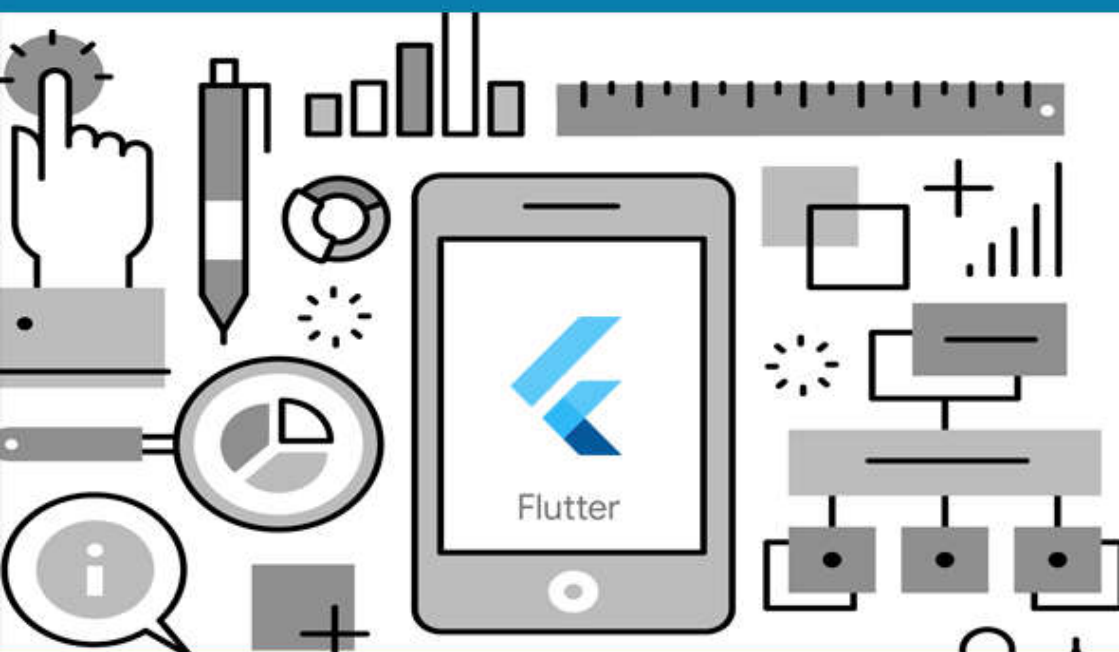


พัฒนา Mobile App ด้วย Flutter



โค้ชเอก

CodingThailand.com

พัฒนา Mobile App ด้วย Flutter

by โค้ชเอก

คู่มือเล่มนี้ผู้เขียนตั้งใจจัดทำขึ้นเพื่อใช้เป็นแนวทางในการเริ่มต้นใช้งาน Flutter ซึ่งจะเน้นเนื้อหาเกี่ยวกับการเตรียมสภาพแวดล้อมต่างๆ และการติดตั้ง Flutter หวังว่าคู่มือเล่มนี้จะเป็นประโยชน์ และประหยัดเวลาการเรียนรู้ของทุกคนครับ

“จงเอาชนะความไม่รู้ ด้วยการพัฒนาตัวเอง และลงมือทำอย่างสม่ำเสมอ”



©2020 CodingThailand.com

สารบัญ

Flutter พื้นฐาน 4

- Flutter คืออะไร
- ความรู้พื้นฐานที่ต้องมี
- ภาพใหญ่ของ Flutter กับการติดต่อกับฐานข้อมูล

การเตรียมตัวก่อนติดตั้ง Flutter 6

- ก่อนใช้งาน Flutter ต้องติดตั้งอะไรบ้าง
- ตรวจสอบโปรแกรมต่างๆ
- การติดตั้ง Android Studio
- ขั้นตอนการสร้าง Emulator

การติดตั้ง Flutter 18

- การติดตั้ง Flutter SDK

การใช้งาน Visual Studio Code 20

- การติดตั้ง Extensions สำหรับ Flutter

การสร้าง และรันโปรเจค

22

- การสร้างโปรเจคใหม่
- การรันโปรเจคเพื่อทดสอบ App
- การรันโปรเจคเพื่อทดสอบ App
- ขั้นตอนการรันโปรเจคโดยใช้ Emulator
- ขั้นตอนการรันโปรเจคบนเครื่องจริง (USB)
- ทดลองเขียน Flutter

Flutter พื้นฐาน

Flutter คืออะไร

Flutter เป็น UI toolkit สำหรับสร้าง User Interface โดยเฉพาะ สามารถ compiled โค้ดเป็น native ได้ทั้ง web, desktop และ mobile app ด้วยการเขียนโค้ดเพียงครั้งเดียว Flutter นั้น เขียนด้วยภาษา Dart สำหรับคนที่ต้องการศึกษา Flutter ก็ควรมีพื้นฐานการเขียน Dart มาก่อน

Note: เว็บไซต์หลักของ Flutter <https://flutter.dev/>

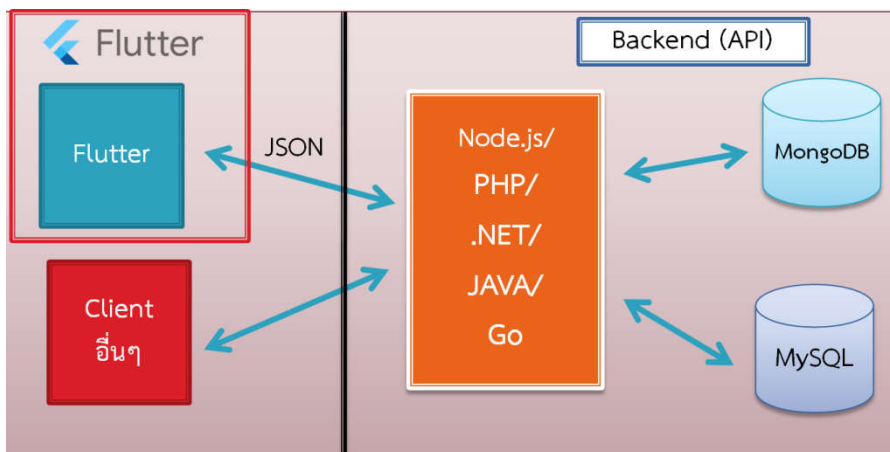
Note: เว็บไซต์หลักของ Dart <https://dart.dev/>

ความรู้พื้นฐานที่ต้องมี

ก่อนเขียน Flutter ควรมีพื้นฐานความรู้ ดังต่อไปนี้

1. หลักการการเขียนโปรแกรมเชิงวัตถุ (OOP) พื้นฐาน
2. การเขียนและใช้งาน Class
3. หลักการ และการเขียน Inheritance
4. หลักการ Encapsulation
5. หลักการ Polymorphism
6. ภาษา Dart พื้นฐาน

ภาพใหญ่ของ Flutter กับการติดต่อกับฐานข้อมูล



จากรูปด้านบนจะเห็นว่า Flutter หากเรามองว่าเป็นส่วนของ Frontend ก็ได้ (Mobile App) หากเราต้องการติดต่อกับฐานข้อมูลก็ต้องมีการเขียน Backend อาจจะเขียนในรูปแบบของ APIs ด้วยภาษาที่เราถนัดได้เลย ไม่ว่าจะเป็น PHP, Node.js, .NET หรือ Java เป็นต้น โดย Backend จะส่งข้อมูลมาในรูปแบบของ JSON ไปให้กับ Flutter นั่นเอง

การเตรียมตัวก่อนติดตั้ง Flutter

ก่อนใช้งาน Flutter ต้องติดตั้งอะไรบ้าง

ก่อนที่จะใช้งาน Flutter นั้น จะต้องติดตั้งโปรแกรมต่างๆ ดังนี้

1. เตรียม Notebook หรือ Mac สำหรับ Notebook แนะนำเป็น Windows 10 แบบ 64bit จะดีมาก (หรือถ้าไม่มีอย่างน้อยต้อง Windows 7 64bit SP1) Ram ต้องไม่น้อยกว่า 6 GB (แนะนำ 8 GB ขึ้นไป)
2. ติดตั้ง git ดาวน์โหลดได้ที่นี้ โดยคลิก next อย่างเดียวจนเสร็จเรียบร้อย
<https://git-scm.com/download>
3. ติดตั้ง Visual Studio Code คลิก next อย่างเดียวจนเสร็จเรียบร้อย ตามลิงก์
<https://code.visualstudio.com/>
4. ติดตั้ง JRE เวอร์ชัน 8 เท่านั้น คลิก next อย่างเดียวจนเสร็จเรียบร้อย
<https://www.oracle.com/java/technologies/javase-jre8-downloads.html>
5. ติดตั้ง Java JDK เวอร์ชัน 8 คลิก next อย่างเดียวจนเสร็จเรียบร้อย
<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
6. ติดตั้ง Android Studio (คลิก next อย่างเดียวจนเสร็จเรียบร้อย)
<https://developer.android.com/studio/>

- เปิด Virtualization Technology (Intel VT-x) ที่ BIOS (กด F2 หรือ Delete หรือ Esc เพื่อเข้า BIOS) มองหาเมนู หรือคำว่า "VT", "Virtualization Technology", หรือ "VT-d." แล้วเปิดใช้งาน จากนั้นบันทึก แล้ว restart เครื่องอีกครั้ง

ตรวจสอบโปรแกรมต่างๆ

หลังจากที่ติดตั้งโปรแกรมต่างๆ จนครบเรียบร้อยแล้ว ก็ลองทดสอบว่าโปรแกรมที่เราได้ติดตั้งนั้น เรียบร้อยดีหรือไม่ โดยให้เปิด **Command Prompt** ขึ้นมาแล้วพิมพ์ตรวจสอบโปรแกรม ดังนี้

```
C:\>java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)

C:\>git --version
git version 2.16.2.windows.1

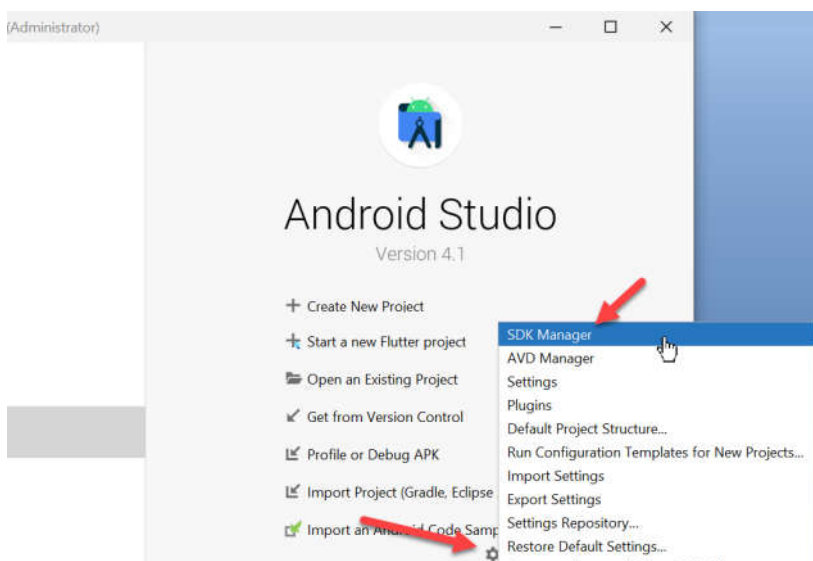
C:\>adb version
Android Debug Bridge version 1.0.40
Version 28.0.2-5303910
Installed as C:\Users\CodingThailand\AppData\Local\Android\Sdk\platform-tools\adb.exe
```

หากพิมพ์แล้วแสดงเวอร์ชันของโปรแกรม แสดงว่าการติดตั้งเรียบร้อยแล้ว

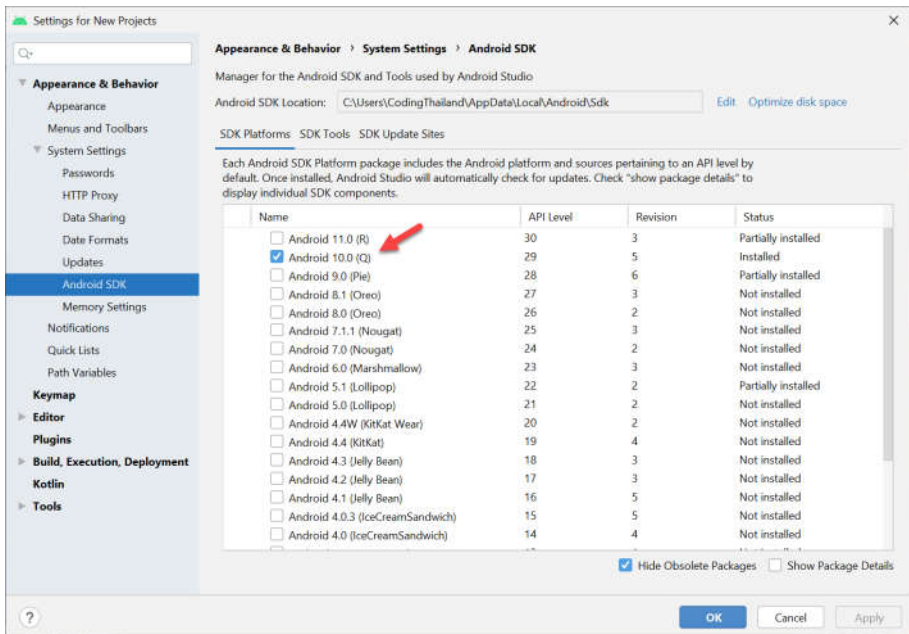
การตั้งค่า Android Studio

หลังจากติดตั้ง Android Studio เรียบร้อยแล้ว ให้ตั้งค่า และตรวจสอบการติดตั้ง SDK และ Tools ต่างๆ ดังนี้

1. เปิด Android Studio ขึ้นมา คลิกเลือกที่เมนู **Configure -> SDK Manager**

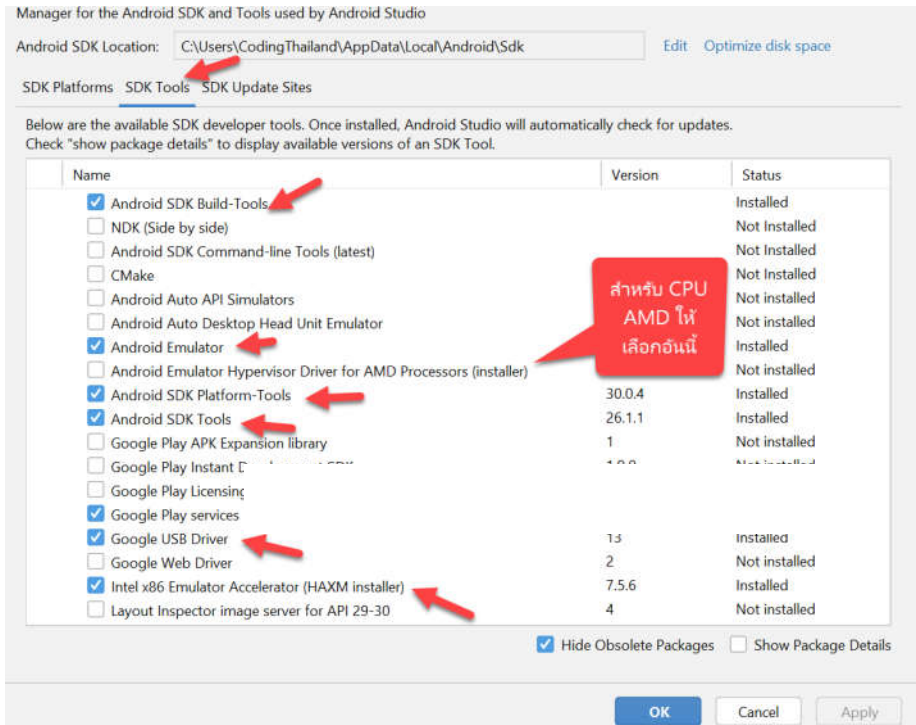


2. ตรวจสอบการติดตั้ง SDK Platforms โดยให้ดูว่ามีการเลือกที่ **Android 10 (Q) API Level 29** แล้วหรือไม่ หากยังไม่มี ก็ให้คลิกเลือกแล้วกดปุ่ม **Apply** เพื่อติดตั้ง



Note: ในหัวข้อนี้ ในอนาคตสามารถขยับเวอร์ชันขึ้นไปอีกได้อีกในอนาคต (ใช้เวอร์ชันล่าสุดเสมอ)

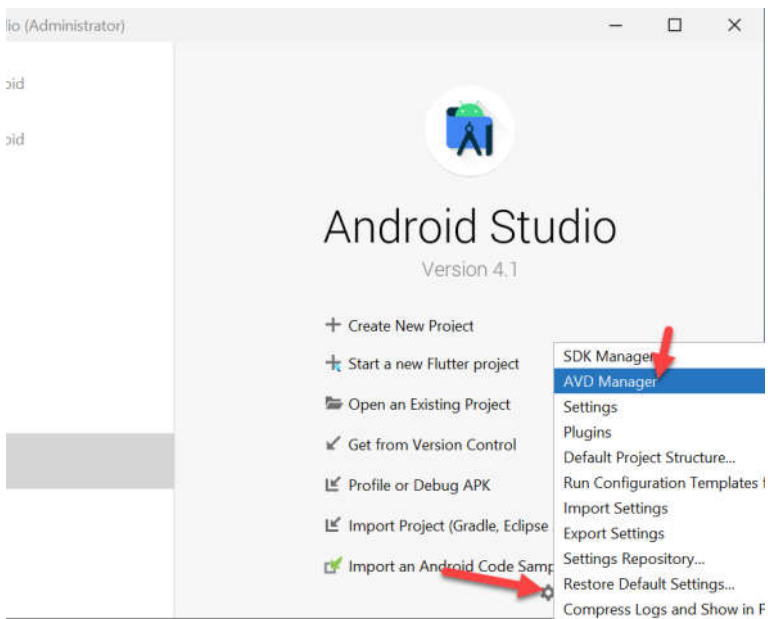
3. คลิกที่ Tab SDK Tools ให้ตรวจสอบว่ามีการเลือกที่ Android SDK Build-Tools แล้วหรือไม่ และคลิกเลือกทั้งหมด ดังรูป จากนั้นกดปุ่ม Apply เพื่อติดตั้ง



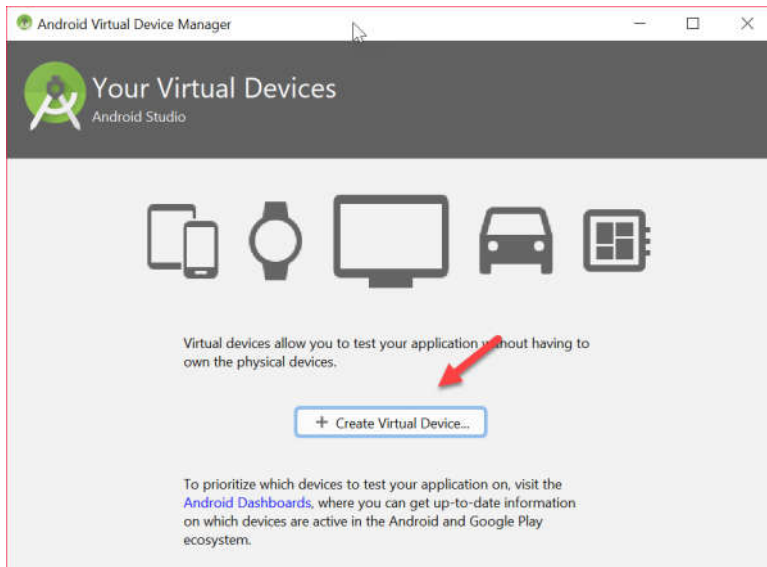
ขั้นตอนการสร้าง Emulator

เราสามารถทดสอบ Mobile App ที่เขียนได้โดยใช้เครื่องจริง (ผ่านสาย USB) หรือจะใช้ Emulator ก็ได้เช่นเดียวกัน ในหัวข้อนี้เราจะมาเรียนรู้ขั้นตอนการสร้าง Emulator ดังนี้

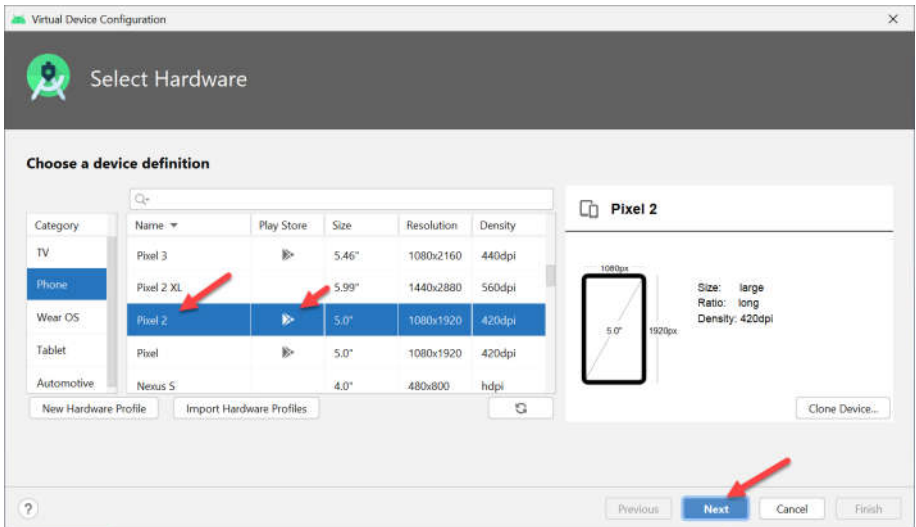
1. เปิด Android Studio ขึ้นมา คลิกเลือกที่เมนู Configure -> AVD Manager



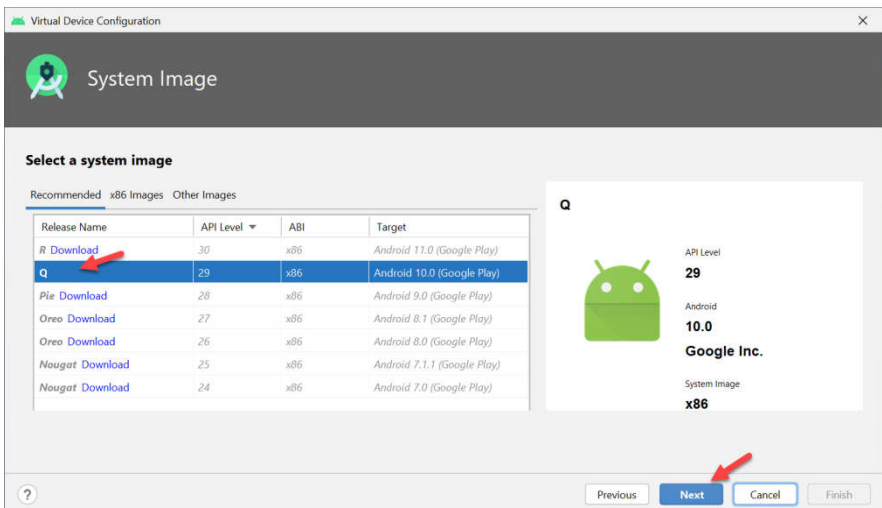
2. คลิกที่ปุ่ม Create Virtual Device...



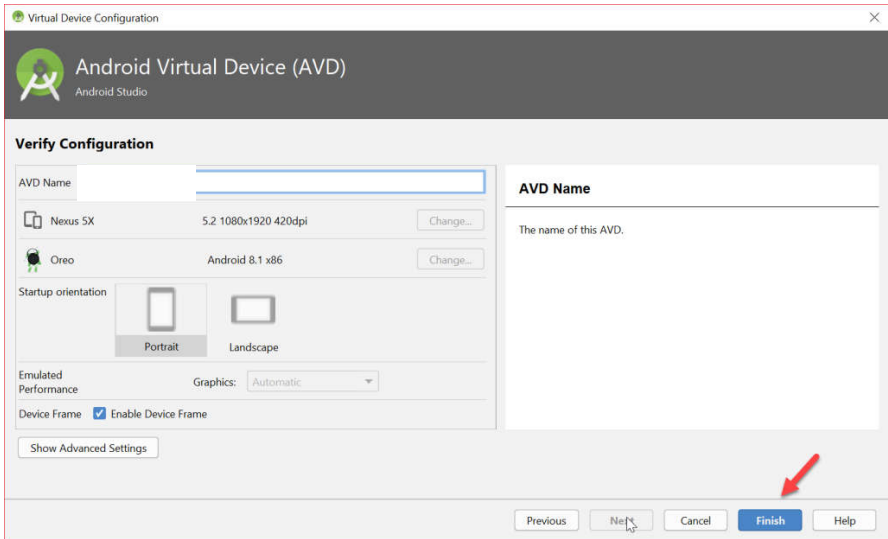
3. คลิกเลือก Pixel 2 (หรือรุ่นอื่นๆ แต่ต้องมีไอคอนที่คล้ายกับ Play Store) แล้วคลิกปุ่ม Next



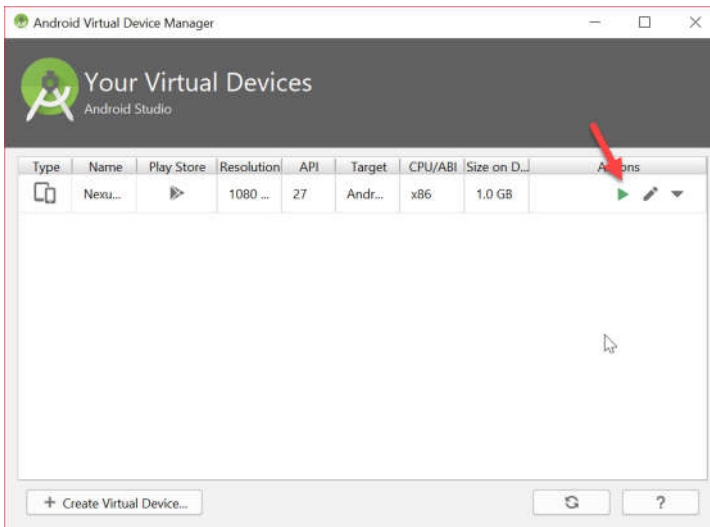
4. คลิกเลือก Q (API Level 29) หรือรุ่นใหม่กว่า (ก่อนเลือกให้คลิกปุ่ม Download ก่อน กรณียังไม่ได้ดาวน์โหลด) จากนั้นคลิกปุ่ม Next (ในขั้นตอนนี้หากอยากเลือกล่าสุดก็ได้ เช่นเดียวกัน)



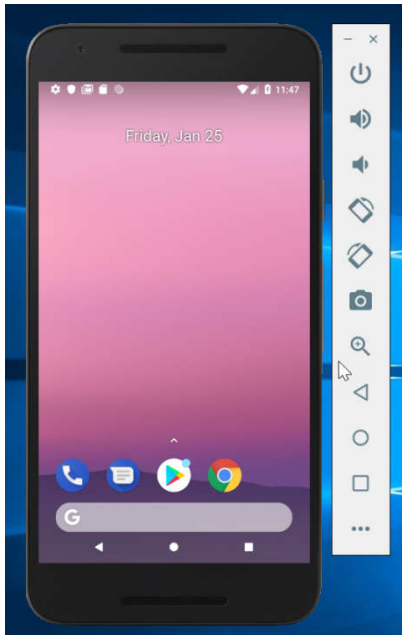
5. คลิกปุ่ม Finish



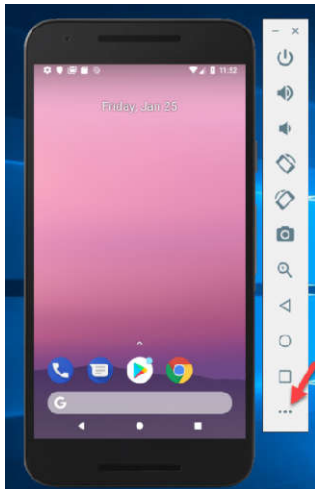
6. คลิกปุ่มสีเขียว เพื่อรัน Emulator ดังรูป



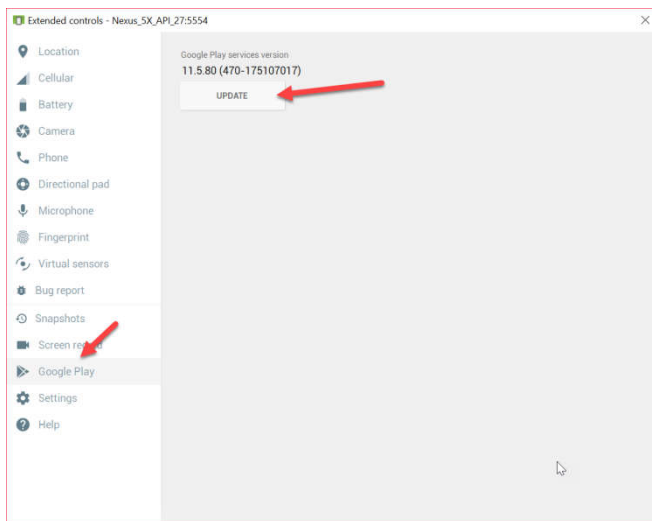
7. เมื่อกดรันแล้วจะได้ Emulator ดังรูป เป็นอันเสร็จเรียบร้อยแล้ว



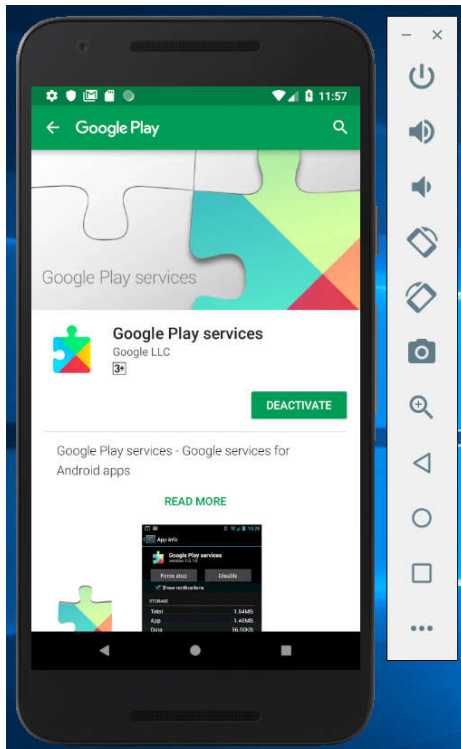
8. หลังจากได้ Emulator มาแล้ว **ควรอัปเดต Google Play Services** ให้ล่าสุดด้วย เพื่อในอนาคตเราต้องใช้ Google Maps API, Location APIs เป็นต้น ให้คลิกที่เมนู ดังรูป



9. คลิกที่เมนู Google Play (ด้านซ้าย) แล้วคลิกที่ปุ่ม UPDATE จากนั้นให้ปิดหน้าต่างนี้ เพื่อกลับไปยัง Emulator



10. ให้ ล็อกอินเข้าระบบของ Google Play รอสักครู่ Google Play Services จะได้รับการอัปเดตเป็นเวอร์ชันล่าสุดเรียบร้อยแล้ว



Note: เพื่อความแน่ใจสามารถไปเช็คเลขเวอร์ชันได้อีกครั้งว่าได้รับการอัปเดตหรือไม่

การติดตั้ง Flutter

1. ดาวน์โหลด Flutter SDK ล่าสุดจากลิงก์นี้

<https://flutter.dev/docs/development/tools/sdk/releases>

2. สำหรับ Windows ให้แตกไฟล์ zip ไว้ที่โฟลเดอร์ `C:\src\flutter` (หรือโฟลเดอร์อื่นที่ต้องการ) ส่วนสำหรับ Mac สามารถไว้ที่ path ไหนก็ได้

3. ต้อง กำหนด path ให้กับระบบปฏิบัติการเพื่อชี้ไปยังโฟลเดอร์ bin (อยู่ในโฟลเดอร์ flutter ในข้อ 2)

สำหรับ Windows ดูได้ที่ <https://flutter.dev/docs/get-started/install/windows#update-your-path>

สำหรับ Mac ดูได้ที่ <https://flutter.dev/docs/get-started/install/macos#update-your-path>

4. ปิด Command Prompt แล้วเปิดใหม่ แล้วทดลองพิมพ์คำสั่งเพื่อตรวจสอบเวอร์ชัน

```
C:\>flutter --version
```



5. เปิด Command Prompt พิมพ์คำสั่ง `flutter doctor` เพื่อตรวจสอบสภาพแวดล้อมการพัฒนาว่าพร้อมหรือไม่ ดังรูป

```

Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, v1.12.13+hotfix.5, on Microsoft
    [Version 10.0.18363.592], locale en-US)

[✓] Android toolchain - develop for Android devices (Android
    version 29.0.2)
[✓] Android Studio (version 3.5)
[✓] VS Code, 64-bit edition (version 1.41.1)
[!] Connected device
    ! No devices available

! Doctor found issues in 1 category.

```

Note: หากมีข้อไหนไม่มีเครื่องหมายถูกสีเขียว ให้แก้ไขให้เรียบร้อย

6. หากถึงขั้นตอนนี้แสดงว่าพร้อมใช้งาน Flutter เรียบร้อยแล้ว!

Note: หากต้องการอัปเดต Flutter SDK ให้ใช้ คำสั่ง **flutter upgrade**

Note: หากต้องการดาวน์โหลด Flutter SDK ให้ใช้

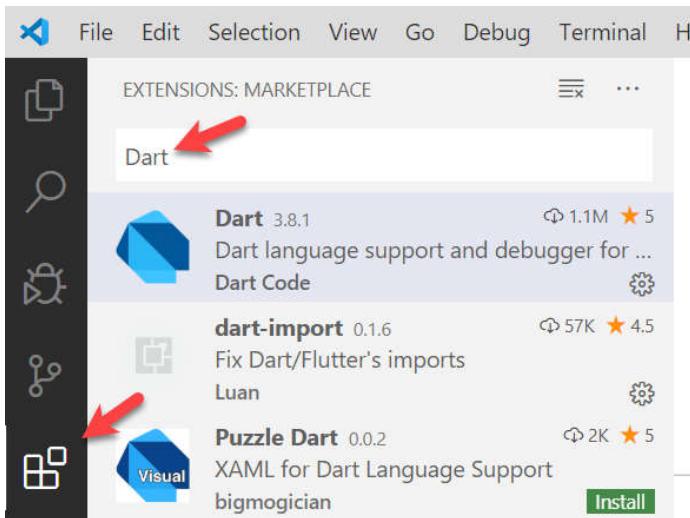
คำสั่ง **flutter version <ชื่อเวอร์ชัน>** เช่น flutter version v1.12.13+hotfix.5

การใช้งาน Visual Studio Code

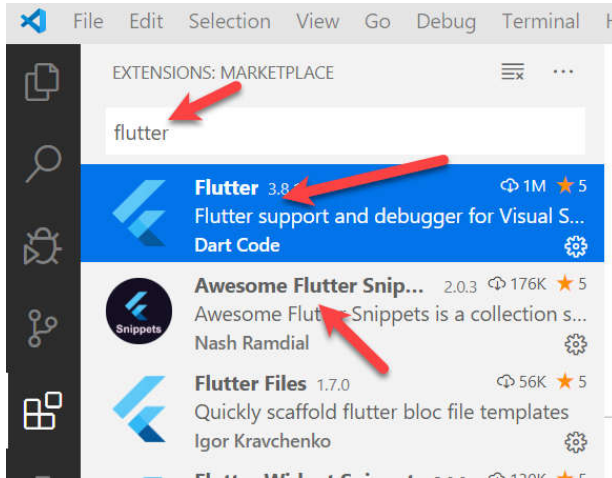
ในหัวข้อนี้เราจะมิตติดตั้ง Extensions เพื่อช่วยให้การเขียน Flutter ได้ง่ายขึ้น

การติดตั้ง Extensions สำหรับ Flutter

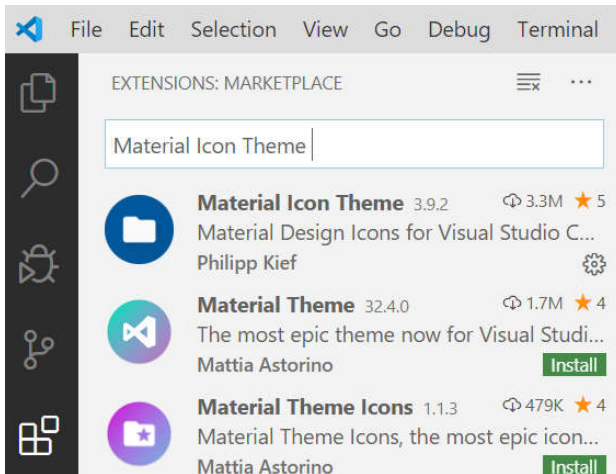
1. เปิดโปรแกรม Visual Studio Code คลิกที่ไอคอน Extensions เพื่อติดตั้ง **Dart** จากนั้นคลิกปุ่ม install



2. พิมพ์คำว่า flutter เพื่อค้นหา จากนั้นติดตั้ง **Flutter** และ **Awesome Flutter Snippet** ดังรูป




3. สุดท้าย ค้นคำว่า **Material Icon Theme** เพื่อติดตั้ง ดังรูป



การสร้าง และรันโปรเจค

การสร้างโปรเจคใหม่

1. สร้างโปรเจคใหม่ ด้วย คำสั่ง `flutter create --org <ชื่อโดเมน> <ชื่อโปรเจค>`
แล้วกด Enter

 Administrator: Command Prompt

```
C:\>flutter create --org com.codingthailand flutterapp■
```














Note: ก่อนเปิด Command Prompt ให้เลือก Run as administrator ด้วยทุกครั้ง

Note: สำหรับเครื่อง Mac ต้องใส่ sudo ข้างหน้าคำสั่งด้วยหากมีปัญหาเรื่อง permission

Note: แนะนำให้ใส่ชื่อโดเมนทุกครั้ง หากแก้ไขทีหลังจะทำได้ยาก

2. เมื่อติดตั้งเสร็จแล้วจะได้โฟลเดอร์ใหม่เป็นชื่อโปรเจคของเรา และไฟล์โค้ดต่างๆ จะถูกเก็บอยู่ในโฟลเดอร์นี้ ดังรูป

Name ^

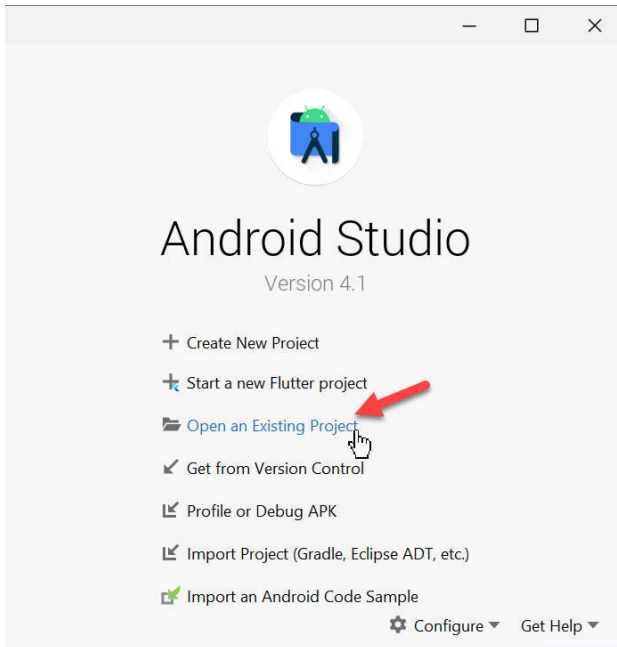
-  .dart_tool
-  .idea
-  android
-  ios
-  lib
-  test
-  .gitignore
-  .metadata
-  .packages
-  flutterbkk.iml
-  pubspec.lock
-  pubspec.yaml
-  README.md

การรันโปรเจคเพื่อทดสอบ App

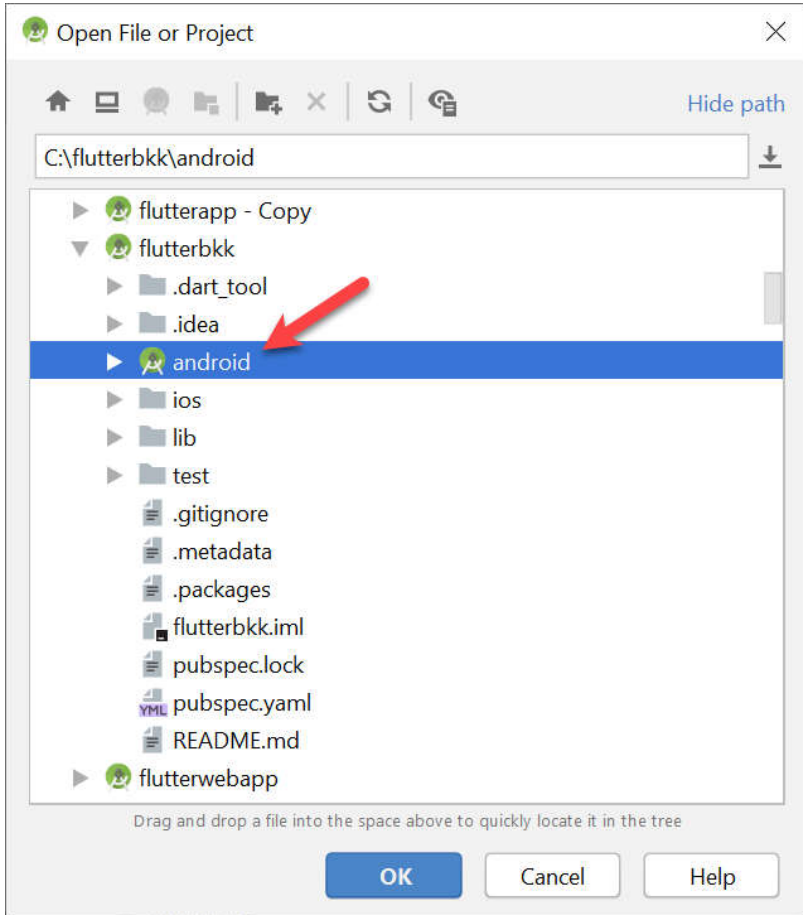
การทดสอบ App นั้นทำได้ 2 วิธีหลักๆ ด้วยกัน ได้แก่ การรันผ่าน Emulator และ การรันโดยใช้เครื่องจริง (USB)

ก่อนที่จะส่งรันโปรเจคแนะนำให้เปิด Android Studio ก่อน เพื่อ build app ให้เรียบร้อย มีขั้นตอน ดังนี้

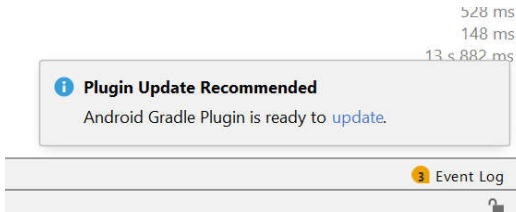
1. เปิดโปรแกรม Android Studio ขึ้นมา คลิกเลือก ที่ **Open and Existing Project** ดังรูป



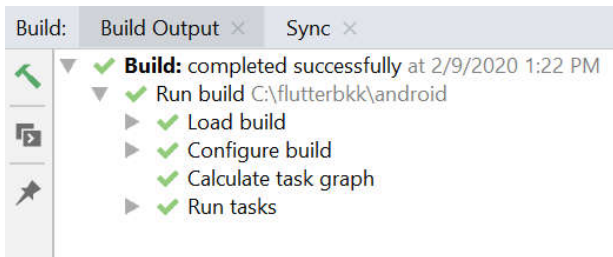
2. เลือกโฟลเดอร์โปรเจกต์ แล้วเลือกที่โฟลเดอร์ **android** จากนั้นคลิกปุ่ม OK เพื่อเปิดโปรเจกต์ขึ้นมา ดังรูป



3. หากมี popup ให้อัปเดต Android Gradle Plugin ให้คลิกปิดได้เลย ไม่ต้องอัปเดต

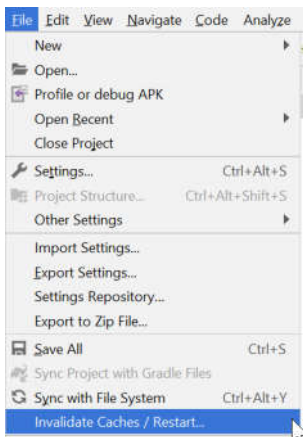


4. รอสักครู่ โปรเจค Android ของเราก็จะถูก build เรียบร้อย ดังรูป



Note: หากมี error แนะนำให้คลิกขวาแล้วเลือก run as administrator เพื่อเปิด

Android Studio อีกครั้งหนึ่ง หรือเลือกที่เมนู File -> Invalidate Caches / Restart...



ขั้นตอนการรันโปรเจกต์โดยใช้ Emulator

1. เปิด Command Prompt ใช้คำสั่ง `cd` เพื่อเข้าไปในโฟลเดอร์โปรเจกต์ ดังรูป

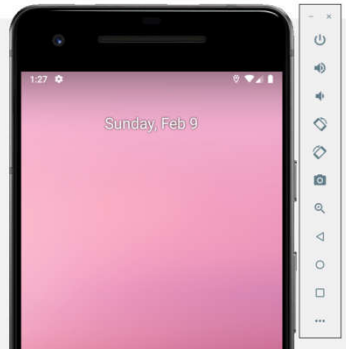
Administrator: Command Prompt

```
C:\>cd flutterbkk  
C:\flutterbkk>
```

2. เปิด Emulator ค้างไว้ จากนั้นใช้คำสั่ง `flutter run` แล้วกด enter

Administrator: Command Prompt

```
C:\flutterbkk>flutter run
```



3. รอสักครู่ โปรเจกต์จะถูกนำไปรันใน Emulator เรียบร้อย ดังรูป

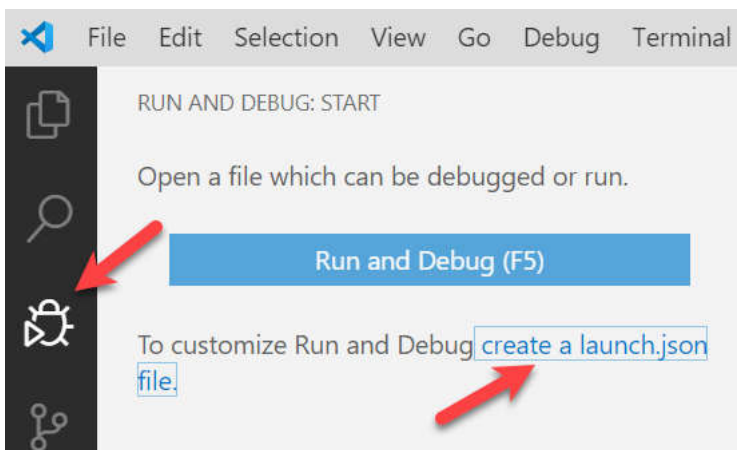


การรันผ่าน Command Prompt อาจจะไม่สะดวกในการพัฒนา แนะนำให้รันผ่าน

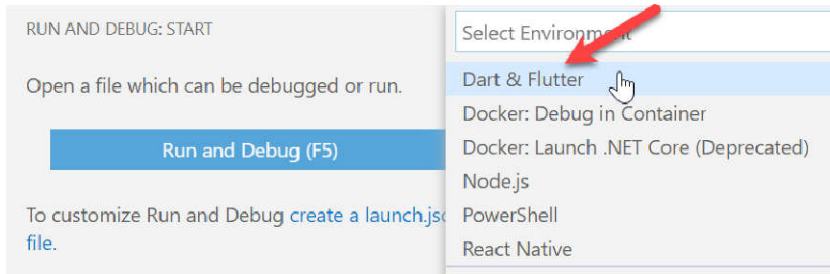
Visual Studio Code แทน ดังนี้

3.1 เปิดโฟลเดอร์โปรเจกต์ใน Visual Studio Code ก่อน

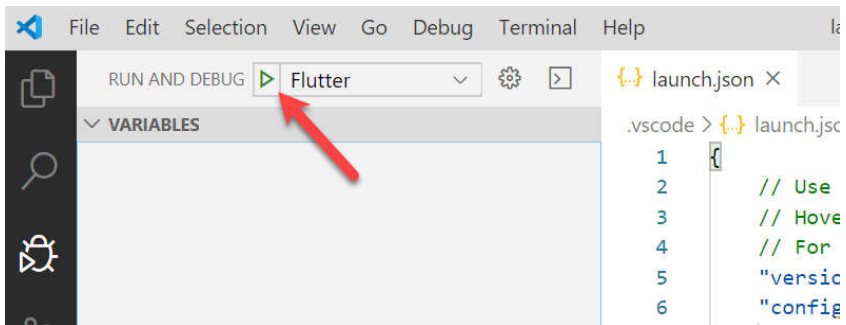
3.2 คลิกที่ไอคอน Run and Debug แล้วคลิกที่ **create a launch.json file**
(ทำครั้งแรกแค่ครั้งเดียว)



3.3 คลิกเลือก Dart & Flutter (ทำครั้งแรกแค่ครั้งเดียว)



3.4 คลิกที่ปุ่ม Start Debugging (สีเขียว) ดังรูป ต่อไปสามารถคลิกปุ่มนี้ได้เลย



ขั้นตอนการรันโปรแกรมบนเครื่องจริง (USB)

การรันบนเครื่องจริงจะต้องเตรียมสาย usb เพื่อต่อเข้ากับคอมพิวเตอร์เสียก่อน จากนั้นทำขั้นตอนดังต่อไปนี้

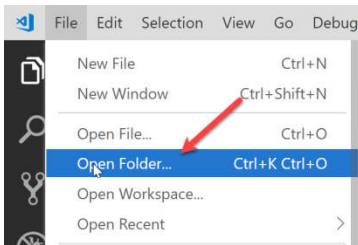
1. ที่โทรศัพท์เครื่องจริง ให้เปิด Debugging over USB (Developer options) โดยเข้าไปที่เมนู Settings -> About phone -> กดที่ Build number 7 ครั้ง จากนั้นกลับไปเมนู Settings -> Developer Options ให้ติ๊กเลือกที่ USB debugging
2. เมื่อเสียบสายโทรศัพท์ให้เรียบร้อย สามารถตรวจสอบว่าโทรศัพท์ได้เชื่อมต่อแล้วหรือไม่ ด้วยคำสั่ง adb devices หรือจะเปิด Android Studio แล้วกดรันตรวจการเชื่อมต่อก็ได้เช่นเดียวกัน (จะเห็นชื่ออุปกรณ์แสดงขึ้นมา)
3. จากนั้น ให้ส่งรันตามปกติ ก็เป็นอันเสร็จเรียบร้อยแล้ว

Note: ควรปิด Emulator ก่อน หากต้องการรันบนเครื่องจริง (USB)

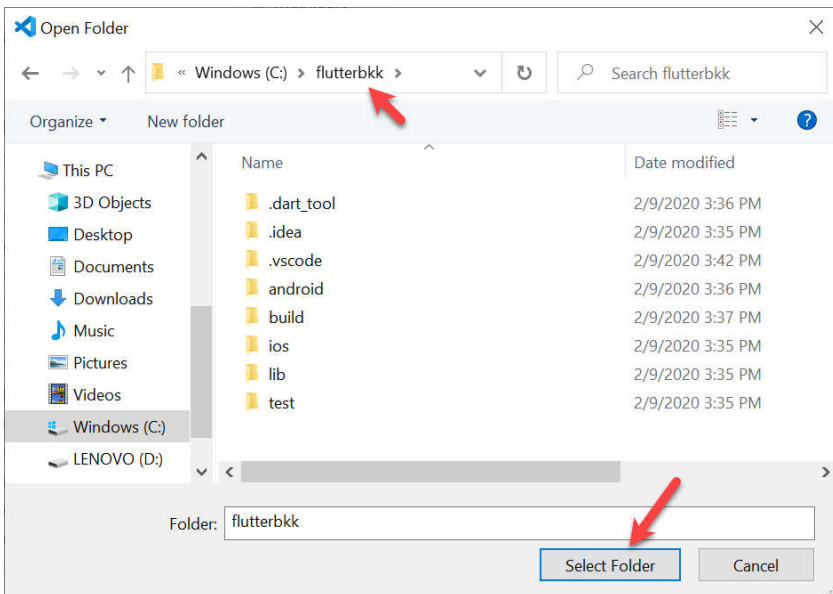
ทดลองเขียน Flutter

ในหัวข้อนี้เราจะมาลองแก้ไขโค้ด เพื่อแสดง Hello World กัน

1. ใช้ Visual Studio Code เพื่อเปิดโฟลเดอร์โปรเจค คลิกที่เมนู File -> Open Folder...



2. คลิกเลือกโฟลเดอร์โปรเจคของตัวเอง



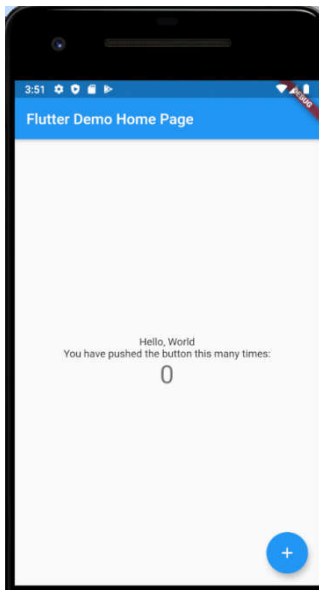
3. เปิดไฟล์ lib/main.dart แล้วทดลองเขียนโค้ดดังนี้ อย่าลืมบันทึกไฟล์

```

lib > main.dart > _MyHomePageState > build
88 // how it positions its children. Here we use mainA
89 // center the children vertically; the main axis he
90 // axis because Columns are vertical (the cross axi
91 // horizontal).
92 mainAxisAlignment: MainAxisAlignment.center,
93 children: <Widget>[
94   Text(
95     'Hello, World',
96   ), // Text
97   Text(
98     'You have pushed the button this many times:',
99   ), // Text

```

4. รันโปรเจกตามปกติ เพื่อดูผลลัพธ์



5. หากรันอยู่แล้ว ลองแก้ไขโค้ดใหม่อีกครั้ง จากนั้นบันทึกไฟล์ จะเห็นผลลัพธ์ทันที หรือจะ reload ใหม่ก็ได้เช่นเดียวกัน