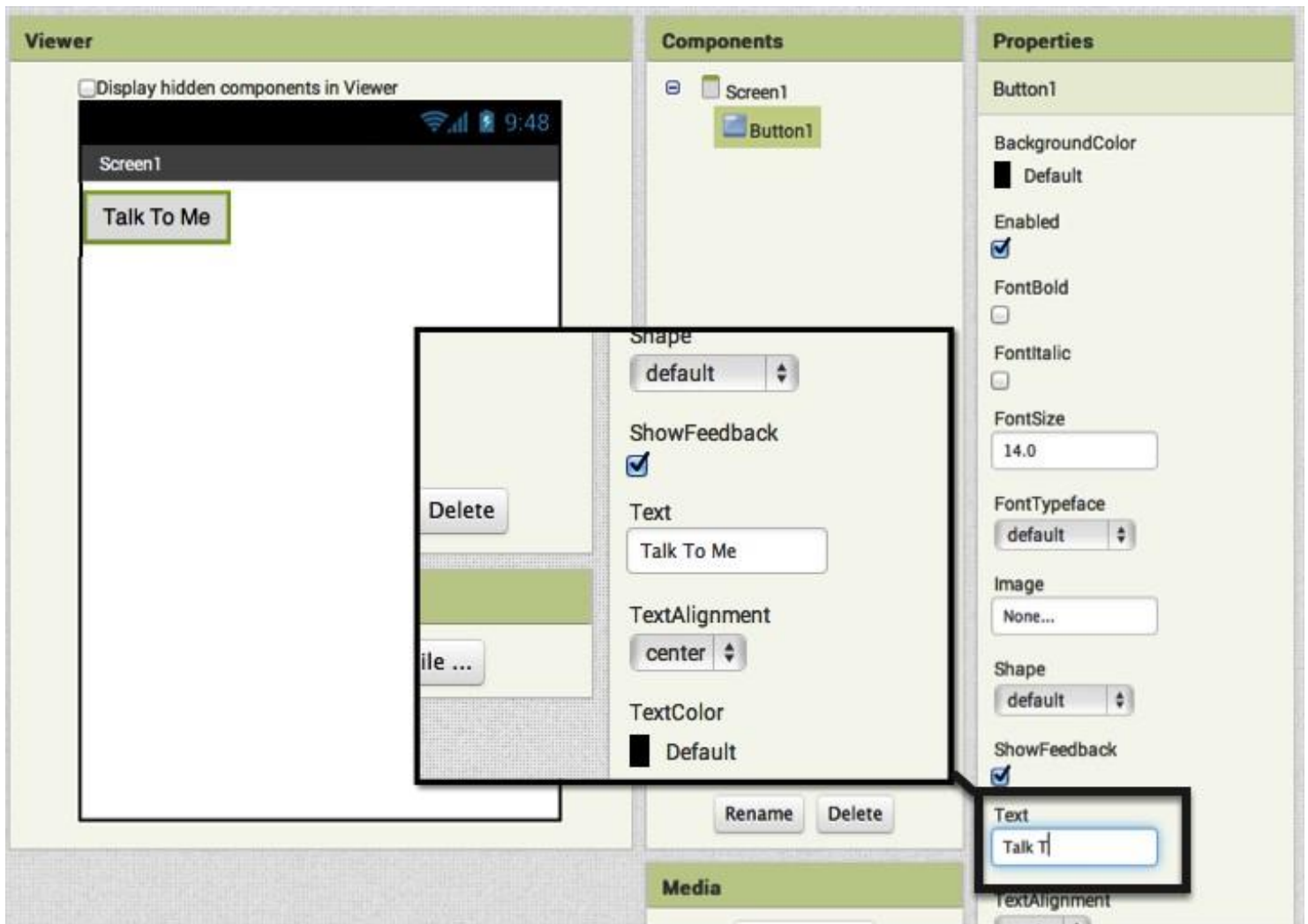


Create a new project and name it as: **“TalkToMe”**

Add a button and Change the Text on the Button

On the properties pane, change the text for the Button. Select the text "Text for Button 1", delete it and type in "Talk To Me". Notice that the text on your app's button changes right away.



Add a Text-to-Speech component to your app

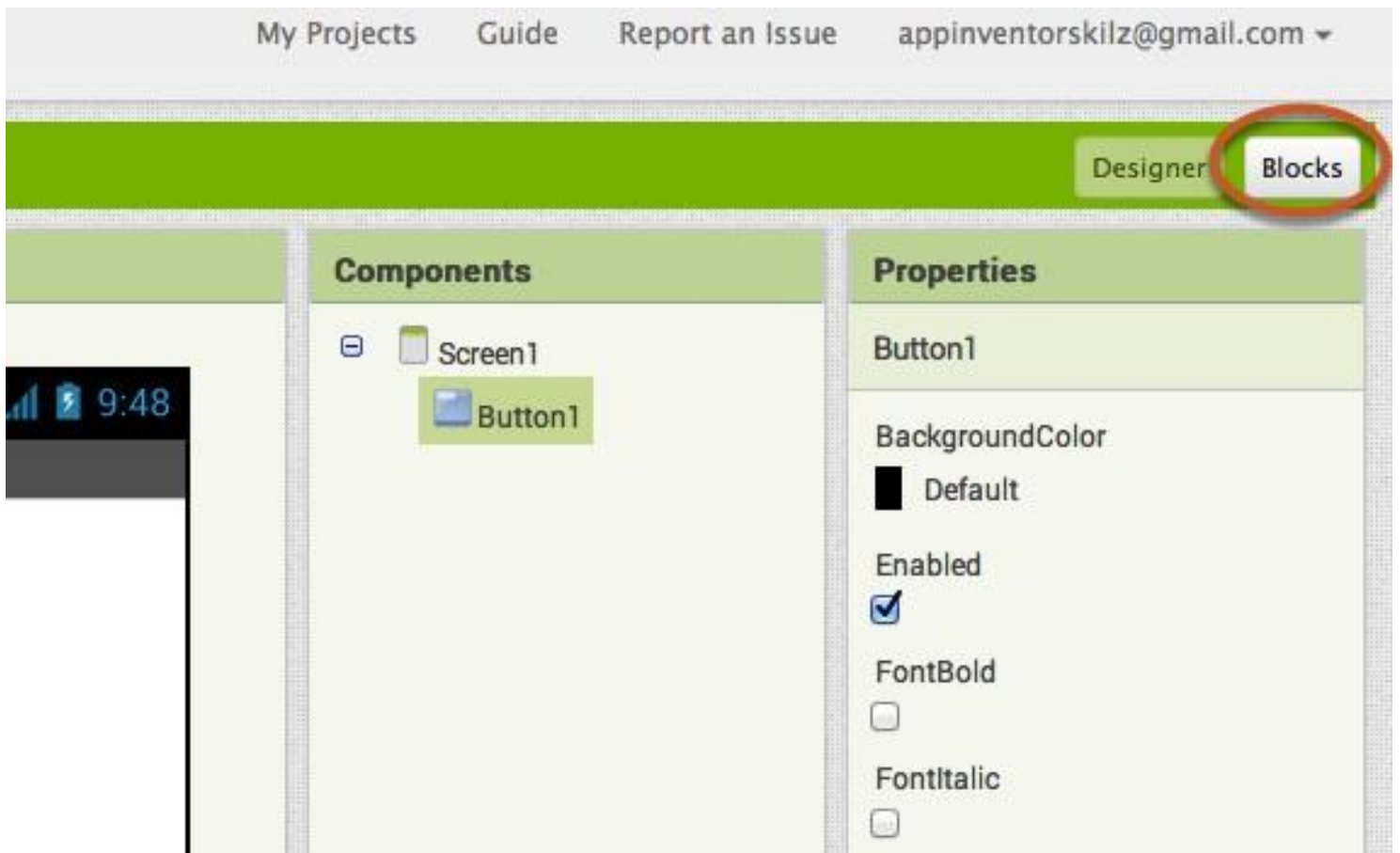
Go to the Media drawer and drag out a TextToSpeech component. Drop it onto the Viewer. Notice that it drops down under "Non-visible components" because it is not something that will show up on the app's user interface. It's more like a tool that is available to the app.

The screenshot displays the LEGO MINDSTORMS software interface, divided into four main sections: Palette, Viewer, Components, and Properties.

- Palette:** The left sidebar contains various component categories. The **Media** category is circled in red. Within this category, the **TextToSpeech** component is also circled in red. An orange arrow points from this component to the Viewer.
- Viewer:** The central workspace shows a mobile app preview. At the top, there's a status bar with the time 9:48. Below it, a button labeled "Talk To Me" is visible. A large grey box with the text "Drop here. Component will automatically show up in Non-visible components area below" is positioned in the center. Below the viewer, a section titled "Non-visible components" contains a single component labeled "TextToSpeech1".
- Components:** The right sidebar shows a tree view of the app's components. It includes "Screen1", "Button1", and "TextToSpeech1". Below this list are "Rename" and "Delete" buttons.
- Properties:** The far right sidebar shows the properties for the selected "TextToSpeech1" component, including fields for "Country" and "Language".

Switch over to the Blocks Editor

It's time to tell your app what to do! Click "Blocks" to move over to the Blocks Editor. Think of the Designer and Blocks buttons like tabs -- you use them to move back and forth between the two areas of App Inventor.



The Blocks Editor

The Blocks Editor is where you program the behavior of your app. There are Built-in blocks that handle things like math, logic, and text. Below that are the blocks that go with each of the components in your app. *In order to get the blocks for a certain component to show up in the Blocks Editor, you first have to add that component to your app through the Designer.*

The screenshot shows the MIT App Inventor 2 interface. At the top, the title bar reads "MIT App Inventor 2 Beta" with navigation options: "Project", "Connect", "Build", and "Help". On the right, there are links for "My Projects", "Guide", "Report an Issue", and an email address "appinventorskilz@gmail.com".

The main workspace is titled "TalkToMe" and includes buttons for "Screen1", "Add Screen ...", and "Remove Screen". On the right side of the workspace, there are tabs for "Designer" and "Blocks".

The "Blocks" panel on the left is organized into three sections:

- Built-in:** Includes categories like Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures.
- Screen1:** Includes components like Button1 and TextToSpeech1.
- Any component:** A catch-all category for blocks from other components.

Two callout boxes provide additional information:

- The first box states: "Built-in Blocks are always available. They handle things like math, text, logic, and control."
- The second box states: "Component Blocks correspond to the components you've chosen for your app."

The "Viewer" area on the right shows a workspace with a trash bin icon and a "Show Warnings" button. A callout box points to the trash bin, stating: "Trash for deleting unneeded blocks." Another callout box points to the workspace area, stating: "Workspace where you assemble the blocks into a program."

Make a button click event

Click on the Button1 drawer. Click and hold the **when Button1.Click do** block. Drag it over to the workspace and drop it there. This is the block that will handle what happens when the button on your app is clicked. It is called an "Event Handler".

The screenshot displays the MIT App Inventor 2 Beta interface for a project named "TalkToMe". The interface is divided into three main sections: "Blocks", "Viewer", and a top navigation bar. The top bar includes the MIT App Inventor logo, the project name "TalkToMe", and navigation options like "Screen1", "Add Screen ...", and "Remove Screen".

The "Blocks" panel on the left shows a list of built-in components categorized by type: Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under the "Control" category, "Screen1" is expanded to show "Button1" and "TextToSpeech1". A red circle with the number "1" highlights the "Button1" component in this list.

The "Viewer" panel on the right shows a list of event handlers for "Button1": "when Button1.Click do", "when Button1.GotFocus do", "when Button1.LongClick do", and "when Button1.LostFocus do". A red circle with the number "2" highlights the "when Button1.Click do" block. An orange arrow points from this block to a separate "when Button1.Click do" block on the right, which is highlighted with a red circle and the number "3".

Below the event handlers, there are several property blocks for "Button1": "Button1.BackgroundColor", "set Button1.BackgroundColor to", "Button1.Enabled", and "set Button1.Enabled to".

Program the TextToSpeech action

Click on the TextToSpeech drawer. Click and hold the **call TextToSpeech1.Speak** block. Drag it over to the workspace and drop it there. This is the block that will make the phone speak. Because it is inside the Button.Click, it will run when the button on your app is clicked.

The screenshot displays the MIT App Inventor 2 Beta interface. The top navigation bar includes the MIT App Inventor logo, the text "MIT App Inventor 2 Beta", and menu items: "Project", "Connect", "Build", "Help", "My Projects", "Guide", and "Report an Issue". Below this is a green header for the project "TalkToMe", with buttons for "Screen1", "Add Screen ...", and "Remove Screen".

The interface is divided into two main sections: "Blocks" on the left and "Viewer" on the right.

Blocks Panel: A list of built-in blocks is shown, categorized by color: Control (orange), Logic (green), Math (blue), Text (red), Lists (light blue), Colors (grey), Variables (orange), and Procedures (purple). Under the "Screen1" category, "Button1" and "TextToSpeech1" are listed. "TextToSpeech1" is circled in orange and labeled with a black circle containing the number "1".

Viewer Panel: The workspace shows several code blocks. A "when TextToSpeech1 .AfterSpeaking" block is at the top. Below it is a "when TextToSpeech1 .BeforeSpeaking" block. A "call TextToSpeech1 .Speak message" block is circled in orange and labeled with a black circle containing the number "2". Below this are blocks for "TextToSpeech1 . Country", "set TextToSpeech1 . Country to", and "TextToSpeech1 . Language".

On the right side of the workspace, a "when Button1 .Click" block is shown, labeled with a black circle containing the number "3". Inside its "do" block is a "call TextToSpeech1 .Speak message" block. An orange arrow points from the "call TextToSpeech1 .Speak message" block in the "BeforeSpeaking" block to the "call TextToSpeech1 .Speak message" block inside the "Button1 .Click" block.

Fill in the message socket on TextToSpeech.Speak Block

Almost done! Now you just need to tell the TextToSpeech.Speak block what to say. To do that, click on the Text drawer, drag out a **text** block and plug it into the socket labeled "message".



The screenshot shows the Scratch IDE interface. The title bar reads "TalkToMe" and includes "Screen1", "Add Screen ...", and "Remove Screen" buttons. The "Blocks" panel on the left shows a tree view with "Built-in" and "Screen1" categories. Under "Built-in", the "Text" block category is highlighted with a red circle. The "Viewer" panel on the right shows a script area with a "when Button1 Click" block and a "do call TextToSpeech1 .Speak" block. The "message" socket of the ".Speak" block is highlighted with a red circle, and a red arrow points to it from a "text" block that is also highlighted with a red circle in the "Text" drawer.

Specify what the app should say when the button is clicked

Click on the text block and type in "Congratulations! You've made your first app." (Feel free to use any phrase you like, this is just a suggestion.)



The screenshot shows the completed script in the Scratch IDE. The script area contains a "when Button1 Click" block followed by a "do call TextToSpeech1 .Speak" block. The "message" socket of the ".Speak" block is now filled with a red "text" block containing the text "Congratulations! You've made your first app.".

Now test it out!

Go to your connected device and click the button. Make sure your volume is up! You should hear the phone speak the phrase out loud. (This works even with the emulator.)

