังนั้น หากไฟล์ CSS เก็บอยู่ในโฟลเดอร์ย่อย ก็ต้องใส่ชื่อโฟลเดอร์ลงในพาธให้ถูกต้องด้วย

```
1 <head>
2   <link rel="stylesheet" href="css/style.css">
3 </head>
```

- บรรทัดที่ 2 ที่แอตทริบิวต์ href กำหนดพาธเป็น "css/style.css" หมายถึง ต้องการโค้ด CSS ที่อยู่ในไฟล์ style.css ที่อยู่ในโฟลเดอร์ css

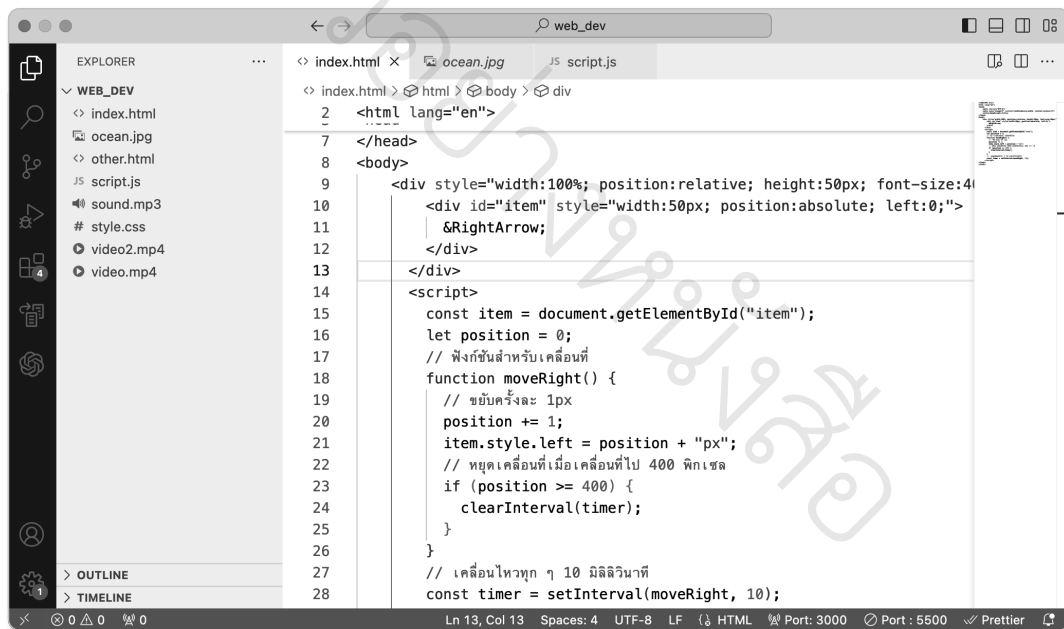
19

แนะนำ JavaScript

JavaScript เป็นภาษาคอมพิวเตอร์ที่ออกแบบมาใช้กับเว็บเพจ ดังนั้น โค้ด JavaScript จึงถูกรันเพื่อใช้งานบนเบราว์เซอร์เป็นหลัก ในบทนี้เป็นการทดลองเขียนและทดสอบโค้ด JavaScript อย่างง่าย ๆ ผ่านทางเว็บไซต์ เช่น <https://codepen.io> และทดสอบโค้ด JavaScript ด้วยวิธีจำลองการทำงานของเซิร์ฟเวอร์ (Live Server)

ภาษา JavaScript

JavaScript เป็นภาษาคอมพิวเตอร์ที่ถูกออกแบบมาให้ทำงานบนเบราว์เซอร์ จุดประสงค์เพื่อให้เว็บเพจสามารถโต้ตอบกับผู้ใช้ได้ หรือมีความสามารถบางอย่างเพิ่มขึ้น เช่น สามารถใช้ JavaScript เพื่อขยายรูปภาพเมื่อผู้ใช้เลื่อนเมาส์ไปทีรูปภาพ, ใช้ JavaScript เพื่อสร้างแอนิเมชันเลื่อนข้อความจากซ้ายไปขวา, ใช้ JavaScript เพื่อนำเนื้อหาใหม่มาแทนที่เนื้อหาเดิมโดยไม่ต้องโหลดเว็บเพจใหม่ เป็นต้น



▲ รูปแสดงตัวอย่างการใช้ HTML ร่วมกับ JavaScript

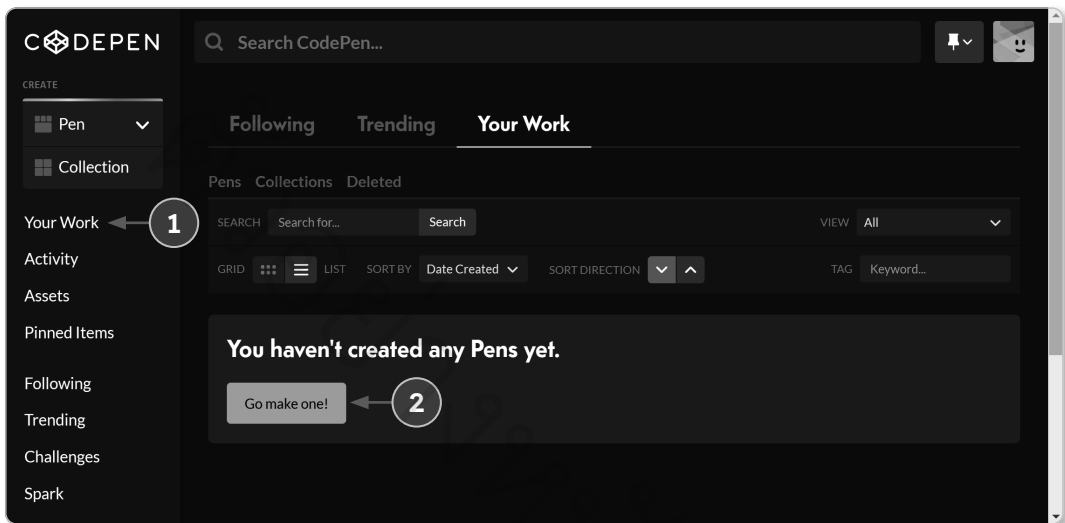
Note

แม้ว่า JavaScript เป็นภาษาคอมพิวเตอร์ที่ถูกออกแบบมาใช้กับเบราว์เซอร์เป็นหลัก แต่ JavaScript ก็ยังสามารถใช้กับสภาพแวดล้อมอื่น ๆ ที่ไม่ใช่เบราว์เซอร์ได้อีกด้วย เช่น ใช้ JavaScript กับ Node.js และ Apache CouchDB เป็นต้น

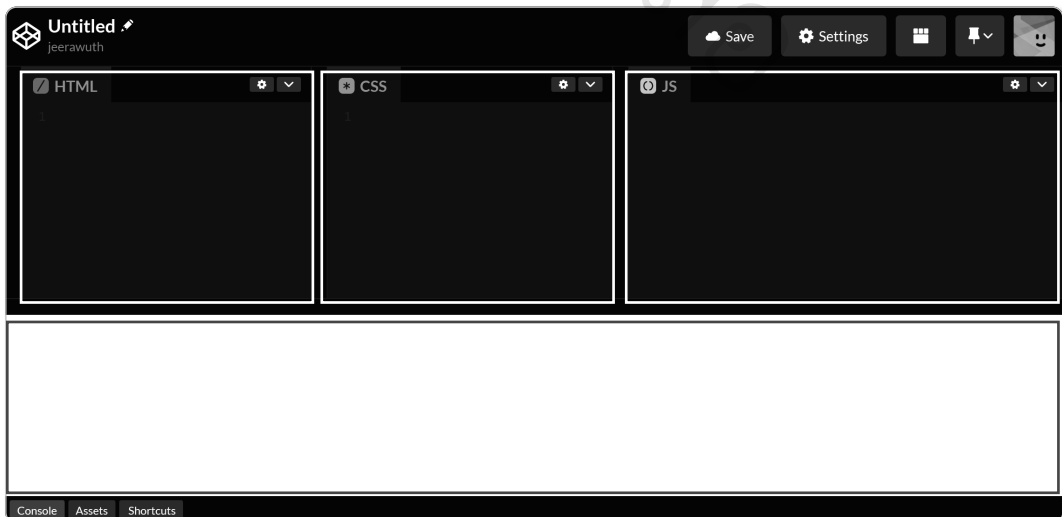
โค้ดแรกกับ JavaScript

คำสั่งแรกของ JavaScript ที่ต้องทราบ คือ คำสั่ง `console.log()` ใช้เพื่อพิมพ์ข้อความออกมาที่หน้าจอ เช่น เมื่อต้องการพิมพ์ข้อความ “Hello JavaScript” ก็ให้ใช้คำสั่ง `console.log(“Hello JavaScript”)` เป็นต้น

1. เปิดเบราว์เซอร์และไปยัง <https://codepen.io> ให้ล็อกอินเข้าสู่ระบบให้เรียบร้อย คลิกเมนู Your Work เพื่อแสดงพื้นที่เก็บงานที่เคยได้ทำเอาไว้ (ในการใช้งานครั้งแรกจะเป็นพื้นที่ว่าง ๆ ยังไม่มีข้อมูลอะไรอยู่)
2. คลิกปุ่ม Go make one! เพื่อเริ่มลงมือเขียนโค้ด หรือถ้าเคยเขียนโค้ดกับ CodePen มาก่อน ก็สามารถแก้ไขโค้ดเดิมที่ได้ทำไว้ก็ได้



3. ระบบจะกำหนดหน้าต่างขึ้นมาจำนวน 4 หน้าต่าง ด้านบนเรียงจากซ้ายไปขวา คือ หน้าต่าง HTML, CSS และ JS ส่วนด้านล่าง คือ หน้าต่างที่ใช้แสดงผลลัพธ์

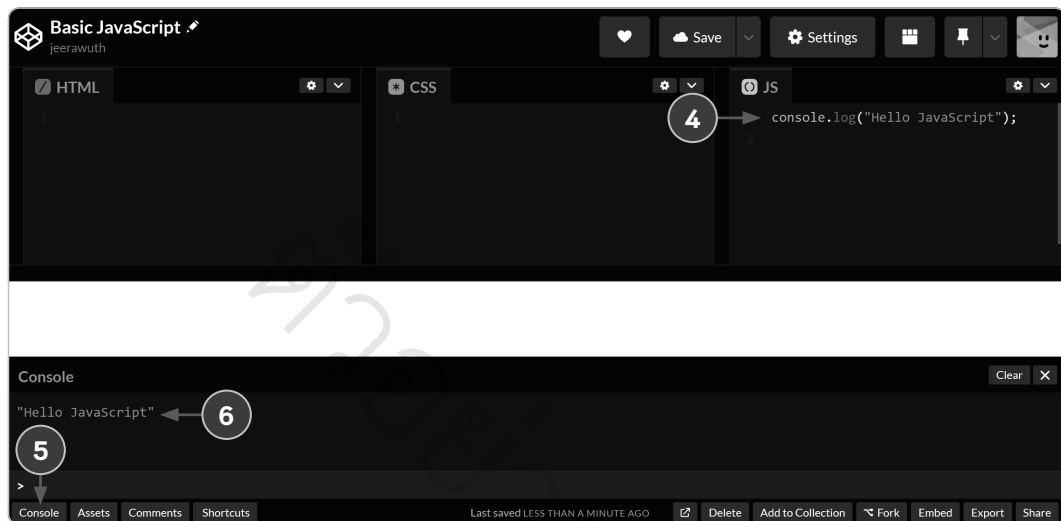


4. ที่หน้าต่าง JS ให้กรอกโค้ดภาษา JavaScript ลงไปดังนี้

```
console.log("Hello JavaScript");
```

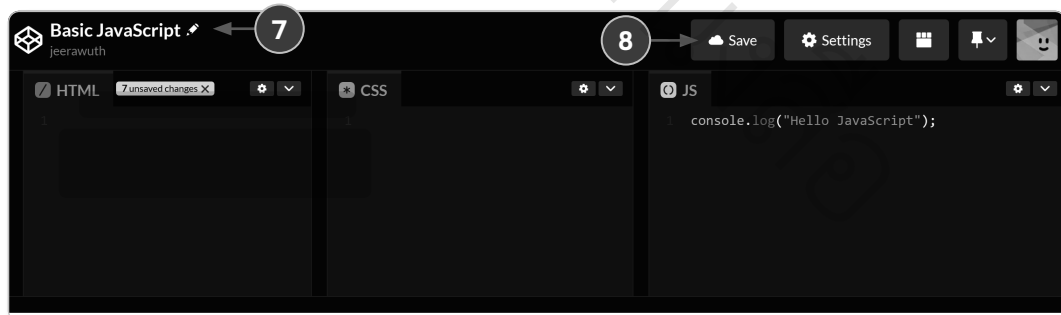
5. คลิกปุ่ม Console เพื่อแสดงหน้าต่างคอนโซล

6. ผลลัพธ์คำว่า Hello JavaScript จะถูกแสดงที่หน้าต่างคอนโซล ซึ่งเกิดจากคำสั่งที่ได้พิมพ์ไว้ในขั้นตอนที่ 4



7. สามารถตั้งชื่อได้ โดยคลิกปุ่ม รูปดินสอ และแก้ไขชื่อตามต้องการ เช่น Basic JavaScript

8. คลิกปุ่ม Save เพื่อบันทึกโค้ด



Note

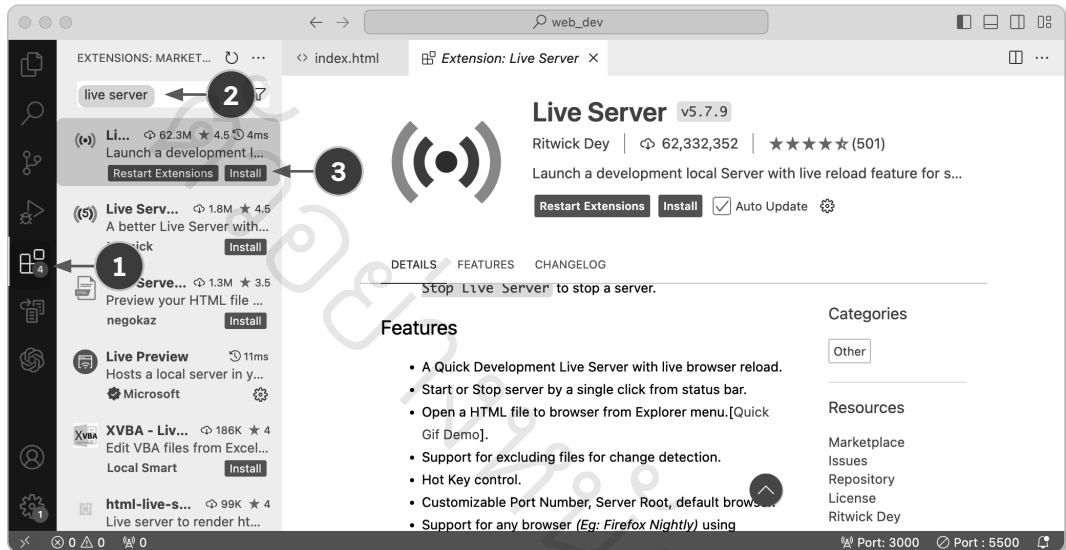
โดยปกติเมื่อบันทึกโค้ดครั้งแรกไปแล้ว ถ้ามีการแก้ไขโค้ด CodePen ก็จะบันทึกให้อัตโนมัติ ซึ่งผู้ใช้สามารถยกเลิกการบันทึกอัตโนมัตินี้ได้

ทดสอบ JavaScript ด้วย Visual Studio Code

การทดสอบ JavaScript ใน Visual Studio Code สามารถทำได้หลายวิธี ในหนังสือเล่มนี้จะทดสอบโดยการจำลองการทำงานของเซิร์ฟเวอร์ด้วย Live Server

ขั้นตอนที่ 1 ติดตั้งส่วนขยายชื่อว่า Live Server ไปยัง Visual Studio Code เสียก่อน ดังนี้

1. คลิกปุ่ม  เพื่อแสดงรายชื่อส่วนขยายทั้งหมดที่ได้ติดตั้งไว้กับ Visual Studio Code
2. กรอกคำค้นหาเป็น live server แล้วคลิกเลือก Live Server ที่พัฒนาโดย Ritwick Dey
3. คลิกปุ่ม install เพื่อติดตั้ง Live Server



ขั้นตอนที่ 2 สร้างไฟล์ HTML และ JavaScript เช่น ไฟล์ index.html และไฟล์ script.js

1. ไฟล์ index.html เป็นไฟล์ HTML ที่ใช้เป็นโครงสร้างของเว็บเพจ เมื่อต้องการใช้ JavaScript กับเว็บเพจปัจจุบัน ให้กำหนดแท็ก `<script src="script.js"></script>` เพื่อนำโค้ด JavaScript ที่อยู่ในไฟล์ script.js มาใช้งาน
2. ไฟล์ script.js เป็นไฟล์ JavaScript (นามสกุล .js) ให้กรอกคำสั่ง JavaScript ที่ต้องการลงไป เช่น console.log("Hello World!!!");

ตัวอย่าง 19.1

แสดงข้อความ Hello World!!!

ตัวอย่างการใช้คำสั่ง JavaScript แสดงข้อความ Hello World !!! ที่หน้าต่างคอนโซล

1. เปิด Visual Studio Code แก้ไขไฟล์ index.html ที่เป็นหน้าโฮมเพจ

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
```

```

5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>JavaScript</title>
7 </head>
8 <body>
9     <h1>Home Page</h1>
10    <script src="script.js"></script>
11 </body>
12 </html>

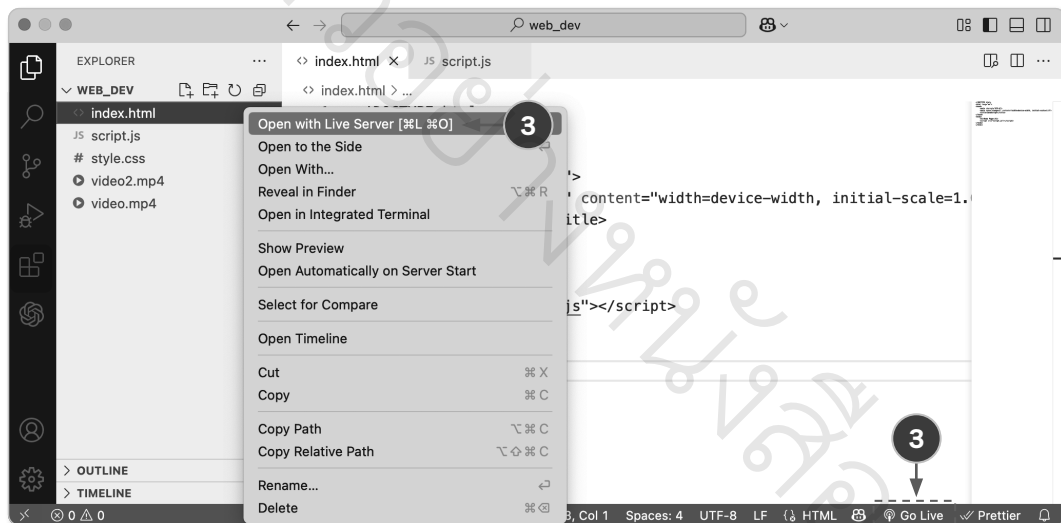
```

- บรรทัดที่ 10 กำหนดแท็ก <script src="script.js"></script> เพื่อใช้โค้ด JavaScript จากไฟล์ script.js

- แก้ไขไฟล์ script.js โดยเขียนคำสั่ง JavaScript ลงไป

```
console.log("Hello World!!!");
```

- ตรวจสอบผลลัพธ์ด้วย Live Server โดยให้คลิกขวาที่ไฟล์ index.html จากนั้นเลือกคำสั่ง Open with Live Server หรือคลิกปุ่ม Go Live ที่บริเวณมุมขวาล่าง เพื่อจำลองการทำงานของไฟล์ index.html บนเบราว์เซอร์



- เบราว์เซอร์ที่เป็นค่าเริ่มต้นจะถูกเรียกขึ้นมาอัตโนมัติ ให้สังเกตแอดเดรสที่เป็นหน้าโฮมเพจ เช่น <http://127.0.0.1:5500/index.html> สามารถใช้ Google Chrome เปิดดูผลลัพธ์ที่แอดเดรส <http://127.0.0.1:5500/index.html> ได้เช่นเดียวกัน
- ที่ Google Chrome ให้เปิดหน้าต่างคอนโซล (แมคโอเอส ให้กดปุ่ม <opt+cmd+j> วินโดวส์ ให้กดปุ่ม <ctrl+shift+j>)
- ที่หน้าต่างคอนโซล (console) จะพบข้อความ Hello World!!! ที่เกิดจากคำสั่ง JavaScript



ถ้าต้องการหยุดการทำงานของ Live Server ให้กลับไปยัง VS Code และคลิกปุ่มซึ่งแสดงหมายเลขพอร์ตของเซิร์ฟเวอร์ เช่น คลิกปุ่ม Port: 5500



Note

ความแตกต่างระหว่างการใช้ Live Server กับ Live Preview ใน VS Code

- Live Preview เหมาะสำหรับแก้ไข HTML/CSS อย่างง่าย ๆ ที่ไฟล์ สามารถดูผลลัพธ์การแก้ไขได้ภายใน VS Code ได้ทันที
- Live Server เป็นการจำลองการทำงานของเว็บเซิร์ฟเวอร์ สามารถใช้เบราว์เซอร์ตรวจสอบการทำงานของแอปพลิเคชันทั้งหมดได้ (รองรับ HTML, CSS และ JavaScript)

กรอกคำสั่ง JavaScript ในแท็ก <script></script>

สามารถเขียนคำสั่ง JavaScript ในไฟล์ HTML แทนการใช้งานโค้ด JavaScript จากภายนอกได้ โดยแทรกโค้ด JavaScript ภายในแท็ก <script></script> การเขียนโค้ด JavaScript วิธีนี้เหมาะสำหรับการเขียนโค้ด JavaScript ที่ต้องการให้มีผลเฉพาะหน้าเว็บเพจปัจจุบันเท่านั้น

ตัวอย่างการเขียนโค้ด JavaScript เพื่อแสดงข้อความ !!!Goodbye World ที่หน้าต่างคอนโซล โดยเราจะใช้วิธีแทรกโค้ด JavaScript ภายในแท็ก <script></script> (ไม่มีการอ้างอิงไฟล์ JavaScript จากภายนอก)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>JavaScript</title>
7 </head>
8 <body>
9   <h1>Home Page</h1>
10  <script>
11    console.log("!!!Goodbye World");
12  </script>
13 </body>
14 </html>
```

- บรรทัดที่ 10-12 กำหนดแท็ก <script></script> กรอบคำสั่ง JavaScript ทั้งหมด

ตัวแปร (Variable)

ตัวแปร (Variable) คือ ชื่อที่ใช้อ้างอิงไปยังพื้นที่ในหน่วยความจำ เพื่อใช้งานข้อมูลที่อยู่ในหน่วยความจำนั้น ๆ เช่น กำหนดชื่อ PI เพื่อใช้อ้างอิงค่าตัวเลข 3.14159 หรือกำหนด my_box เพื่ออ้างอิงไปยังตัวเลข 10 เป็นต้น

การกำหนดค่าให้กับตัวแปรจะเริ่มจากไคยเวิร์ด var จากนั้นให้กำหนดชื่อตัวแปรไว้ด้านซ้าย ตามด้วยเครื่องหมายเท่ากับ (=) และกำหนดค่าที่ต้องการกำหนดให้กับตัวแปรลงไป

```
var variable_name = data
```

- var คือ ไคยเวิร์ดสำหรับใช้ประกาศตัวแปร
- variable_name คือ ชื่อของตัวแปร
- data คือ ข้อมูลหรือออบเจกต์ที่ต้องการกำหนดให้กับตัวแปร

Note

การประกาศตัวแปรใน JavaScript สามารถใช้ไคยเวิร์ด var, let หรือ const ก็ได้ ผู้อ่านสามารถศึกษาความแตกต่างของการใช้ ไคยเวิร์ด var, let และ const ได้ในบทถัดไป

พื้นฐานการใช้งาน React

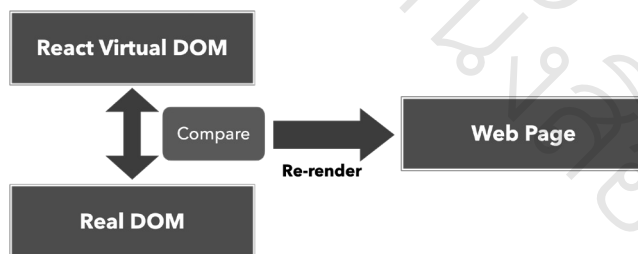
React เป็น JavaScript เฟรมเวิร์ค ใช้สำหรับสร้างเว็บแอปพลิเคชันในฝั่งไคลเอนต์ React สามารถสร้างส่วนติดต่อกับผู้ใช้ (UI) โดยที่ไม่ต้องเขียน JavaScript เพื่อแก้ไขเอลิเมนต์ต่าง ๆ บนเว็บเพจด้วยตนเองทั้งหมด นักพัฒนาเพียงแค่เตรียมข้อมูล และจัดโครงสร้างของเว็บเพจเอาไว้ เมื่อข้อมูลเปลี่ยนแปลง React ก็จะทำหน้าที่อัปเดตเนื้อหาบนเว็บเพจให้โดยอัตโนมัติ

React คืออะไร

React คือ JavaScript โลบารี ที่ถูกพัฒนาขึ้นโดย Facebook โดยมีจุดประสงค์สำหรับสร้างส่วนติดต่อกับผู้ใช้ (User Interface หรือ UI) บนเว็บเพจหรือบนเว็บแอปพลิเคชัน

React จะแบ่งเว็บเพจออกเป็นส่วนย่อย เรียกว่า คอมโพเนนต์ (Components) นักพัฒนาสามารถนำคอมโพเนนต์ต่าง ๆ มาประกอบเป็นหน้าเว็บเพจ หากหน้าเพจมีส่วนที่มีเนื้อหาซ้ำก็สามารถนำคอมโพเนนต์เก่ามาใช้งานได้ ด้วยการแบ่งเนื้อหาออกเป็นส่วนย่อย จึงทำให้การพัฒนาแอปพลิเคชันทำได้ง่าย ซึ่งสะดวกกว่าการเขียนโค้ด JavaScript เพื่อจัดการกับเนื้อหาบนหน้าเว็บเพจด้วยตนเองทั้งหมด

React จะใช้วิธีเก็บโครงสร้างของเว็บเพจไว้ในหน่วยความจำ เรียกว่า Virtual DOM โดยจะนำไปเปรียบเทียบกับเอลิเมนต์ที่อยู่ใน DOM จริง ด้วยวิธีนี้ React จึงสามารถอัปเดตเฉพาะส่วนของเว็บเพจที่มีการเปลี่ยนแปลงไปจากเดิม โดยไม่จำเป็นต้องอัปเดตข้อมูลทั้งหน้าเว็บเพจ ทำให้การแสดงผลและการโต้ตอบระหว่างผู้ใช้ในระหว่างใช้งานแอปพลิเคชันสามารถทำได้อย่างรวดเร็วยิ่งขึ้น



ด้วยคุณสมบัติต่าง ๆ ของ React นักพัฒนาจึงนิยมนำ React มาใช้สร้างเว็บแอปพลิเคชันที่ใช้เพียงหน้าเว็บเพจเดียว (เรียกว่า Single Page Application : SPA) ที่สามารถแสดงผลและโต้ตอบกับผู้ใช้โดยไม่ต้องโหลดหน้าเว็บใหม่ ทำให้แอปพลิเคชันทำงานได้รวดเร็วและมีประสิทธิภาพ

สร้างและรัน React โดยใช้ Vite

Vite หรือ Vitejs (มาจากคำภาษาฝรั่งเศส อ่านว่า วีต) เป็นเครื่องมืออีกตัวหนึ่งที่สามารถนำมาใช้สร้าง React โปรเจกต์ ข้อดีของการใช้ Vite คือ เรื่องของความเร็วและประสิทธิภาพที่เหนือกว่าการใช้เครื่องมือแบบเดิม (create-react-app) อยู่พอสมควร ดังนั้น ในปัจจุบันนักพัฒนาหลายท่านจึงเปลี่ยนไปใช้งาน Vite กันมากขึ้น

การสร้าง React โปรเจกต์โดยใช้ Vite นั้น จะต้องติดตั้ง Node.js ลงไปก่อน จากนั้นให้ทำตามขั้นตอน ดังนี้

1. เปิดหน้าต่าง Command Prompt (ในวินโดวส์) หรือหน้าต่าง Terminal (ในแมคโอเอส)
2. พิมพ์คำสั่ง `npm create vite@latest first-app` เพื่อสร้างโปรเจกต์โดยเก็บไว้ในโฟลเดอร์ `first-app` (สามารถใช้โฟลเดอร์อื่นแทน `first-app` ได้)

```
react_dev -- zsh -- 92x15
thinkbeyond@MacBook-Pro-khxng-ThinkBeyond react_dev % npm create vite@latest first-app
```

3. เลือกเฟรมเวิร์คเป็น React และกดปุ่ม <Enter>

```
react_dev -- node - npm create vite@latest first-app __CFBundleIdentifier=com.apple.Terminal TMPDIR=/var/folders/g5/54_m6d15285_12hx56tc...
Select a framework:
  o Vanilla
  o Vue
  o React
  o Preact
  o Lit
  o Svelte
  o Solid
  o Qwik
```

4. เลือกภาษาที่ใช้พัฒนาเป็น JavaScript และกดปุ่ม <Enter>

```
react_dev -- node - npm create vite@latest first-app __CFBundleIdentifier=com.apple.Terminal TMPDIR=/var/folders/g5/54_m6d15285_12hx56tc...
Select a framework:
  React
Select a variant:
  o TypeScript
  o TypeScript + SWC
  o JavaScript
  o JavaScript + SWC
  o React Router v7
```

5. เข้าไปยังโฟลเดอร์ที่เก็บโปรเจกต์ เช่น ถ้าโฟลเดอร์โปรเจกต์คือ `first-app` ให้พิมพ์ `cd first-app` แล้วกดปุ่ม <Enter>
6. พิมพ์คำสั่ง `npm install` แล้วกดปุ่ม <Enter> เพื่อติดตั้ง dependencies (แพ็คเกจที่จำเป็นต้องใช้งาน)
7. ทดลองรันแอปพลิเคชัน โดยใช้คำสั่ง `npm run dev`

```
first-app -- zsh -- 92x15
cd first-app
npm install
npm run dev
thinkbeyond@MacBook-Pro-khxng-ThinkBeyond react_dev % cd first-app
thinkbeyond@MacBook-Pro-khxng-ThinkBeyond first-app % npm install
added 150 packages, and audited 151 packages in 28s
30 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
thinkbeyond@MacBook-Pro-khxng-ThinkBeyond first-app % npm run dev
```

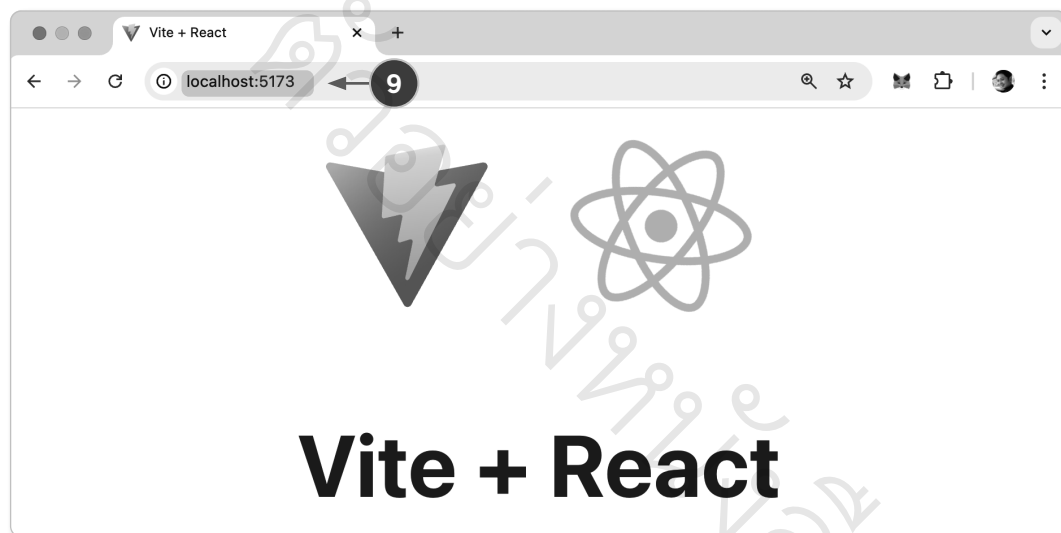
8. จะปรากฏข้อความบอกว่า Vite พร้อมใช้งานแล้ว โดยสามารถดูผลลัพธ์ได้จาก URL และพอร์ตที่กำหนด เช่น `http://localhost:5173/`

```
first-app — esbuild • npm run dev __CFBundleIdentifier=com.apple.Terminal TMPDIR=/var/folders/gs/54_m6d15285_12hx56tc_l280000gn/T/XP...

VITE v6.2.6 ready in 299 ms

→ Local: http://localhost:5173/ ← 8
→ Network: use --host to expose
→ press h + enter to show help
```

9. ใช้เบราว์เซอร์ไปยังแอดเดรส `http://localhost:5173/` จะพบ React แอปพลิเคชันที่สร้างจาก Vite (ให้สังเกตว่า Vite จะใช้หมายเลขพอร์ตที่เป็นค่าเริ่มต้น คือ 5173)



Tip

สามารถสร้าง React โปรเจกต์อย่างรวดเร็วโดยใช้คำสั่ง `npm create vite@latest your-app -- --template react` เพื่อใช้ Vite สร้างโปรเจกต์ไปยังโฟลเดอร์ `your-app` และใช้เทมเพลตสำหรับใช้งานกับ React

```
npm create vite@latest your-app -- --template react
```

NOTE

ก่อนการสร้าง React โปรเจกต์ร่วมกับ Vite จะต้องติดตั้ง Node.js ลงไปในเครื่องคอมพิวเตอร์ให้เรียบร้อย หลังจากติดตั้งจะได้ JavaScript Runtime และเครื่องมือที่จำเป็นสำหรับใช้งานร่วมกับ React และ Vite เช่น เครื่องมือสำหรับจัดการกับแพ็คเกจที่เรียกว่า npm (Node Package Manager) ผู้อ่านสามารถดาวน์โหลด Node.js ได้จากเว็บ <https://nodejs.org>

หยุดการทำงานของเว็บเซิร์ฟเวอร์

การใช้คำสั่ง `npm run dev` เพื่อรัน React แอปพลิเคชัน จะมีการกำหนดสภาพแวดล้อมเพื่อจำลองการทำงานของเว็บเซิร์ฟเวอร์ ทำให้นักพัฒนาสามารถทดสอบ React จากคอมพิวเตอร์ส่วนตัวได้ โดยไม่จำเป็นต้องไปฝากไว้กับเว็บเซิร์ฟเวอร์จริงที่อยู่ในอินเทอร์เน็ต

เมื่อใดที่ต้องการยกเลิกการทดสอบแอปพลิเคชัน ก็ให้ปิดระบบจำลองเว็บเซิร์ฟเวอร์ ด้วยการกดปุ่ม `<Ctrl+C>`

1. ไปยังหน้าต่าง Terminal (หรือ Command Prompt) ที่ได้สั่งรันแอปพลิเคชันเอาไว้ (หน้าต่างที่ใช้คำสั่ง `npm run dev`)
2. กดปุ่ม `<Ctrl+C>` เพื่อสั่งปิดการทำงานของเซิร์ฟเวอร์

```

first-app — zsh — 92x15

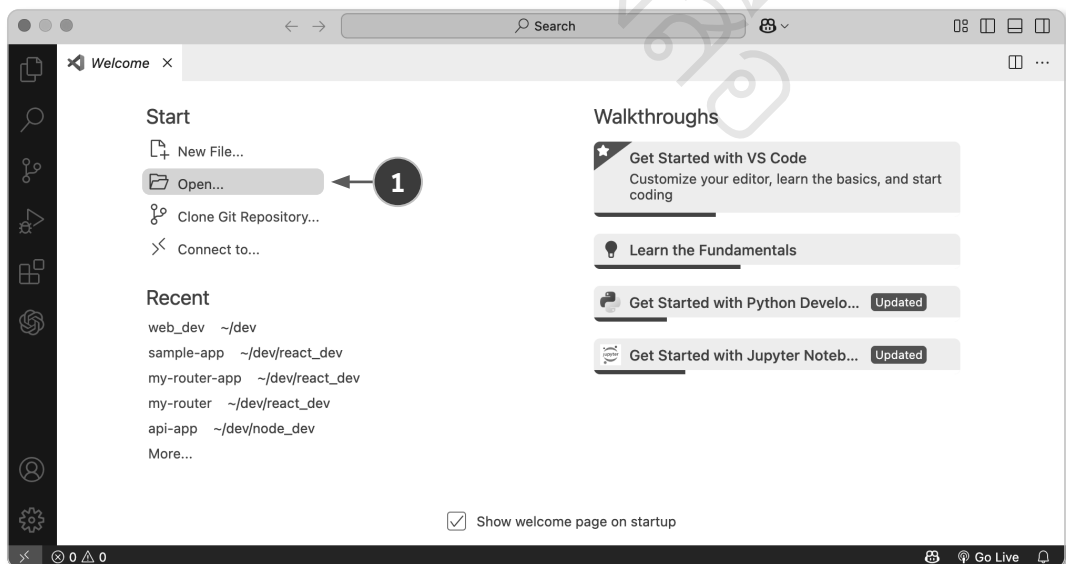
VITE v6.2.6 ready in 299 ms
  → Local:   http://localhost:5173/
  → Network: use --host to expose
  → press h + enter to show help
^C
thinkbeyond@MacBook-Pro-khxng-ThinkBeyond first-app %

```

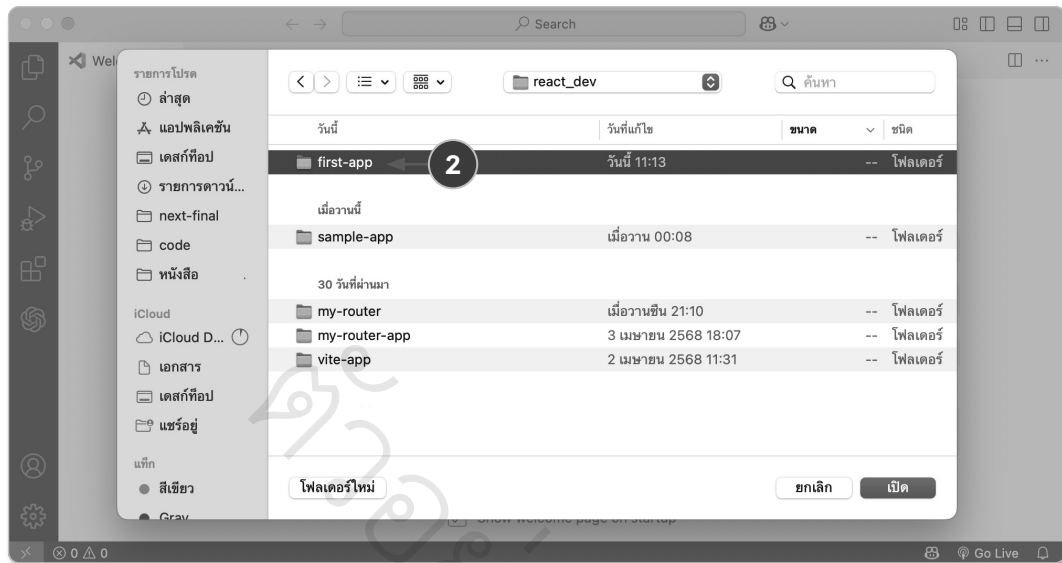
เปิดโปรเจกต์มาแก้ไข

เนื่องจาก React ก็คือ JavaScript เฟรมเวิร์ค ดังนั้น ขั้นตอนการสร้างหรือแก้ไขโค้ด จึงใช้รูปแบบเดียวกับการแก้ไขโค้ด JavaScript ทั่วไป ซึ่งในหนังสือเล่มนี้จะแก้ไขโค้ดโดยใช้ Visual Studio Code เป็นหลัก

1. เปิดโปรแกรม Visual Studio Code จากนั้นให้คลิกที่ Open...



2. คลิกเลือกโฟลเดอร์ที่เก็บ React แอปพลิเคชันเอาไว้ สำหรับในตัวอย่างนี้ ได้เลือกโฟลเดอร์ first-app



3. ที่โฟลเดอร์ src จะเก็บไฟล์ที่ใช้ควบคุมการทำงานของแอปพลิเคชัน

4. ที่ไฟล์ vite.config.js ใช้ตั้งค่าและปรับแต่งการทำงานของ Vite โปรเจกต์ เช่น การตั้งค่าปลั๊กอิน เป็นต้น



Tip

สามารถเปิดโปรเจกต์อย่างรวดเร็ว ด้วยวิธีพิมพ์คำสั่ง ดังนี้

1. เปิดหน้าต่าง Terminal (หรือ Command Prompt) เข้าไปยังโฟลเดอร์ที่เก็บแอปพลิเคชันเอาไว้ โดยพิมพ์คำสั่ง `cd` ตามด้วยชื่อโฟลเดอร์ เช่น พิมพ์ `cd first-app`
2. พิมพ์คำสั่ง `code .` เพื่อเปิดโฟลเดอร์ด้วย Visual Studio Code

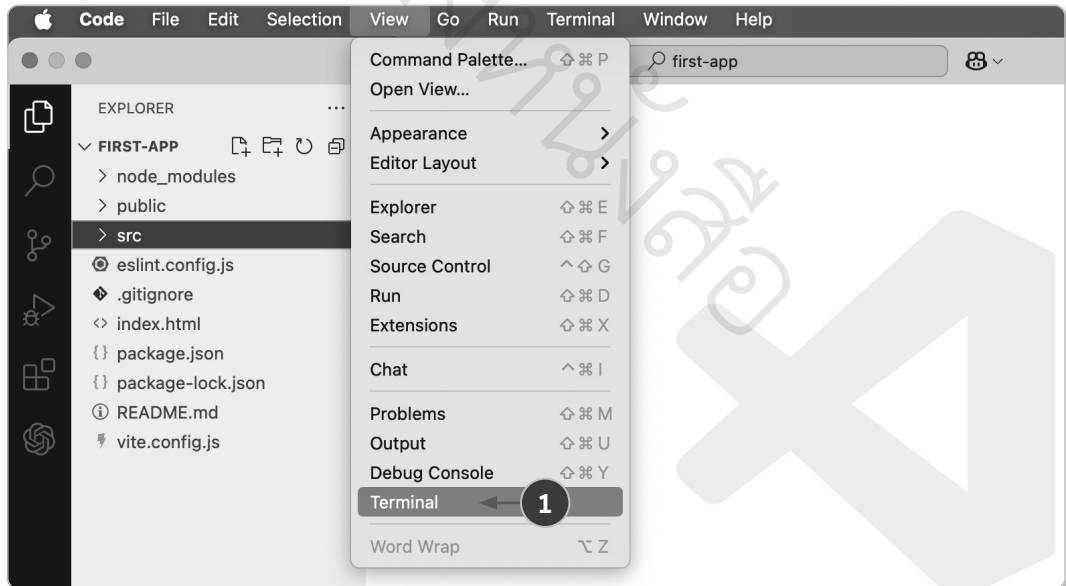
```

first-app — zsh — 85x13
thinkbeyond@MacBook-Pro-khxng-ThinkBeyond react_dev %
thinkbeyond@MacBook-Pro-khxng-ThinkBeyond react_dev % cd first-app ← 1
thinkbeyond@MacBook-Pro-khxng-ThinkBeyond first-app % code . ← 2
  
```

อัปเดตการแก้ไขแบบเรียลไทม์

ขณะที่ React แอปพลิเคชันกำลังรันอยู่ หากมีการแก้ไขโค้ดและบันทึกไฟล์ เบราวเซอร์จะอัปเดตเนื้อหาที่อยู่ในเพจทันทีโดยไม่ต้องสั่งรีสตาร์ทเซิร์ฟเวอร์ใหม่

1. ใช้ Visual Studio Code เปิดโปรเจกต์ที่ต้องการ จากนั้นแสดงหน้าต่าง Terminal โดยคลิกเมนู View > Terminal

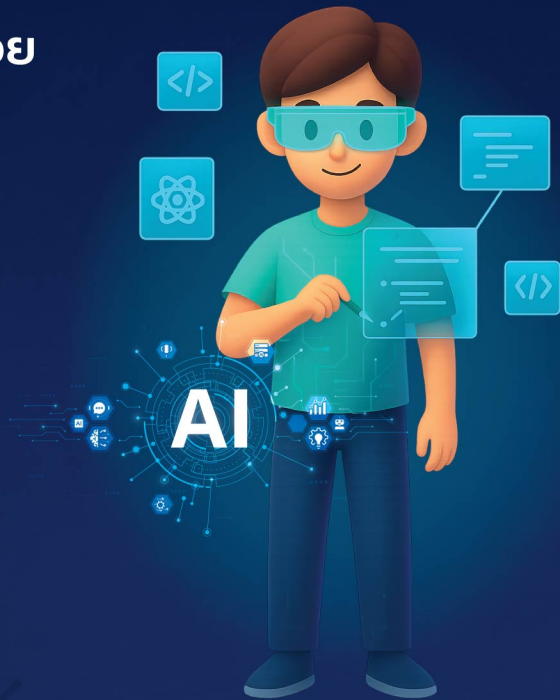


สร้างเว็บแบบไม่รู้อะไรเลย ด้วย **AI Coding**

ทุกคนสามารถ ใช้ AI เขียนโค้ดได้
แต่ไม่ใช่ทุกคนจะเข้าใจโค้ดที่เขียน

เรียนรู้การพัฒนาเว็บจากศูนย์ โดยใช้ AI
ช่วยเขียนโค้ดอย่างมืออาชีพ

อ่านและทำตามได้แบบ
Step-By-Step
เก่งได้ แม้ไม่มีพื้นฐานมาก่อน



- พื้นฐาน HTML + HTML5 ที่จำเป็นต้องทราบ
- พื้นฐาน CSS สำหรับตกแต่งสไตล์ให้กับเว็บเพจ
- พื้นฐาน JavaScript สำหรับการสร้างและพัฒนาเว็บ
- การใช้งาน Tailwind CSS ตกแต่งเว็บให้เว็บสวยโดนใจ
- แนะนำหลักการของ DOM และการใช้ JavaScript จัดการ DOM
- การสร้างแอปพลิเคชันในฝั่ง Frontend ด้วย React
- การใช้ React Router กำหนดระบบเราท์ติ้ง

- วิธีสร้างแอปพลิเคชันสำหรับติดต่อกับ API
- เรียนรู้การใช้งาน AI สำหรับโปรแกรมเมอร์
- เทคนิคการใช้ Prompt เพื่อให้ได้ผลลัพธ์ตามที่ต้องการ
- แนะนำ AI Code Editor เช่น Lovable AI, Cursor AI, Windsurf AI และ GitHub Copilot
- ตัวอย่างการใช้งาน GitHub Copilot เขียนโค้ด และแก้ไขโค้ด
- ตัวอย่างการใช้ AI สร้าง API เพื่อแสดงข้อมูลรายการสินค้า
- ตัวอย่างการสร้าง React แอปพลิเคชันติดต่อกับ API



ซื้อสะดวก ส่งถึงบ้านที่ Shopee และ Lazada หรือผ่านทาง
ร้านหนังสือออนไลน์ www.thinkbeyondbook.com



thinkbeyond books



9 786164 496019

ราคา 495 บาท