

การพัฒนาแอปพลิเคชันบนเว็บไซต์ Kodular

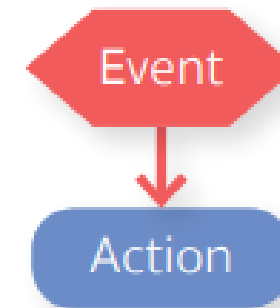
เรื่อง

การเขียนโปรแกรม

# การเขียนโปรแกรม

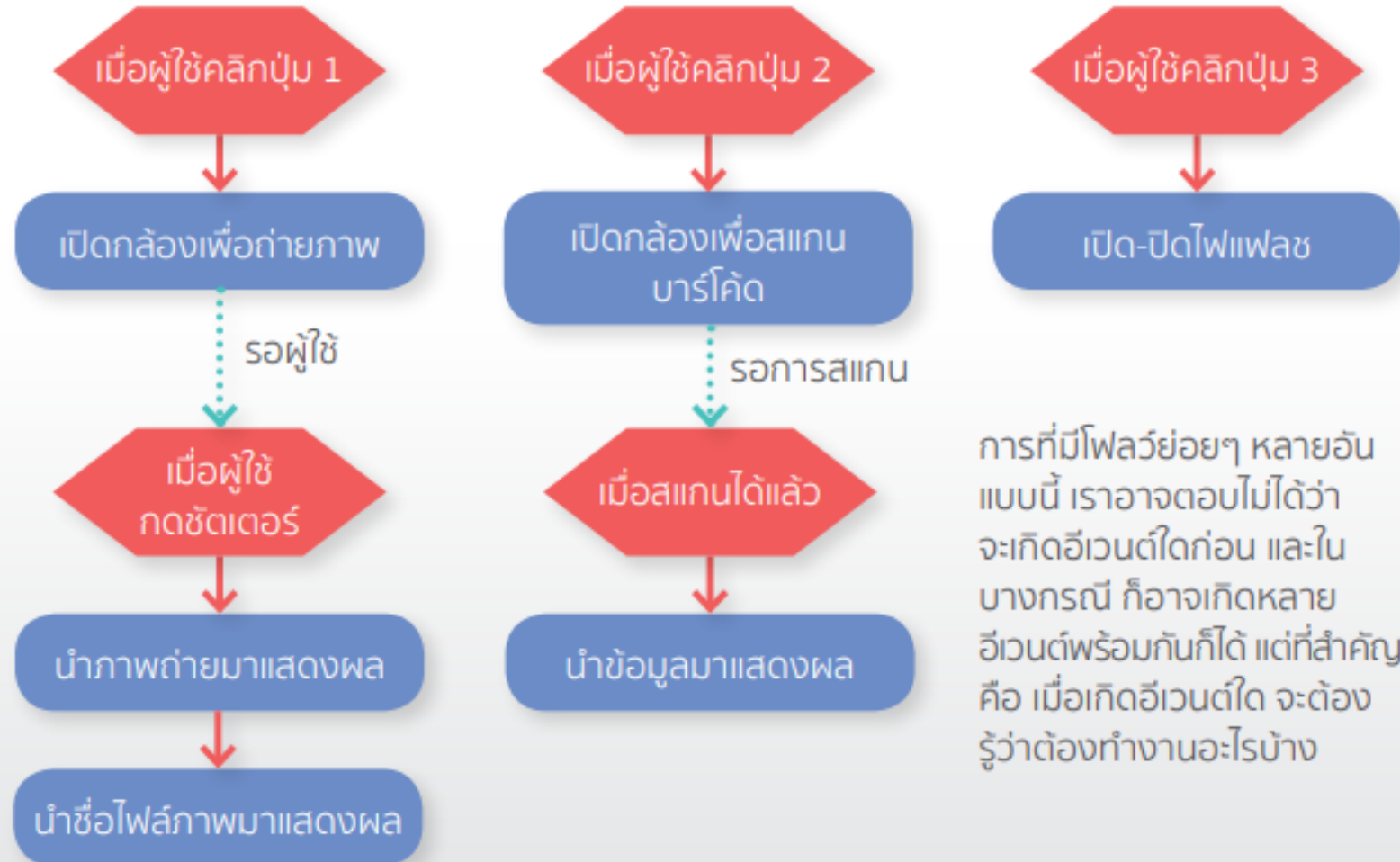
เมื่อนำคอมพิวเตอร์มาวาง และจัดเลย์เอาต์หน้าจอเรียบร้อยแล้ว ขั้นตอนต่อไป คือ การเขียนโปรแกรมควบคุมการทำงานของแอป ซึ่งการทำงานของโปรแกรมจะ เริ่มต้นด้วยอีเวนต์ (**event**) คือ เมื่อเกิดเหตุการณ์ใดขึ้น จะให้แอปทำอะไรบ้าง นั้นหมายความว่า เราจะต้องเขียนโปรแกรมเพื่อให้แอปทำงานตอบสนองต่ออีเวนต์ที่เกิดขึ้นนั่นเอง ซึ่งจะต่างจากเขียนโปรแกรมแบบเดิมที่เป็นโฟลว์เดียว โดยจะมีโฟลว์การทำงานย่อยๆ หลายอัน แต่ละโฟลว์เริ่มต้นด้วยอีเวนต์ ซึ่งอีเวนต์ อาจเกิดขึ้นได้หลายทาง ได้แก่

- **เกิดจากผู้ใช้** เช่น การกดปุ่ม การป้อนข้อมูล
- **เกิดต่อเนื่องจากบล็อกคำสั่งก่อนหน้า** เช่น เมื่อสั่งให้เปิดกล้องเพื่อถ่ายภาพ การถ่ายภาพจะยังไม่เกิดขึ้นจนกว่าผู้ใช้จะกดปุ่มชัตเตอร์ จึงจะเกิดอีเวนต์ตามมา หรือการสั่งให้อ่านข้อมูลจากเว็บ ซึ่งจะไม่ได้ข้อมูลในทันทีที่สั่ง ต้องรอจนกระทั่งเว็บนั้นส่งข้อมูลกลับมาให้ จึงเกิดอีเวนต์ว่าได้รับข้อมูลแล้ว
- **เกิดขึ้นโดยอัตโนมัติ** เช่น อีเวนต์ที่เกิดขึ้นเมื่อแอปเริ่มทำงาน หรือเมื่อคอมพิวเตอร์พร้อมใช้งาน



# การเขียนโปรแกรม

เมื่อลองเขียนโปรแกรมการทำงานของแอปนี้ ก็จะเป็นดังภาพ คือ ประกอบด้วยฟลว์ย่อย 5 อัน แต่ละฟลว์เริ่มต้นด้วยอีเวนต์

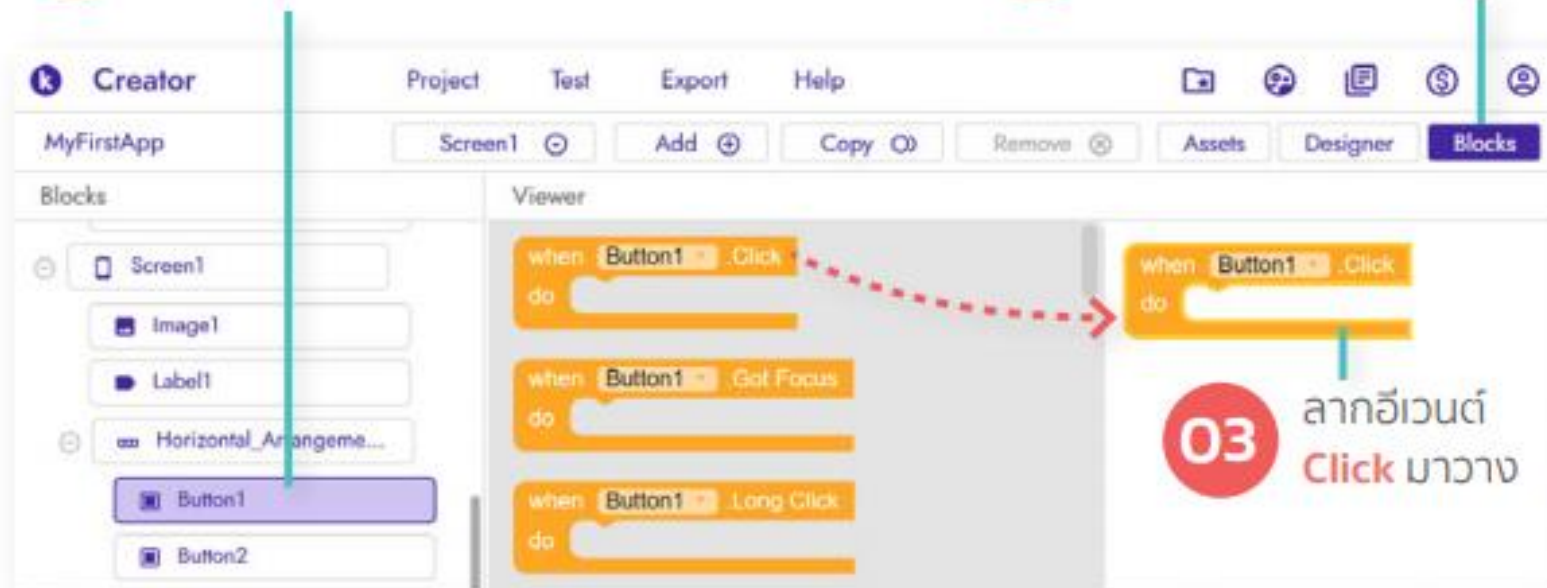


# การเขียนโปรแกรม

การเขียนโปรแกรมใน **Kodular** ทำได้ง่ายๆ โดยการลากบล็อกคำสั่งมาเรียงต่อกัน เริ่มต้นด้วยอีเวนต์

**02** คลิกเลือก **Button1**  
ซึ่งเป็นปุ่มถ่ายภาพ

**01** คลิกปุ่ม **Blocks** เพื่อไปยัง  
หน้า **Block Editor**



**Click** เป็นอีเวนต์สำคัญของคอมโพเนนต์ **Button** ซึ่งใช้บ่อยที่สุด โดยอีเวนต์นี้จะเกิดขึ้นเมื่อผู้ใช้คลิกปุ่ม

**Take Picture** เป็นคำสั่งในคอมโพเนนต์ **Camera** ใช้สั่งให้เปิดกล้องเพื่อถ่ายภาพนิ่ง

# การเขียนโปรแกรม

04 คลิกเลือก **Camera1**

05 ลากคำสั่ง **Take Picture** มาใส่ในอีเวนต์ **Click**

06 ลากอีเวนต์ **After Picture** มาวาง

04 คลิกเลือก **Camera1**

**After Picture** เป็นอีเวนต์ที่เกิดขึ้น เมื่อผู้ใช้กดปุ่ม ชัตเตอร์เพื่อถ่ายภาพ ในอีเวนต์นี้มีตัวแปร **image** ซึ่งเก็บชื่อไฟล์ภาพถ่าย และโฟลเดอร์ที่เก็บภาพ

# การเขียนโปรแกรม



07 คลิกเลือก **Image1**

08 ลาก **set Picture** มาใส่ในอีเวนต์ **After Picture** เพื่อกำหนดชื่อของไฟล์ภาพที่จะแสดงที่คอมพิวเตอร์ **Image** นี้

09 เลื่อนเมาส์ไปชี้ที่ตัวแปร **image** ในอีเวนต์ **After Picture** รอสักครู่ คำสั่ง **get** และ **set** จะปรากฏ



10 ให้ลาก **get image** มาวางต่อท้าย เป็นการกำหนดว่า ให้ใช้ภาพที่ถ่ายได้นั้น มาแสดงผลด้วยคอมพิวเตอร์ **Image1**



# การเขียนโปรแกรม

The screenshot shows a programming environment with a 'Blocks' panel on the left and a 'Viewer' panel on the right. In the 'Blocks' panel, the 'Label1' block is selected. In the 'Viewer' panel, a script is visible with several 'set' blocks for 'Label1' (Marquee, Rotation Angle, Text, Text Color). To the right, there are two event-driven scripts: one for 'Button1 Click' that calls 'Camera1 Take Picture', and another for 'Camera1 After Picture' that sets 'Image1 Picture' to 'get image' and then sets 'Label1 Text' to the value of 'Image1 Picture'. A red dashed line connects the 'get image' block to the 'set Label1 Text' block, indicating data flow. A teal line points from the 'Label1' block in the 'Blocks' panel to the 'set Label1 Text' block in the 'Viewer' panel.

11

คลิกเลือก **Label1**

12

ลาก **set Text** มาต่อจาก **set Picture** เพื่อกำหนดข้อความที่จะแสดงที่ **Label**

# การเขียนโปรแกรม

```
when Camera1 . After Picture  
image  
do  
  set Image1 . Picture . to  
  set Label1 . Text . to
```

**13** คลิกขวาที่บล็อก **get image** แล้วเลือกคำสั่ง **Duplicate** จากเมนูลัด เพื่อสร้างบล็อก **get image** ขึ้นมาอีกหนึ่ง



```
when Camera1 . After Picture  
image  
do  
  set Image1 . Picture . to  
  set Label1 . Text . to
```

**14** ลาก **get image** ที่ได้จากการ **Duplicate** มาต่อท้าย

บล็อกคำสั่งส่วนนี้ ทำเพื่อแสดงชื่อไฟล์ภาพถ่าย และโฟลเดอร์ที่เก็บ ผ่านทาง **Label1** เพื่อให้ผู้ใช้ทราบว่าไฟล์ภาพถ่ายนั้นจัดเก็บไว้ที่ไหน



# การเขียนโปรแกรม

**Do Scan** เป็นคำสั่งในคอมโพเนนต์ **Barcode Scanner** ใช้สั่งให้เปิดกล้องเพื่อสแกนบาร์โค้ด

15

นำอีเวนต์ **Click** ของปุ่ม **Button2** มาวาง  
ปุ่มนี้เป็นปุ่มสั่งสแกนบาร์โค้ด

The screenshot displays a visual programming interface with two main panels: 'Blocks' and 'Viewer'. The 'Blocks' panel on the left contains a list of components including 'Screen1', 'Image1', 'Label1', 'Horizontal\_Arrangeme...', 'Button1', 'Button2', 'Button3', 'Camera1', 'Barcode\_Scanner1', and 'Any component'. The 'Barcode\_Scanner1' component is highlighted with a blue selection bar. The 'Viewer' panel on the right shows a sequence of logic blocks. A 'when Barcode\_Scanner1 After Scan' block is connected to a 'result' block. Below it is a 'do' block containing a 'call Barcode\_Scanner1 Do Scan' block. Further down are 'Barcode\_Scanner1 Result', 'Barcode\_Scanner1 Use External Scanner', 'set Barcode\_Scanner1 Use External Scanner to', and 'Barcode\_Scanner1' blocks. A 'when Button2 Click' block is connected to the 'call Barcode\_Scanner1 Do Scan' block via a dashed red line. Another 'when Barcode\_Scanner1 After Scan' block with a 'result' block is also shown, connected to the 'Barcode\_Scanner1 Result' block in the viewer.

16

คลิก **Barcode\_Scanner1**

17

ลากคำสั่ง **Do Scan** ไปใส่  
ในอีเวนต์ **Click**

18

ลากอีเวนต์ **After Scan** มาวาง

**After Scan** เป็นอีเวนต์ที่เกิดขึ้นเมื่อสามารถ  
สแกนบาร์โค้ดได้สำเร็จ ในอีเวนต์นี้มีตัวแปร  
**result** ซึ่งเก็บข้อมูลที่ได้จากการสแกน

# การเขียนโปรแกรม

19 นำ **set Text** จาก **Label1**  
มาวางในอีเวนต์ **After Scan**

```
when Barcode_Scanner1 . After Scan
  result
do
  set Label1 . Text to
```

บล็อกคำสั่งส่วนนี้ ใช้แสดงข้อมูลที่ได้จากการ  
สแกนบาร์โค้ด ผ่านทาง **Label1**

21 นำ **get result** มาต่อท้าย

```
when Barcode_Scanner1 . After Scan
  result
do
  set [get result] to [get result]
  set result to
```

20 เลื่อนเมาส์ไปชี้ที่ตัวแปร **result**  
รอสักครู่ คำสั่ง **get / set** จะปรากฏ

# การเขียนโปรแกรม

22 นำอีเวนต์ **Click** ของปุ่ม **Button3** มาวาง

24 นำบล็อก **Toggle Light** มาวางในอีเวนต์

23 คลิกเลือก **Camera1**

**Toggle Light** เป็นคำสั่งในคอมพิวเตอร์ **Camera** สำหรับเปิด-ปิดไฟแฟลช โดยการสั่งงานแต่ละครั้ง จะทำให้เปิด-ปิด สลับกัน

# สรุปการทำงานของโปรแกรม

โปรแกรมทั้งหมดของแอปนี้ ประกอบด้วยโฟลว์การทำงานย่อย 5 อัน ตามที่เราออกแบบไว้ตั้งแต่แรก (หน้า 34) และใช้บล็อกคำสั่งไปทั้งสิ้นเพียง 14 บล็อกเท่านั้น โดยเราสามารถสรุปการทำงานของโปรแกรมได้ดังนี้

เมื่อผู้ใช้คลิกปุ่ม **Button1** ก็จะเกิดอีเวนต์ **Click** ขึ้น ภายในอีเวนต์ เราสั่งให้เปิดกล้อง เพื่อถ่ายภาพด้วยคำสั่ง **Take Picture**

```
when Button1 .Click
do call Camera1 .Take Picture
```

และเมื่อผู้ใช้คลิกปุ่มชัตเตอร์เพื่อถ่ายภาพ ก็จะเกิดอีเวนต์ **After Picture** ตามมา เรานำชื่อไฟล์ภาพและโฟลเดอร์ในตัวแปร **image** มากำหนดหรือพเพอร์ดี **Picture** เพื่อแสดงภาพที่ถ่ายด้วย **Image1** และนำชื่อไฟล์นั้นไปกำหนดหรือพเพอร์ดี **Text** เพื่อแสดงผลทาง **Label1**

```
when Camera1 .After Picture
image
do set Image1 . Picture to get image
set Label1 . Text to get image
```

# สรุปการทำงานของโปรแกรม

ในกรณีที่ผู้ใช้คลิกปุ่ม **Button2** ก็จะเกิด  
อีเวนต์ **Click** ขึ้น ภายในอีเวนต์ เราสั่งให้  
เปิดกล้องเพื่อสแกนบาร์โค้ดด้วยคำสั่ง  
**Do Scan**

```
when Button2 .Click  
do call Barcode_Scanner1 .Do Scan
```

เมื่อสามารถสแกนบาร์โค้ดได้แล้ว ก็จะเกิด  
อีเวนต์ **After Scan** พร้อมข้อมูลที่สแกน  
ได้อยู่ในตัวแปร **result** เรานำข้อมูลนั้นมา  
กำหนดหรือพเพอร์ตี **Text** เพื่อแสดง  
ข้อมูลทาง **Label1**

```
when Barcode_Scanner1 .After Scan  
result  
do set Label1 . Text to (get result)
```

สุดท้าย ในกรณีที่ผู้ใช้คลิกปุ่ม **Button3**  
ก็จะเกิดอีเวนต์ **Click** เช่นกัน ภายในอีเวนต์  
เราสั่งให้เปิด-ปิดไฟแฟลชของกล้องด้วย  
คำสั่ง **Toggle Light**

```
when Button3 .Click  
do call Camera1 .Toggle Light
```