

Introduction to Design (Track 3)

3.2 Hands-on: Counter App

Zilu Liang

www.zilu-liang.net/id3

Important!!

Please disinfectize your hands before entering the classroom!

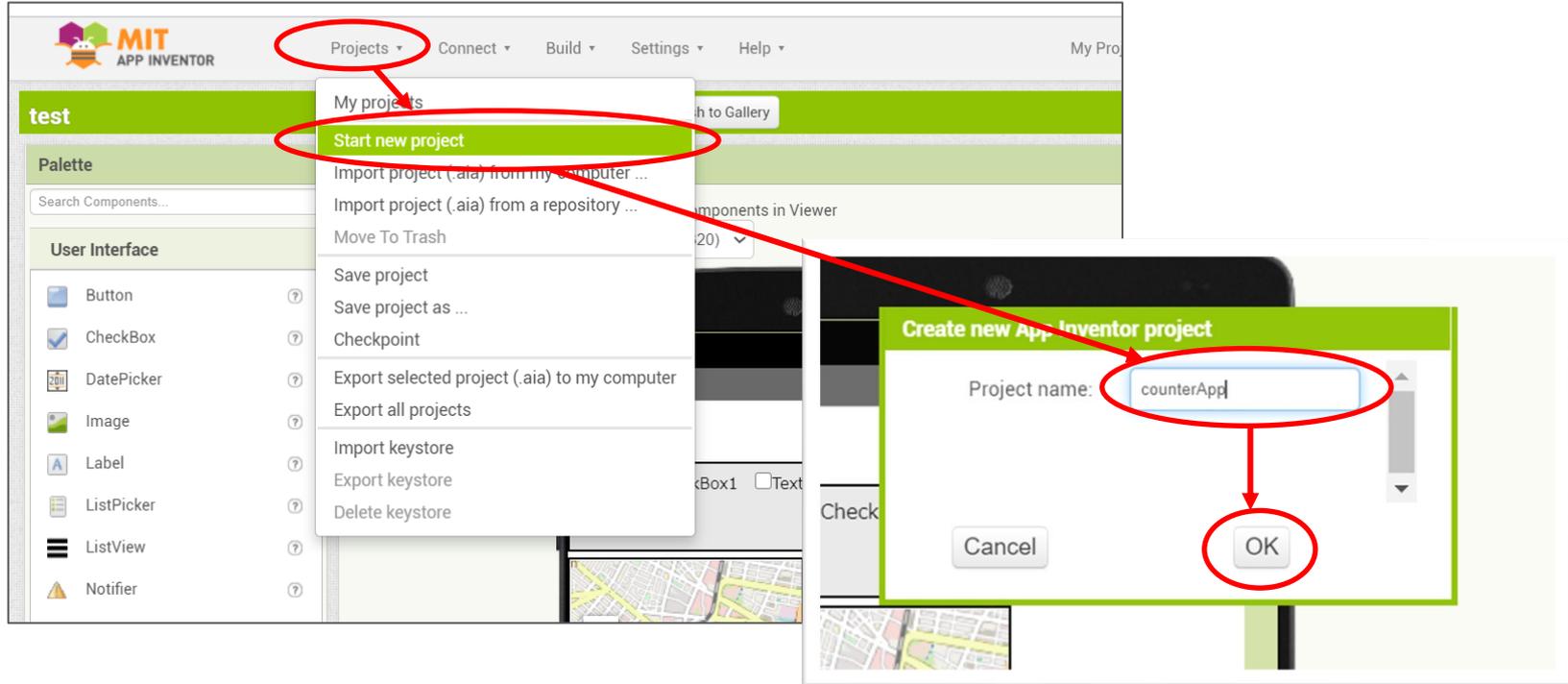
入室前にアルコールを使用して手指消毒を行ってください。

Please disinfectize your chair and table!

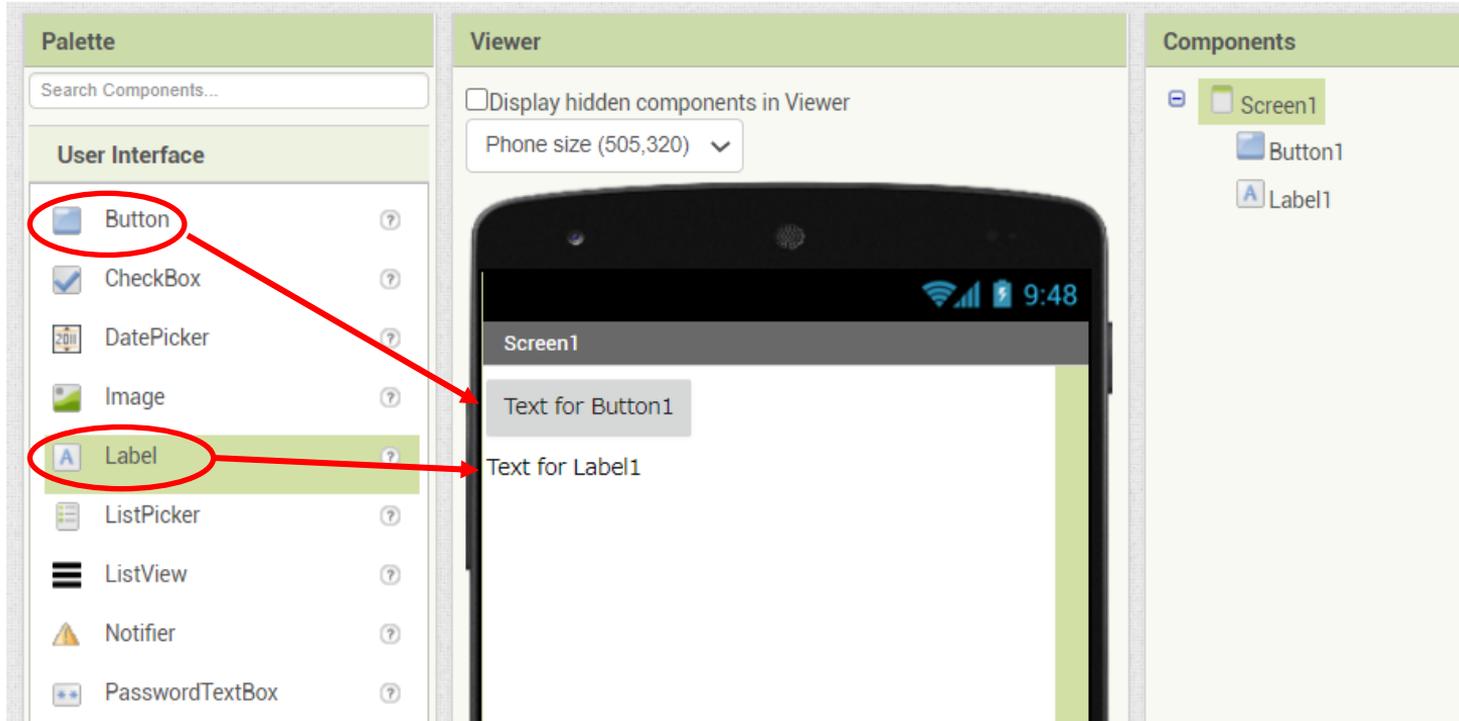
- ①ペーパーにアルコールを噴霧してください。
- ②アルコールが噴霧されたペーパーで、使用箇所（テーブル、椅子など）を拭き取ってください。
- ③使用済のペーパーは廊下のごみ箱に捨ててください。



Create a new project

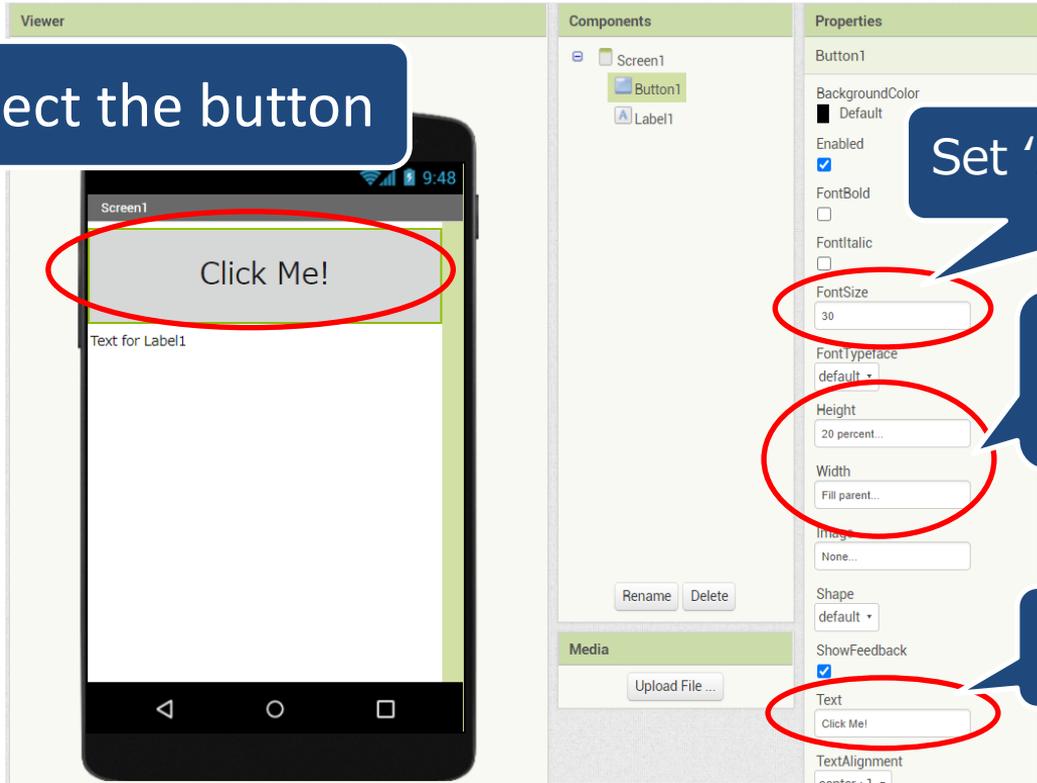


Add a button and a label



Set property variables of the button

Select the button



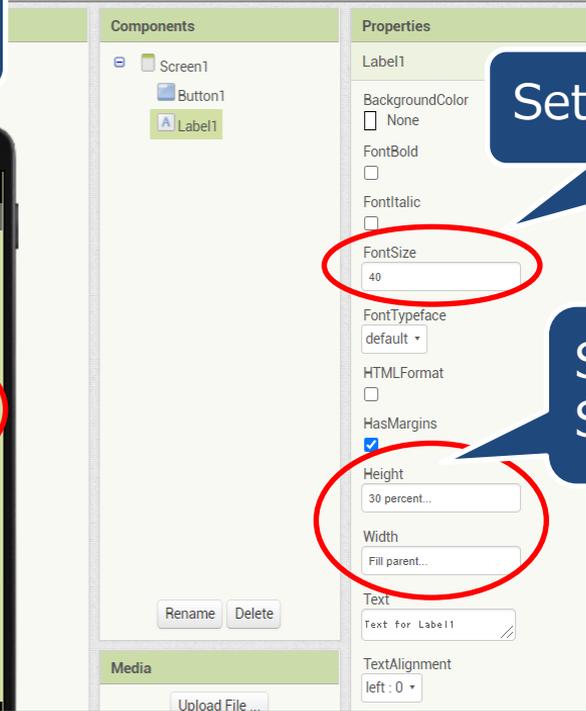
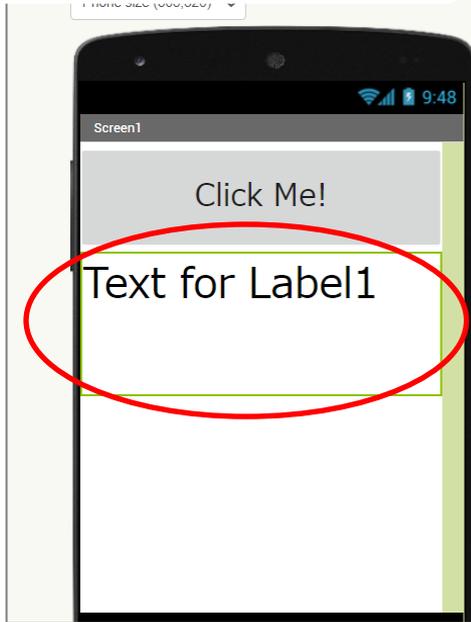
Set *'FontSize'* to *'30'*

Set *'Height'* to *'20 percent'*
Set *'Width'* to *'Fill parent'*

Set *'Text'* to *'Click Me!'*

Set property variables of the label

Select the label



The screenshot shows the development tool's interface. On the left, the 'Components' panel lists 'Screen1', 'Button1', and 'Label1'. On the right, the 'Properties' panel for 'Label1' is visible. The 'FontSize' property is set to '40' and the 'Height' property is set to '30 percent...'. Both the 'FontSize' and 'Height' input fields are circled in red. Other properties like 'BackgroundColor', 'FontBold', 'FontItalic', 'FontTypeface', 'HTMLFormat', 'HasMargins', 'Width', 'Text', and 'TextAlignment' are also visible.

Set '*FontSize*' to '*40*'

Set '*Height*' to '*30 percent*'
Set '*Width*' to '*Fill parent*'

Rename the button

The image shows a software development interface with two main panels: 'Viewer' on the left and 'Components' on the right. In the 'Components' panel, a tree view shows a hierarchy starting with 'Screen1', which contains 'Button1' and 'Label1'. 'Button1' is circled in red. A red arrow points from 'Button1' down to a dialog box. The dialog box has two input fields: 'Old name:' containing 'Button1' and 'New name:' containing 'click_button1'. The 'New name' field is also circled in red. Below the input fields are 'Cancel' and 'OK' buttons. A second red arrow points from the 'New name' field to the 'Rename' button in the 'Components' panel, which is also circled in red. A third red arrow points from the 'Rename' button back to the 'New name' field. Three blue callout boxes with white text provide instructions: 'Input a new name for the button, such as 'click_button'' (pointing to the 'New name' field), 'Select the button' (pointing to 'Button1' in the components list), and 'Click "Rename"' (pointing to the 'Rename' button).

Input a new name for the button, such as 'click_button'

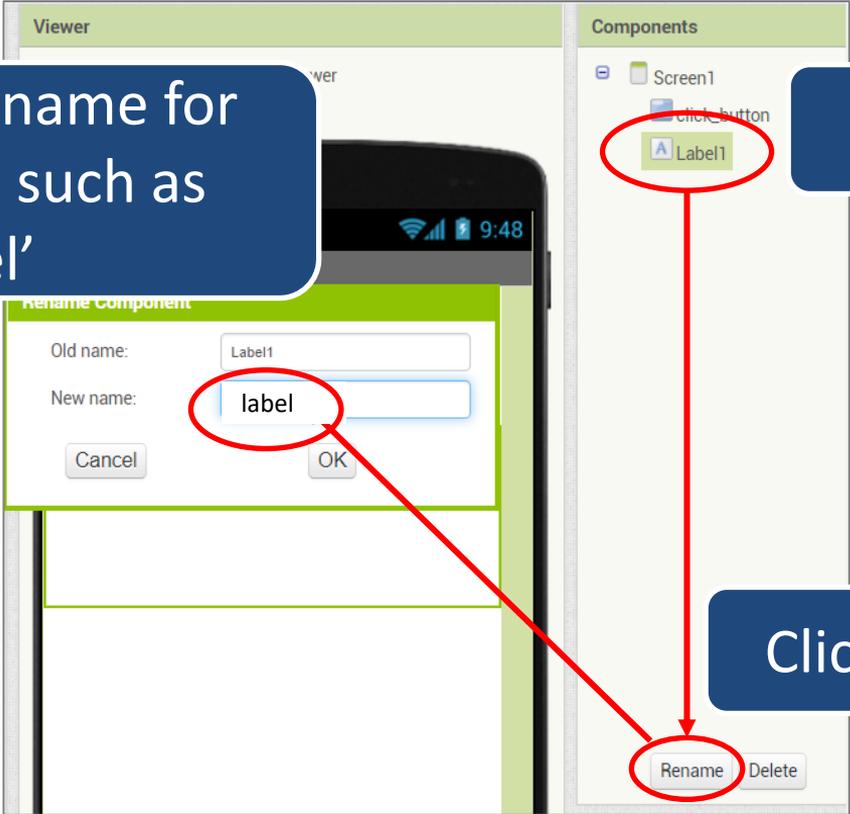
Select the button

Click "Rename"

Rename the label

Input a new name for the button, such as 'label'

Select the label



Click "Raname"

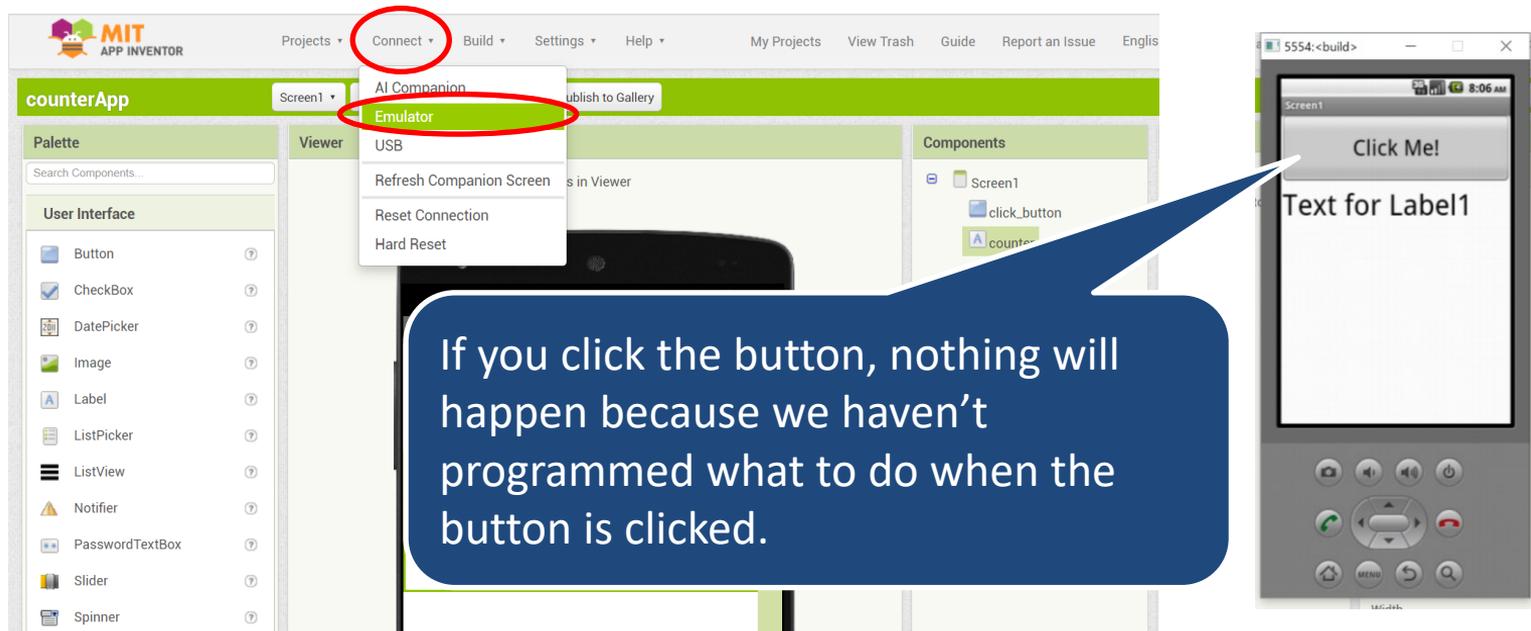
Test App on Emulator

Click on the aiStarter on your computer, you should see the following window open.

```
aiStarter
127.0.0.1 -- [21/Apr/2021 17:00:23] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:23] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:24] "GET /start/ HTTP/1.1" 200 0
127.0.0.1 -- [21/Apr/2021 17:00:24] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:24] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:24] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:24] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:24] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:24] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:25] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:26] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:27] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:28] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:29] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:30] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:31] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:32] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:33] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:34] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:35] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:36] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:37] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:38] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:39] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:40] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:41] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:42] "GET /echeck/ HTTP/1.1" 200 40
127.0.0.1 -- [21/Apr/2021 17:00:43] "GET /echeck/ HTTP/1.1" 200 67
Device = emulator-5554
127.0.0.1 -- [21/Apr/2021 17:01:09] "GET /restart/emulator-5554 HTTP/1.1" 200 0
```

Test app on emulator

Go back to App Inventor and connect to the emulator



The image shows the MIT App Inventor web interface. The 'Connect' menu is open, and the 'Emulator' option is highlighted with a red circle. A blue callout box points to the emulator window, which displays a mobile app with a 'Click Me!' button and a 'Text for Label1' label. The app is running on a virtual device with a status bar at the top showing the time as 8:06 AM.

If you click the button, nothing will happen because we haven't programmed what to do when the button is clicked.

Switch to blocks view

counterApp

Screen1 ▾ Add Screen ... Remove Screen Publish to Gallery

Designer Blocks

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Dictionaries
 - Colors
 - Variables
 - Procedures
- Screen1
 - click_button
 - label
- Any component

Viewer

Show Warnings

Add and initialize a global variable

We need a variable to store the counter value. Since this counter value should be used anywhere in the app, it requires a global variable.

The screenshot shows the App Inventor interface for an application named "counterApp". The interface is divided into two main sections: "Blocks" on the left and "Viewer" on the right. In the "Blocks" section, the "Variables" category is highlighted with a red circle. A red arrow points from this category to the "initialize global name to" block in the "Viewer" section, which is also circled in red. A tooltip next to this block reads "Creates a global variable and gives it the value of the attached blocks." Another red arrow points from the "initialize global name to" block to a separate, larger view of the same block on the right. Below the circled block in the "Viewer" section, there are other blocks: a "get" block, a "set to" block, and two "initialize local name to" blocks, each with a gear icon and a "name" field.

Initialize the global variable

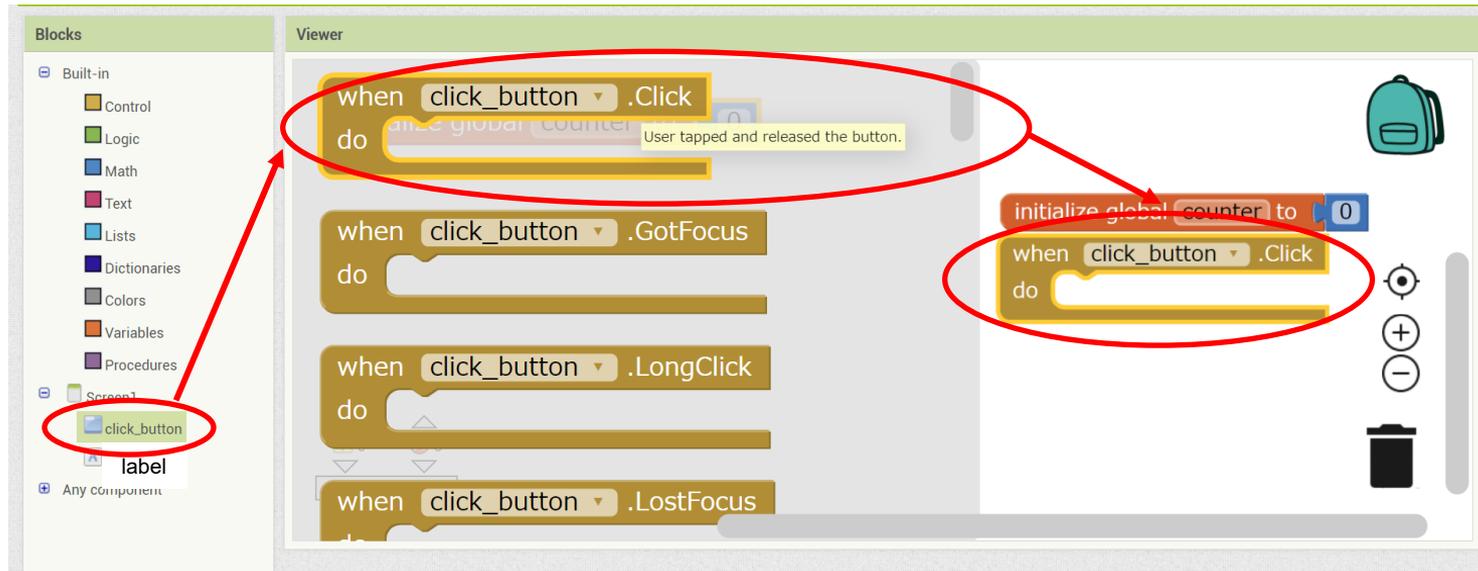
All variables need to be initialized. Change the variable name to 'counter' and set its value to '0'.

Set the variable value to '0', so the variable is going to start as 0 when we click the button.

Change variable name to 'counter'

Add click handler to the button

When we click the button, the 'counter' variable should add one. We need to add a click handler to the button. The click handler takes care of what should be done when the button is clicked.



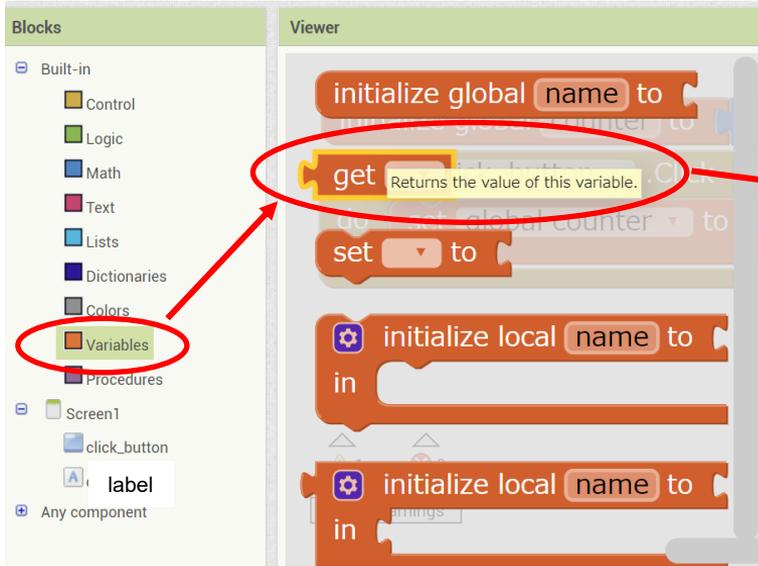
Set the click handler

Now we need to set the click handler to increase the 'counter' variable.

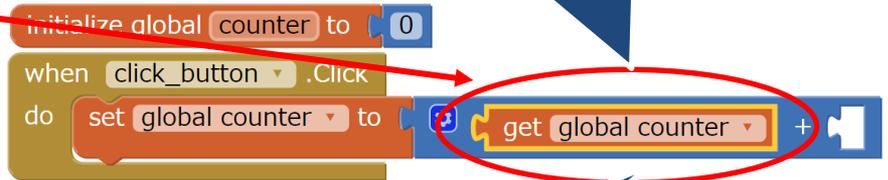
The screenshot shows a visual programming interface with a 'Blocks' panel on the left and a 'Viewer' panel on the right. The 'Blocks' panel has a 'Variables' category circled in red. The 'Viewer' panel shows a sequence of code blocks: 'initialize global name to', 'initialize global counter to', 'get click_button .Click', 'do', 'set to', 'initialize local name to', 'in', 'initialize local name to', 'in'. A blue callout bubble points to the 'set to' block with the text 'Make sure to put the "set to" block inside the click handler'. Another blue callout bubble points to the 'set global counter to' block with the text 'Select 'global counter''.

Make sure to put the "set to" block inside the click handler

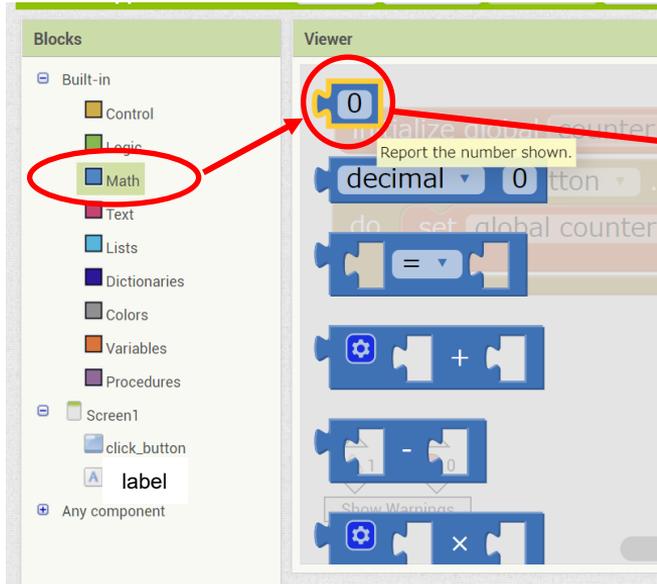
Select 'global counter'



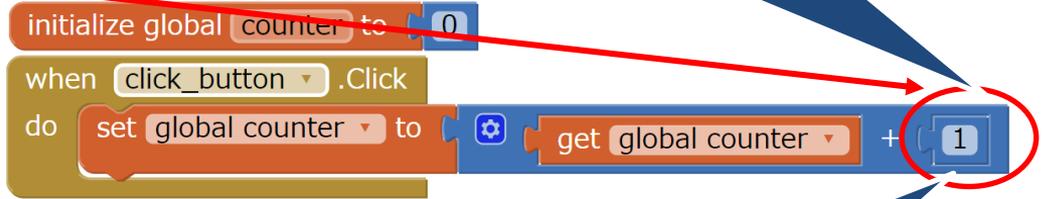
Make sure to put the 'get' block inside the 'sum' block



Select 'global counter' here



Make sure to put the 'number' block inside the 'sum' block



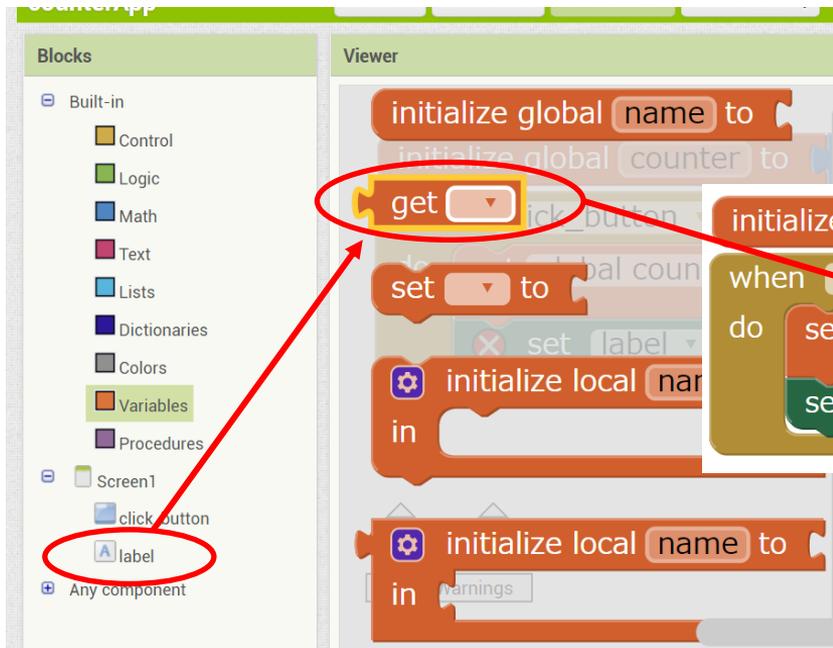
Set the 'number' to 1

Set the label

After we change the counter's value, we need to display it in the label. So we need to set the label text.



The screenshot shows a visual programming environment with a 'Blocks' panel on the left and a 'Viewer' panel on the right. The 'Blocks' panel lists categories like Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. Under 'Text', a 'label' block is circled in red. The 'Viewer' panel shows a sequence of code blocks: 'initialize global counter to 0', 'when click_button .Click do', 'set global counter to [get global counter + 1]', and 'set label . Text to'. The 'set label . Text to' block is circled in red. A blue callout box with a white arrow pointing to the 'set label . Text to' block contains the text: 'Make sure to put the 'set text' block inside the click handler.'

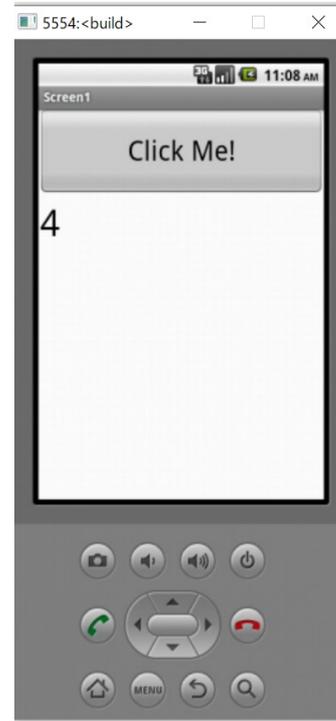


Make sure to put the 'get' block behind the 'set counter text' block



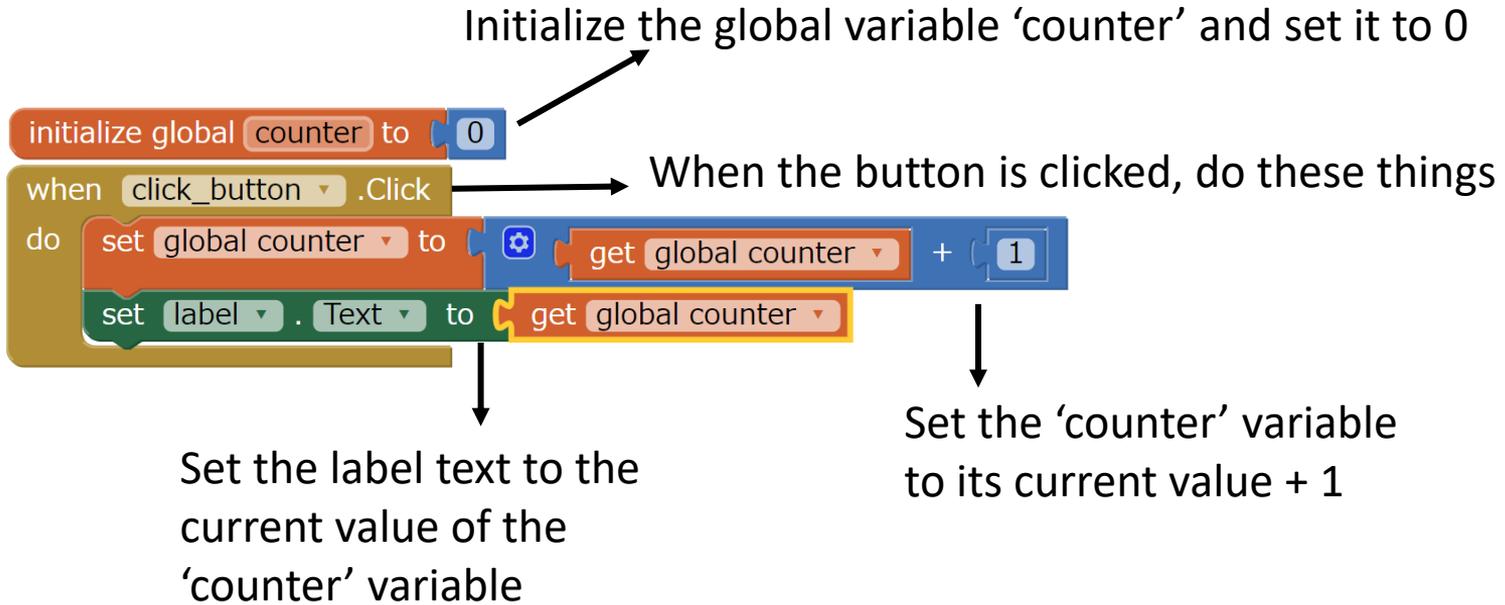
Select 'global counter'

Test on Emulator



Code Anatomy

Now let's review what the code is actually doing.



Assignment

- Complete the hands-on tasks in the tutorial
 - ✓ If you finish all the steps in class, show your Counter App to one of the instructors before you leave
 - ✓ If you cannot finish all the steps, you can work on them after class and show your Counter App to one of the instructors in the class next week
- (Optional) If you have time, why not challenging the following tasks
 1. Remove the text shown in the label 'Text for Label1'
 2. Change the counter to increase by 2 instead of 1 on each click
 3. Add another counter button that counts independently of the first button
 4. Add another button that resets the counter to 0

KUAS

KYOTO UNIVERSITY of ADVANCED SCIENCE

京都先端科学大学