



Visual Studio Code

สำหรับผู้เริ่มต้น

Visual Studio Code



- เป็นโปรแกรมประเภท **Code Editor** หรือเครื่องมือสำหรับใช้ในการเขียนโปรแกรมที่ได้รับความนิยมในปัจจุบัน เหมาะสำหรับนักพัฒนาทุกระดับ
- รองรับการทำงานบน Windows , macOS , Linux
- รองรับการเขียนโปรแกรมได้หลายภาษา
- ใช้งานได้ฟรี มีขนาดเล็กและทำงานได้อย่างรวดเร็ว
- มี **ส่วนเสริม (Extension)** ช่วยอำนวยความสะดวกในการเขียนโปรแกรม

ภาพรวมเนื้อหา



- พื้นฐานการใช้งาน Visual Studio Code
- Command Palette
- การจัดการโฟลเดอร์และไฟล์
- การจัดการ Editor (พื้นที่สำหรับใช้ในการเขียนโค้ด)

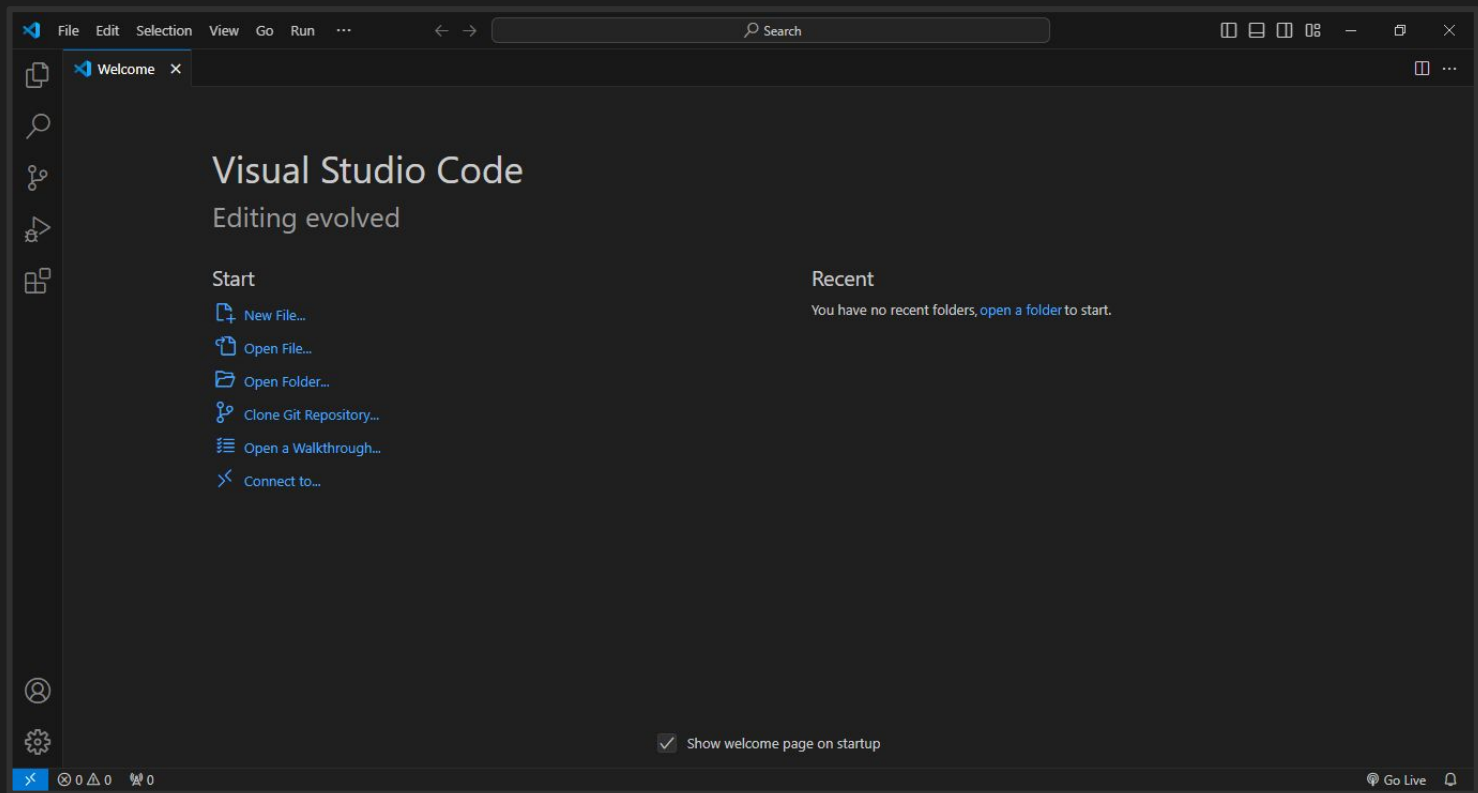
ภาพรวมเนื้อหา

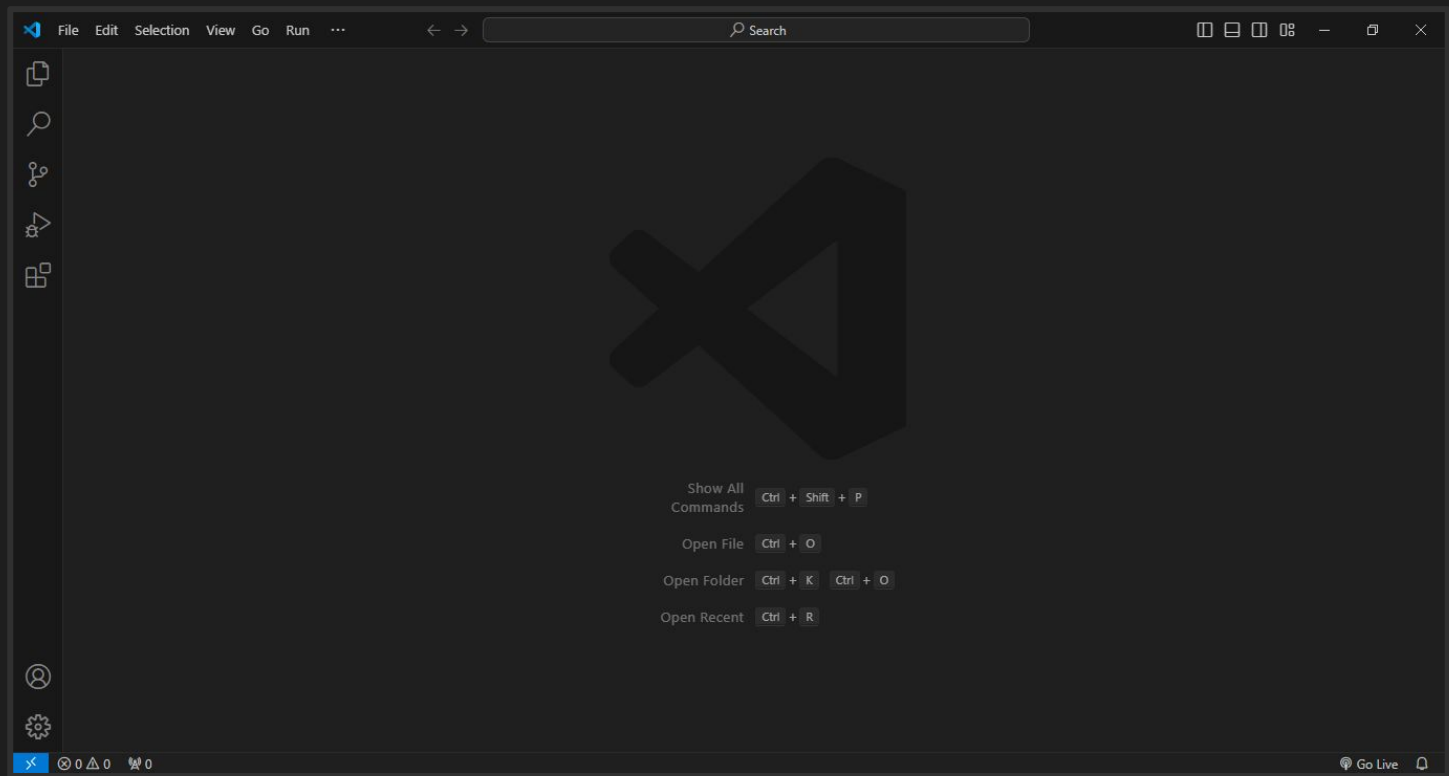


- Workspace
- Terminal
- Extension
- Profiles
- คีย์ลัดพื้นฐาน



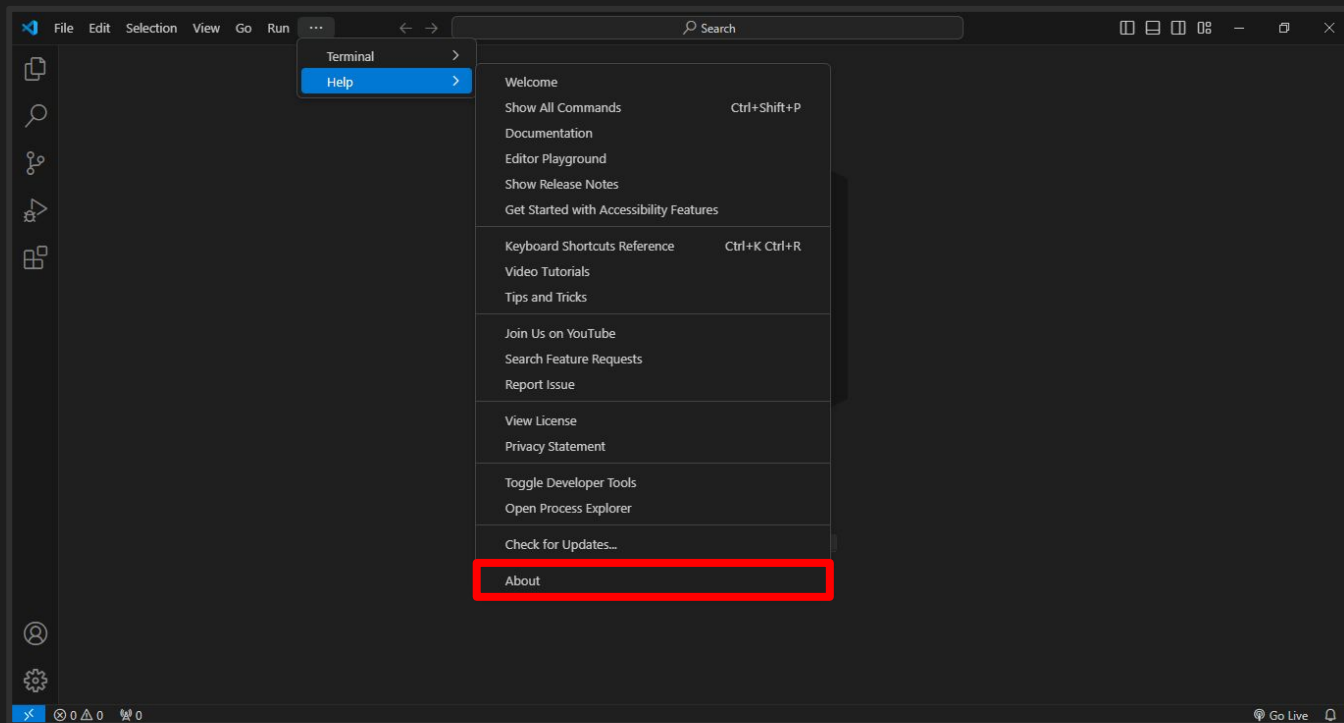
หน้าจอโปรแกรม (User Interface)



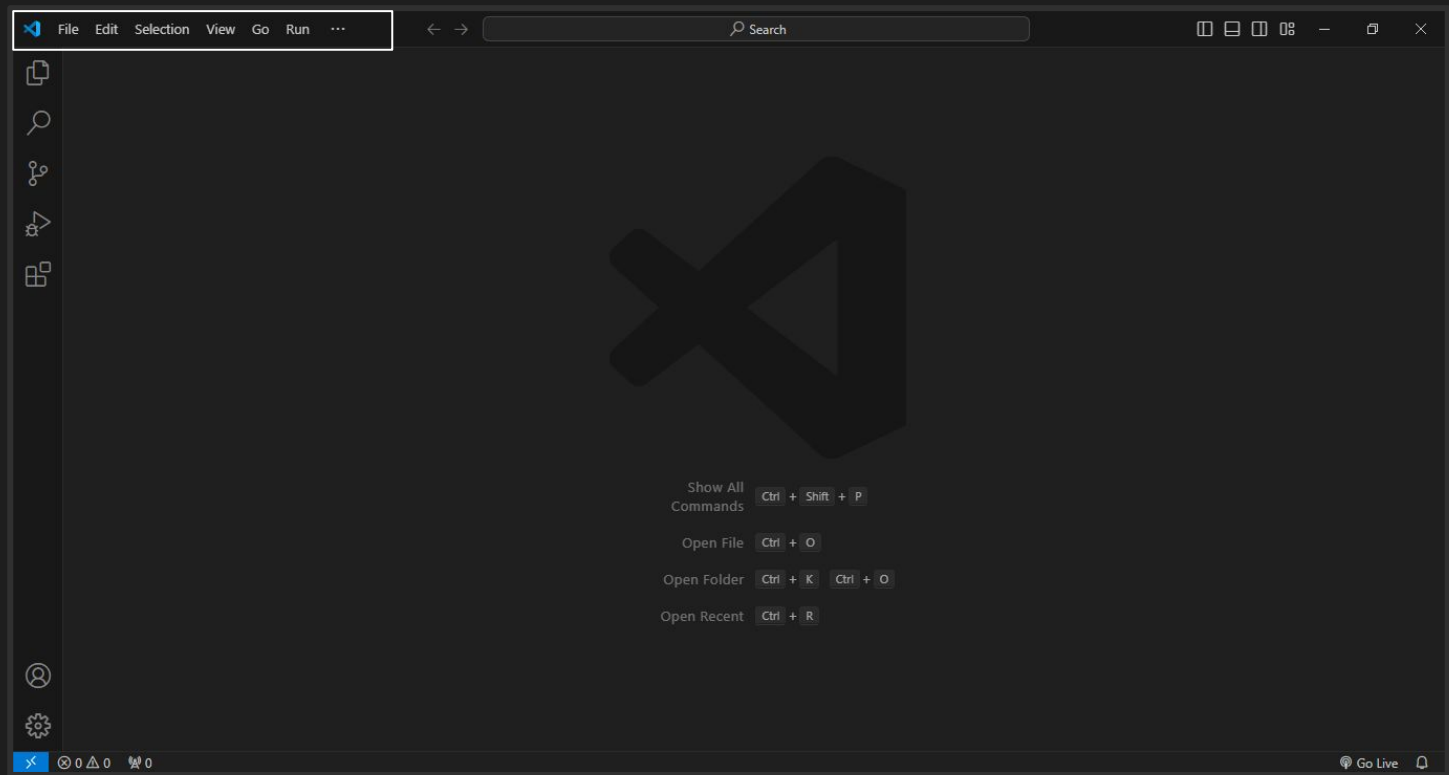


***หมายเหตุ :** หน้าจอโปรแกรมอาจมีการปรับเปลี่ยนตามเวอร์ชันที่ใช้งาน

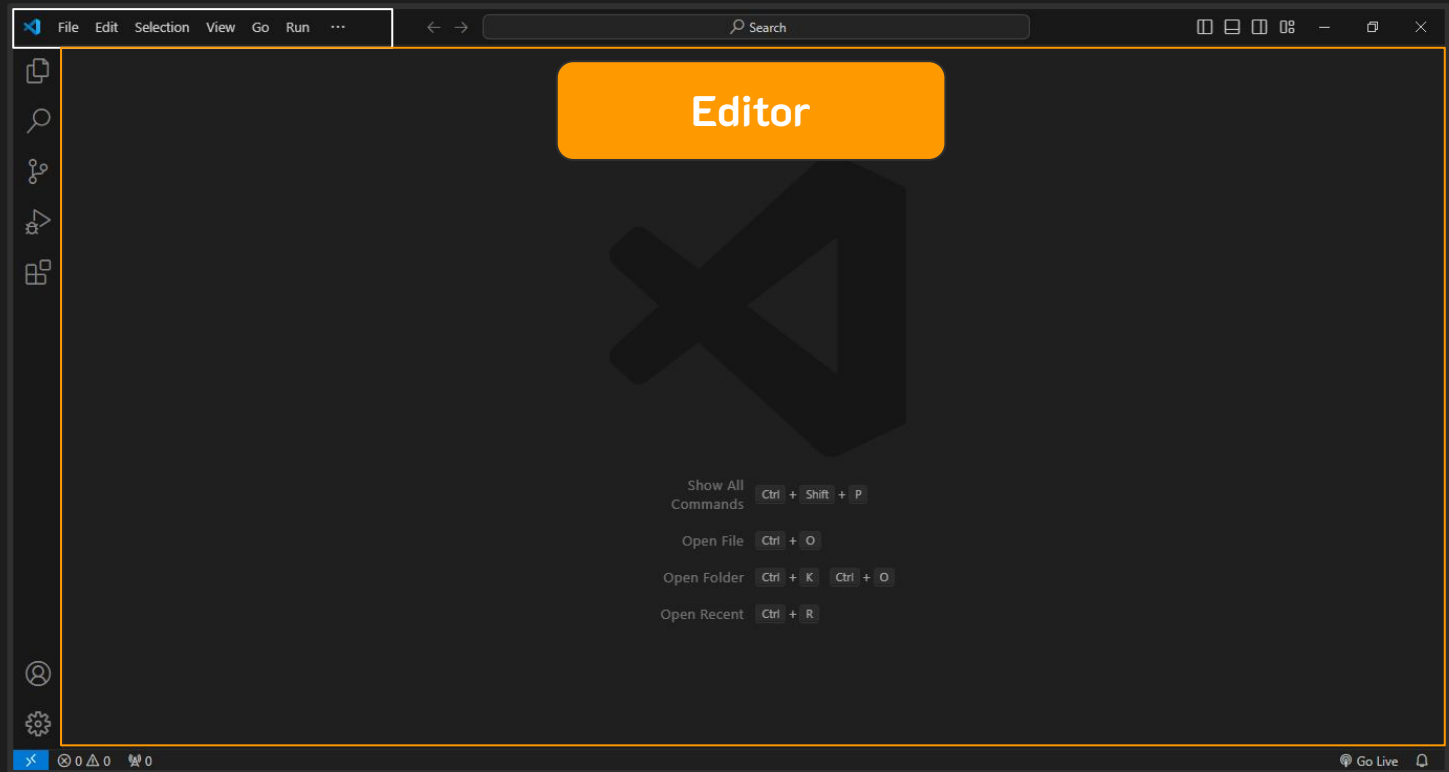
การตรวจสอบเวอร์ชัน



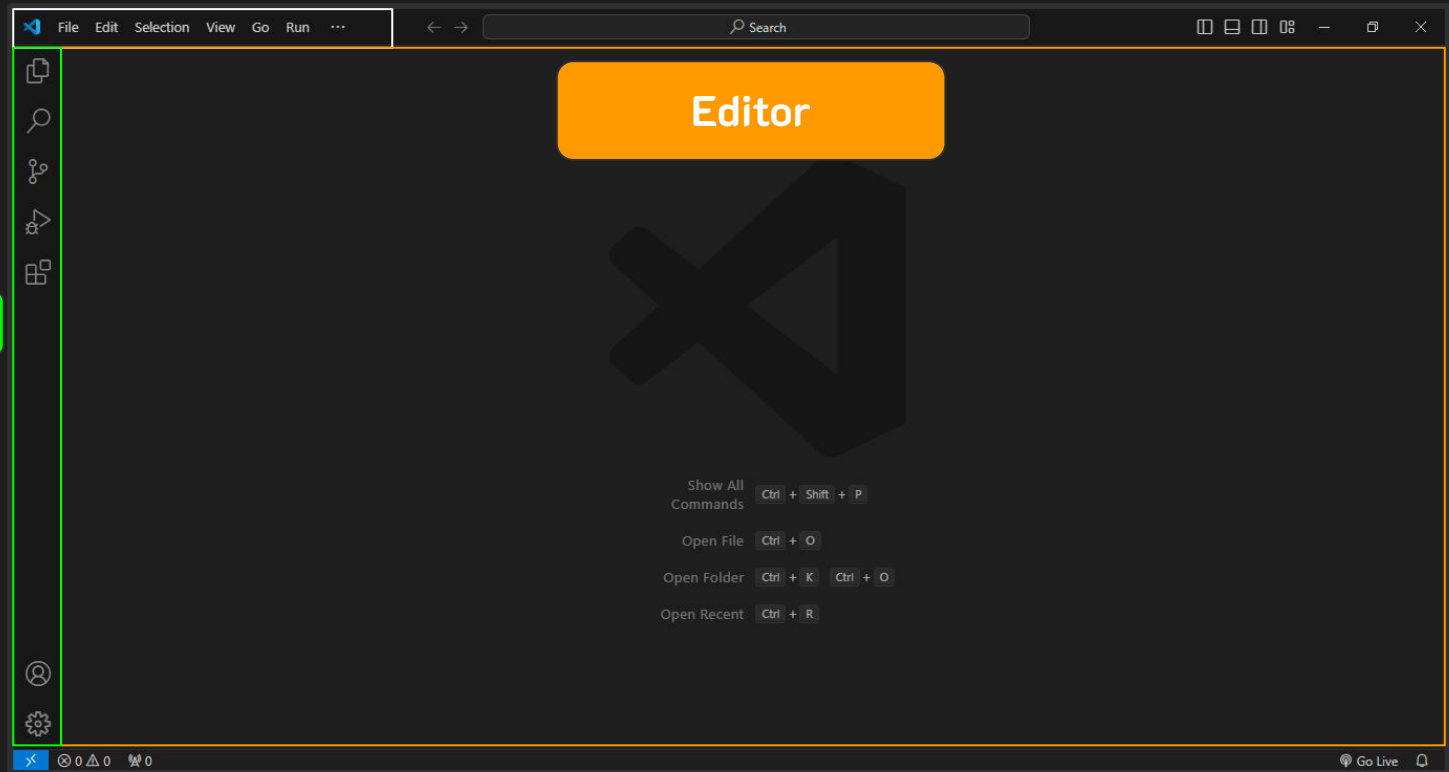
Menu Bar



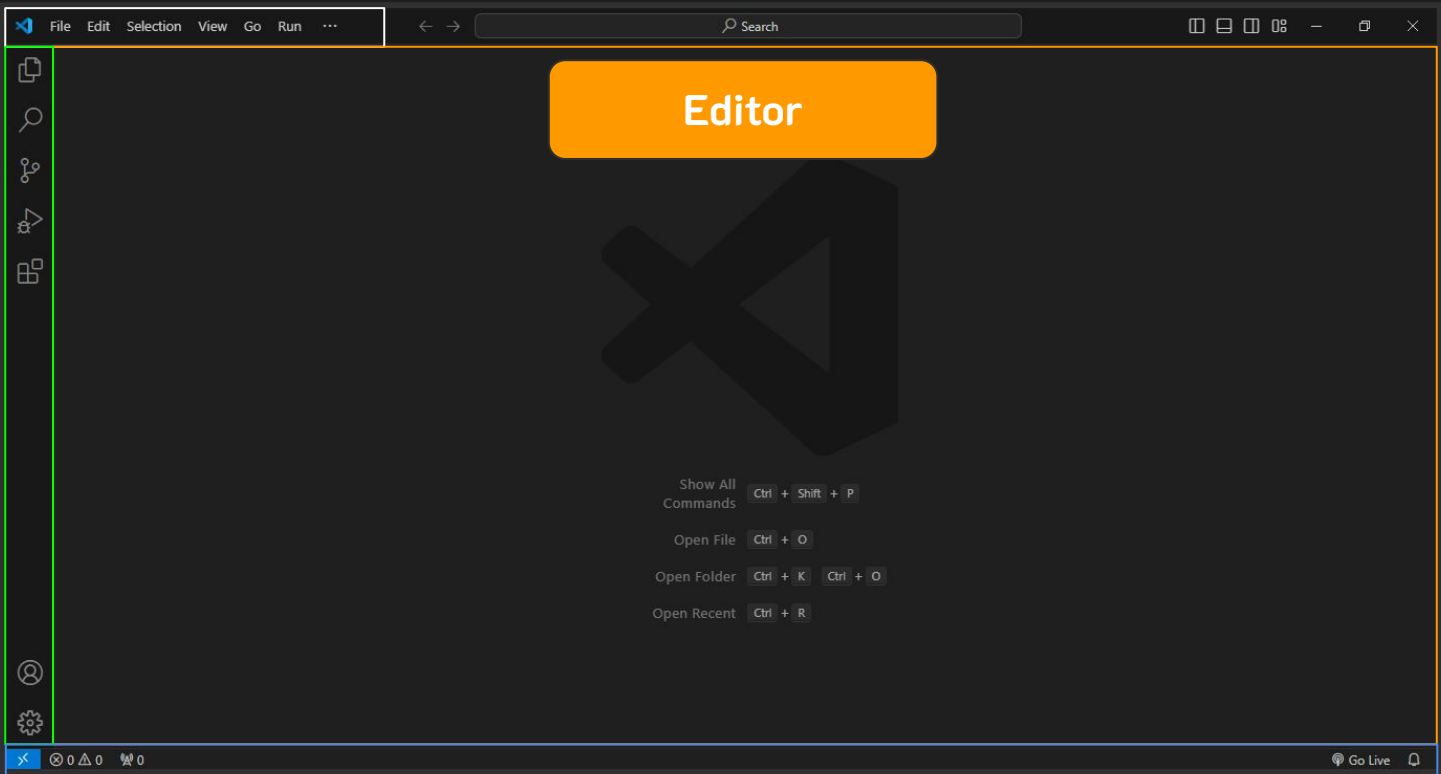
Menu Bar



Menu Bar



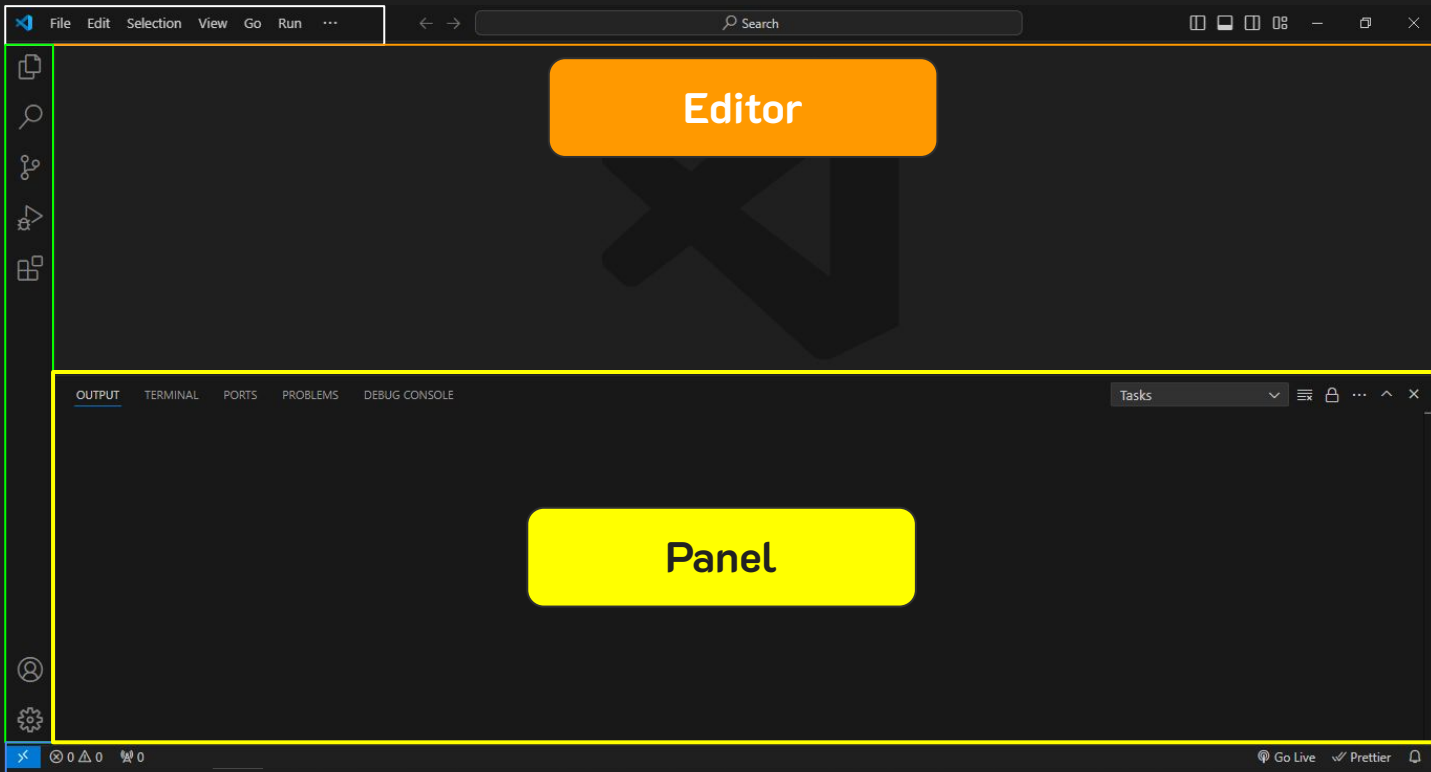
Menu Bar



Activity Bar

Status Bar

Menu Bar



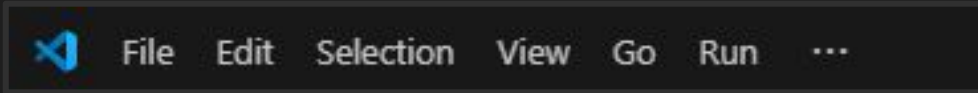
Activity Bar

Editor

Panel

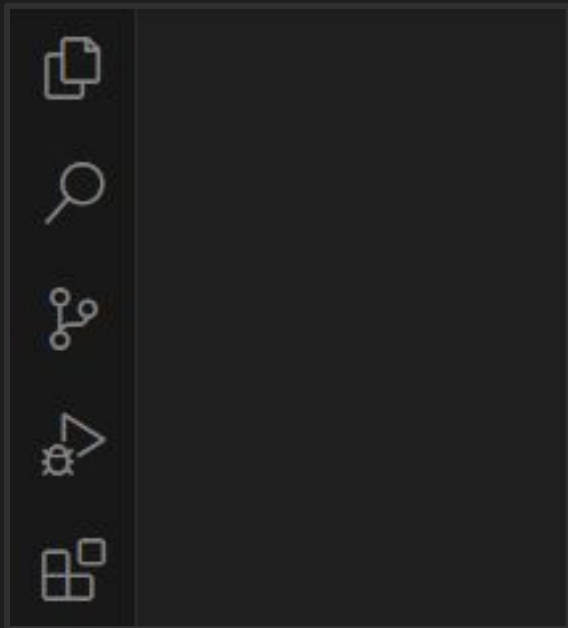
Status Bar

Menu Bar



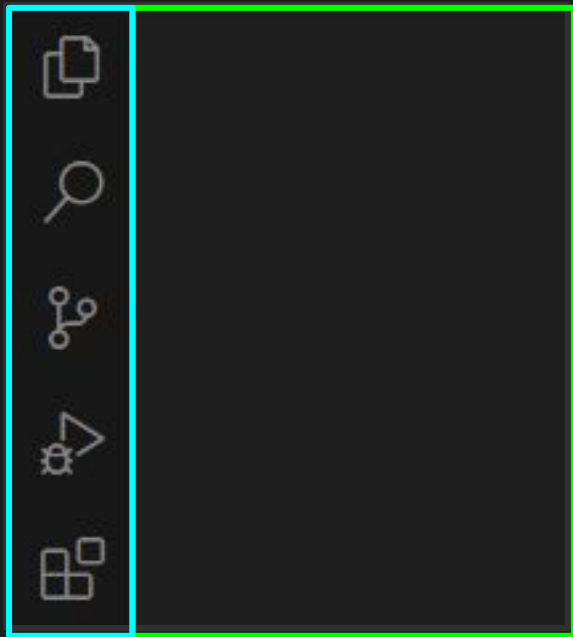
- Menu Bar เป็นแถบเมนูที่รวมคำสั่งทั้งหมดใน Visual Studio Code

Primary Sidebar



Primary Side Bar แถบเมนูที่อยู่ด้าน
ข้างโปรแกรม สำหรับแสดงรายละเอียดการ
ทำงานของเครื่องมือที่อยู่ใน **Activity Bar**

Primary Sidebar

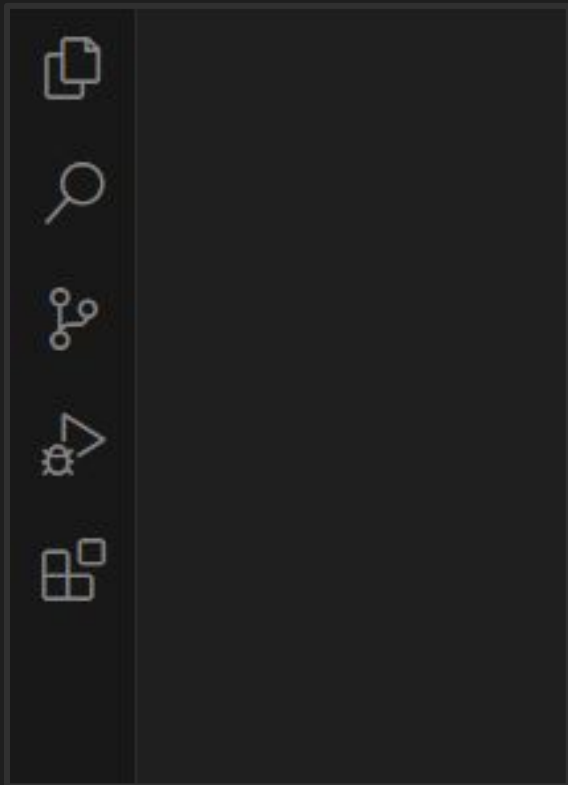


Activity Bar

Primary Side Bar

Primary Side Bar แถบเมนูที่อยู่ด้านข้างโปรแกรม สำหรับแสดงรายละเอียดการทำงานของเครื่องมือที่อยู่ใน **Activity Bar**

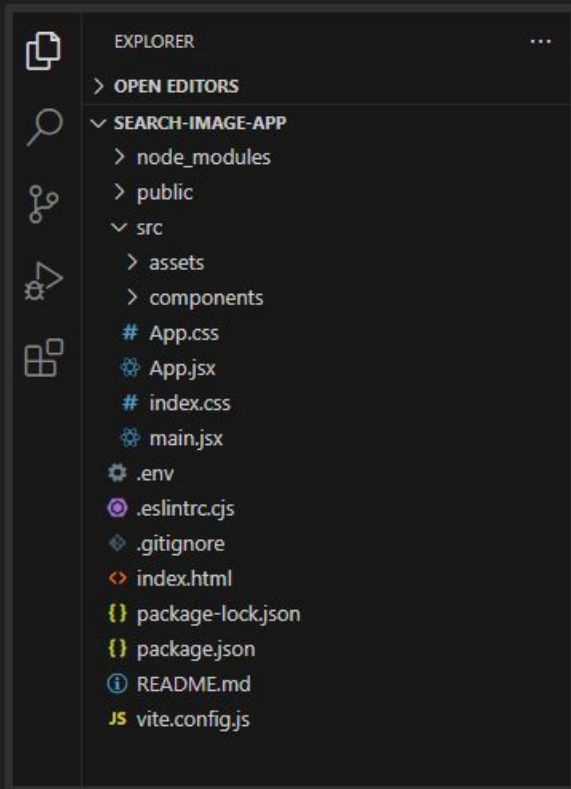
Activity Bar



Activity Bar แถบเครื่องมือประกอบด้วย
เครื่องมือ ดังนี้

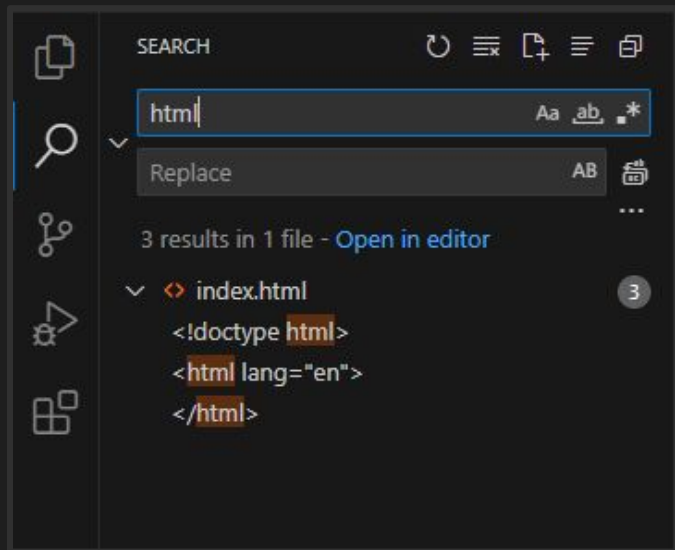
- Explorer
- Search
- Source Control
- Run and Debug
- Extensions

Activity Bar



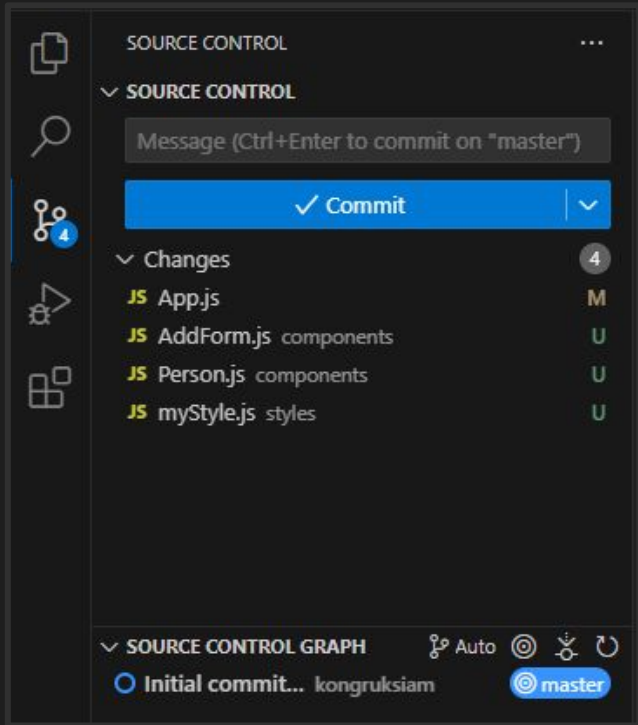
- **Explorer** ส่วนที่ใช้สำหรับแสดงรายชื่อ รวมถึงจัดการโฟลเดอร์และไฟล์ต่างๆ โดยจะแสดงโฟลเดอร์และไฟล์ทั้งหมด เป็นลำดับชั้นทำให้เข้าถึงข้อมูลในส่วนต่างๆ ได้ง่ายและรวดเร็ว

Activity Bar



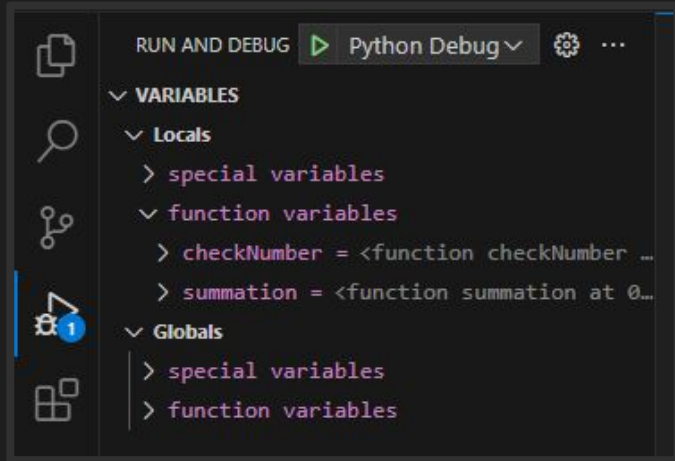
- **Search** ส่วนที่ใช้สำหรับค้นหาข้อมูลต่างๆที่กำลังเปิดใช้งานในโปรแกรม

Activity Bar



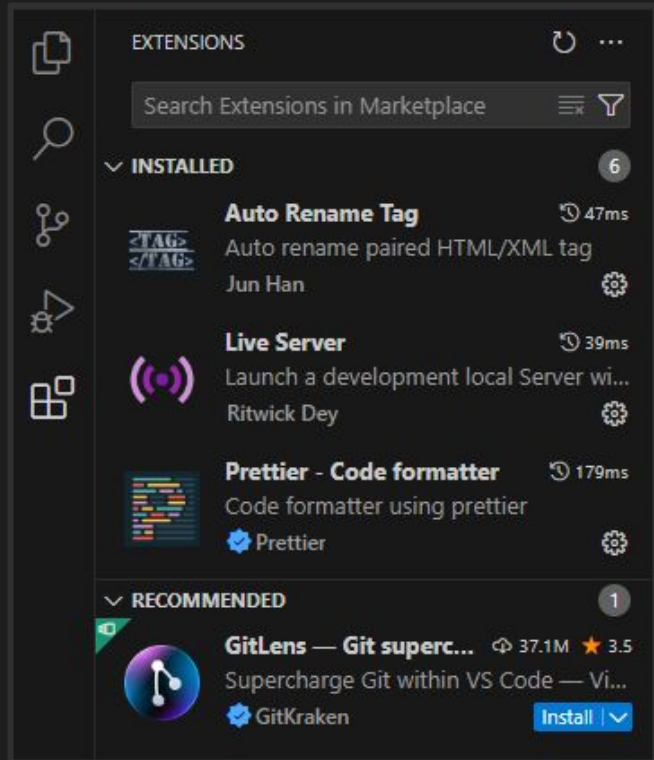
- **Source Control** ส่วนที่ใช้
ควบคุมการทำงานเกี่ยวกับ
Source Code ผ่าน Git ,
GitHub

Activity Bar

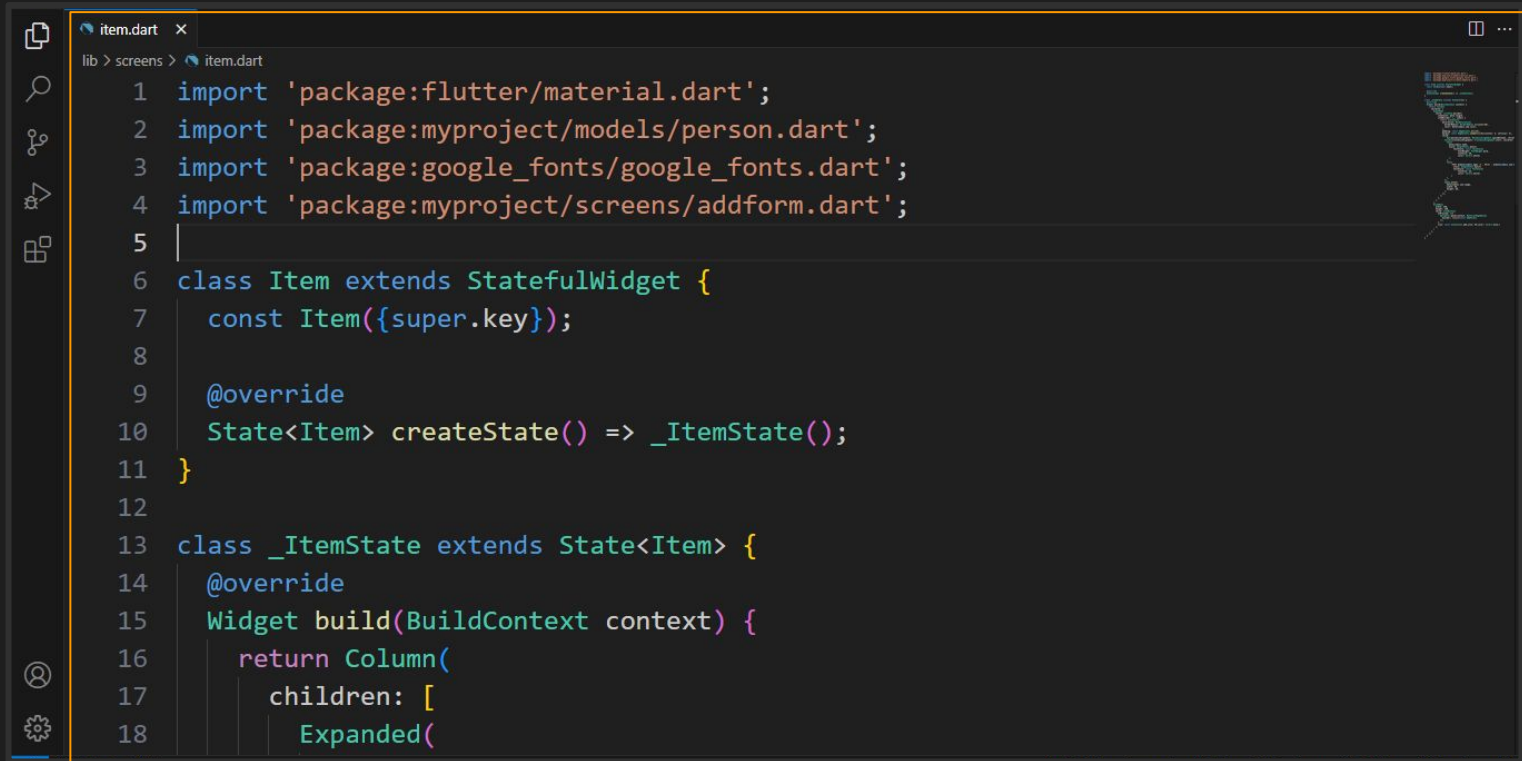


- **Run and Debug** ส่วนที่ใช้
สำหรับการรันและตรวจสอบ
การทำงานของโปรแกรม

Activity Bar

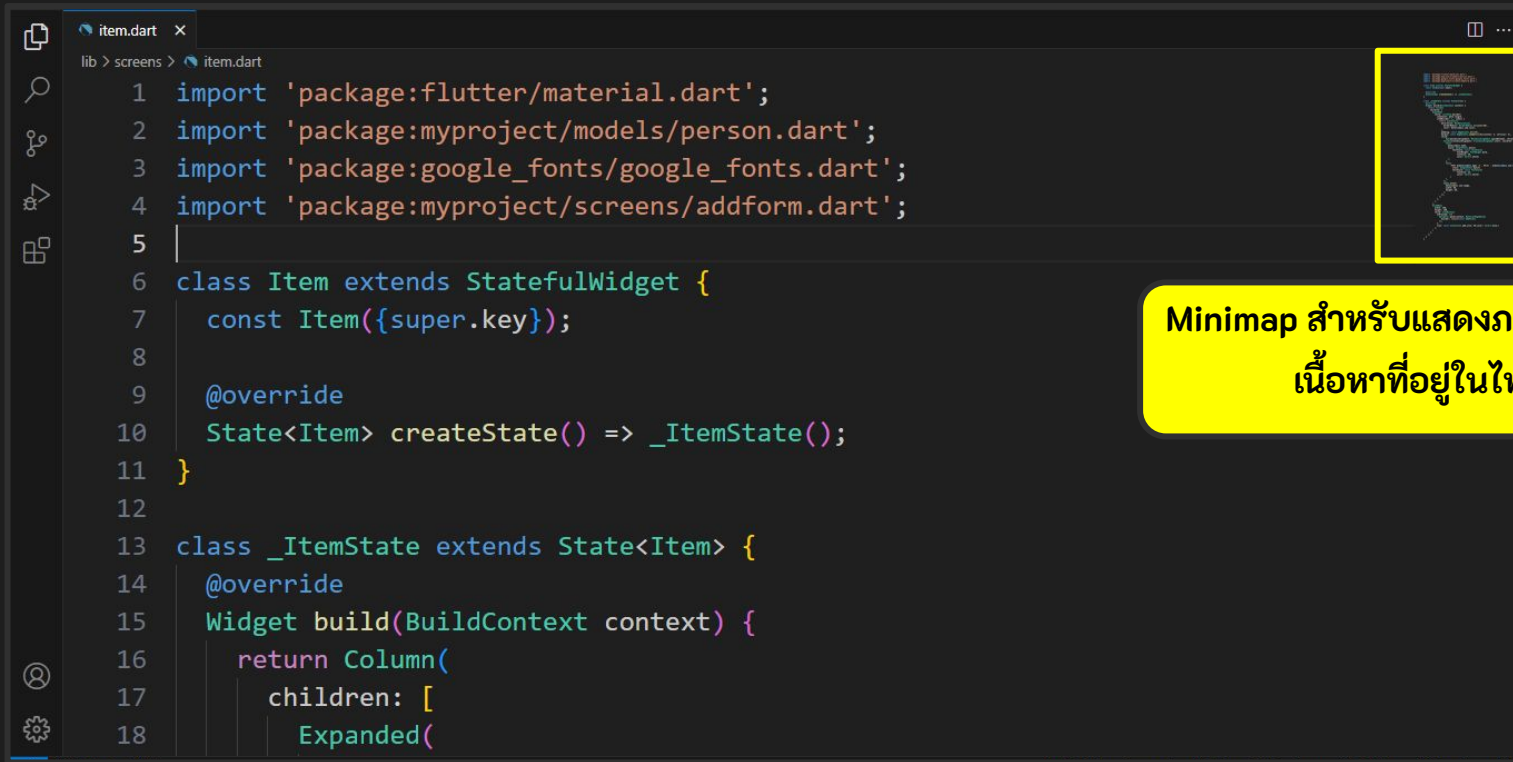


- **Extension** จัดการส่วนขยาย
หรือส่วนเสริมภายในโปรแกรม
เพื่อเพิ่มประสิทธิภาพการทำงาน



```
item.dart x
lib > screens > item.dart
1 import 'package:flutter/material.dart';
2 import 'package:myproject/models/person.dart';
3 import 'package:google_fonts/google_fonts.dart';
4 import 'package:myproject/screens/addform.dart';
5
6 class Item extends StatefulWidget {
7   const Item({super.key});
8
9   @override
10  State<Item> createState() => _ItemState();
11 }
12
13 class _ItemState extends State<Item> {
14   @override
15   Widget build(BuildContext context) {
16     return Column(
17       children: [
18         Expanded(
```

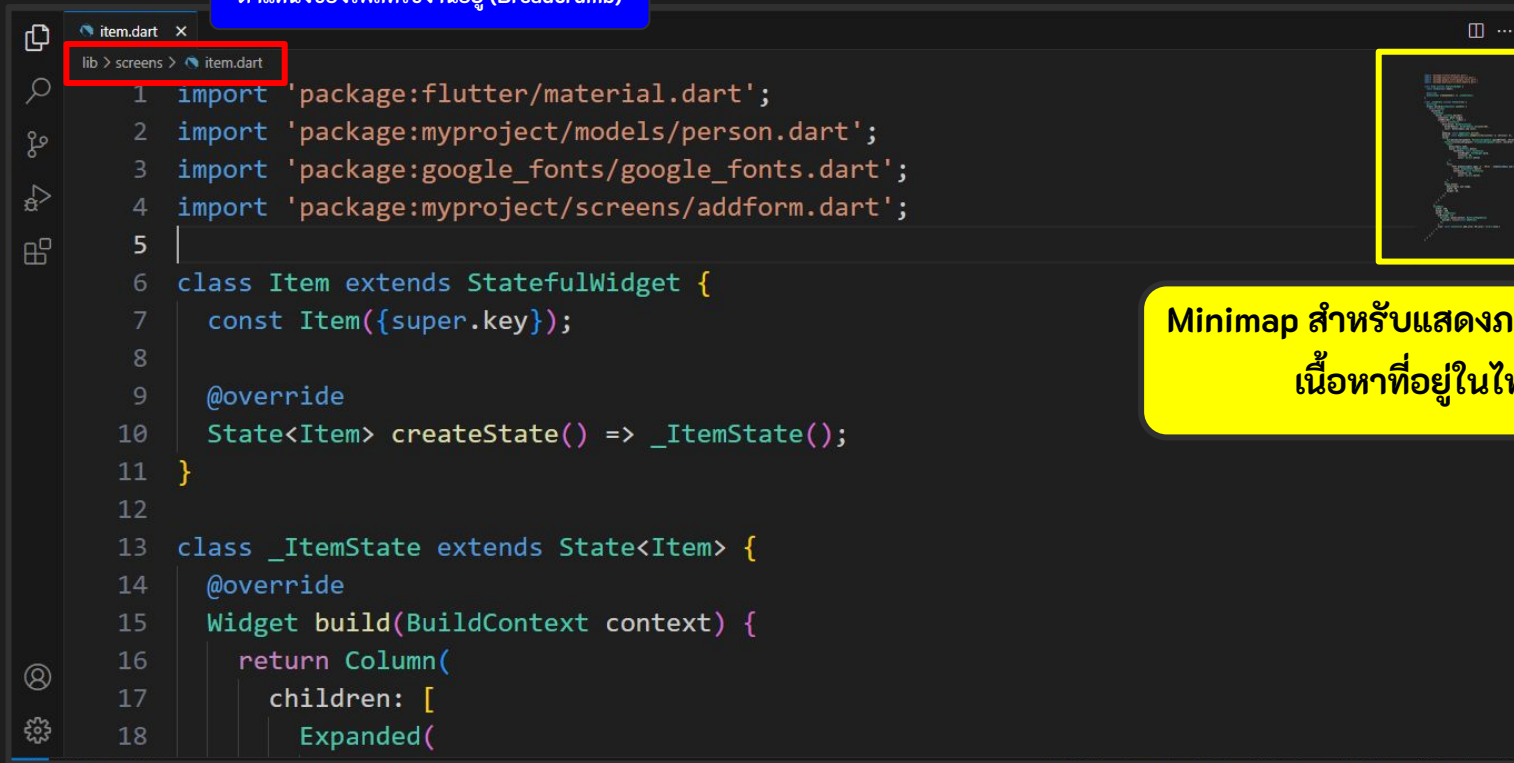
- **Editor** เป็นพื้นที่สำหรับใช้ในการเขียนโค้ด



Minimap สำหรับแสดงภาพรวมของ
เนื้อหาที่อยู่ในไฟล์

- **Editor** เป็นพื้นที่สำหรับการเขียนโค้ด

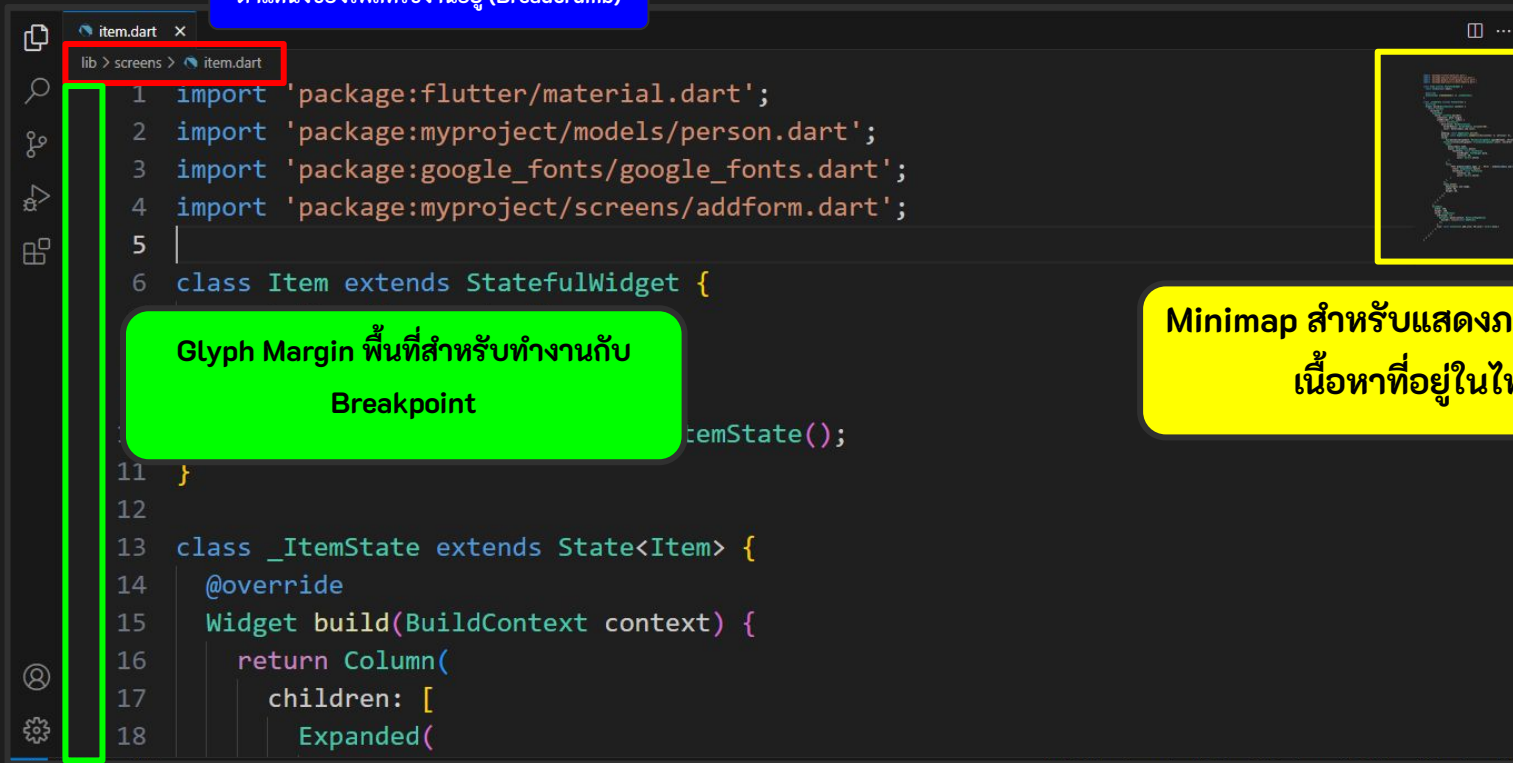
ตำแหน่งของไฟล์ที่ใช้งานอยู่ (Breadcrumb)



Minimap สำหรับแสดงภาพรวมของ
เนื้อหาที่อยู่ในไฟล์

- **Editor** เป็นพื้นที่สำหรับการเขียนโค้ด

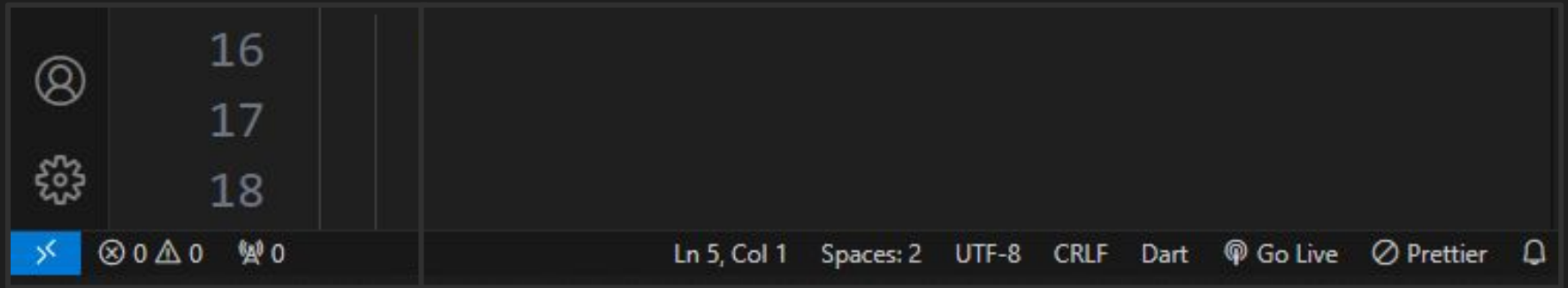
ตำแหน่งของไฟล์ที่ใช้งานอยู่ (Breadcrumb)



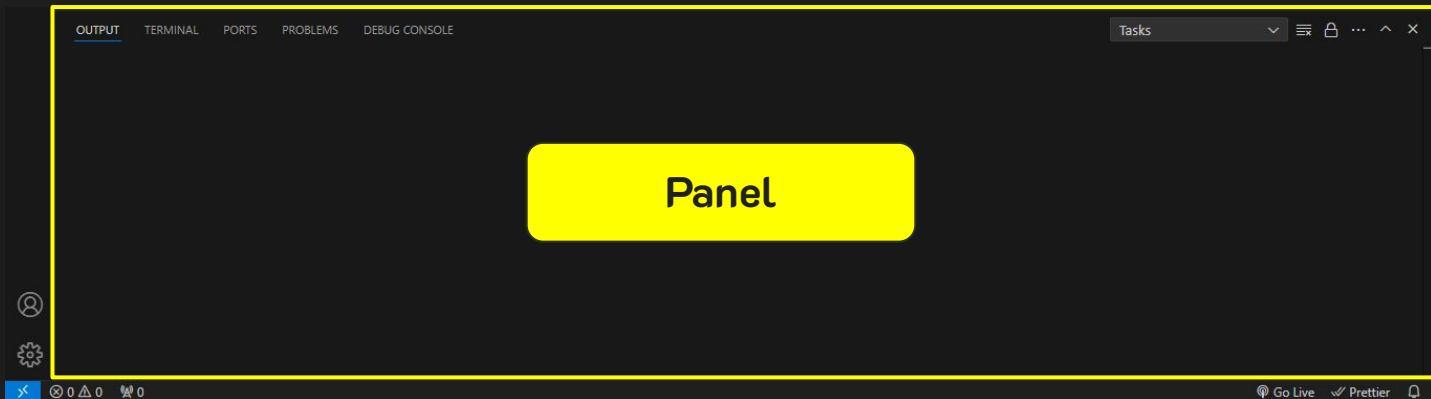
Glyph Margin พื้นที่สำหรับทำงานกับ
Breakpoint

Minimap สำหรับแสดงภาพรวมของ
เนื้อหาที่อยู่ในไฟล์

- **Editor** เป็นพื้นที่สำหรับใช้ในการเขียนโค้ด



- **Status Bar** พื้นที่สำหรับแสดงรายละเอียดต่างๆ เช่น จำนวนบรรทัด , ภาษาที่เขียน , Extension ที่เกี่ยวข้อง เป็นต้น



- **Panel** พื้นที่แสดงรายละเอียดต่างๆ เช่น การแสดงผล (Output) , การแจ้งข้อผิดพลาด (Problem) , การตรวจสอบ (Debug) และ Terminal เป็นต้น

การใช้งานเบื้องต้น



- การตั้งค่า Theme
- การตั้งค่าขนาดข้อความ
- ซ่อน-แสดง Minimap
- เปิด-ปิด Glyph Margin

การใช้งานเบื้องต้น



- Full Screen - โหมดเต็มจอ
- Zen Mode - โหมดโฟกัส (ไม่มี Tab ด้านข้าง)
- การย่อ-ขยายผ่านคีย์บอร์ด (Zoom)
- การย่อ-ขยายผ่านเมาส์ (Mouse Wheel Zoom)

ข้อควรรู้เกี่ยวกับการตั้งค่า



Editor: **Font Size**

Controls the font size in pixels.

14

- ค่าเริ่มต้น

ข้อควรรู้เกี่ยวกับการตั้งค่า



Editor: Font Size

Controls the font size in pixels.

25

- ค่าเริ่มต้น
- เปลี่ยนแปลงการตั้งค่า

ข้อควรรู้เกี่ยวกับการตั้งค่า



Editor: Font Size

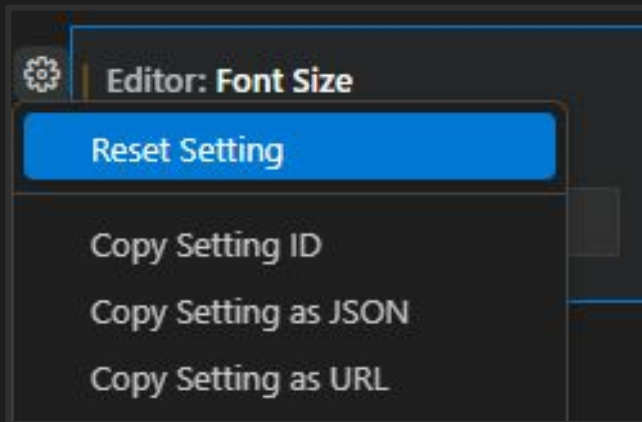
Controls the font size in pixels.

25

เส้นกำกับ
เพื่อบอกว่ามี
การเปลี่ยนแปลง
การตั้งค่า

- ค่าเริ่มต้น
- เปลี่ยนแปลงการตั้งค่า

ข้อควรรู้เกี่ยวกับการตั้งค่า



- ค่าเริ่มต้น
- เปลี่ยนแปลงการตั้งค่า
- รีเซ็ตการตั้งค่า

Command Palette

ส่วนที่ใช้ในการแสดงคำสั่งทั้งหมดหรือคำสั่งที่มีการใช้งานบ่อยเป็น
เครื่องมือทางลัดที่อำนวยความสะดวกในการเรียกใช้งานคำสั่งต่างๆ
ภายในโปรแกรมได้รวดเร็วมากยิ่งขึ้น ใช้การคลิกที่น้อยลง เช่น การตั้ง
ค่า , การจัดการไฟล์ การรีเซ็ตตำแหน่ง View , อื่นๆ เป็นต้น

แสดง Command Palette

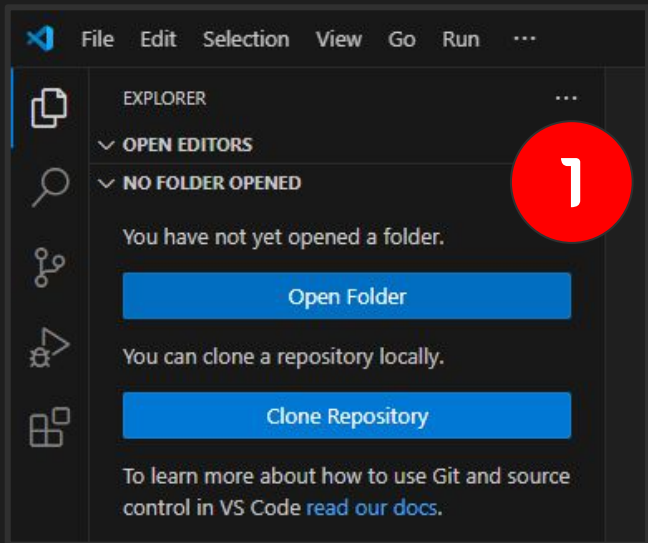
1. กด F1 หรือ Ctrl + Shift+P
2. View -> Command Palette
3. Manage -> Command Palette

จัดการโฟลเดอร์

- เปิด - ปิดโฟลเดอร์
- สร้าง - ลบโฟลเดอร์
- เปลี่ยนชื่อโฟลเดอร์
- โฟลเดอร์ย่อย
- การย้ายโฟลเดอร์ (Drag and Drop)

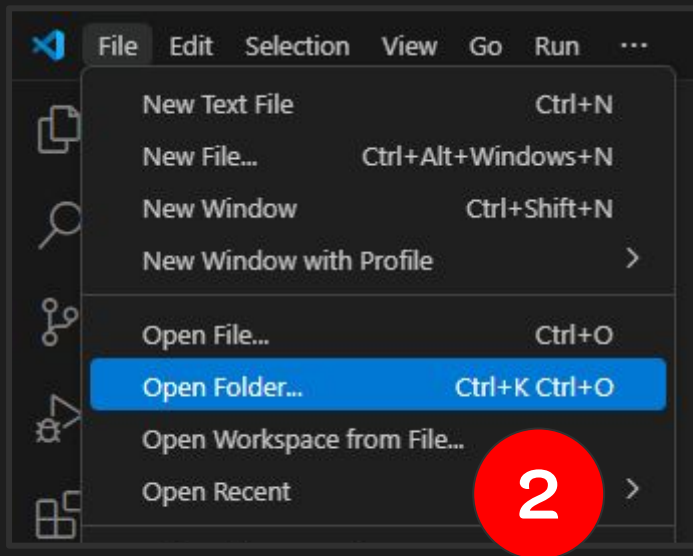


เปิดโฟลเดอร์



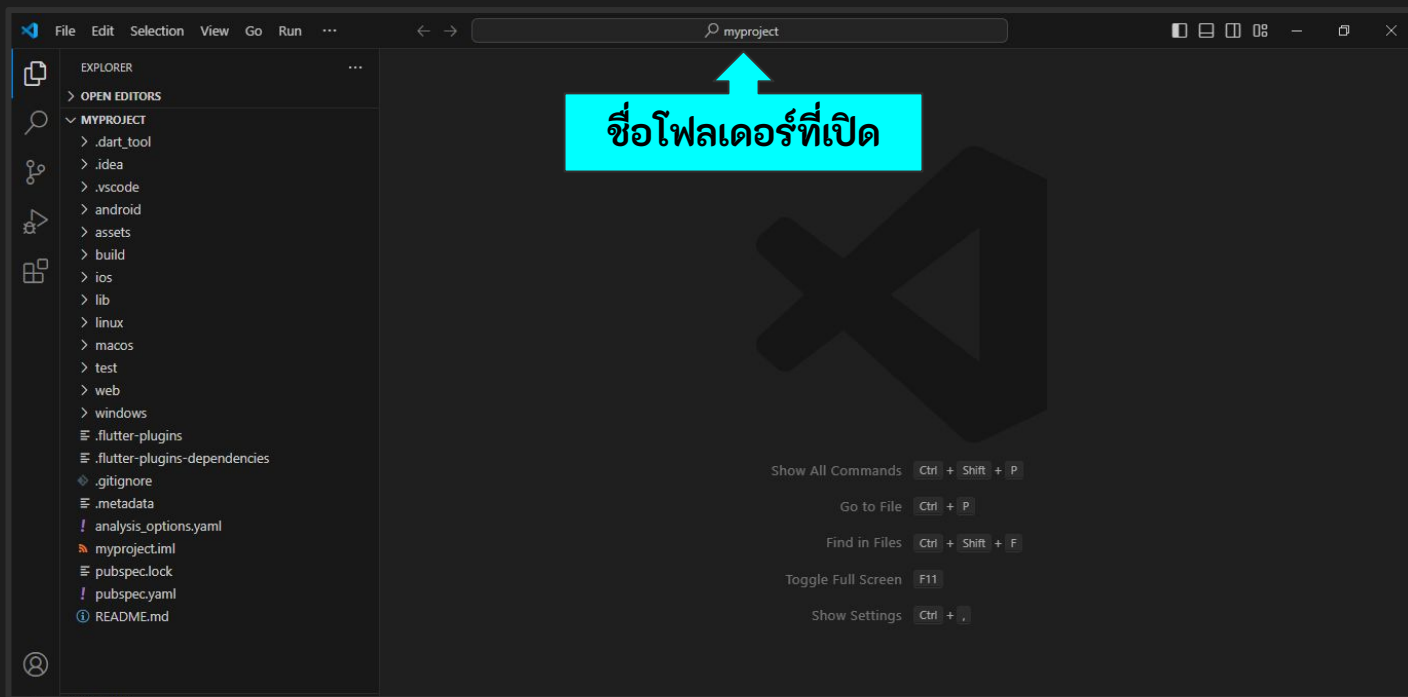
- Explorer -> Open Folder

เปิดโฟลเดอร์

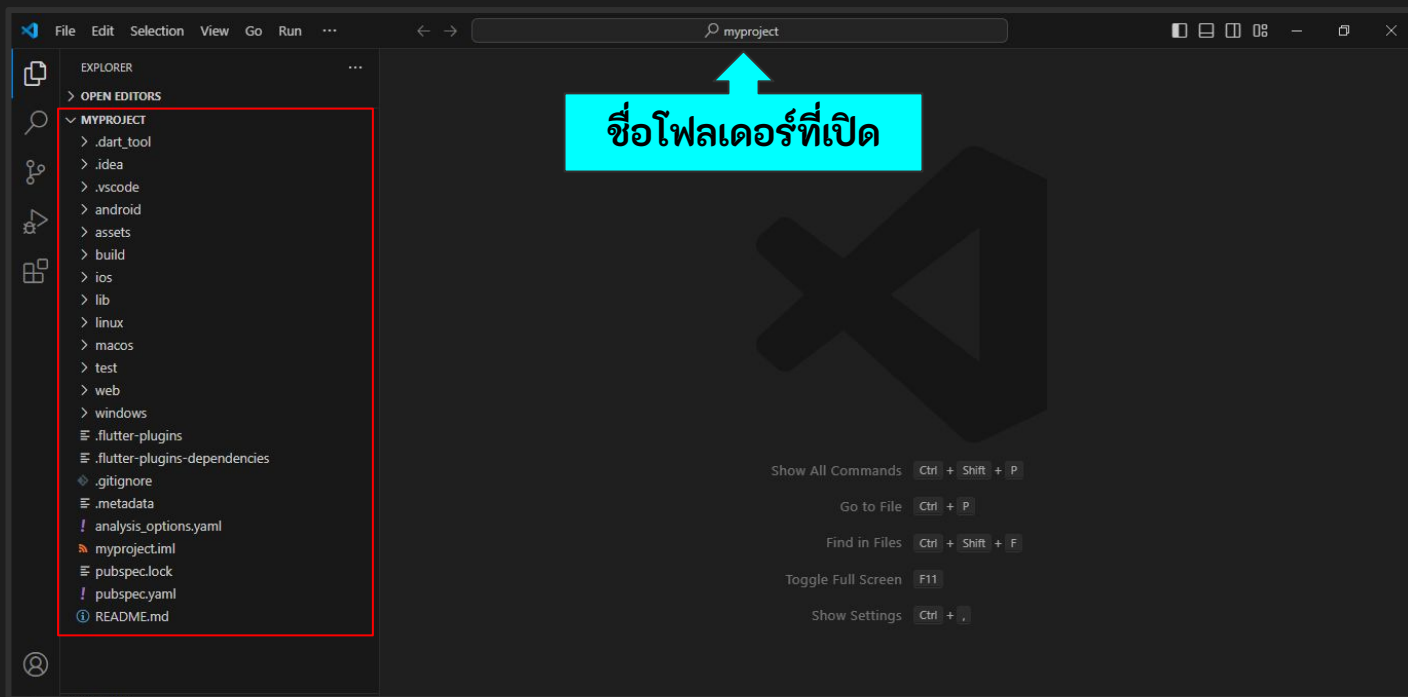


- File -> Open Folder

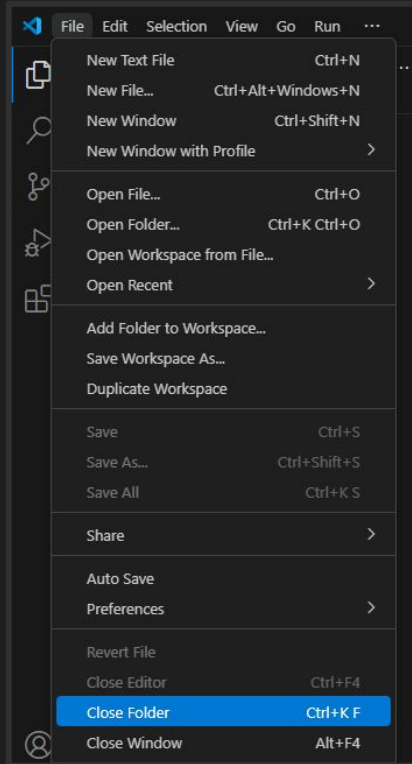
เปิดโฟลเดอร์



เปิดโฟลเดอร์

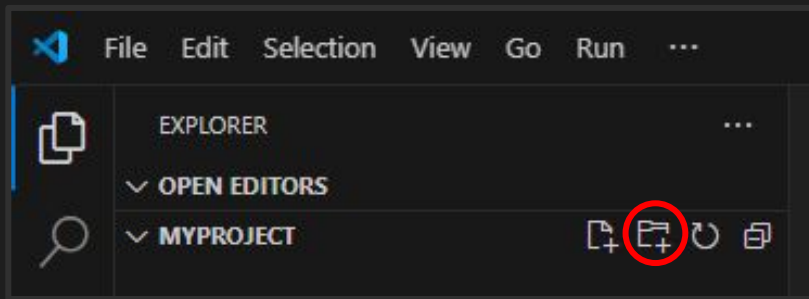


ปิดโฟลเดอร์



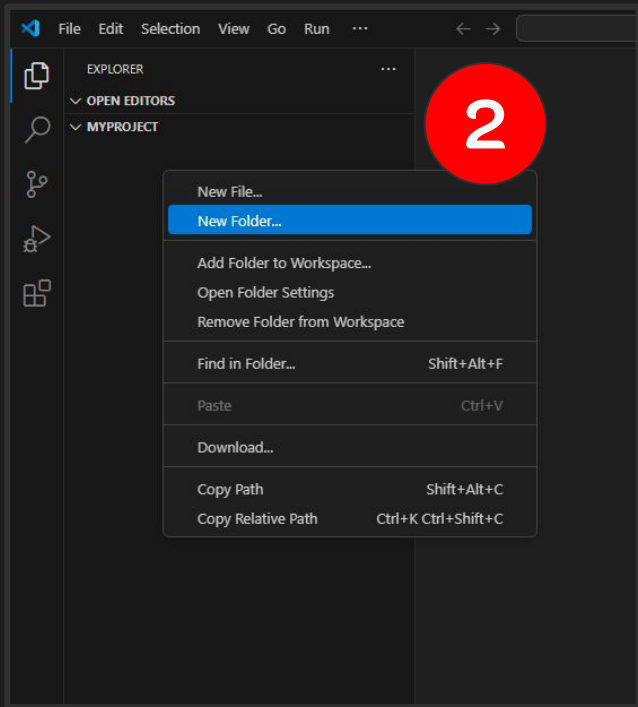
- File -> Close Folder
- หรือกดคีย์ลัด Ctrl+K F

สร้างโฟลเดอร์



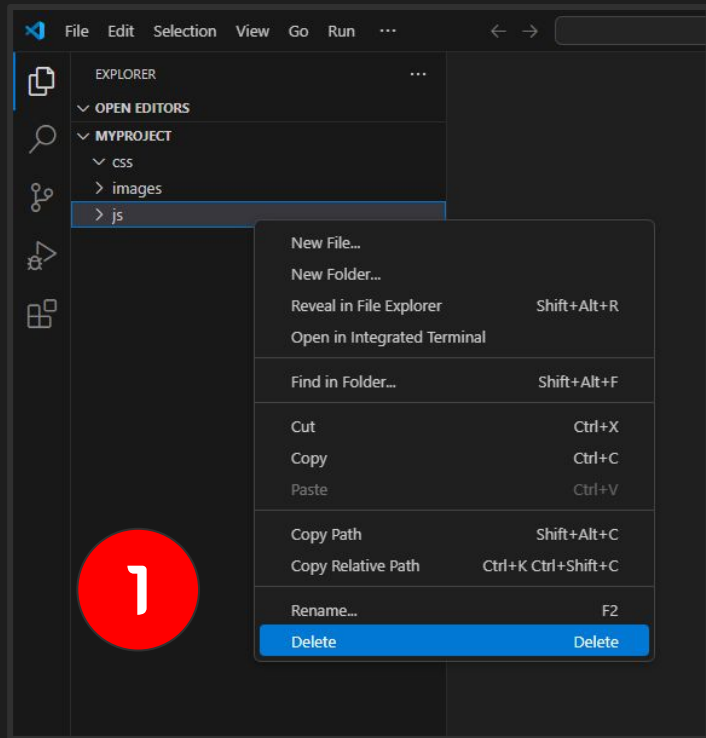
- กดไอคอน New Folder
- ตั้งชื่อโฟลเดอร์

สร้างโฟลเดอร์



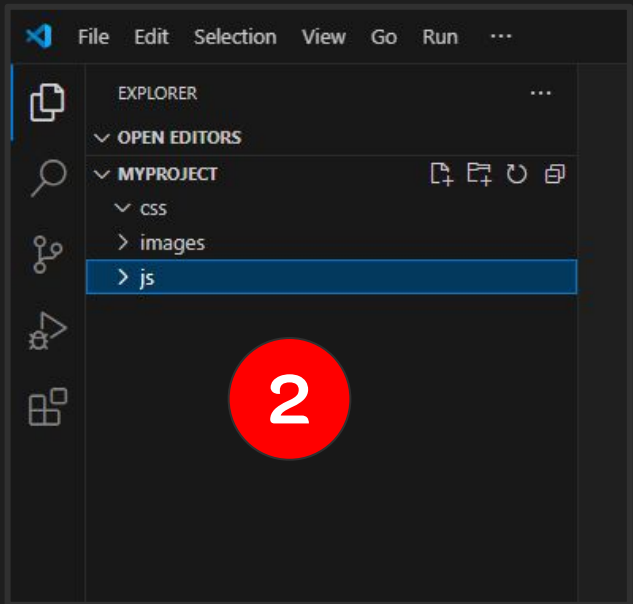
- คลิกขวาพื้นที่ว่าง Explorer
- New Folder...
- ตั้งชื่อโฟลเดอร์

ลบโฟลเดอร์



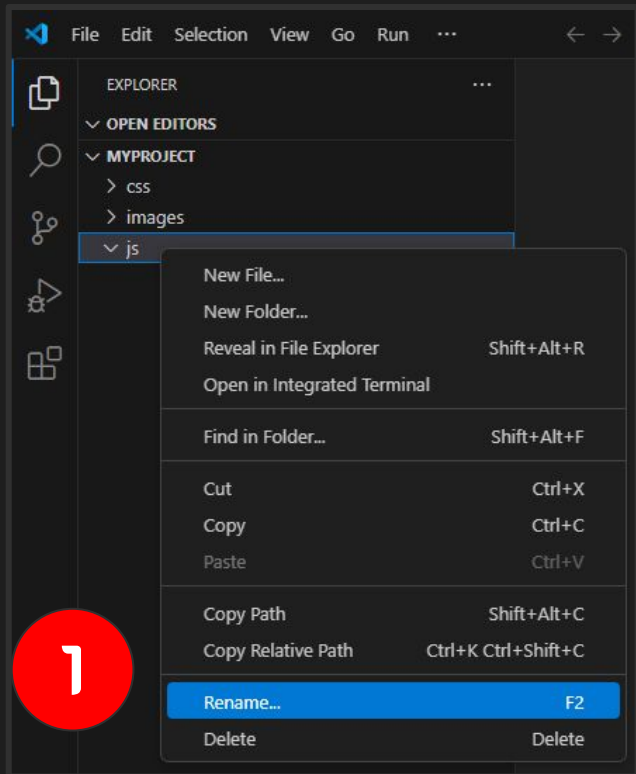
- คลิกขวาโฟลเดอร์ที่ต้องการลบ
- เลือกเมนู Delete
- ยืนยันการลบ

ลบโฟลเดอร์



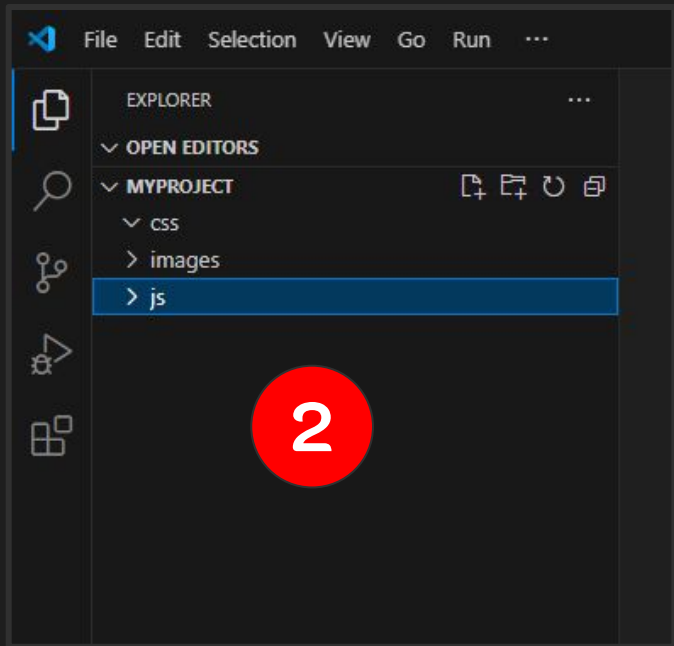
- เลือกโฟลเดอร์ที่ต้องการลบ
(กรณีลบหลายโฟลเดอร์ให้กด Ctrl ร่วมด้วย)
- กดปุ่ม Delete
- ยืนยันการลบ

เปลี่ยนชื่อโฟลเดอร์



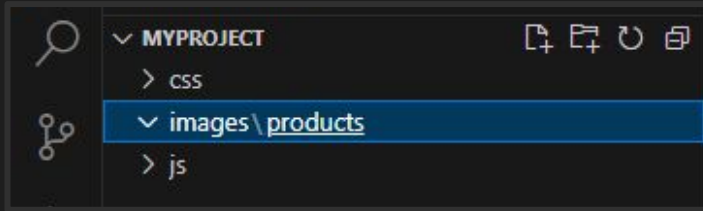
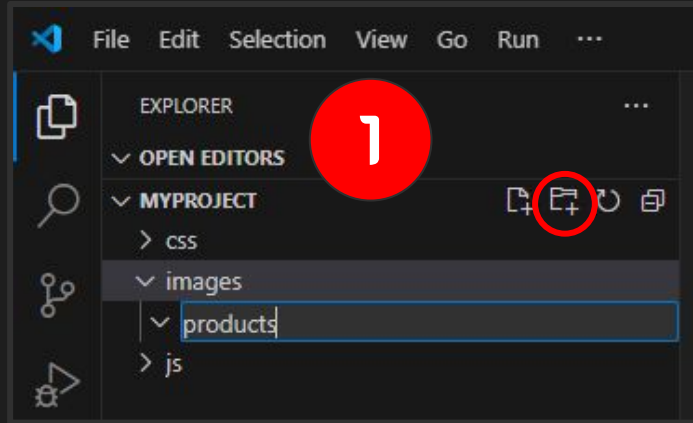
- คลิกขวาโฟลเดอร์ที่ต้องการเปลี่ยนชื่อ
- เลือกเมนู Rename...
- ตั้งชื่อใหม่

เปลี่ยนชื่อโฟลเดอร์



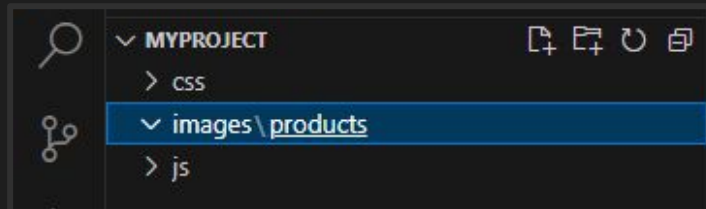
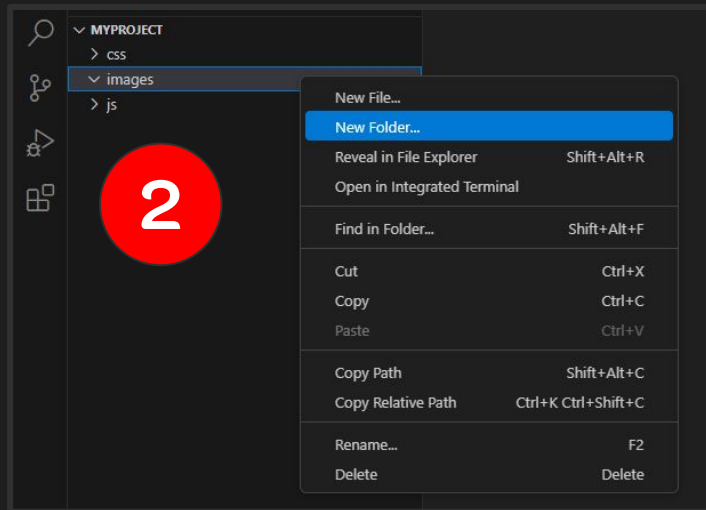
- เลือกโฟลเดอร์ที่ต้องการเปลี่ยนชื่อ
- กดปุ่ม F2
- ตั้งชื่อใหม่

สร้างโฟลเดอร์ย่อย



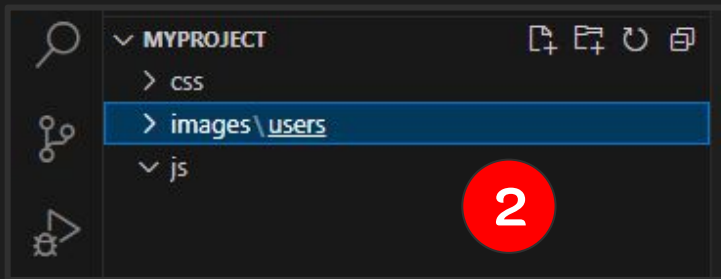
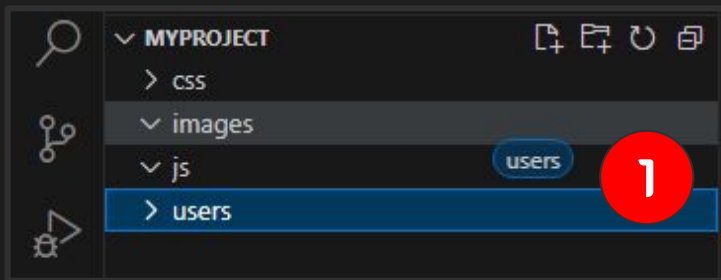
- เลือกโฟลเดอร์หลัก
- กดไอคอน New Folder
- ตั้งชื่อโฟลเดอร์ย่อย

สร้างโฟลเดอร์ย่อย



- คลิกขวาที่โฟลเดอร์หลัก
- เลือกเมนู New Folder
- ตั้งชื่อโฟลเดอร์ย่อย

การย้ายโฟลเดอร์ (Drag and Drop)



1. เลือกโฟลเดอร์ที่ต้องการย้าย

(กรณีย้ายหลายโฟลเดอร์ให้กด Ctrl ร่วมด้วย)

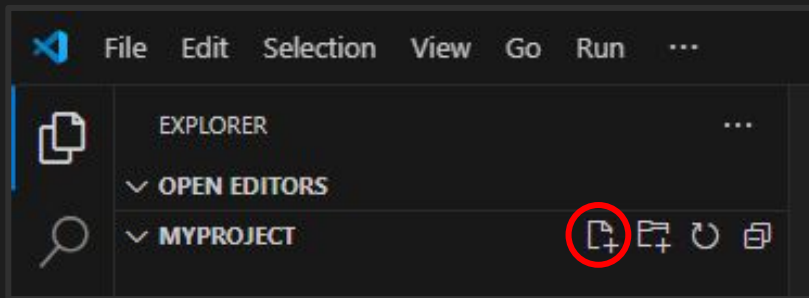
2. ลาก-วางไปยังตำแหน่งโฟลเดอร์ปลายทางที่ต้องการ

จัดการไฟล์

- สร้างไฟล์ใหม่ - เปลี่ยนชื่อไฟล์
- การบันทึกไฟล์
- การลบไฟล์
- การเปิด-ปิดไฟล์
- การย้ายไฟล์ (Drag and Drop)
- สร้างโฟลเดอร์และไฟล์ในครั้งเดียว

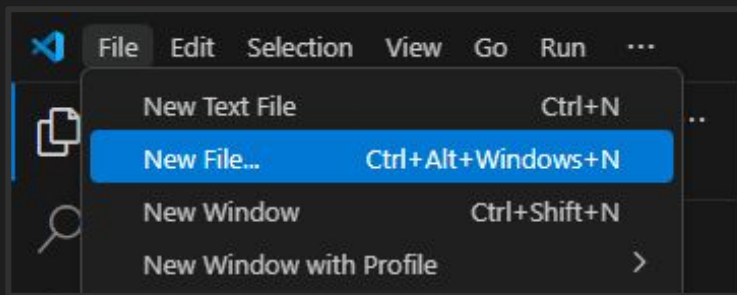


สร้างไฟล์ใหม่



- กดไอคอน New File
- กำหนดชื่อไฟล์และนามสกุลไฟล์

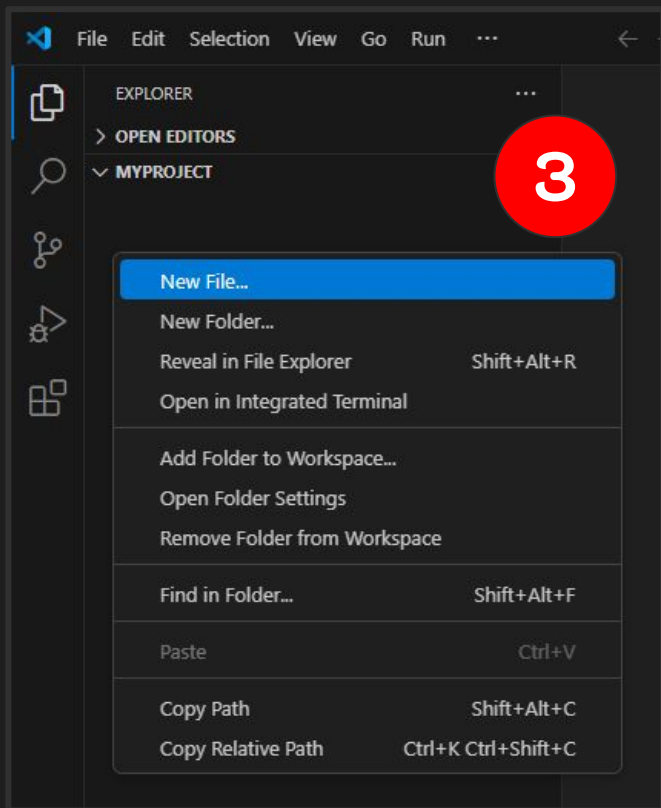
สร้างไฟล์ใหม่



2

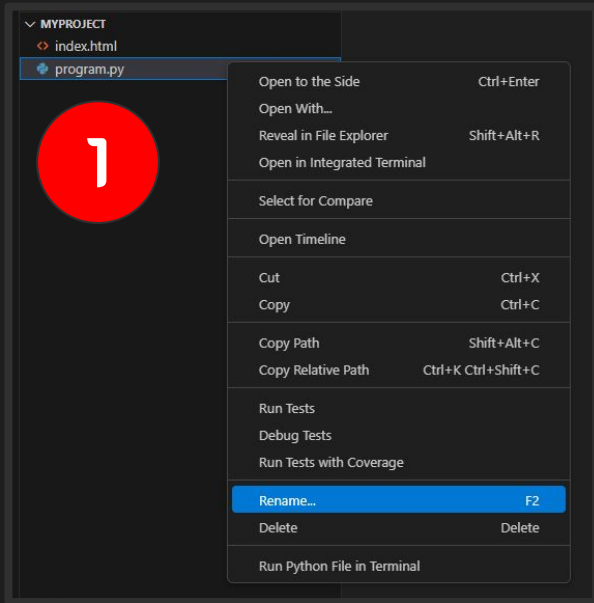
- File -> New File...
- กำหนดรูปแบบไฟล์

สร้างไฟล์ใหม่



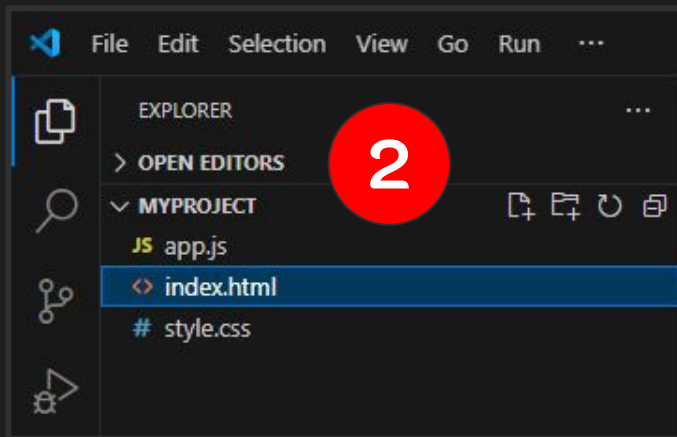
- คลิกขวาพื้นที่ว่าง Explorer
- เลือกเมนู New File...
- กำหนดชื่อไฟล์และนามสกุลไฟล์

เปลี่ยนชื่อไฟล์



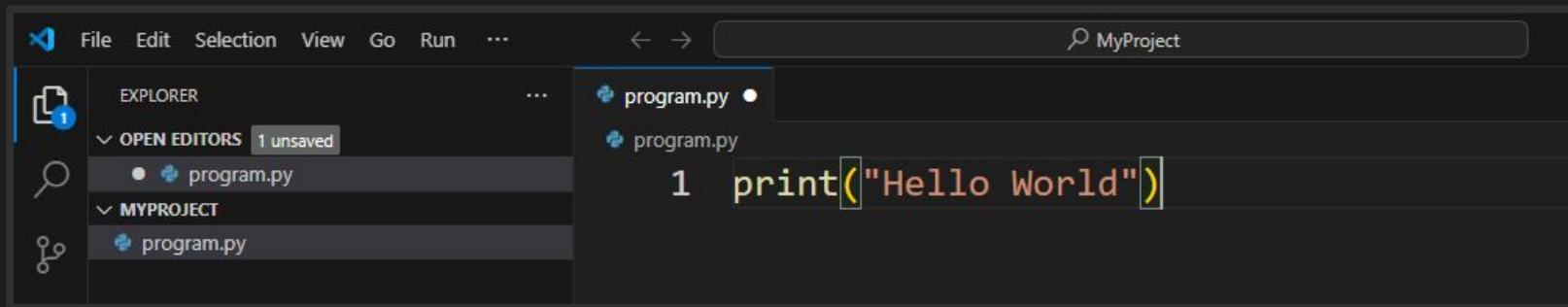
- คลิกขวาไฟล์ที่ต้องการเปลี่ยนชื่อ
- เลือกเมนู Rename
- ตั้งชื่อไฟล์ใหม่

เปลี่ยนชื่อไฟล์



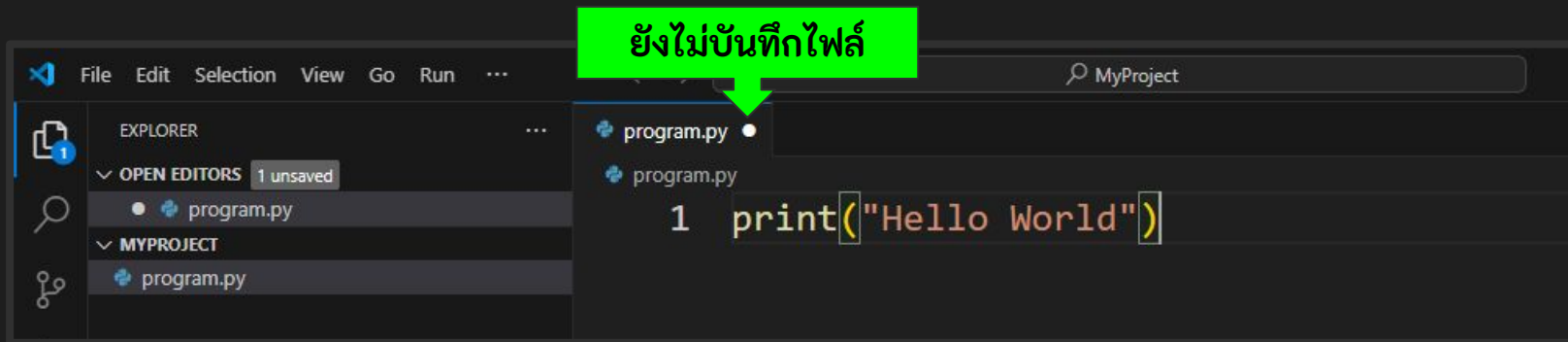
- เลือกไฟล์ที่ต้องการเปลี่ยนชื่อ
- กดปุ่ม F2
- ตั้งชื่อไฟล์ใหม่

การบันทึกไฟล์



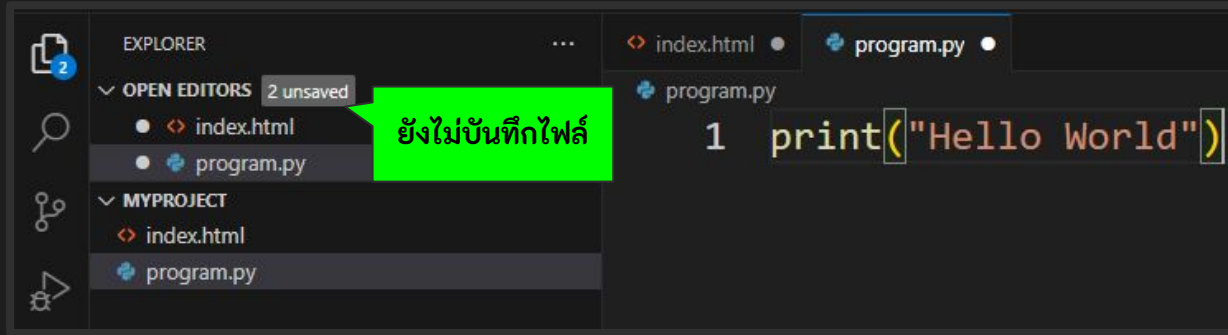
- ไฟล์ที่ยังไม่ถูกบันทึกจะมีเครื่องหมายจุดสีขาว่าต่อท้ายชื่อไฟล์
- Open Editors จะแจ้งจำนวนไฟล์ที่ยังไม่มีการบันทึก

การบันทึกไฟล์



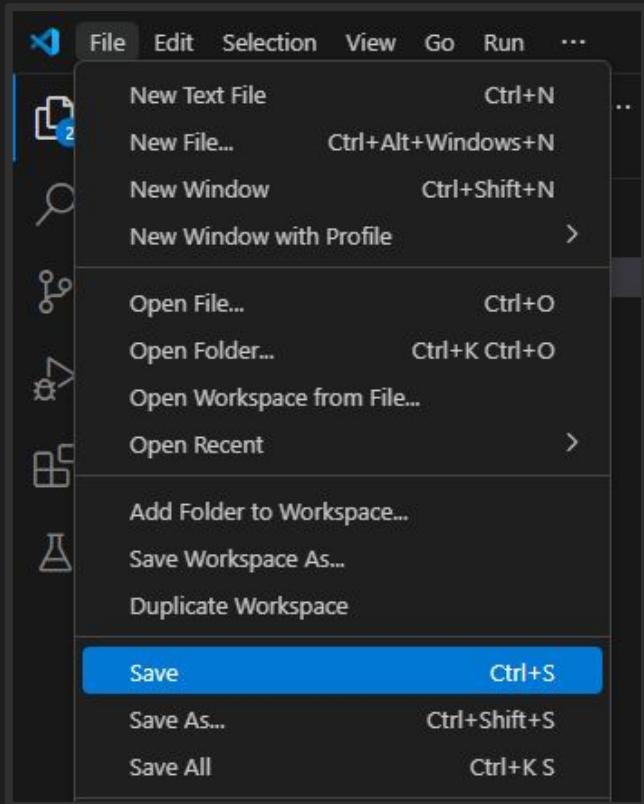
- ไฟล์ที่ยังไม่ถูกบันทึกจะมีเครื่องหมายจุดสีขาวต่อท้ายชื่อไฟล์
- Open Editors จะแจ้งจำนวนไฟล์ที่ยังไม่มีการบันทึก

การบันทึกไฟล์



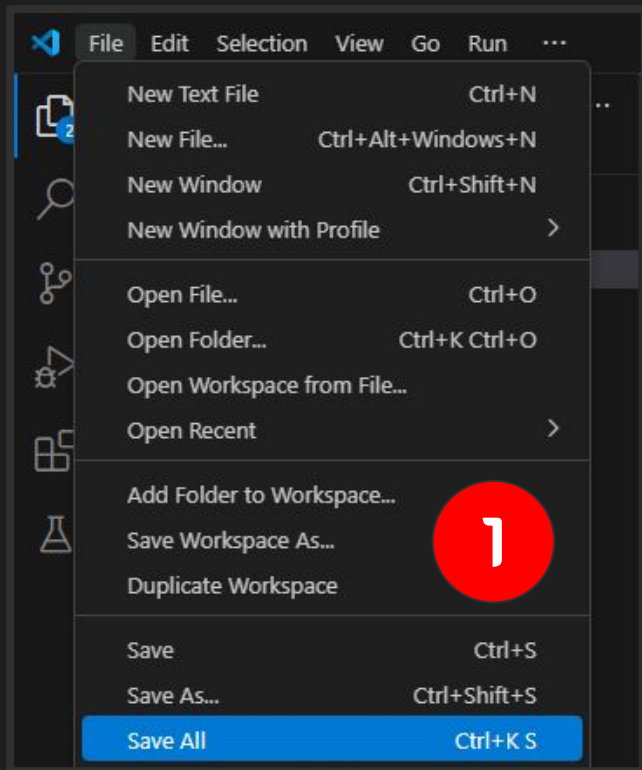
- ไฟล์ที่ยังไม่ถูกบันทึกจะมีเครื่องหมายจุดสีขาว่าต่อท้ายชื่อไฟล์
- Open Editors จะแจ้งจำนวนไฟล์ที่ยังไม่มีการบันทึก

การบันทึกไฟล์ (เฉพาะไฟล์ที่สนใจ)



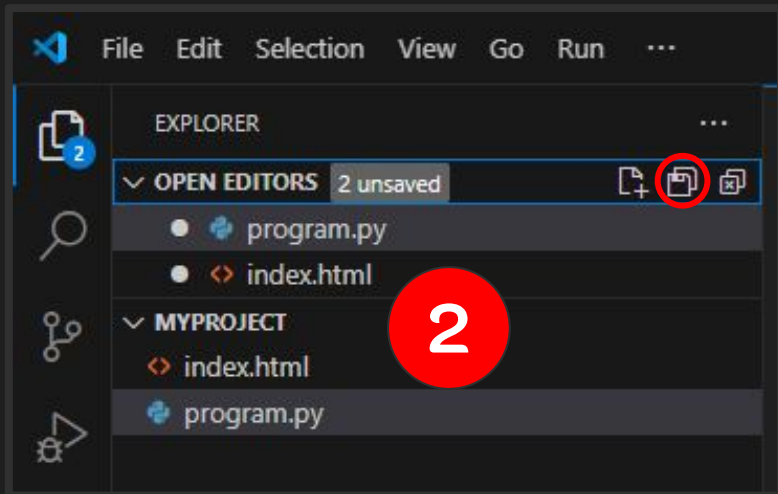
- File -> Save
- กดคีย์ลัด Ctrl + S

การบันทึกไฟล์ (ทั้งหมด)



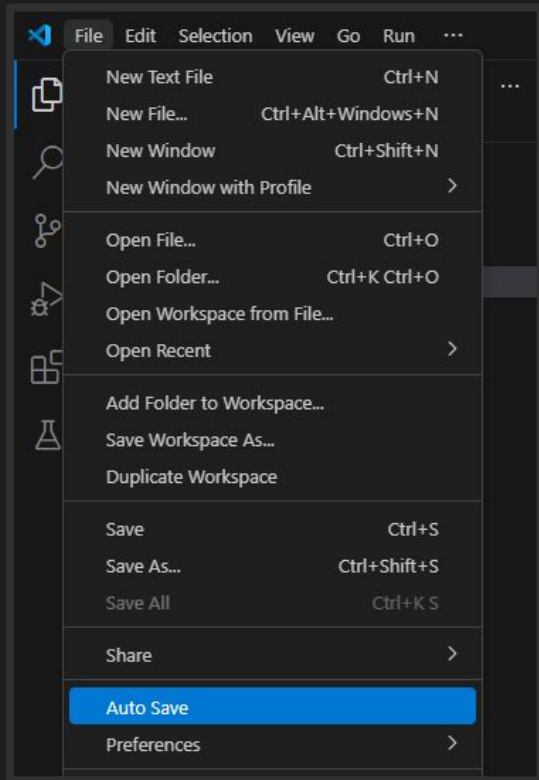
- File -> Save All
- กดคีย์ลัด Ctrl + K S

การบันทึกไฟล์ (ทั้งหมด)



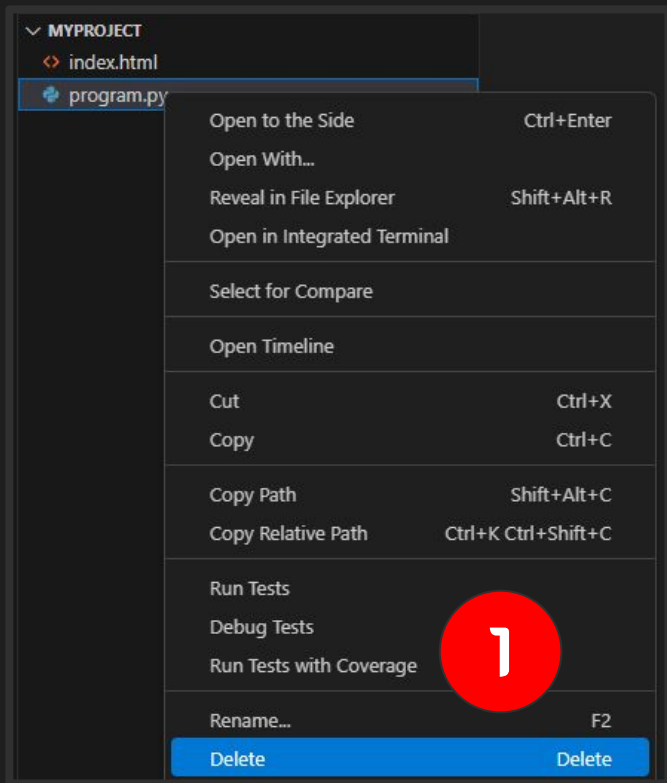
- Open Editors
- กดไอคอน Save All

ตั้งค่าบันทึกอัตโนมัติ (Auto Save)



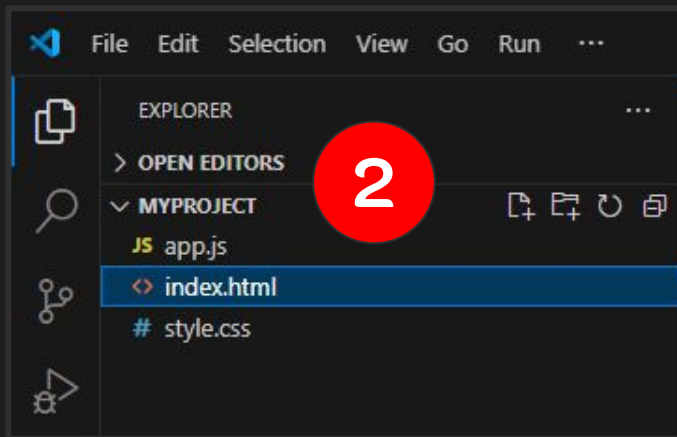
- เมนู File
- Auto Save

การลบไฟล์



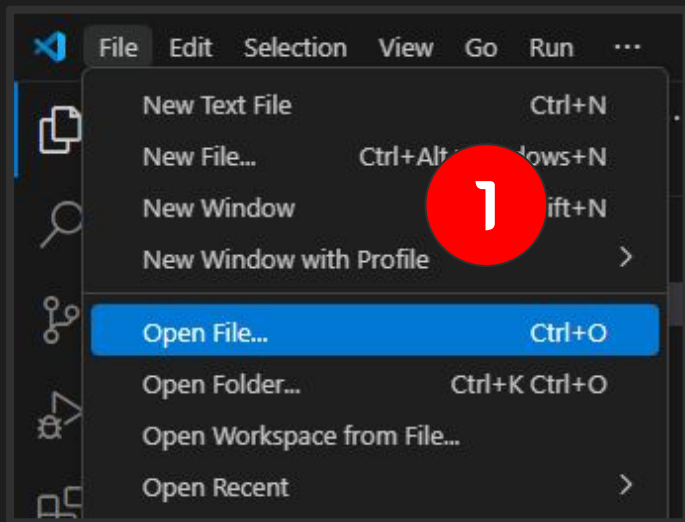
- คลิกขวาไฟล์ที่ต้องการลบ
- เลือกเมนู Delete
- ยืนยันการลบ

การลบไฟล์



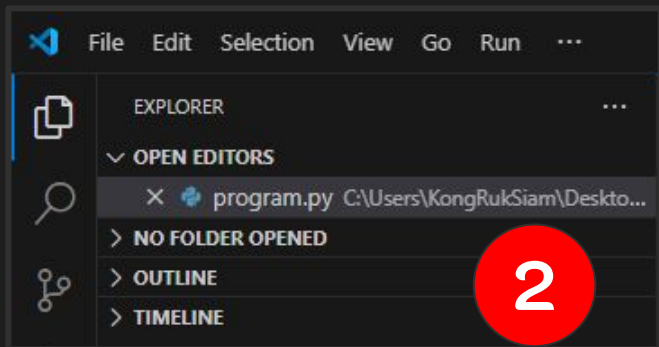
- เลือกไฟล์ที่ต้องการลบ
(กรณีลบหลายไฟล์ให้กด Ctrl ร่วมด้วย)
- กดปุ่ม Delete
- ยืนยันการลบ

การเปิดไฟล์



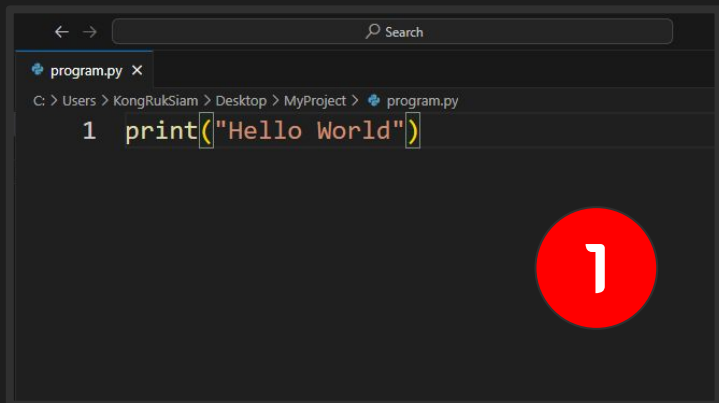
- File -> Open File
- เลือกไฟล์ที่ต้องการเปิดใน Editor

การเปิดไฟล์



- กดคีย์ลัด Ctrl + O
- เลือกไฟล์ที่ต้องการเปิด

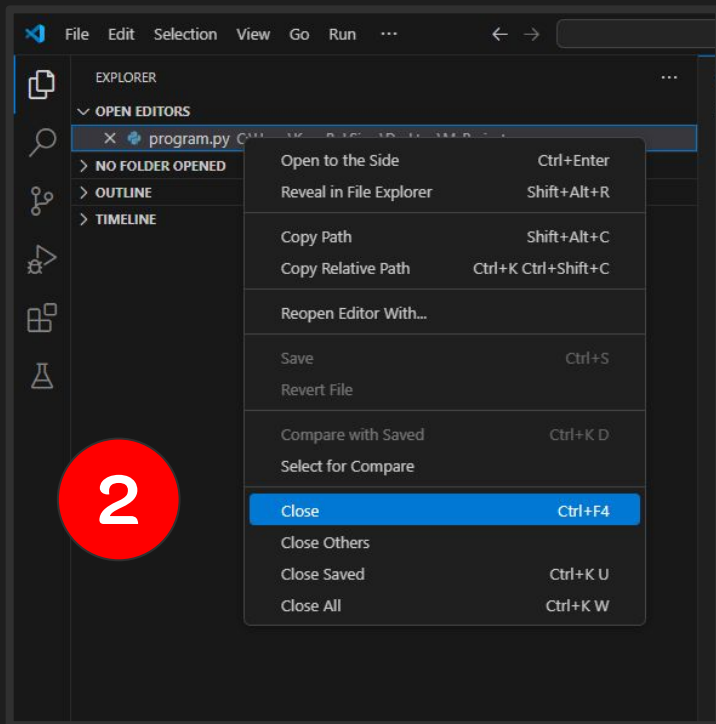
การปิดไฟล์



A screenshot of a code editor window. The title bar shows 'program.py' with a close button. The address bar shows the file path: 'C: > Users > KongRuksiam > Desktop > MyProject > program.py'. The code area contains a single line: '1 print("Hello World")'. A red circle with the number '1' is overlaid on the bottom right of the code editor.

- กดเครื่องหมายกากบาท (x) ในไฟล์ที่ต้องการปิด

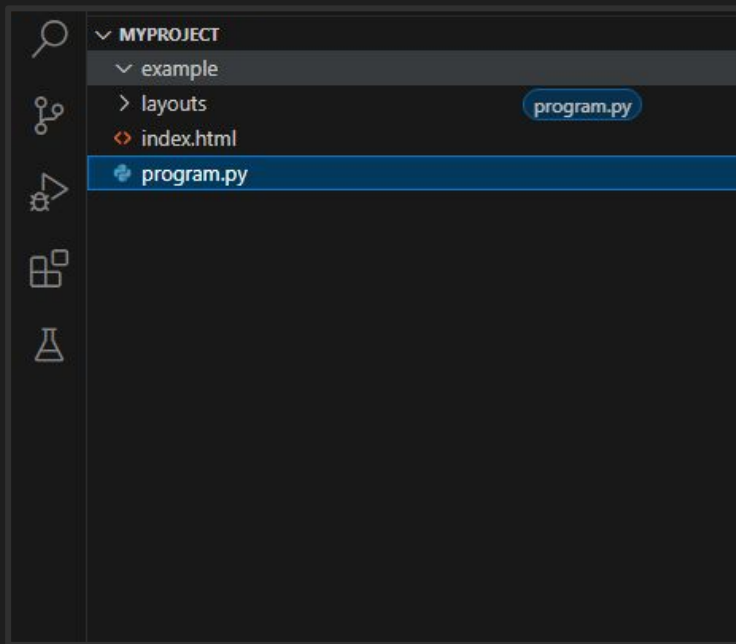
การปิดไฟล์



- คลิกขวาที่ไฟล์ -> เลือกเมนู Close
- หรือกดคีย์ลัด Ctrl + F4

(กรณีปิดหลายไฟล์ให้เลือกเมนู Close All)

การย้ายไฟล์ (Drag and Drop)



1. เลือกไฟล์ที่ต้องการย้าย

(กรณีย้ายหลายไฟล์ให้กด Ctrl ร่วมด้วย)

2. ลาก-วางไปยังตำแหน่งโฟลเดอร์

หรือปลายทางที่ต้องการ

การสร้างโฟลเดอร์และไฟล์ในครั้งเดียว

โครงสร้างคำสั่ง

- ชื่อโฟลเดอร์หลัก / ชื่อโฟลเดอร์ย่อย / ชื่อไฟล์.นามสกุลไฟล์

ตัวอย่าง

- `layouts/pages/footer.html`

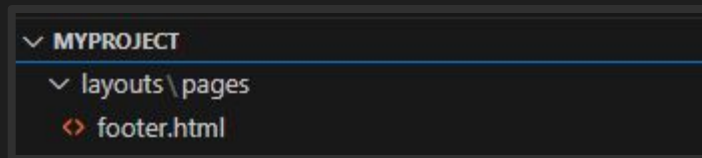
การสร้างโฟลเดอร์และไฟล์ในครั้งเดียว

โครงสร้างคำสั่ง

- ชื่อโฟลเดอร์หลัก / ชื่อโฟลเดอร์ย่อย / ชื่อไฟล์.นามสกุลไฟล์

ตัวอย่าง

- layouts/pages/footer.html



จัดการ Editor

- Previews Mode
- การจัดการแท็บ (Tabs)
- Sticky Scroll
- Editor Groups

จัดการ Editor

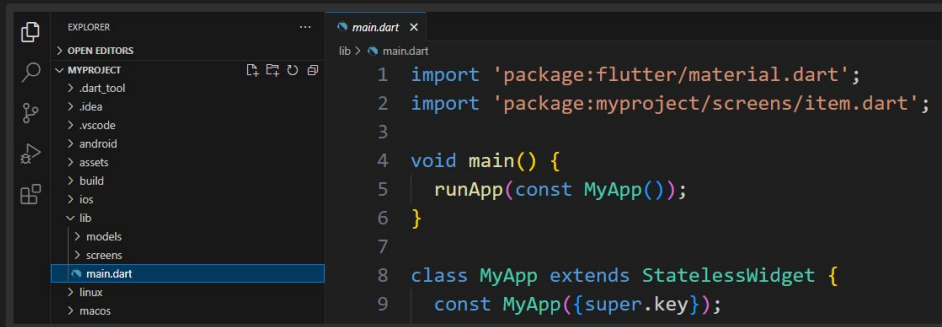
- Editor Layout
- Sort Lines
- Transform

Previews Mode

เป็นโหมดแสดงผลตัวอย่างเนื้อหาในไฟล์ที่เปิดใช้งานใน Editor
เมื่อมีการเปิดไฟล์ใหม่ก็จะแทนที่เนื้อหาเดิมใน Editor ให้อัตโนมัติ

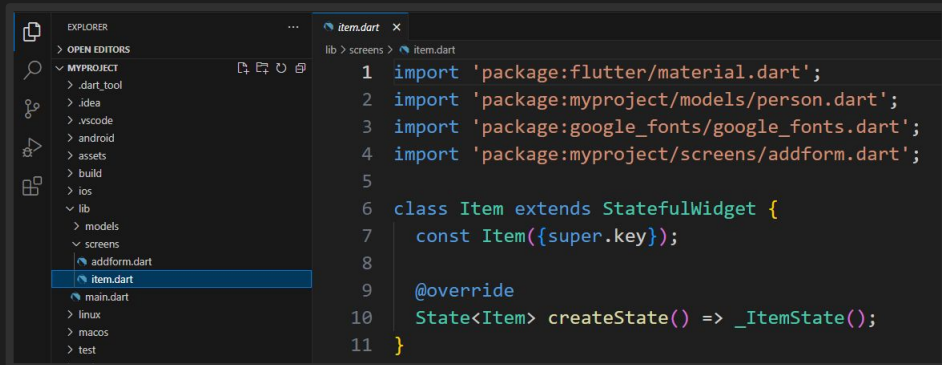
ซึ่งมีความสะดวกอย่างมากในการดูภาพรวมและแก้ไขเนื้อหาใน
ไฟล์ที่เปิดใช้งานภายในแท็บเดียว

Previews Mode



The screenshot shows the VS Code interface with the Explorer sidebar on the left. The 'main.dart' file is selected in the Explorer. The main editor area displays the code for 'main.dart' with line numbers 1 through 9. The code includes imports for Flutter material and a custom item screen, a main function that runs the app, and a StatelessWidget class named MyApp.

```
1 import 'package:flutter/material.dart';
2 import 'package:myproject/screens/item.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
```

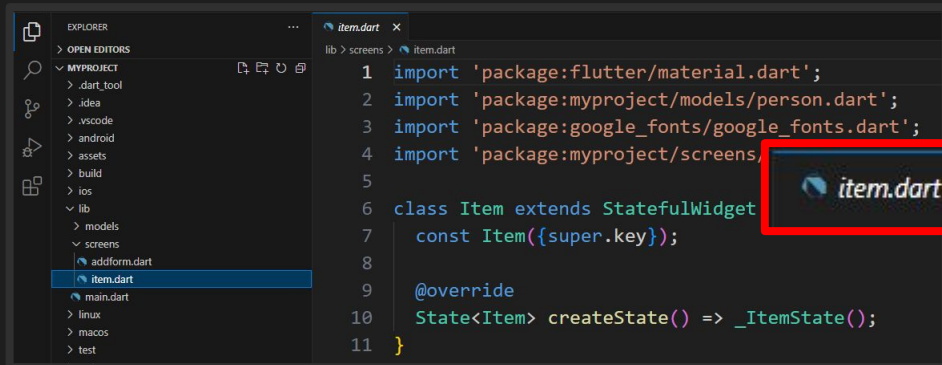
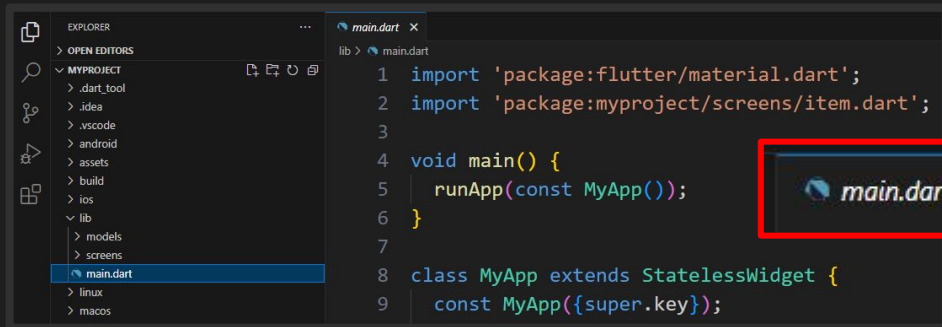


The screenshot shows the VS Code interface with the Explorer sidebar on the left. The 'item.dart' file is selected in the Explorer. The main editor area displays the code for 'item.dart' with line numbers 1 through 11. The code includes imports for Flutter material, a custom person model, Google Fonts, and a custom add form screen, and a StatefulWidget class named Item.

```
1 import 'package:flutter/material.dart';
2 import 'package:myproject/models/person.dart';
3 import 'package:google_fonts/google_fonts.dart';
4 import 'package:myproject/screens/addform.dart';
5
6 class Item extends StatefulWidget {
7   const Item({super.key});
8
9   @override
10  State<Item> createState() => _ItemState();
11 }
```

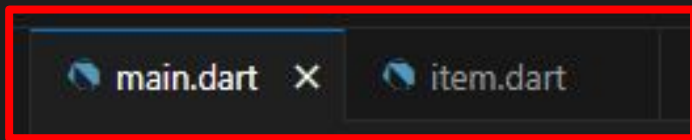
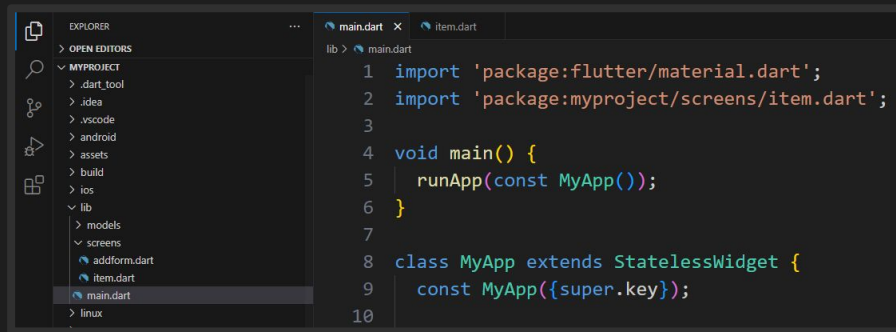
สามารถดูเนื้อหาภายในไฟล์
ที่ต้องการผ่านแท็บเดียวกัน
ได้เลย

Previews Mode



ไฟล์ที่เปิดผ่าน Preview Mode
ชื่อไฟล์จะเป็นตัวอักษร
ตัวเอียง

Previews Mode



ถ้าต้องการเปิดไฟล์ในแท็บ Editor

แบบถาวร โดยไม่มีการแทนที่เนื้อหาให้

ดำเนินการดังนี้

1. Double Click ที่ไฟล์
2. Double Click ที่แท็บ Editor

*ชื่อไฟล์ที่เปิดใน Editor จะเป็นตัวอักษรปกติ

*กรณีที่มีการแก้ไขข้อมูลในไฟล์จะเป็นโหมดปกติ

การจัดการแท็บ (Tabs)

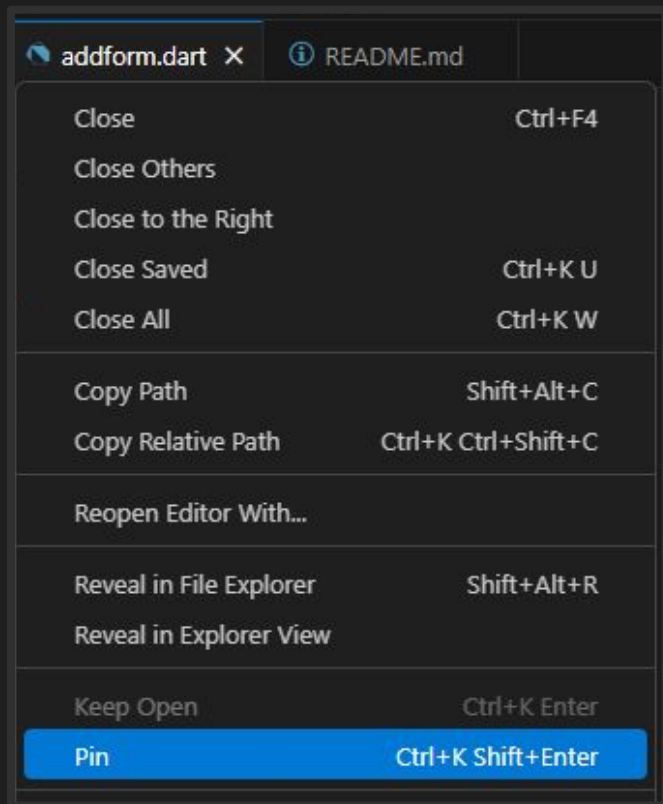
- กำหนดรูปแบบการแสดงผลแท็บ
- Pin Tabs
- Wrap Tabs

กำหนดรูปแบบการแสดงผลแท็บ

View -> Appearance -> Tab Bar

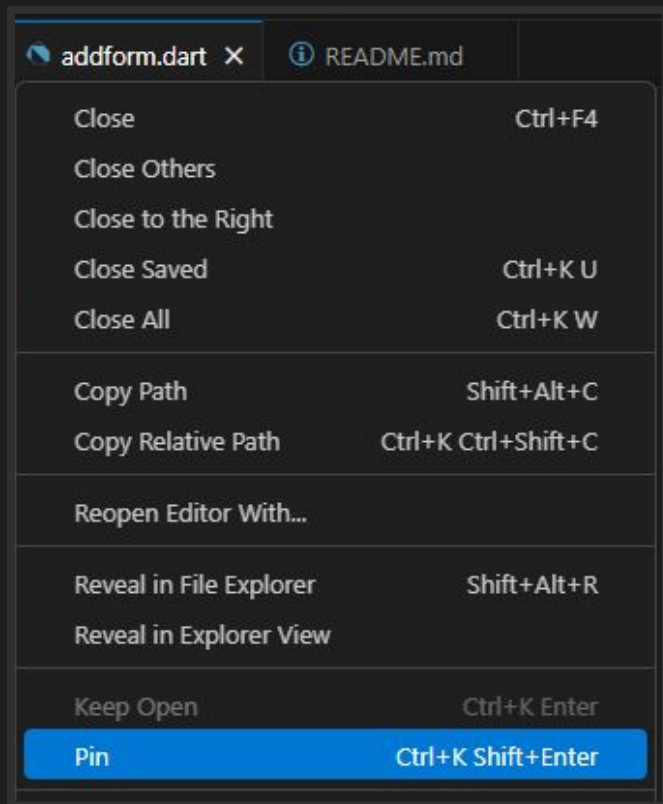
- Single (แท็บเดียว)
- Multiple Tabs (หลายแท็บ)
- Hidden (ซ่อนแท็บ)

Pin Tabs



- **Pin** คือ การปักหมุดแท็บ Editor ที่มีความสำคัญ ให้ย้ายไปอยู่ลำดับแรก หรือด้านหน้าสุด

Pin Tabs



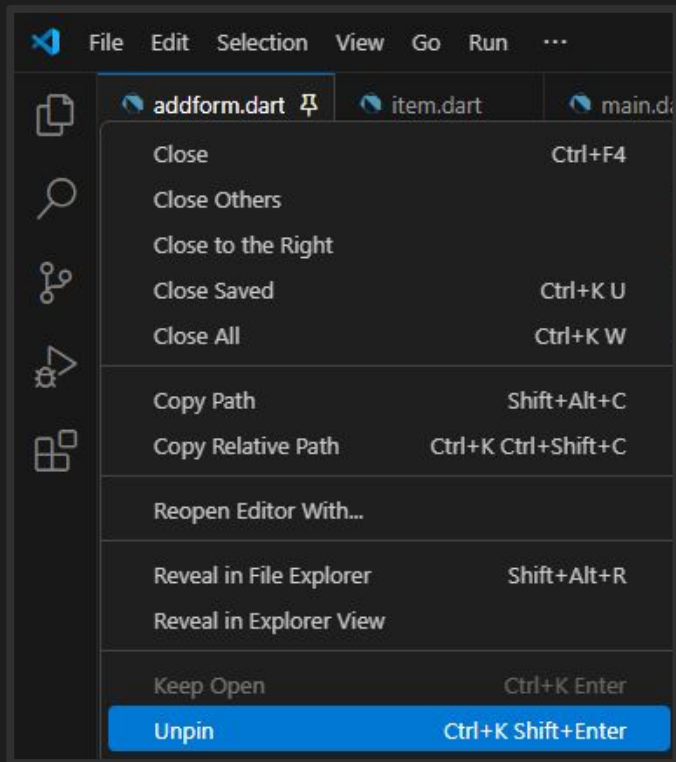
- การปักหมุด

คลิกขวาที่แท็บ -> Pin

- การยกเลิก

คลิกขวาที่แท็บ -> Unpin

Pin Tabs



- การปักหมุด

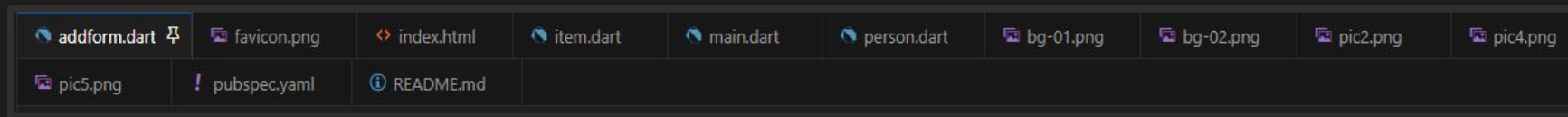
คลิกขวาที่แท็บ -> Pin

- การยกเลิก

คลิกขวาที่แท็บ -> Unpin

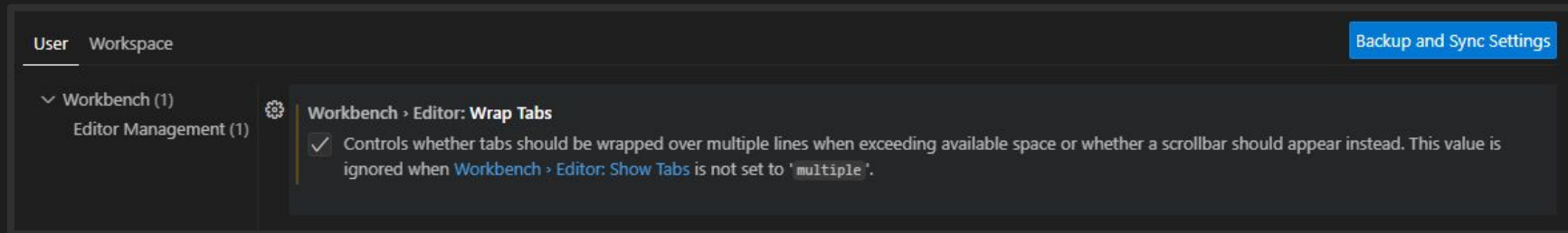
Wrap Tabs

สำหรับจัดการแท็บที่เกินพื้นที่ ให้ไปอยู่ในบรรทัดถัดไป
เพื่อให้แท็บการทำงานทั้งหมดอยู่ในระดับเดียวกัน



การตั้งค่า Wrap Tabs

- Manage -> Settings -> Wrap Tabs
- เปิดใช้งาน (Enabled)



Sticky Scroll

```
lib > models > person.dart
3  enum Job {
11  final Color color;
12  final String title;
13  final String image;
14  }
15
16  class Person {
17    Person({required this.name,
18    String name;
19    int age;
20    Job job;
21  }
```

การกำหนดให้ตรึงคำสั่งติดกับขอบ

Editor เมื่อมีการเลื่อนไปถึง

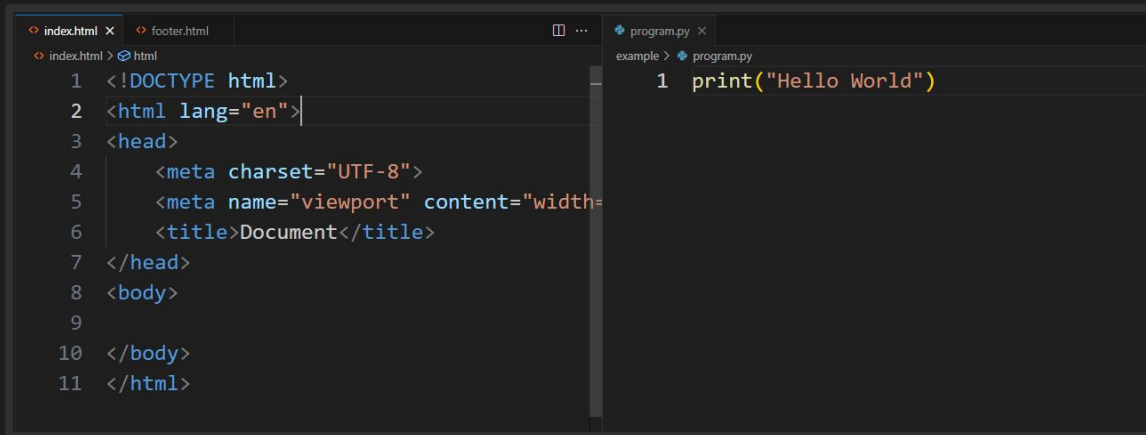
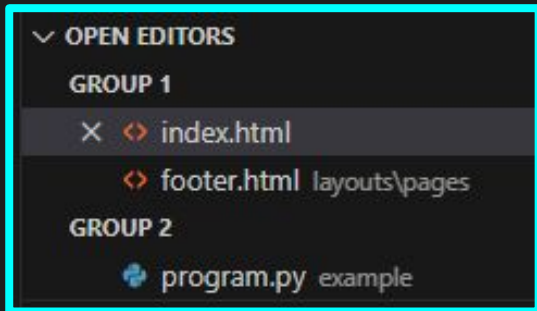
ใช้สำหรับบอกขอบเขตการทำงาน

การตั้งค่าเปิด-ปิด Sticky Scroll

```
lib > models > person.dart
3  enum Job {
11  final Color color;
12  final String title;
13  final String image;
14  }
15
16  class Person {
17    Person({required this.name,
18    String name;
19    int age;
20    Job job;
21  }
```

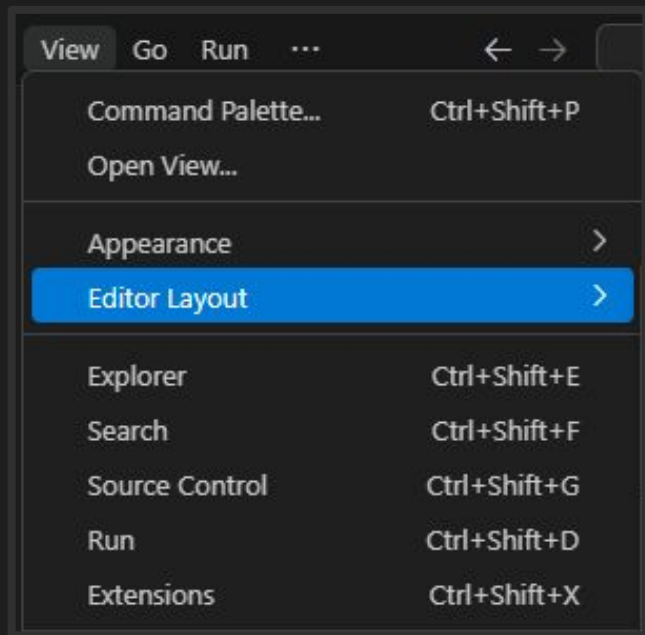
- Views -> Appearance
- เลือก Sticky Scroll

Editor Group



การแยกหน้าจอทำงานของ Editor ออกเป็นกลุ่มหรือ
หมวดหมู่เพื่อให้ง่ายต่อการบริหารจัดการ Editor

Editor Layout



การกำหนดเค้าโครงการแสดงผล

การทำงานของ Editor โดยแบ่ง

Editor ออกเป็นหลายๆส่วนตาม

รูปแบบมาตรฐานที่กำหนด

Editor Layout

Split Up	Ctrl+K Ctrl+\
Split Down	
Split Left	
Split Right	
Move Editor into New Window	
Copy Editor into New Window	Ctrl+K O
Single	
Two Columns	
Three Columns	
Two Rows	
Three Rows	
Grid (2x2)	
Two Rows Right	
Two Columns Bottom	
Flip Layout	Shift+Alt+O

การกำหนดเค้าโครงการแสดงผล

การทำงานของ Editor โดยแบ่ง

Editor ออกเป็นหลายๆส่วนตาม

รูปแบบมาตรฐานที่กำหนด

Sort Lines

เรียงลำดับข้อมูลที่อยู่ในไฟล์ผ่าน Command Palette

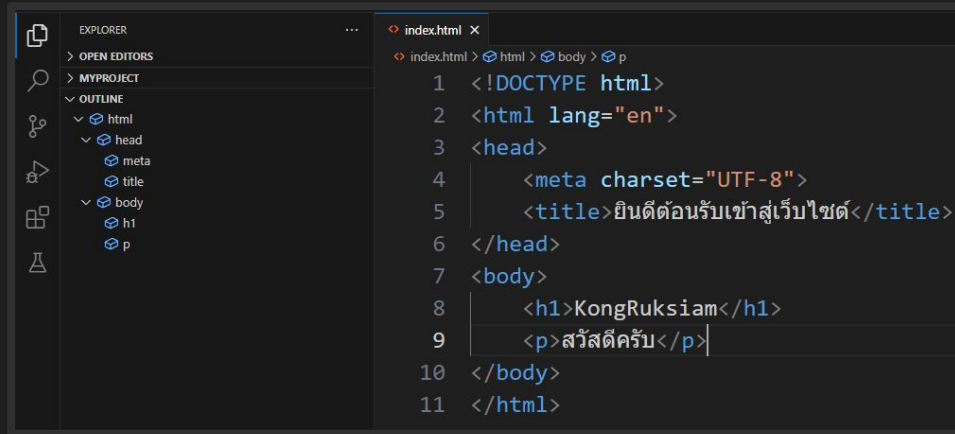
- Ascending (น้อยไปมาก)
- Descending (มากไปน้อย)

Transform

การเปลี่ยนรูปแบบตัวอักษรที่อยู่ในไฟล์ ตัวอย่าง เช่น

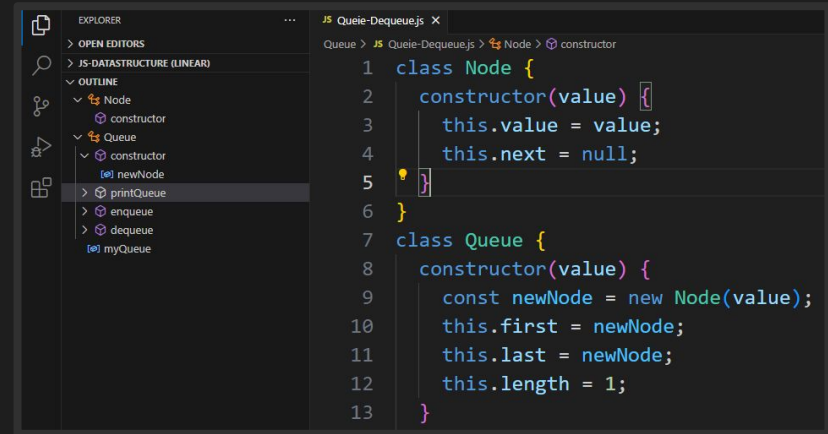
- Uppercase (ตัวพิมพ์ใหญ่)
- Lowercase (ตัวพิมพ์เล็ก)
- อื่นๆ

Outline และ Timeline



This screenshot shows the VS Code interface with the 'index.html' file open. The Explorer sidebar on the left shows the file structure. The Outline view on the right lists the DOM elements of the page, including the root 'html' element, 'head' (with 'meta' and 'title'), and 'body' (with 'h1' and 'p'). The main editor displays the HTML code corresponding to these elements.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>ยินดีต้อนรับเข้าสู่เว็บไซต์</title>
6 </head>
7 <body>
8   <h1>KongRuksiam</h1>
9   <p>สวัสดีครับ</p>
10 </body>
11 </html>
```

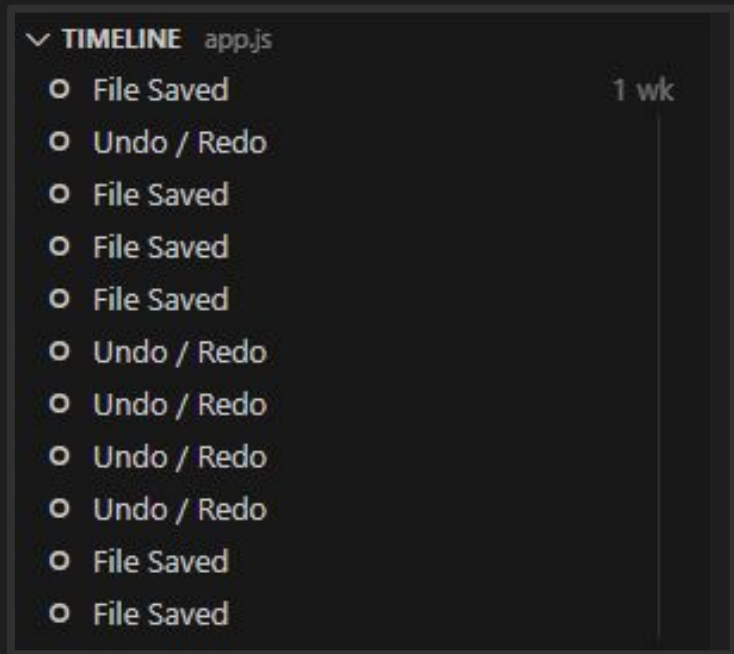


This screenshot shows the VS Code interface with the 'Queue-Dequeues.js' file open. The Explorer sidebar on the left shows the file structure. The Outline view on the right lists the classes and methods in the file, including 'Node' (with 'constructor' and 'newNode') and 'Queue' (with 'enqueue', 'dequeue', and 'myQueue'). The main editor displays the JavaScript code for these classes.

```
1 class Node {
2   constructor(value) {
3     this.value = value;
4     this.next = null;
5   }
6 }
7 class Queue {
8   constructor(value) {
9     const newNode = new Node(value);
10    this.first = newNode;
11    this.last = newNode;
12    this.length = 1;
13  }
```

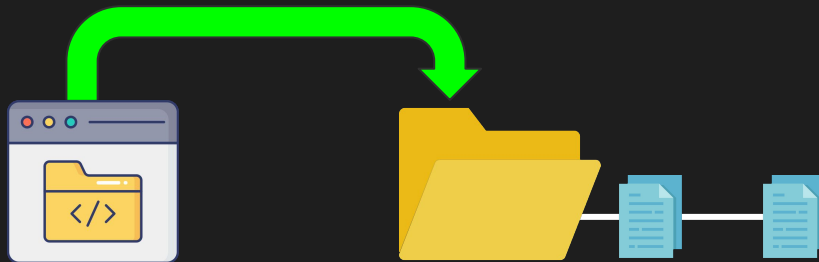
Outline คือ ส่วนที่แสดงภาพรวมและเข้าถึงโครงสร้างคำสั่งของ
ไฟล์ที่เปิดใน Editor ในลักษณะของลำดับชั้น

Outline และ Timeline



Timeline คือ ส่วนที่ช่วยติดตาม
ประวัติของไฟล์ , ขั้นตอนการทำงาน
ในไฟล์ที่เปิดใน Editor รวมถึง
เปรียบเทียบคำสั่งก่อน-หลังแก้ไข
เนื้อหาในไฟล์

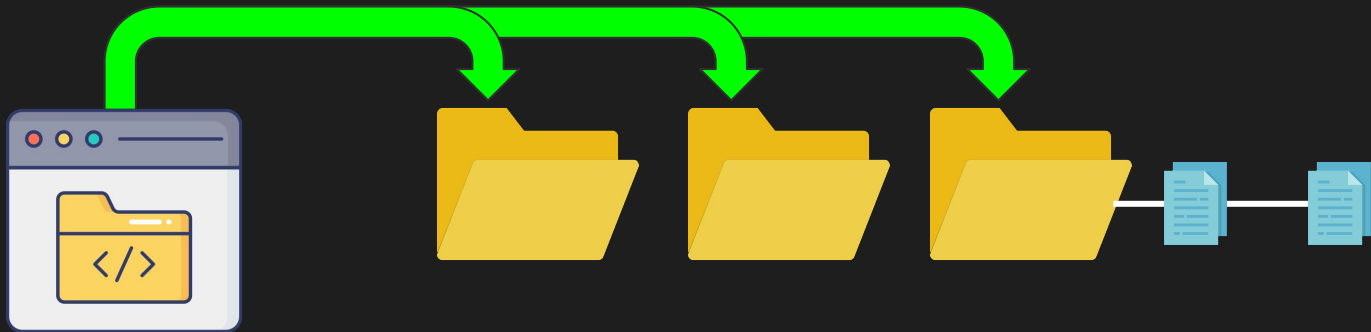
Workspaces



Single-Folder Workspaces

Workspaces คือ ส่วนที่ใช้บริหารจัดการโฟลเดอร์ที่จะนำมาทำงาน
ร่วมกันอย่างเต็มรูปแบบ

Workspaces



Multi-Root Workspaces

Workspaces คือ ส่วนที่ใช้บริหารจัดการโฟลเดอร์ที่จะนำมาทำงาน
ร่วมกันอย่างเต็มรูปแบบ

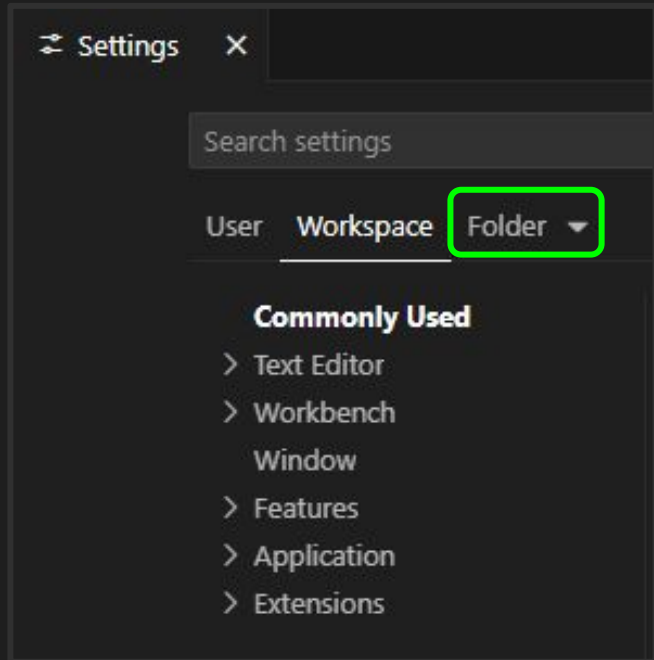
ขั้นตอนการสร้าง Workspace

- File -> Add Folder To Workspace
- File -> Save Workspace As...
- ตั้งชื่อ Workspace (.code-workspace)

โครงสร้างไฟล์ .code-workspace

```
{  
  "folders": [  
    {  
      "path": "ตำแหน่งโฟลเดอร์"  
    },  
  ]  
}
```

การปรับแต่ง Workspace



- Manage->Settings
- แท็บ Workspace
- เลือกโฟลเดอร์ใน Workspace
- ตั้งค่าหรือปรับแต่ง

(ข้อมูลการตั้งค่าจะอยู่ในโฟลเดอร์ .vscode)

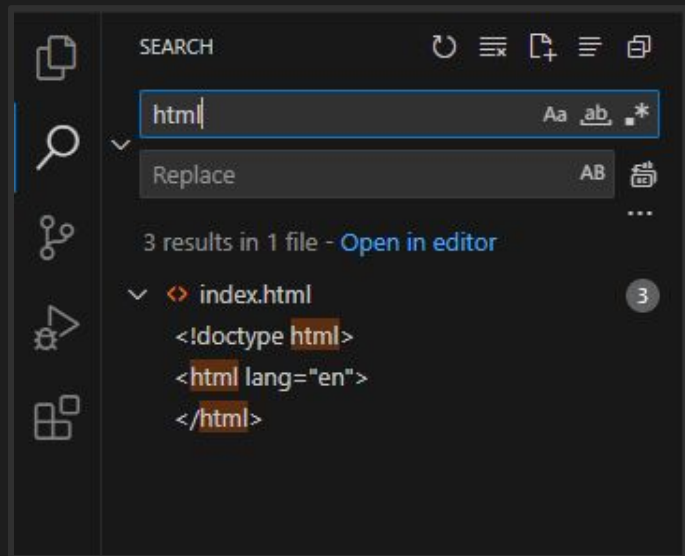
การเปิด Workspace

- File -> Open Workspace from File..
- เลือกไฟล์ Workspace (.code-workspace)

การปิด Workspace

- File -> Close Workspace

Search และ Replace



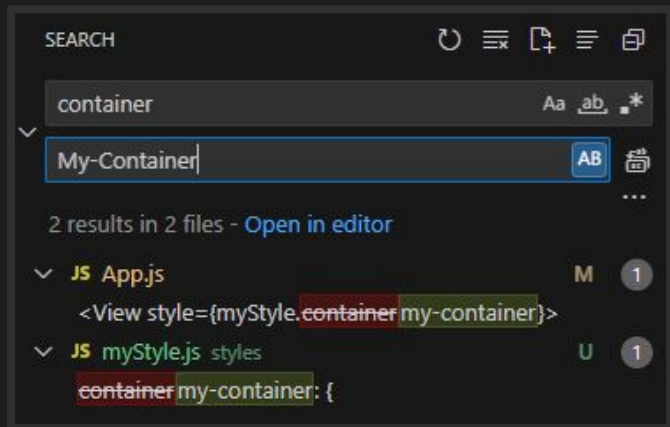
- **Search** ส่วนที่ใช้สำหรับค้นหาข้อมูลต่างๆที่กำลังเปิดใช้งานในโปรแกรม

รูปแบบการค้นหา

- **Match Case** ค้นหาโดยจำแนกตัวอักษรพิมพ์เล็ก - พิมพ์ใหญ่
- **Match Whole Word** ค้นหาข้อความที่เหมือนกันทุกอย่าง
- **Use Regular Expression** ค้นหาผ่านการกำหนด Pattern หรือ

รูปแบบกลุ่มคำ (ต้องมีพื้นฐาน Regex)

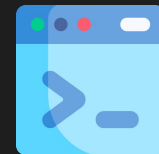
Search และ Replace



Replace ส่วนที่ใช้สำหรับการแทนที่
เนื้อหา ประกอบด้วย

- **Preserve Case** แทนที่และกำหนดเงื่อนไข
ตามรูปแบบตัวอักษรพิมพ์เล็ก-พิมพ์ใหญ่
- **Replace All** แทนที่ทั้งหมด

Terminal



- ในการเขียนโปรแกรมบางครั้งต้องอาศัยการเขียนคำสั่งผ่าน **Command Line** หรือ **Terminal** เพื่อดำเนินการต่างๆ ซึ่งอาจจะเป็นคำสั่งเฉพาะหรือคำสั่งที่เป็นของระบบปฏิบัติการ เช่น การจัดการไฟล์เดอร์ การติดตั้งไลบรารี รวมถึงการรันและทดสอบโปรแกรม ตัวอย่างคำสั่งพื้นฐาน เช่น
- **cd** ไปยัง directory ที่กำหนด , **cls** สำหรับเคลียร์หน้าจอ

ตัวอย่างคำสั่งของ VSCode

- `code --help` ดูรายละเอียดคำสั่งทั้งหมดใน Visual Studio Code
- `code --version` เรียกดูเวอร์ชันของ Visual Studio Code
- `code .` หมายถึง เปิด directory ปัจจุบัน
- `code ตำแหน่งไฟล์ที่ N` หมายถึง เปิดไฟล์ที่กำหนดตามตำแหน่งที่ระบุ

Terminal

การสร้าง Terminal

- Terminal -> New Terminal

การลบ Terminal

- กด Kill Terminal

แสดง Terminal

- View -> Terminal

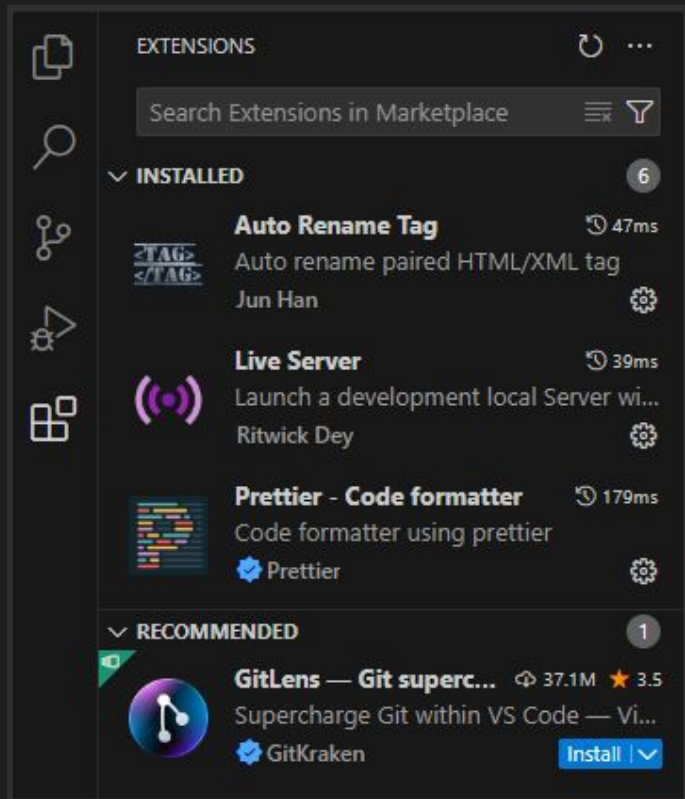
ซ่อน Terminal

- กดไอคอน x ที่ Terminal

การกำหนดรูปแบบ Terminal

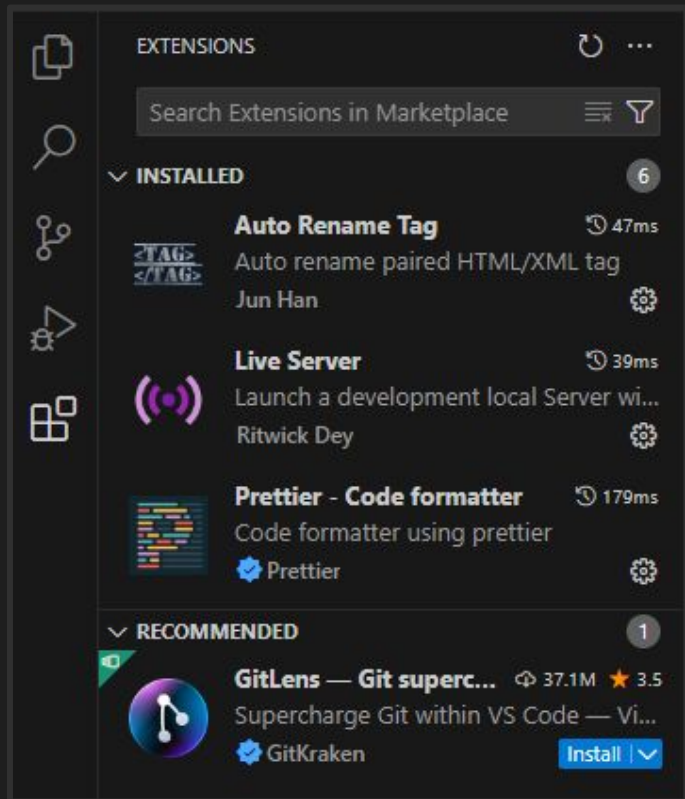
- View -> Terminal
- เมนู Launch Profile...
- เลือก Select Default Profile
- เลือกรูปแบบ Terminal ที่ต้องการ

Extension



- **Extension** ส่วนขยายหรือส่วนเสริมภายในโปรแกรมสำหรับเพิ่มประสิทธิภาพการทำงาน

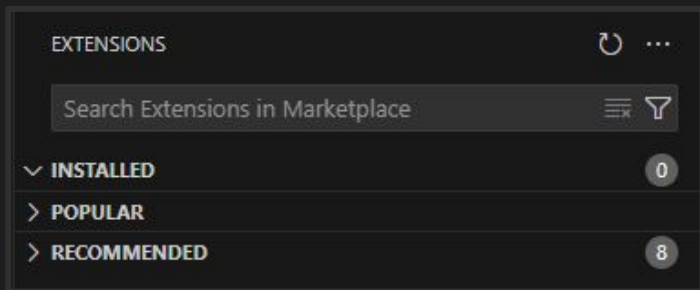
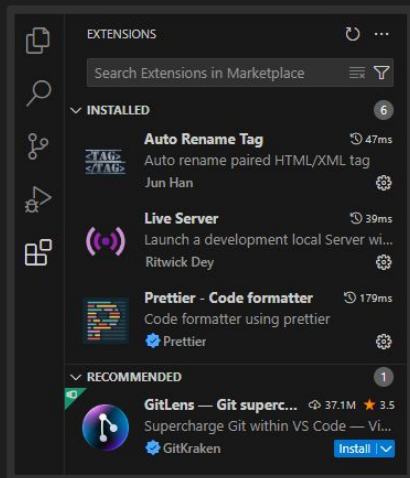
Extension



การแสดง Extension

1. ไอคอน Extension ใน Activity Bar
2. View -> Extension

Extension



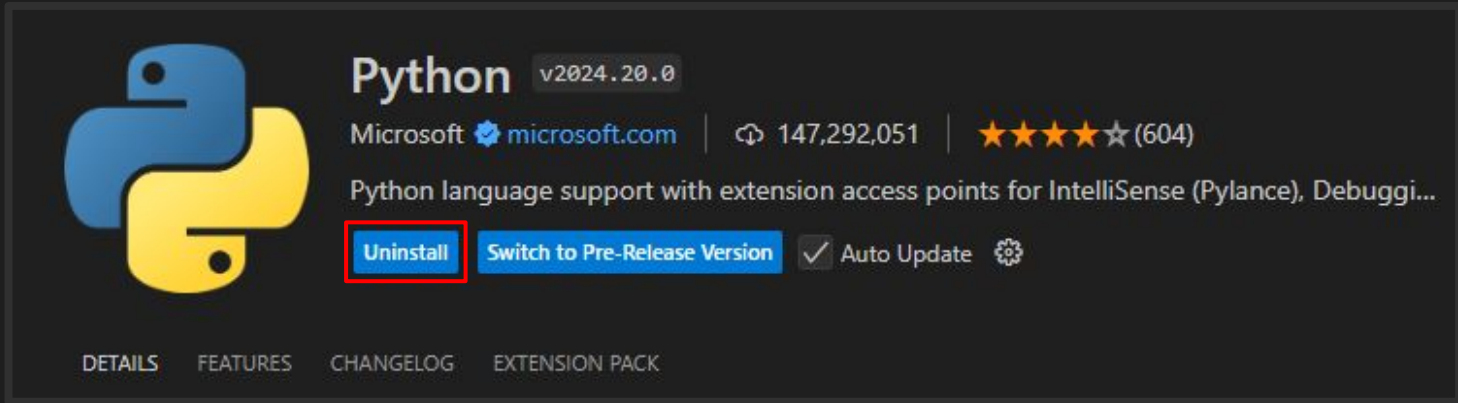
- **Search** ช่องค้นหา Extension ที่เราสนใจ
- **Installed** แสดงรายชื่อ Extension ที่ติดตั้งใน Visual Studio Code แล้ว
- **Popular & Recommended** แสดงรายชื่อ และแนะนำ Extension ที่ได้รับความนิยม

ติดตั้ง Extension



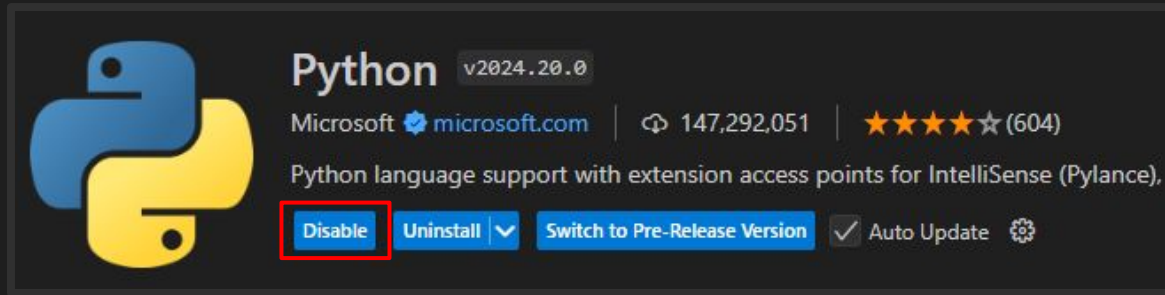
- เลือก Extension
- กดเมนู Install (ต้องเชื่อมต่ออินเทอร์เน็ต)

ถอนติดตั้ง Extension



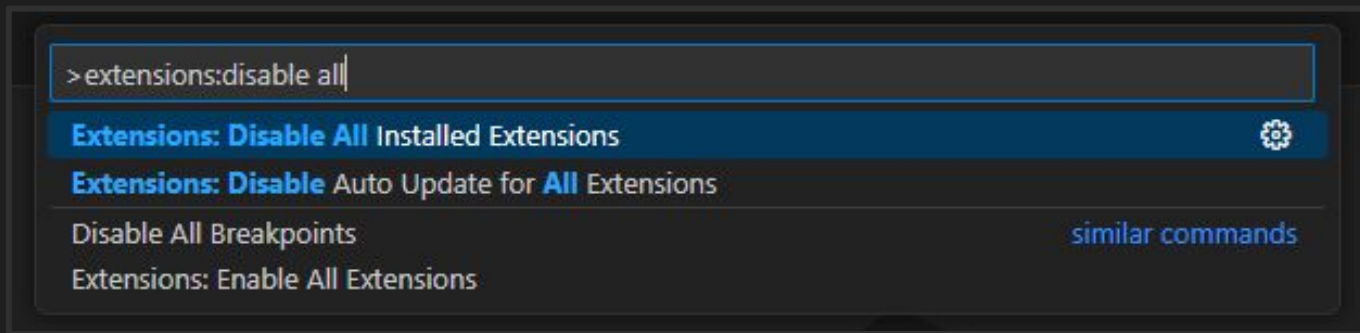
1. เลือก Extension -> Uninstall
2. เลือก Extension -> Manage -> Uninstall

ปิดการใช้งาน Extension



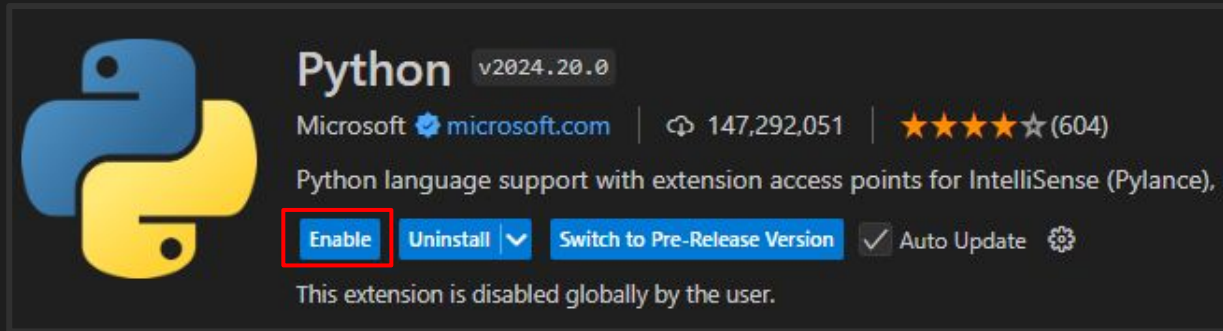
1. เลือก Extension -> Disable
2. เลือก Extension -> Manage -> Disable

ปิดการใช้งาน Extension (ทุกตัว)



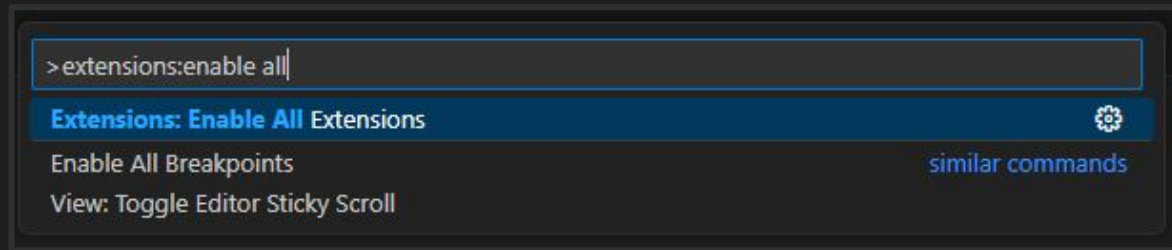
1. เปิด Command Palette
2. พิมพ์คำสั่ง Disable All Installed Extensions

เปิดการใช้งาน Extension



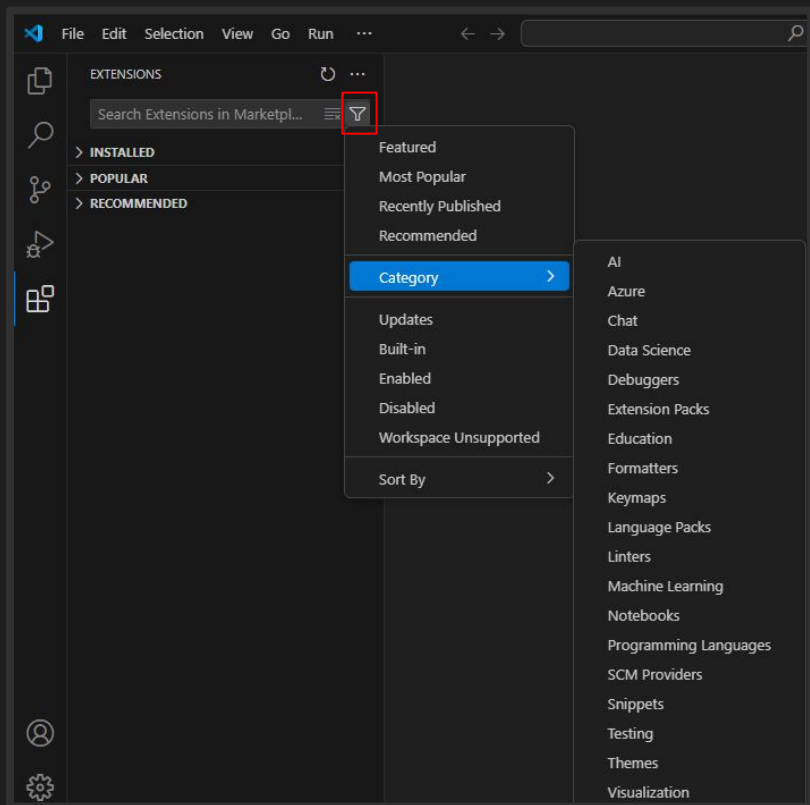
1. เลือก Extension -> Enable
2. เลือก Extension -> Manage -> Enable

เปิดการใช้งาน Extension (ทุกคน)



1. เปิด Command Palette
2. พิมพ์คำสั่ง Enable All Extensions

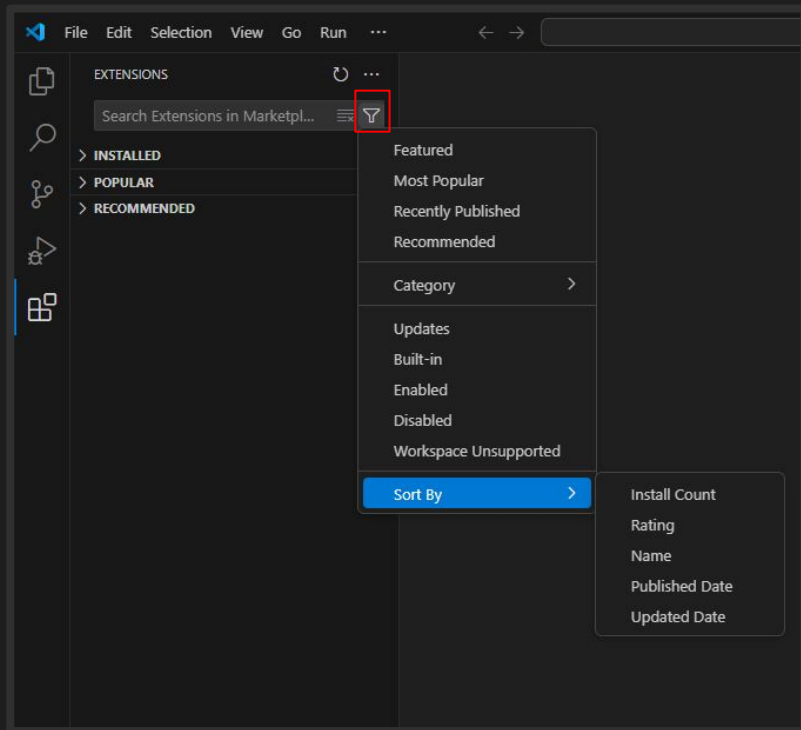
Filter Extension



Filter เป็นตัวกรอง Extension
ที่เกี่ยวข้องโดยจำแนกตามหมวดหมู่
ต่างๆ เช่น

- AI
- Programming
- Testing
- อื่นๆ

Sort Extension



Sort เป็นการจัดเรียง Extension
ตามรูปแบบที่กำหนด เช่น

- จำนวนการติดตั้ง
- เรตติ้ง
- ชื่อ
- อื่นๆ

Profiles

เป็นระบบที่อยู่ใน Visual Studio Code ใช้สำหรับรวบรวม
สิ่งต่างๆ เข้าไว้ด้วยกัน เพื่อให้เหมาะกับการจัดการโปรเจกต์
ที่ได้รับมอบหมายหรือการกำหนดสภาพแวดล้อมตามที่
นักพัฒนาต้องการ เช่น การตั้งค่า , การติดตั้ง Extension ต่างๆ

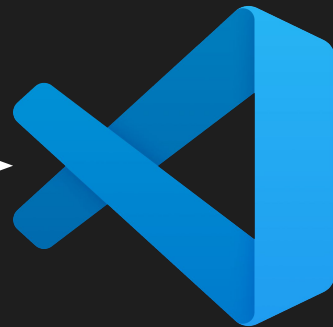
ปัญหาที่เกิดขึ้น

การพัฒนาระบบเราไม่ได้เขียนโปรแกรมแค่ภาษาเดียว เช่น

- ระบบ A ใช้ภาษา Python
- ระบบ B ใช้ภาษา JavaScript
- ระบบ C ใช้ภาษา Java



ปัญหาที่เกิดขึ้น



ปัญหาที่เกิดขึ้น

ส่งผลให้นักพัฒนาต้องตั้งค่าธรรมเนียมติดตั้ง Extension

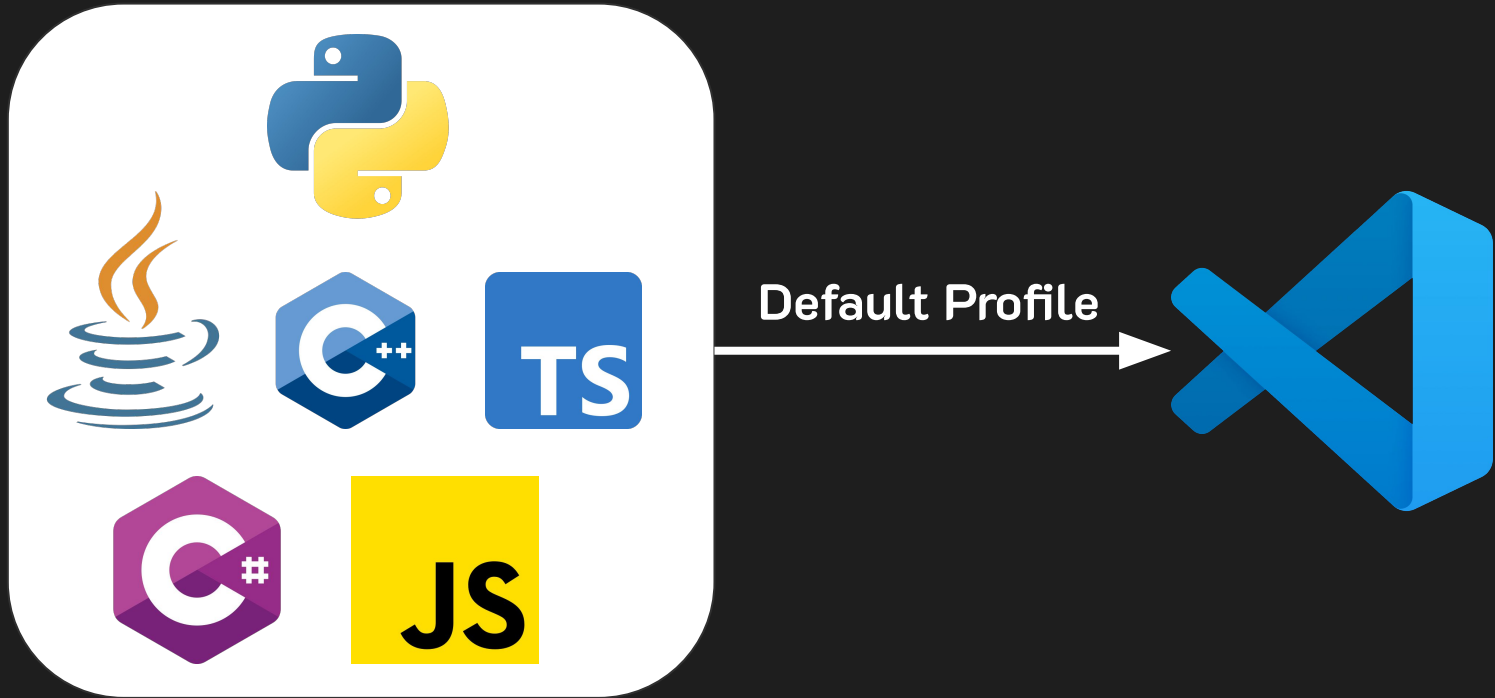
มากมายเพื่อให้สามารถพัฒนาระบบและเขียนโปรแกรม

ภาษาดังกล่าวได้ ทำให้เกิดความยุ่งยากในการบริหารจัดการ

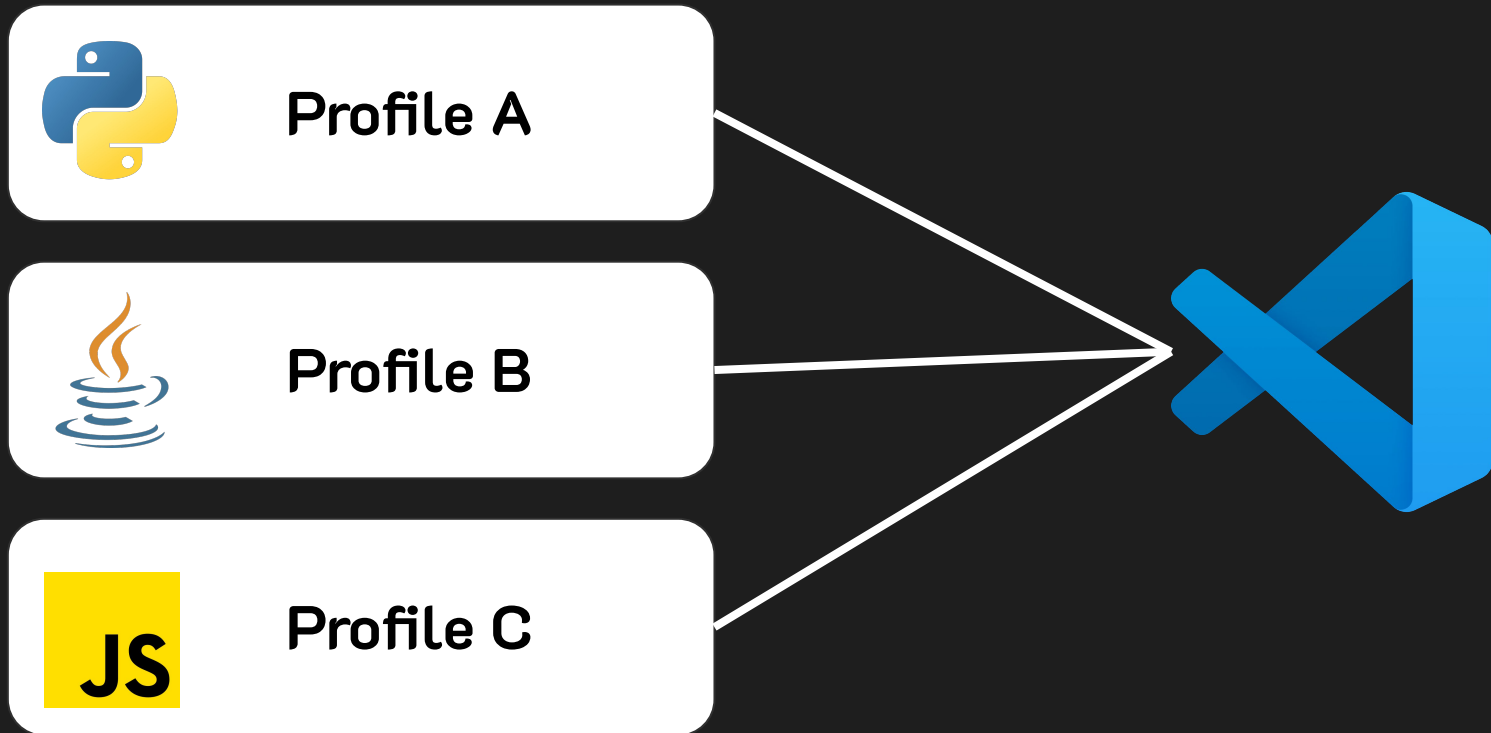
Profiles

ในการทำงาน Visual Studio Code เราสามารถแยกกลุ่มการตั้งค่าต่างๆ หรือติดตั้ง Extensions ให้เหมาะสมกับการเขียนโปรแกรมหรือการพัฒนาระบบตามสภาพแวดล้อมที่เราต้องการได้ผ่าน **ระบบ Profile** (โปรไฟล์เริ่มต้นคือ Default Profile)

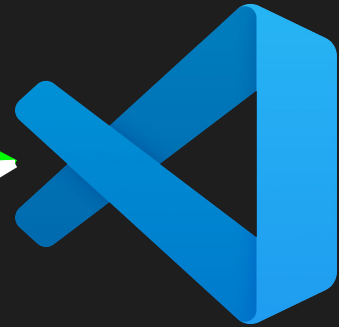
ใช้ระบบ Profiles



ใช้ระบบ Profiles

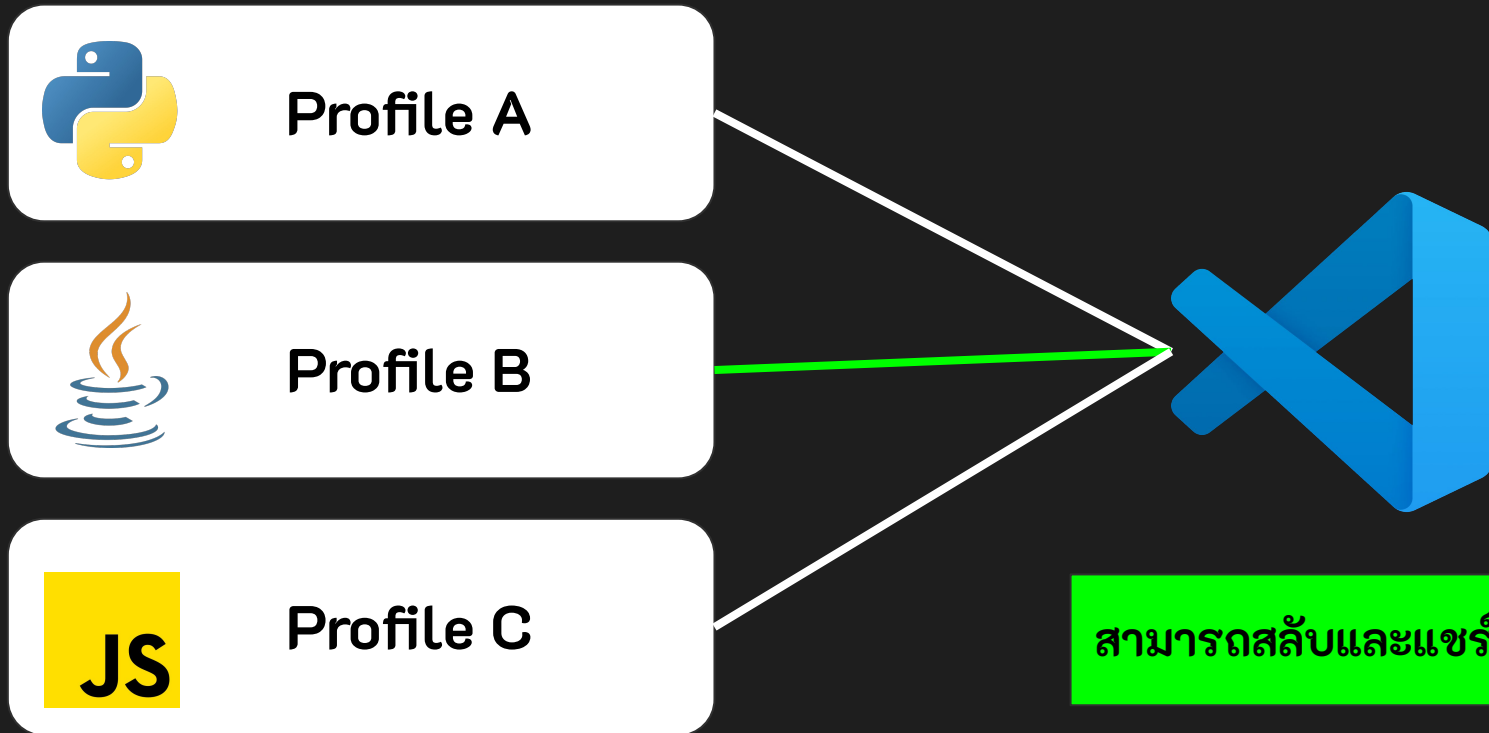


ใช้ระบบ Profiles



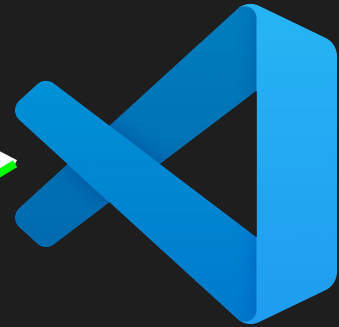
สามารถสลับและแชร์โปรไฟล์ได้

ใช้ระบบ Profiles



สามารถสลับและแชร์โปรไฟล์ได้

ใช้ระบบ Profiles



สามารถสลับและแชร์โปรไฟล์ได้

การสร้าง Profiles

- Settings -> Profiles -> New Profile
- ตั้งชื่อ Profile
- กำหนด Icons ให้ Profile
- ตั้งค่า Contents
- กด Preview เพื่อดูตัวอย่าง หรือ กด Create เพื่อสร้าง Profile

Profiles Contents

- **Setting** คือ การตั้งค่าต่างๆที่ใช้งานร่วมกับ Profile
- **Keyboard Shortcut** คือ การตั้งค่าคีย์ลัดที่ใช้งานร่วมกับ Profile
- **Tasks** คือ การตั้งค่า Task ที่ใช้งานร่วมกับ Profiles
- **Snippets** คือ การตั้งค่า Snippets (คำสั่งสำเร็จรูป) ที่ใช้งานร่วมกับ Profile
- **Extension** คือ ส่วนขยายที่ใช้งานร่วมกับ Profile

Profiles Contents

Contents Source

- **Default** อ้างอิง Content จาก Default Profile
- **None** ไม่กำหนด Content

การสลับ Profiles

1. File -> Preferences -> Profile -> เลือก Profile
2. Manage -> Profile -> เลือก Profile
3. Manage -> Profile -> Profiles -> เลือก Profile -> Use this Profile....
4. Command Palette -> พิมพ์คำสั่ง Switch Profile -> เลือก Profile

การส่งออก Profile (Export)

1. เลือก Profile
2. กด More Actions...
3. Export -> เลือกรูปแบบการส่งออก (Local , GitHub)

การนำเข้า Profile (Import)

1. Settings -> Profiles -> New Profile
2. กด More Actions...
3. Import Profile... -> ชื่อไฟล์.code-profile
4. กด Preview เพื่อดูตัวอย่าง หรือ กด Create เพื่อสร้าง Profile

การลบ Profile (Delete)

1. เลือก Profile
2. กด More Actions...
3. เลือกเมนู Delete
4. ยืนยันการลบ

Profile Template

Profile สำเร็จรูป ที่อยู่ใน Visual Studio Code (Built-In Profiles)

ซึ่งได้มีการจัดเตรียมเครื่องมือพื้นฐานพร้อมนำไปใช้งานได้เลย

โดยใน Template นั้นได้มีการกำหนดการตั้งค่าพื้นฐาน , Extension , Snippets ที่เหมาะสมกับงานหรือรูปแบบการเขียนโปรแกรมของแต่ละภาษาเรียบร้อยแล้ว เช่น Python , Java , Node.js เป็นต้น

การสร้าง Profile Template

1. Settings -> Profiles -> New Profile
2. กด More Actions...
3. From Template
4. เลือก Template ที่ต้องการ