

CHAPTER 1

Introduction to Programming and App Inventor

starting out with >>> **APP INVENTOR**
FOR ANDROID



Getting Hands-On with App Inventor

Programming with Blocks

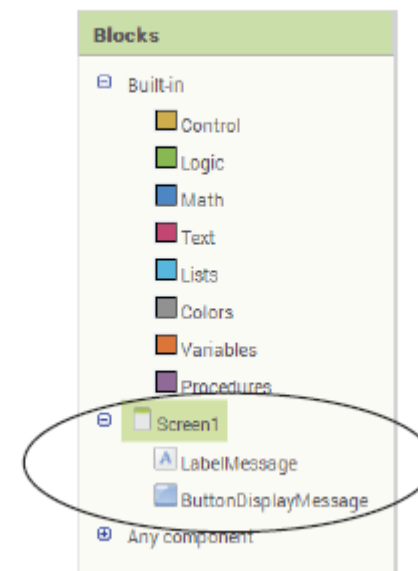
- Apps are *event-driven* programs.
- When an app is running, it waits for a specific event to happen.
- An event is an action such as the user clicking a button or sliding his or her finger across the device's screen.
- An incoming text message is also an event.
- An event also occurs when the user tilts or shakes the phone.

Getting Hands-On with App Inventor

Programming with Blocks

- The Hello World app has a Button component named ButtonDisplayMessage.
- And a Label component named LabelMessage
- We need a block that executes when user clicks the ButtonDisplayMessage component.

Figure 1-59 The Component Entries in the Blocks Column (Source: MIT App Inventor 2)

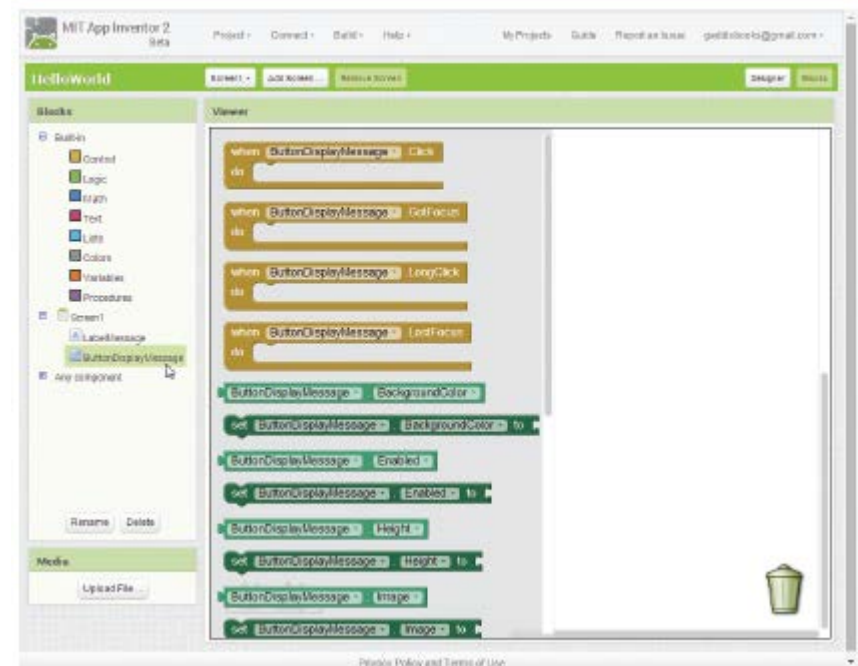


Getting Hands-On with App Inventor

Programming with Blocks

Because you want to create a block that executes when the `ButtonDisplayMessage` component is clicked, you need to click the `ButtonDisplayMessage` entry.

Figure 1-60 The `ButtonDisplayMessage` Drawer Open (Source: MIT App Inventor 2)



Getting Hands-On with App Inventor

Programming with Blocks

There are numerous blocks in this drawer. Here is a summary of the meaning of the colors.

- ***Brown blocks*** are called *event handlers*.
 - An *event handler* is a block that automatically executes when a specific event takes place.
- ***Light green blocks*** represent values that are related to the component.
- ***Dark green blocks*** perform actions with the component.

Getting Hands-On with App Inventor

Figure 1-61 The when ButtonDisplayMessage.Click do Block
(Source: MIT App Inventor 2)



Programming with Blocks

- Figure 1-61 Shows the topmost block inside the drawer. It reads:

`when ButtonDisplayMessage.Click do`

- This means when the ButtonDisplayMessage is clicked, *do this block*.

Getting Hands-On with App Inventor

Figure 1-63 You Complete the Block by Snapping Other Blocks into the Empty Space. (Source: MIT App Inventor 2)



Programming with Blocks

- The `ButtonDisplayMessage.Click` do block has an odd-shaped space in the middle.
- You can snap another block or set of blocks into this space.

Getting Hands-On with App Inventor

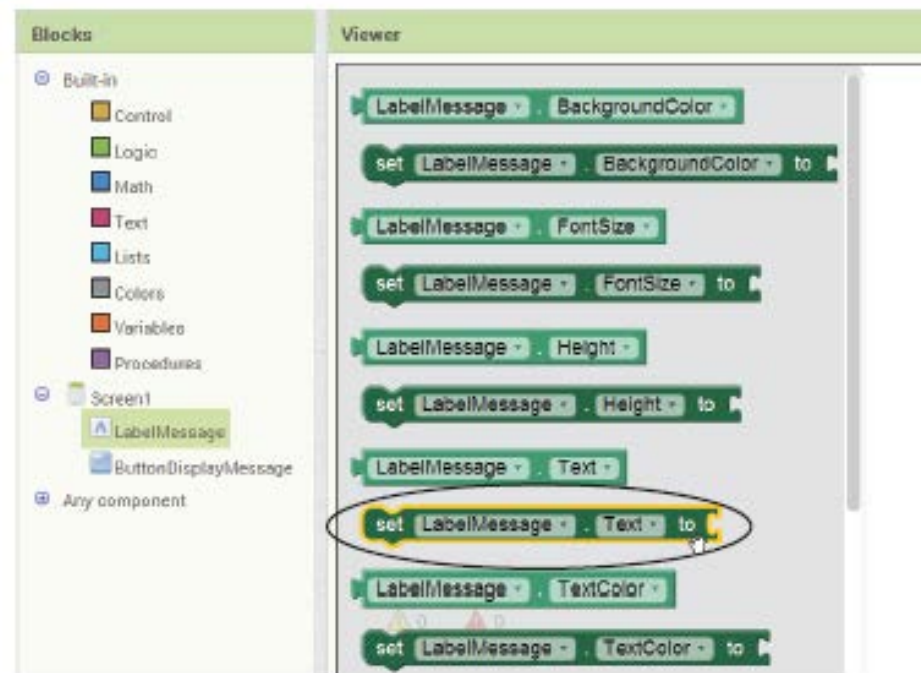
Programming with Blocks

- A `Label` component has a `Text` property.
- Find a block that sets the `LabelMessage` component's `Text` property.
- Snap it into the `when ButtonDisplayMessage.Click do` block.
- Open the drawer containing the blocks for the `LabelMessage` component. It reads:

```
set LabelMessage.Text to
```


Getting Hands-On with App Inventor

Figure 1-64 The set LabelMessage.Text to Block (Source: MIT App Inventor 2)



Getting Hands-On with App Inventor

Programming in Blocks

- To insert the block you click and drag it to the empty space inside the
`when ButtonDisplayMessage.Click do block.`
- The `set LabelMessage.Text to block` is not a complete instruction.
- Noticed the opening on the right edge of the `set LabelMessage.Text to block`.
- You need to snap another block specifying a value into the socket.

Getting Hands-On with App Inventor

Figure 1-65 The `set LabelMessage.Text` to Block Inserted

(Source: MIT App Inventor 2)

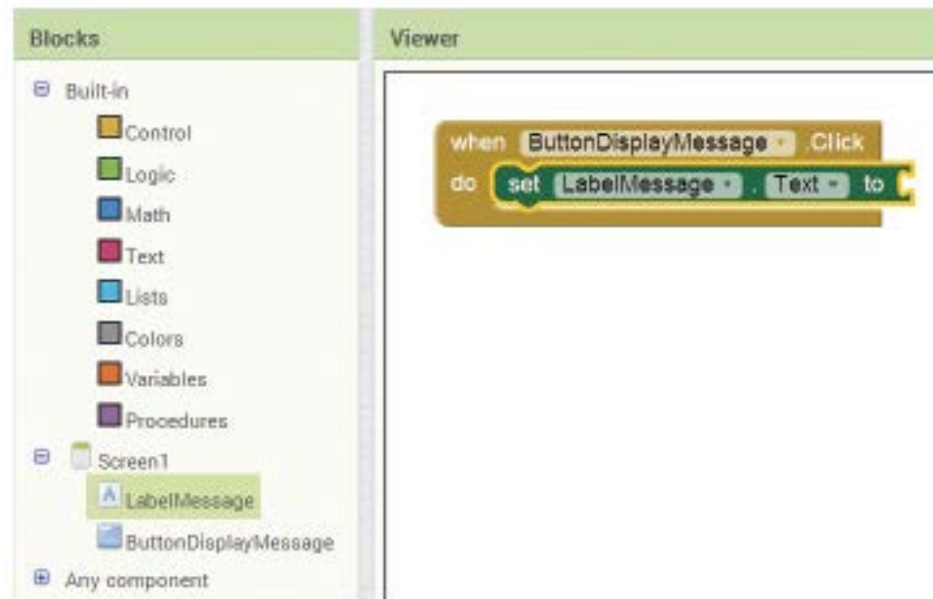


Figure 1-66 An Empty Socket (Source: MIT App Inventor 2)



Getting Hands-On with App Inventor

Programming in Blocks

- Click *Text* under *Built-in* in the Blocks column.
- A drawer will open as shown in Figure 1-67.
- In the figure is the *text string* block.
- Click the empty space, type *Hello World* then press enter.

Getting Hands-On with App Inventor

Figure 1-67 The Built-in text string Block (Source: MIT App Inventor 2)

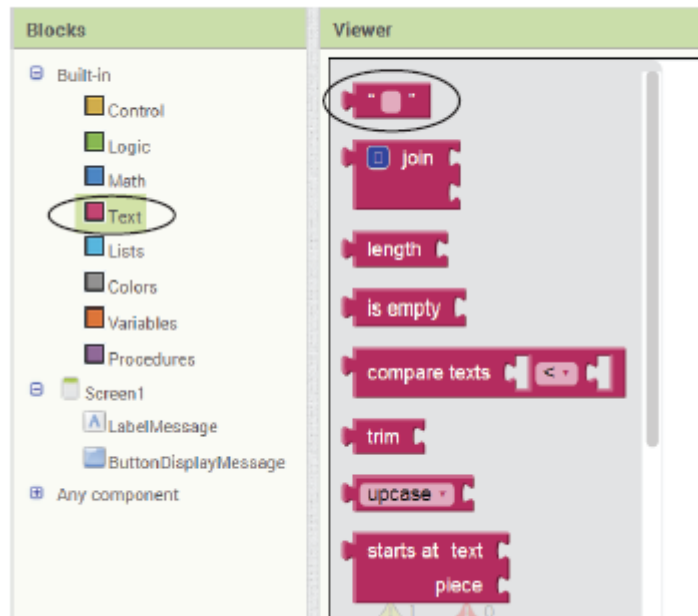


Figure 1-68 The Text String Block Snapped into the Socket of the set LabelMessage.Text to Block (Source: MIT App Inventor 2)

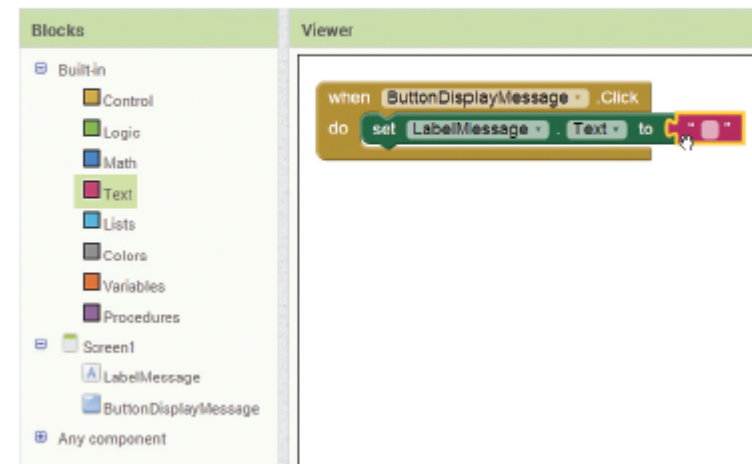


Figure 1-69 Changing the Value of the Text String Block to Hello World (Source: MIT App Inventor 2)

