



## TalkToMe: A beginner App Inventor app

This step-by-step picture tutorial will guide you through making a talking app.

To get started, sign up for a free Google Account:  
<http://accounts.google.com/signup>

(If you already have a Google Account, skip this step.)

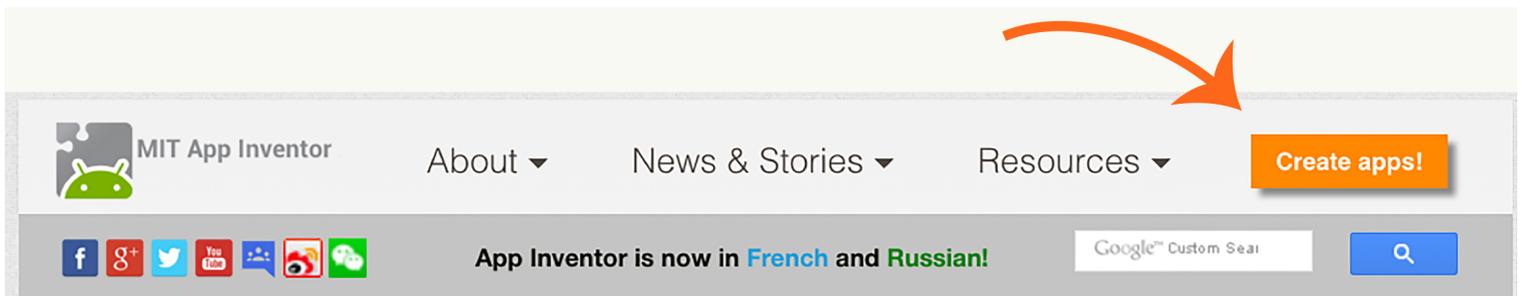
Create your Google Account

The screenshot shows the Google Account creation form with the following fields and sections:

- Name:** Two input fields for "First" and "Last" names.
- Choose your username:** An input field with "@gmail.com" as a placeholder and a link "I prefer to use my current email address".
- Create a password:** An input field.
- Confirm your password:** An input field.
- Birthday:** A "Month" dropdown menu, and "Day" and "Year" input fields.
- Gender:** A dropdown menu with "I am..." as the selected option.
- Mobile phone:** A dropdown menu for country code (showing "US") and an input field for the phone number.
- Your current email address:** An input field.

Go to the App Inventor home page: [www.appinventor.mit.edu](http://www.appinventor.mit.edu)

Click the orange "Create Apps" button in the menu bar.





MIT App Inventor  
appinventor.mit.edu

**Log in to App Inventor with your Gmail (or Google) username and password.**

Use an existing gmail account or school-based google account to log in to ai2.appinventor.mit.edu

(Example Gmail shown below.)

Google

One account. All of Google.

Sign in with your Google Account

kelliak@[redacted].com  
.....  
Sign In  
 Stay signed in    [Need help?](#)

[Create an account](#)

Choose a Google account if you have more than one. Click "Allow".

Google accounts

**The application MIT AppInventor Version 2 is requesting permission to access your Google Account.**

**Please select an account that you would like to use.**

- kelliak@[redacted].com
- kelliak@[redacted].com
- kelliak@[redacted].com

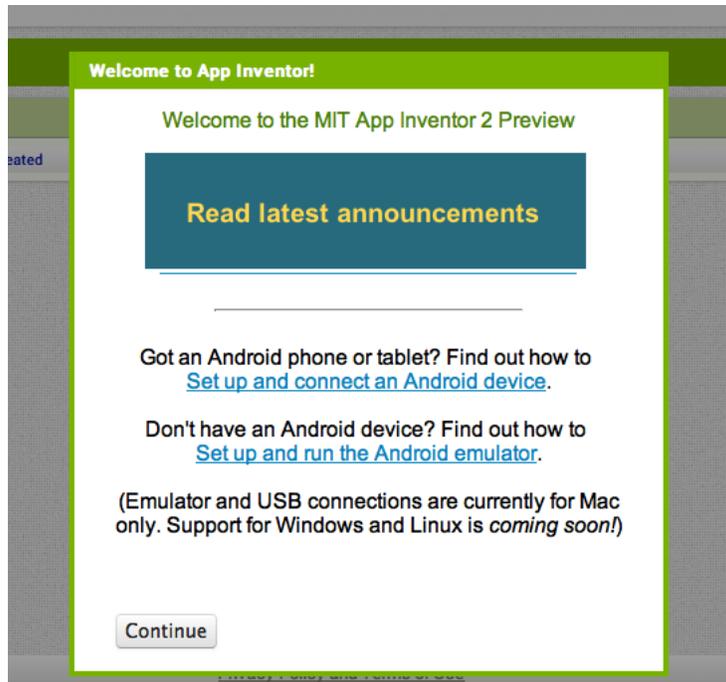
Google is not affiliated with the contents of **MIT AppInventor Version 2** or its owners. If you sign in, Google will share your email address with **MIT AppInventor Version 2** but not your password or any other personal information.

[Sign in to another account](#)

©2011 Google - [Google Home](#) - [Terms of Service](#) - [Privacy Policy](#) - [Help](#)



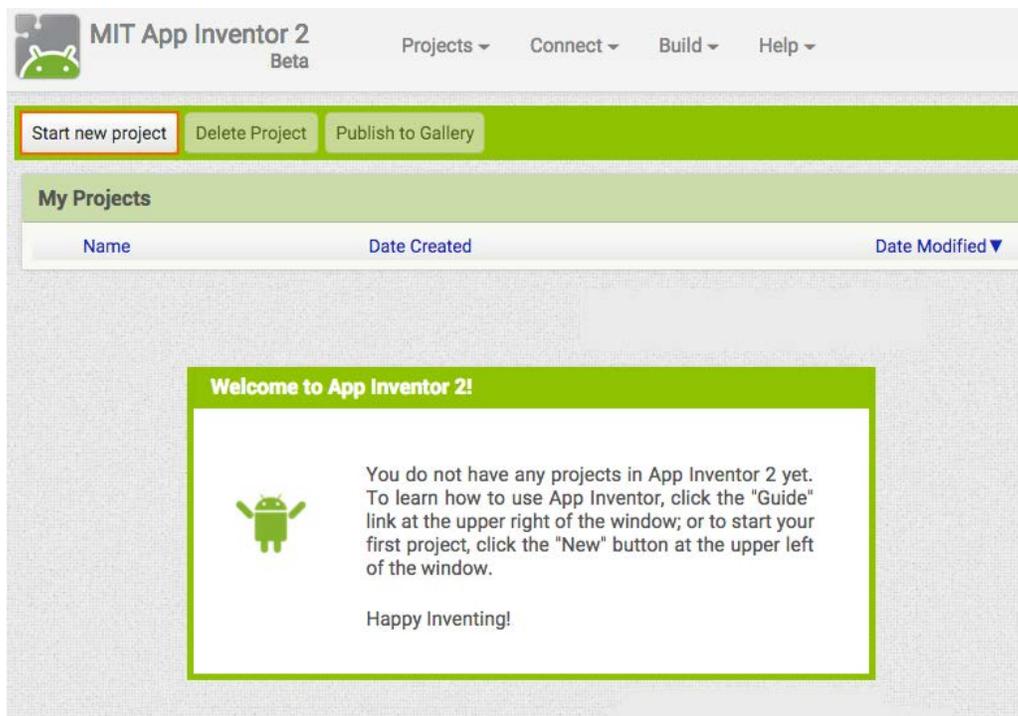
Read the App Inventor announcements, then click "Continue".



If you don't have any projects created in App Inventor, you will land in the Projects View.

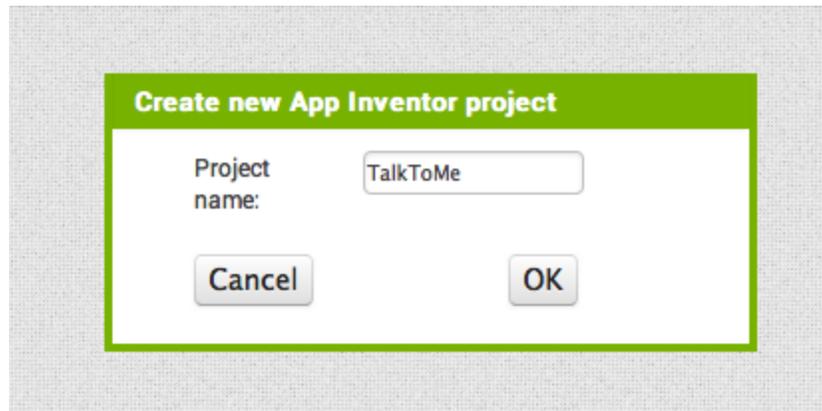
Start a new project by clicking the "Start new project" button.

(If you have already created projects, App Inventor will open the most recent project.)



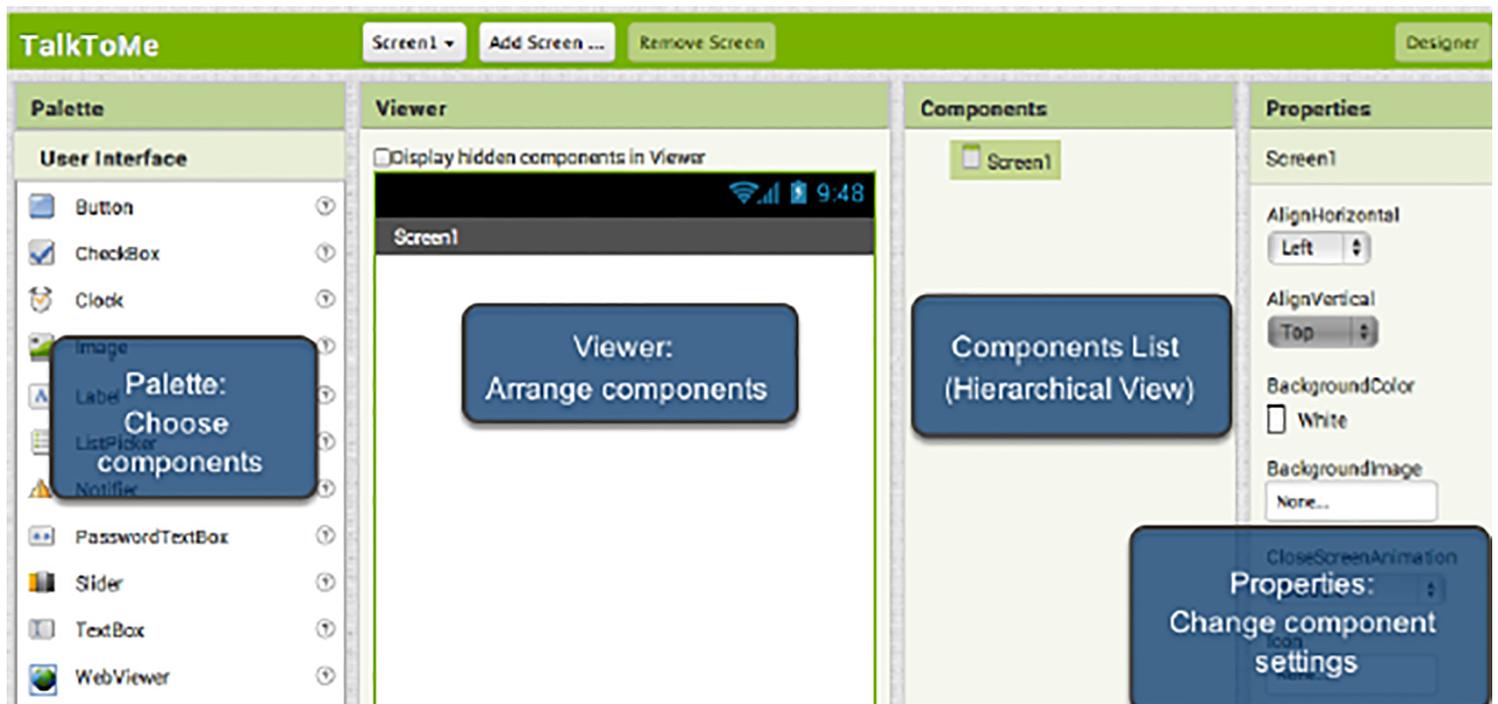
## Name the project "TalkToMe" (no spaces)

Type in the project name (underscores are allowed, spaces are not) and click OK.



## App Inventor opens the Designer window

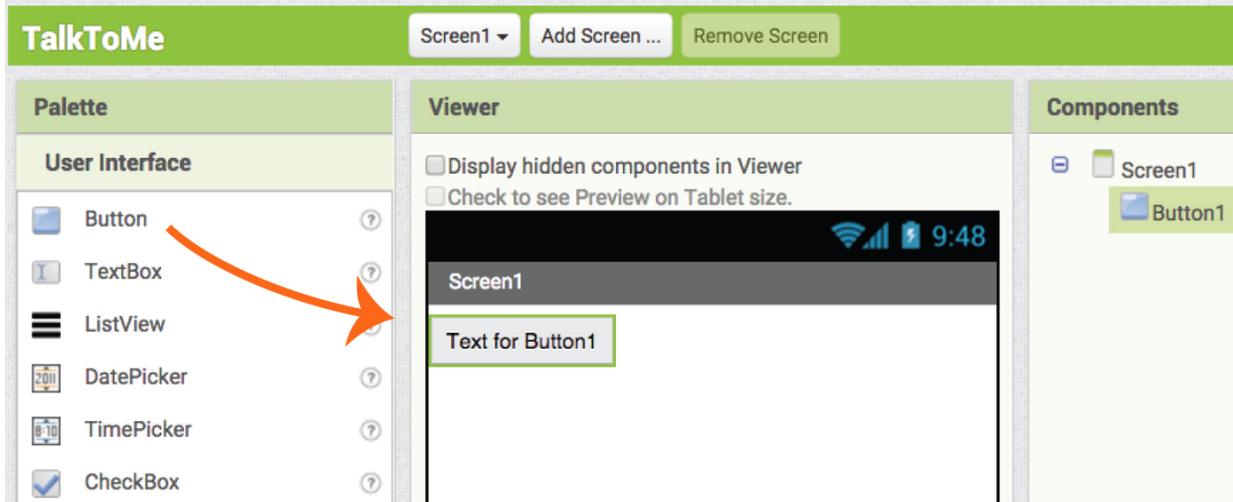
The "Designer" is where you create the Graphical User Interface (GUI) or the look and feel of your app. You choose components like Buttons, Images, and Text boxes, and functionalities like Text-to-Speech, Sensors, and GPS.



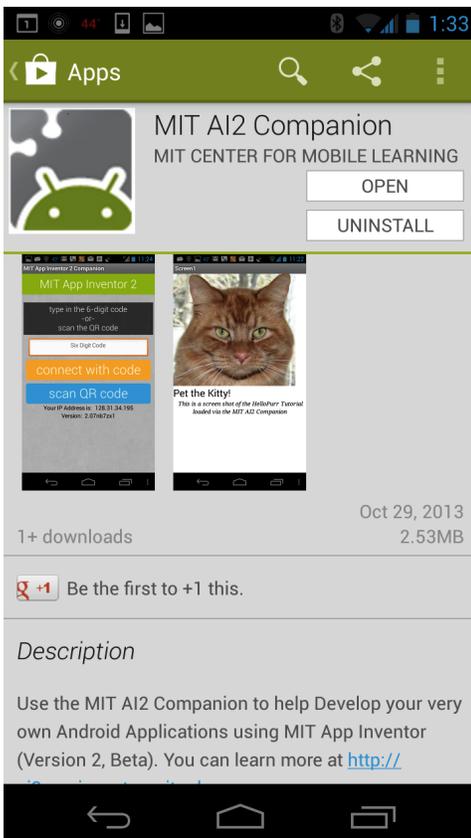


## Add a Button

Click and hold on the word "Button" in the Palette. Drag your mouse over to the Viewer. Release the mouse. A new button will appear on the Viewer.



## Download the MIT AI2 Companion App from the Google Play Store and install it on your phone or tablet.



Getting the app through Google Play Store is preferred because you will get automatic updates.

If your mobile device has a QR code reader app installed, you can scan the QR code image below. It will take you to the AI2 Companion app in the Google Play store where you can download it by clicking the "install" button. Then, you will find the app in your Downloads folder on your device.



OR

Search directly for "MIT AI2 Companion" on Google Play Store, <https://play.google.com/store> and then install the app by clicking "install".



MIT App Inventor

appinventor.mit.edu

**If you were unable to download the AI2 Companion App from the Google Play Store, you can follow these step to download it directly to your device.**

NOTE: Direct APK download (requires manual updates)

Go into your phone's settings, choose "Security"

Scroll down and allow "Unknown Sources" by checking the box.

(This allows apps that are not from the Play Store to be installed on the device.)

Scan this QR code



OR

Type this URL into a web browser on your device: <http://appinv.us/companion>.

The AI2 Companion app will automatically download.

Regardless of which method you use, scanning the QR code or directly typing the link into your device's browser, you will see a message similar to this:

This type of file can harm your device. ✕  
Do you want to keep  
MITAI2Companion.apk anyway?

CANCEL

OK

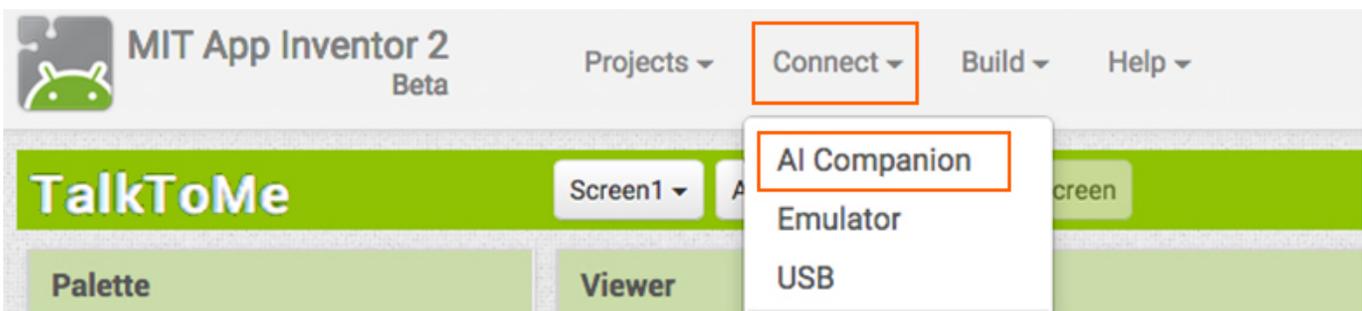
Click "OK". (Don't worry, the AI2 Companion app will not harm your mobile device.)

**If you don't have a mobile device, you can still use App Inventor by connecting with the emulator. Visit: <http://appinventor.mit.edu/explore/ai2/setup.html> and follow the instructions under Option 2.**

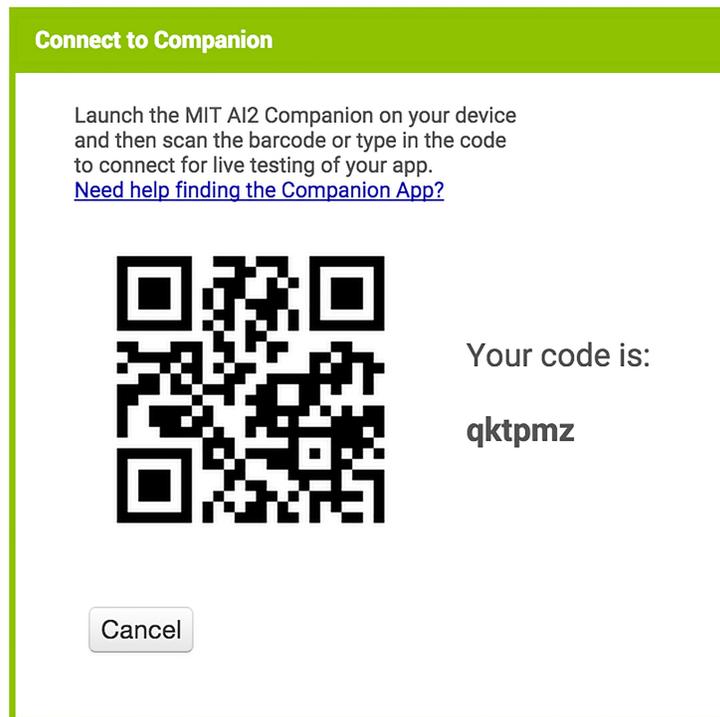
**Once you've installed the AI2 Companion app, you can connect your App Inventor project to your phone or tablet for live testing**

While you're building an app on your computer, you can test it on a connected Android phone or tablet.  
**Be sure your computer and mobile device are connected to the same WiFi network.**

Return to the Designer Window on your computer.  
Click Connect and choose AI Companion from the drop down menu



A QR code and a 6 character code will appear on the screen of your computer screen.



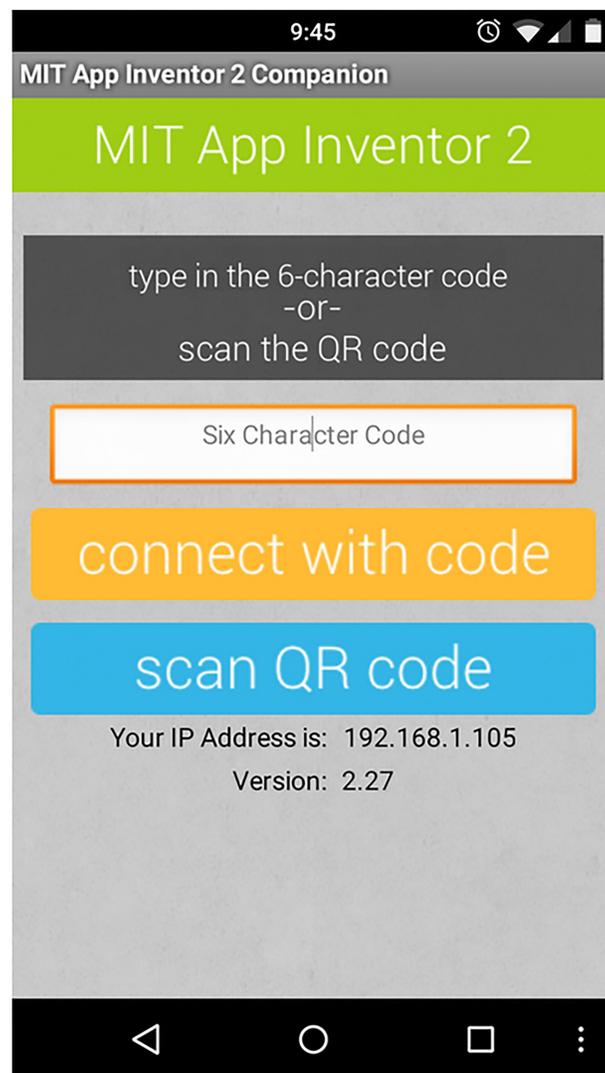


Open the AI2 Companion app on your device by clicking on the app icon.

A screen (like the one shown below) will appear with the option to scan the QR code or type in the six character code.

If you choose to scan the code, press the blue "scan QR code" button for the scanner to launch. Scan the QR code. Wait a few seconds for your app to open on your mobile device.

If you choose to use the code, type it into the white text box, click the orange button afterwards.



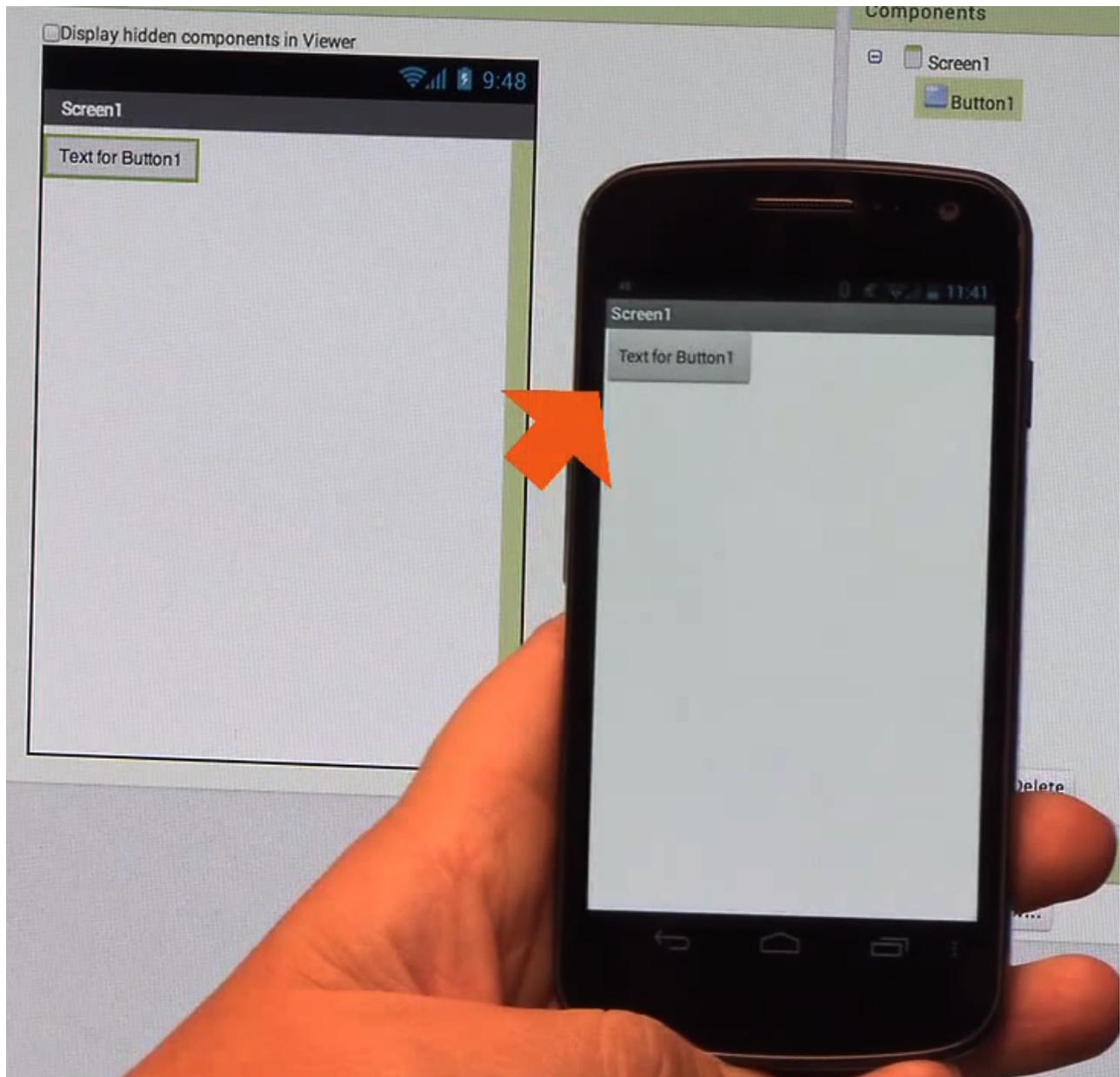
If you can not connect over wifi, go to the Setup Instructions on the App Inventor Website to find out how to connect with a USB cable. <http://appinventor.mit.edu/explore/ai2/setup.html>

## See your app on your connected device

You will know that your connection is successful when you see your app on the connected device.

Since our app only has a button, that is what you will see on your mobile device.

As you add more components to the project, your app will update on your computer and your phone.

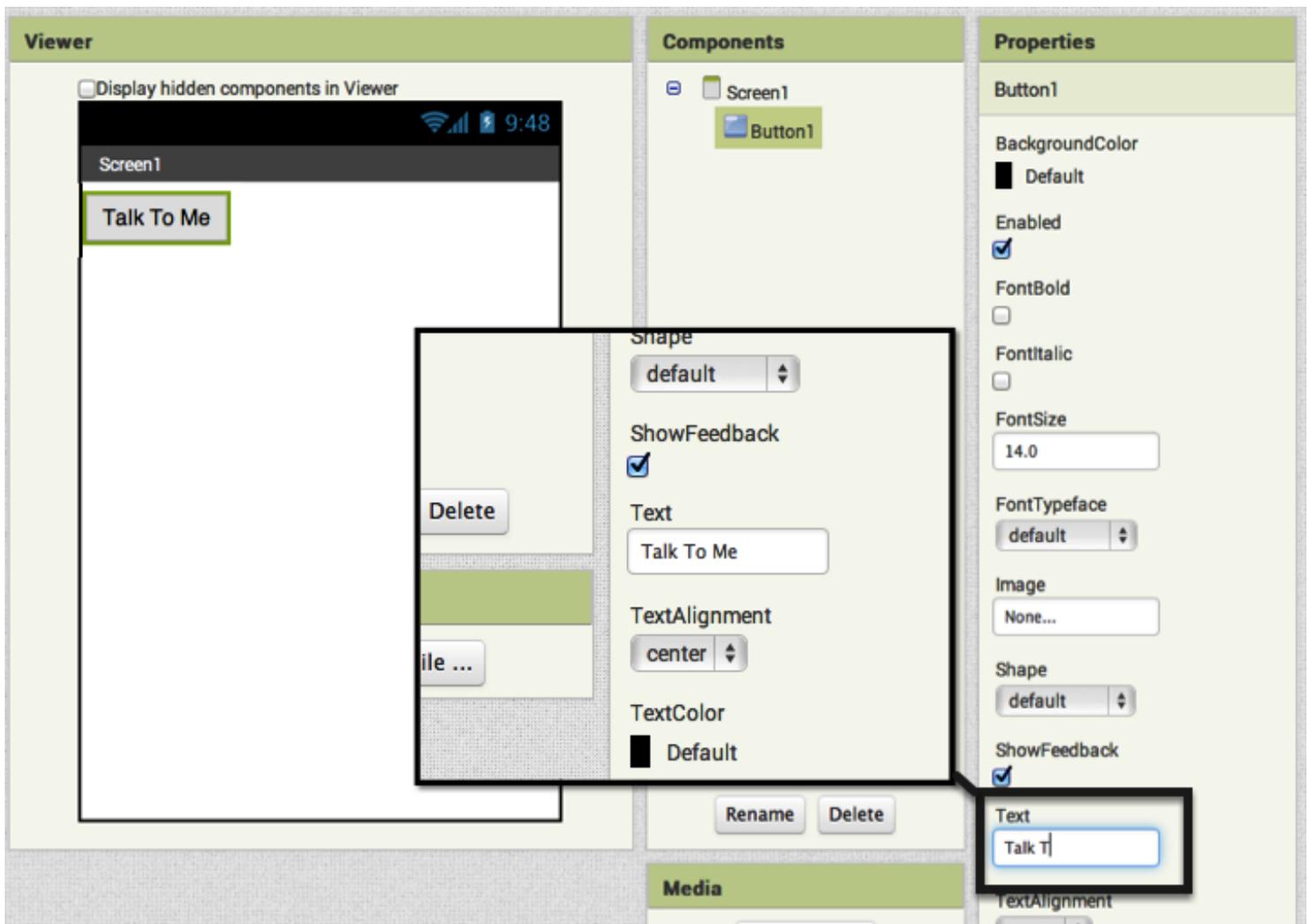


## Change the Text on the Button

In the properties panel, change the text for the Button.

Under the Text property, select "Text for Button 1", delete it and type in "Talk To Me".

Notice that the text on your app's button changes right away too.

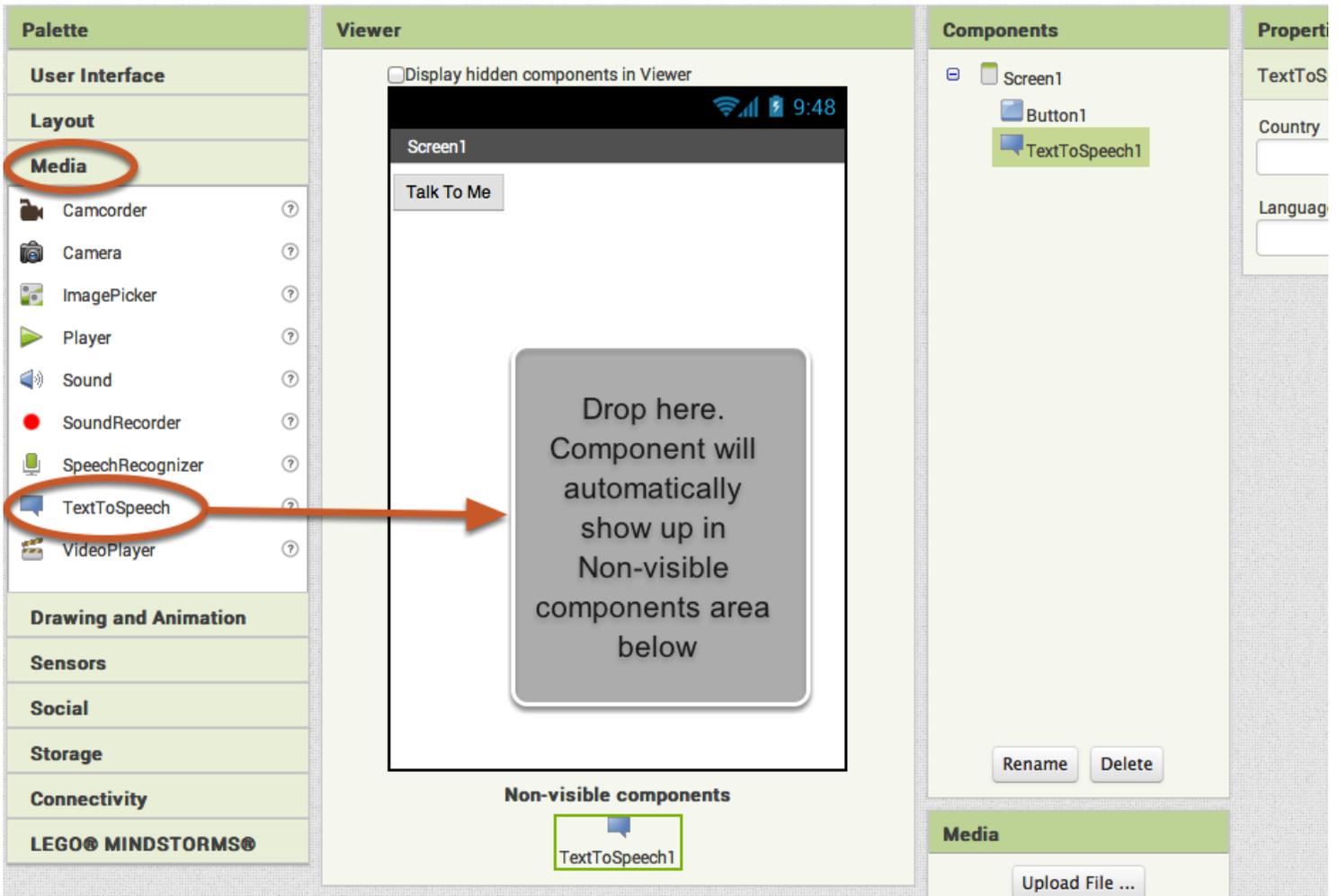


## Add a Text-to-Speech component to your app

Go to the Media drawer in the Palette and drag out a TextToSpeech component.

Drag and 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 MIT App Inventor interface with three main panels: Palette, Viewer, and Components. The Palette panel on the left is divided into several categories: User Interface, Layout, Media, Drawing and Animation, Sensors, Social, Storage, Connectivity, and LEGO® MINDSTORMS®. The Media category is highlighted with a red circle, and the TextToSpeech component is also circled in red. An orange arrow points from the TextToSpeech component in the Palette to the Viewer panel. The Viewer panel shows a mobile app interface with a button labeled "Talk To Me". Below the main app view, there is a "Non-visible components" area containing a single component labeled "TextToSpeech1". A grey box with the text "Drop here. Component will automatically show up in Non-visible components area below" is positioned over the Viewer area. The Components panel on the right shows a tree view with "Screen1" containing "Button1" and "TextToSpeech1". The Properties panel on the far right shows properties for "TextToSpeech1", including "Country" and "Language".

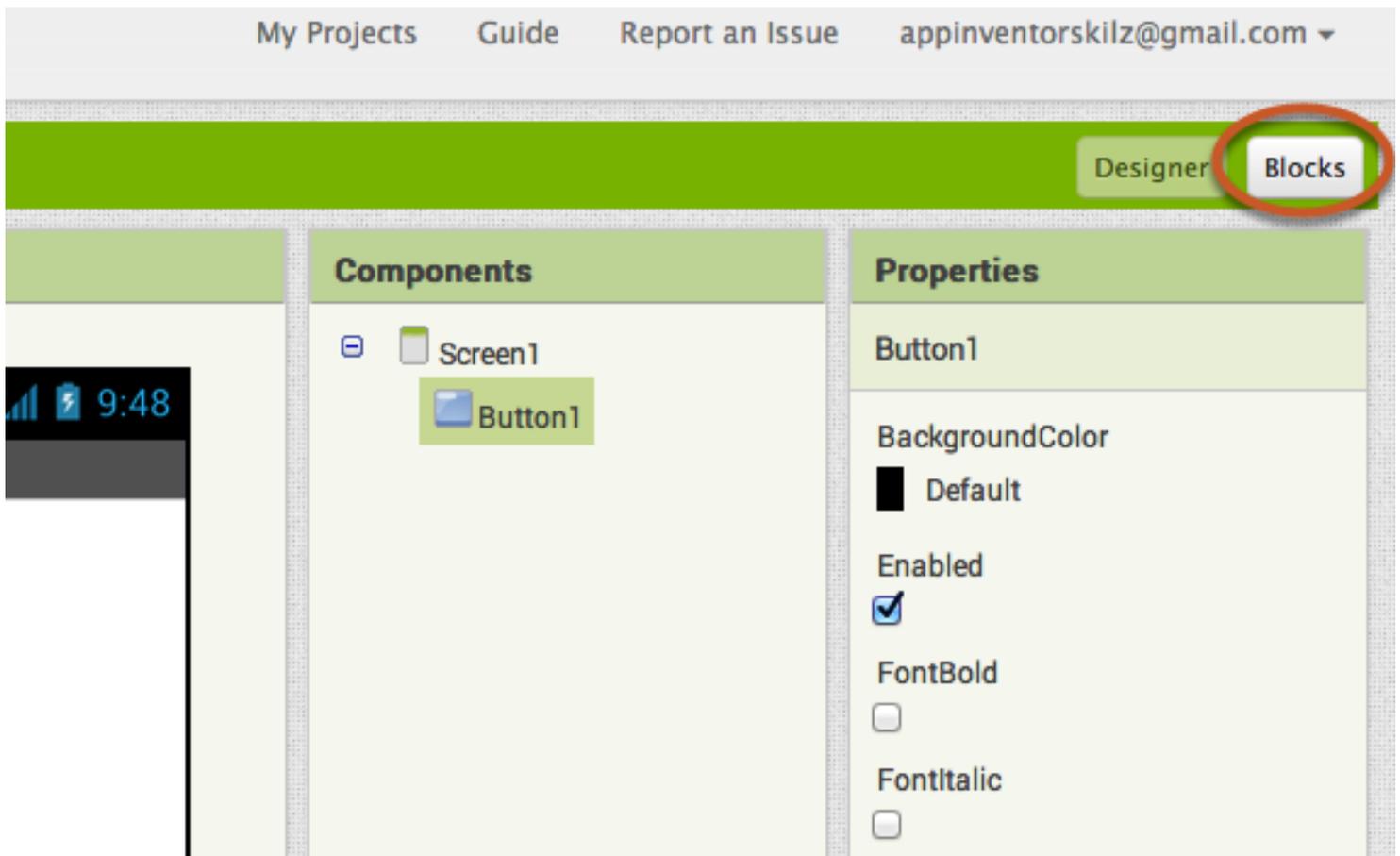


## Switch over to the Blocks Editor

It's time to tell your app what to do. The Blocks Editor is where you program the behavior of your app.

Click the button "Blocks" to move over to the Blocks Editor.

You will often toggle between the Designer and Blocks Editor as you develop apps.





## The Blocks Editor

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 you add to your app.

*(In order to get the blocks for a certain component to show up in the Blocks Editor, you first add that component to your app in the Designer.)*

The screenshot shows the MIT App Inventor 2 interface. At the top, the title bar reads "MIT App Inventor 2 Beta" and includes navigation menus for "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". Below the title bar, the app name "TalkToMe" is displayed, along with "Screen1" and buttons for "Add Screen ..." and "Remove Screen". On the far right, there are tabs for "Designer" and "Blocks".

The main workspace is divided into two sections:

- Blocks:** A vertical sidebar on the left containing a tree view of available blocks. It is organized into categories: "Built-in" (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), "Screen1" (Button1, TextToSpeech1), and "Any component".
- Viewer:** A large central area where blocks are assembled. It contains three callout boxes:
  - A grey box on the left stating: "Built-in Blocks are always available. They handle things like math, text, logic, and control." This box is positioned next to the "Built-in" category in the Blocks sidebar.
  - A larger grey box below it stating: "Component Blocks correspond to the components you've chosen for your app." This box is positioned next to the "Screen1" and "Any component" categories.
  - A tan box on the right with four red arrows pointing outwards, stating: "Workspace where you assemble the blocks into a program."
  - A tan box at the bottom right with a trash can icon, stating: "Trash for deleting unneeded blocks."

At the bottom of the workspace, there are warning indicators: a yellow triangle with "0" and a red triangle with "0", followed by a "Show Warnings" button. At the bottom left of the interface, there are "Rename" and "Delete" buttons.



## Make a button click event

Click on the Button1 drawer.

Click and hold the **when Button1.Click do** event block.

Drag it over to the Viewer and drop it there.

This block will launch when the button on your app is clicked.

It is called an "Event Handler".

The screenshot shows the MIT App Inventor 2 Beta interface. The top navigation bar includes "Project", "Connect", "Build", and "Help" menus, along with "My Projects", "Guide", and "Rep" links. The main workspace is titled "TalkToMe" and contains a "Screen1" drawer with "Add Screen ..." and "Remove Screen" buttons. The "Blocks" panel on the left is categorized into "Built-in" (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures) and "Any component". The "Screen1" drawer is expanded, and the "Button1" component is circled in red, labeled with a "1". The "Viewer" panel on the right shows a list of event blocks for "Button1": ".Click", ".GotFocus", ".LongClick", and ".LostFocus". The ".Click" event block is circled in red, labeled with a "2". An orange arrow points from this block to a yellow event block in the "Viewer" panel, labeled with a "3". The yellow block is a "when Button1.Click do" event handler.



## Program the TextToSpeech action

Click on the TextToSpeech drawer.

Click and hold the **call TextToSpeech1.Speak** block.

Drag it over to the Viewer 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 shows the MIT App Inventor 2 Beta interface. The top navigation bar includes 'Project', 'Connect', 'Build', and 'Help' menus, along with 'My Projects', 'Guide', and 'Report an Issue' links. The main workspace is titled 'TalkToMe' and contains a 'Screen1' drawer with 'Add Screen ...' and 'Remove Screen' buttons. On the left, the 'Blocks' palette is visible, with 'TextToSpeech1' highlighted under the 'Screen1' category, marked with a circled '1'. In the center, the 'Viewer' area shows a 'when TextToSpeech1 .AfterSpeaking' block with a 'result' block and a 'do' block. Below it is a 'when TextToSpeech1 .BeforeSpeaking' block with a 'do' block. A 'call TextToSpeech1 .Speak message' block is being dragged from the 'TextToSpeech1' drawer to the 'do' block of the 'when Button1 .Click' event, marked with a circled '2'. The 'when Button1 .Click' event block is marked with a circled '3'. Below the event blocks, there are three 'TextToSpeech1' blocks: '. Country', 'set TextToSpeech1 . Country to', and '. Language'.



## Fill in the message socket on TextToSpeech.Speak Block

Now you need to tell the TextToSpeech.Speak block what to say.

Click on the Text drawer, drag out a **text** block and plug it into the socket labeled "message".

The screenshot shows the MIT App Inventor interface. The top bar is green and contains the app name 'TalkToMe', a dropdown menu for 'Screen1', and buttons for 'Add Screen ...' and 'Remove Screen'. Below this is a 'Blocks' panel on the left and a 'Viewer' panel on the right. The 'Blocks' panel has a 'Built-in' section with various drawers: Control, Logic, Math, Text (highlighted with a red circle), Lists, Colors, Variables, and Procedures. Below this is a 'Screen1' section with 'Button1' and 'TextToSpeech1'. The 'Viewer' panel shows a script starting with 'when Button1 Click' followed by 'do call TextToSpeech1 .Speak'. The 'message' socket of the .Speak block is highlighted with a red circle, and an orange arrow points to it from a 'text' block in the 'Text' drawer, which is also highlighted with a red circle.

## 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.)

The screenshot shows the MIT App Inventor script editor. The script starts with 'when Button1 Click' followed by 'do call TextToSpeech1 .Speak'. The 'message' socket of the .Speak block is filled with the text 'Congratulations! You've made your first app.'

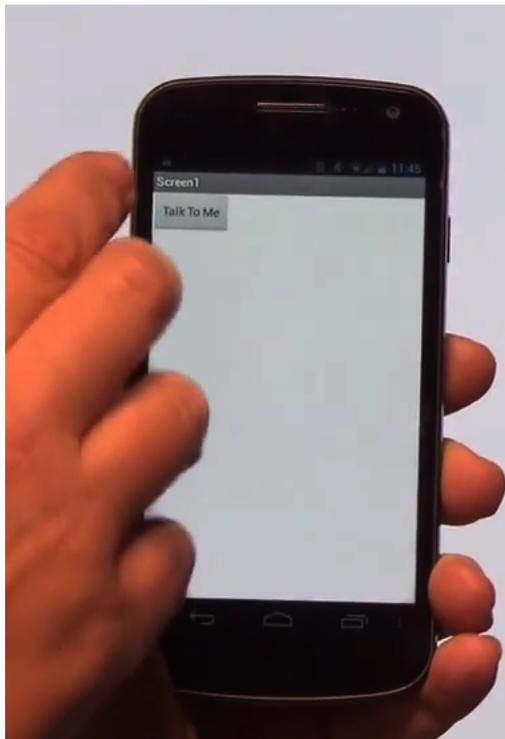


### Now test it out!

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.)



### Next up: TalkToMe Part 2

You will expand the app to respond to shaking and to let users enter their own text.

Great job!