



MIT App Inventor
appinventor.mit.edu

TalkToMe: Your first App Inventor app

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

To get started, go to App Inventor on the web.

Go directly to ai2.appinventor.mit.edu, or click the orange "Create" button from the App Inventor website.

MIT App Inventor

Home Blog ▼ Support ▼ **Create**

Follow Us: [f](#) [t](#) [y](#) [m](#)

Google™ Custom Search

Your ideas.
Your designs.
Your apps.

Invent Now

Get Started
Follow these simple steps to build your first app.
Get Started

Create
Design and program your own apps using MIT App Inventor.
Create

Tutorials
Step-by-step guides show you how to build all kinds of apps.
Tutorials



MIT App Inventor
appinventor.mit.edu

Log in to App Inventor with a gmail (or google) user name and password.

Use an existing gmail account or school-based google account to log in to ai2.appinventor.mit.edu
To set up a brand new gmail account, go to accounts.google.com/SignUp



One account. All of Google.

Sign in with your Google Account

appinventorskilz@gmail.com

.....

Sign in

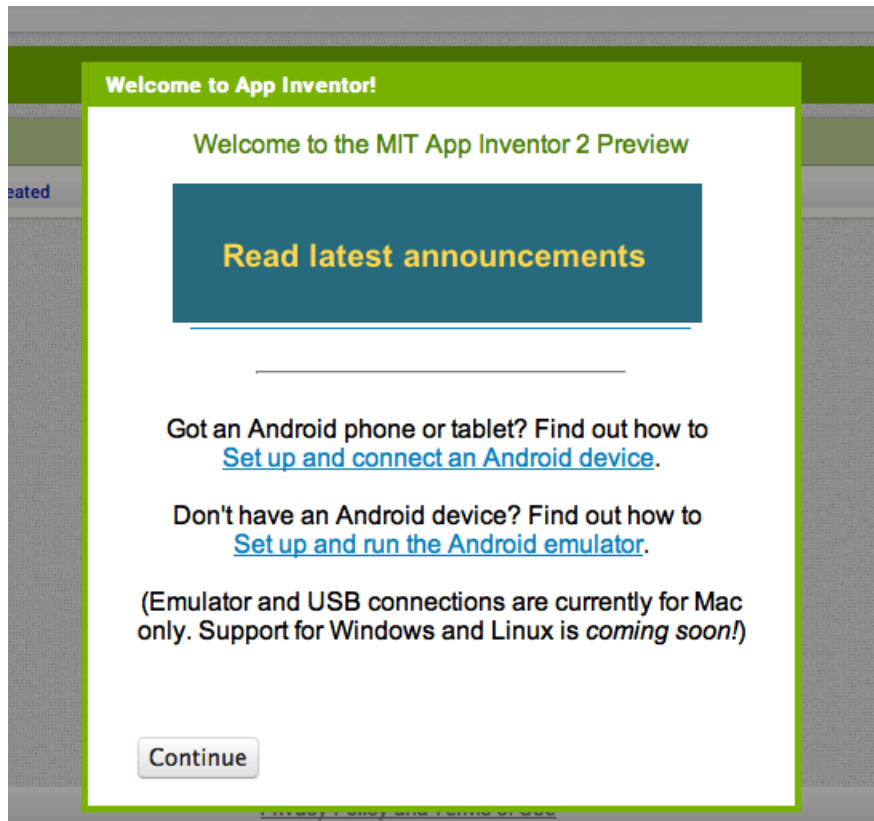
☒ Stay signed in [Need help?](#)

[Create an account](#)



MIT App Inventor
appinventor.mit.edu

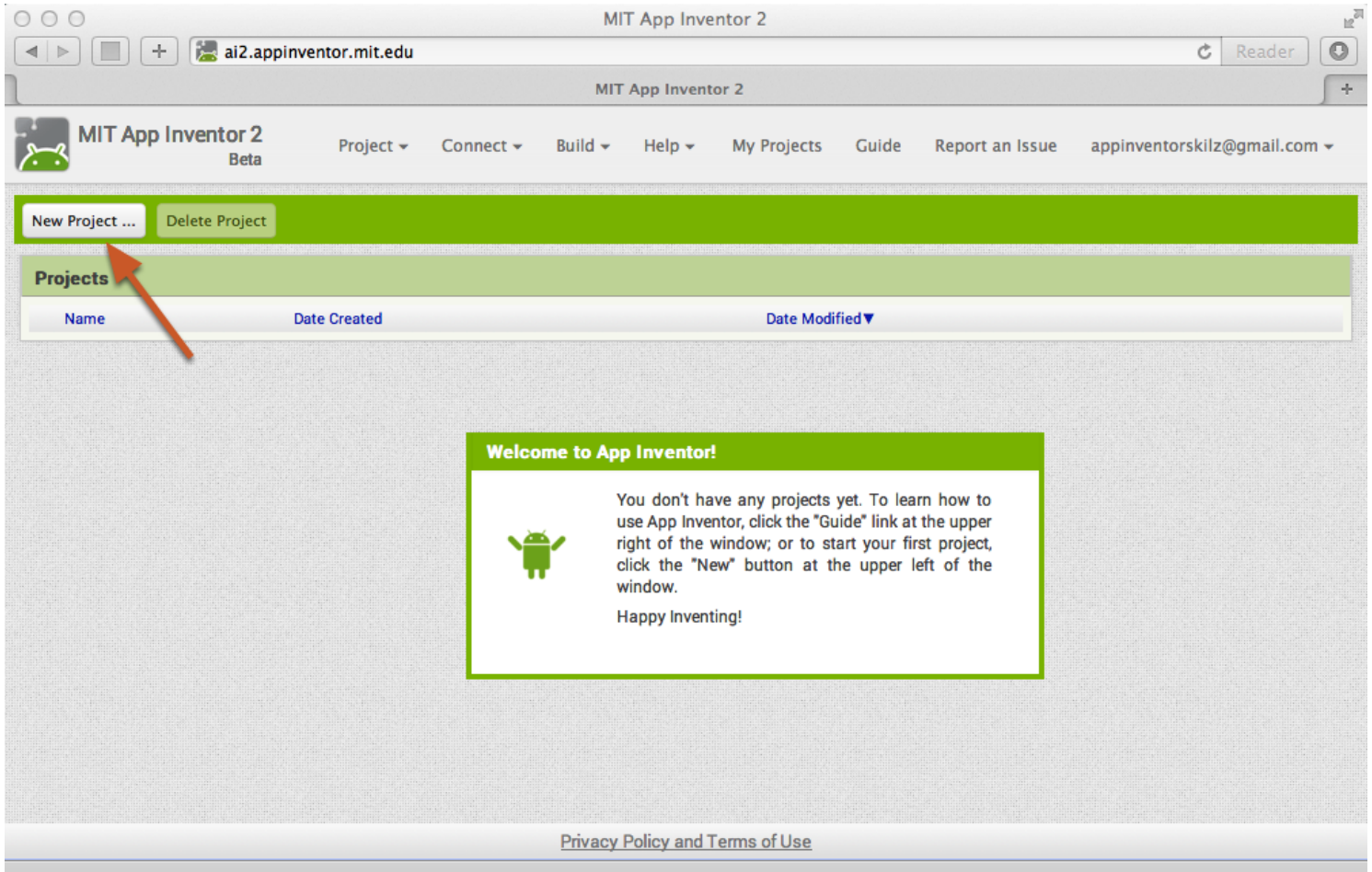
Click "Continue" to dismiss the splash screen.





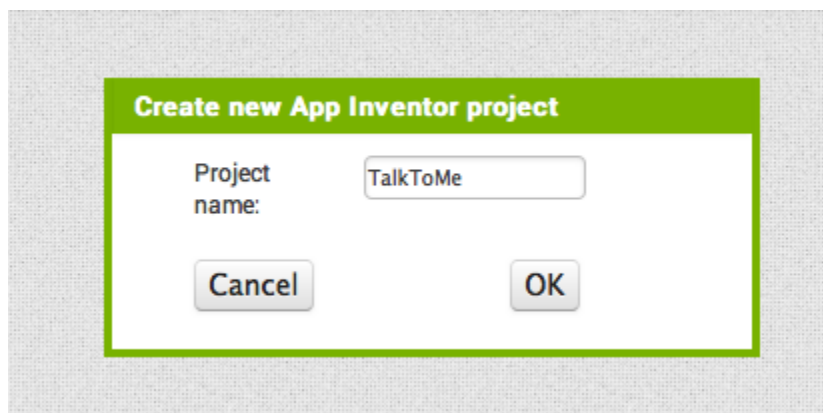
MIT App Inventor
appinventor.mit.edu

Start a new project.



Name the project "TalkToMe" (no spaces!)

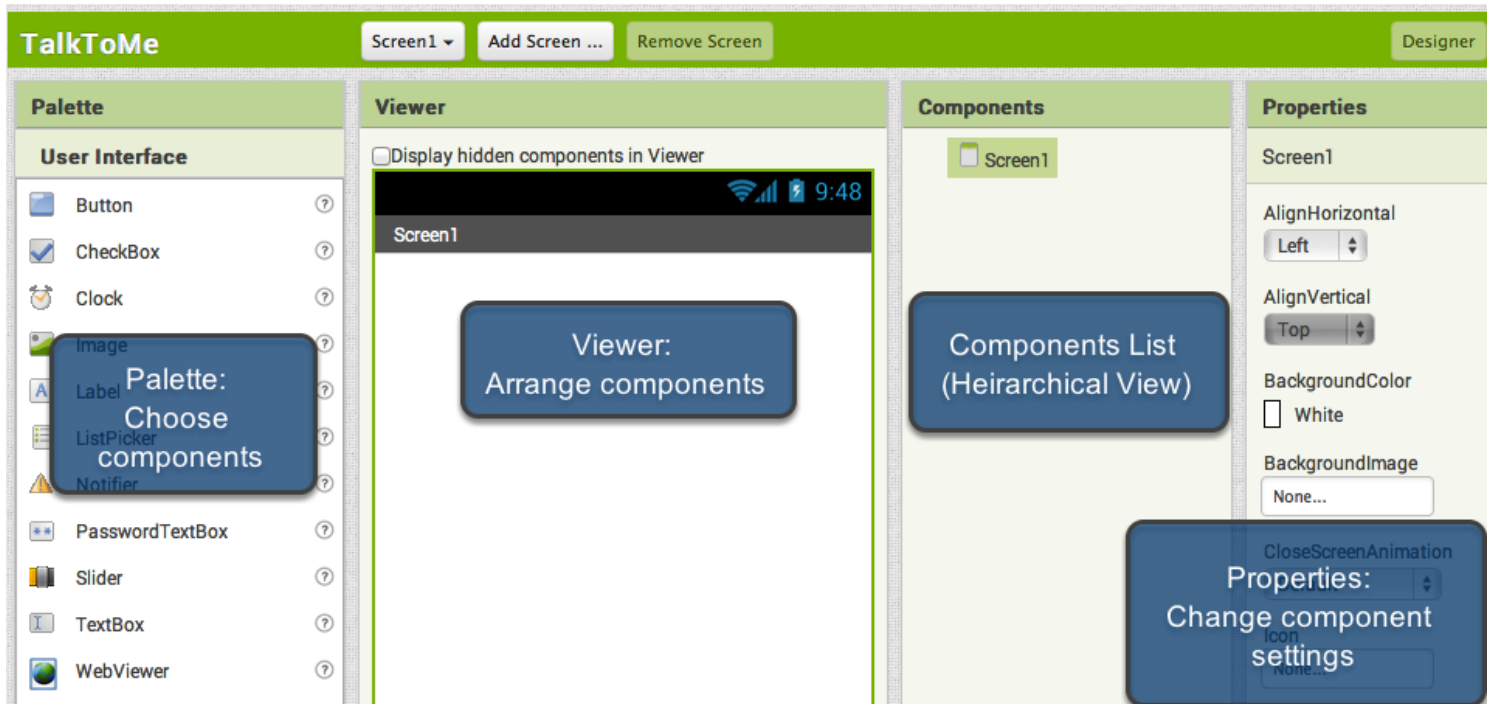
Type in the project name (underscores are allowed, spaces are not) and click OK.





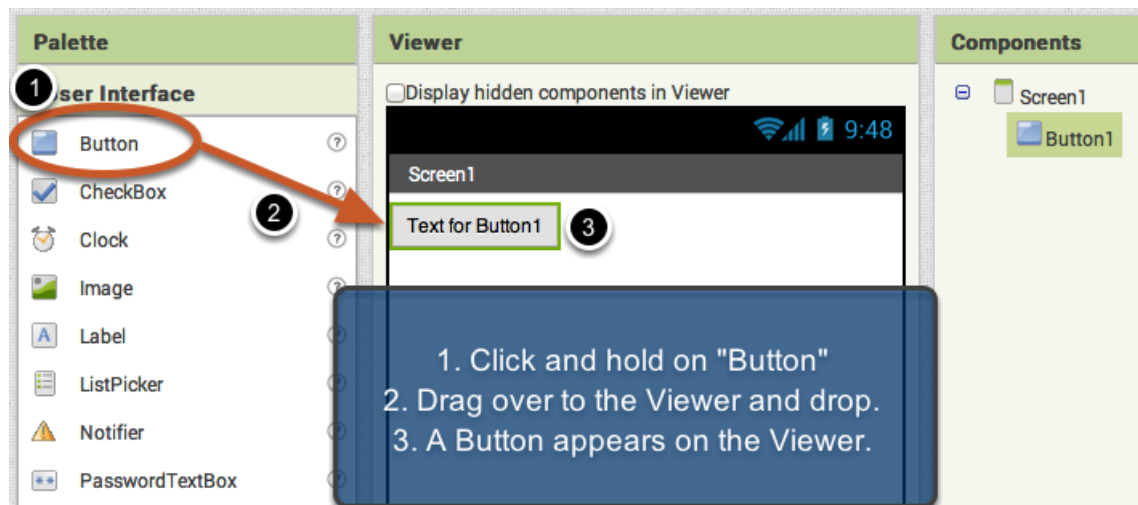
You are now in the Designer, where you lay out the "user interface" of your app.

The Design Window, or simply "Designer" is where you lay out the look and feel of your app, and specify what functionalities it should have. You choose things for the user interface things like Buttons, Images, and Text boxes, and functionalities like Text-to-Speech, Sensors, and GPS.



Add a Button

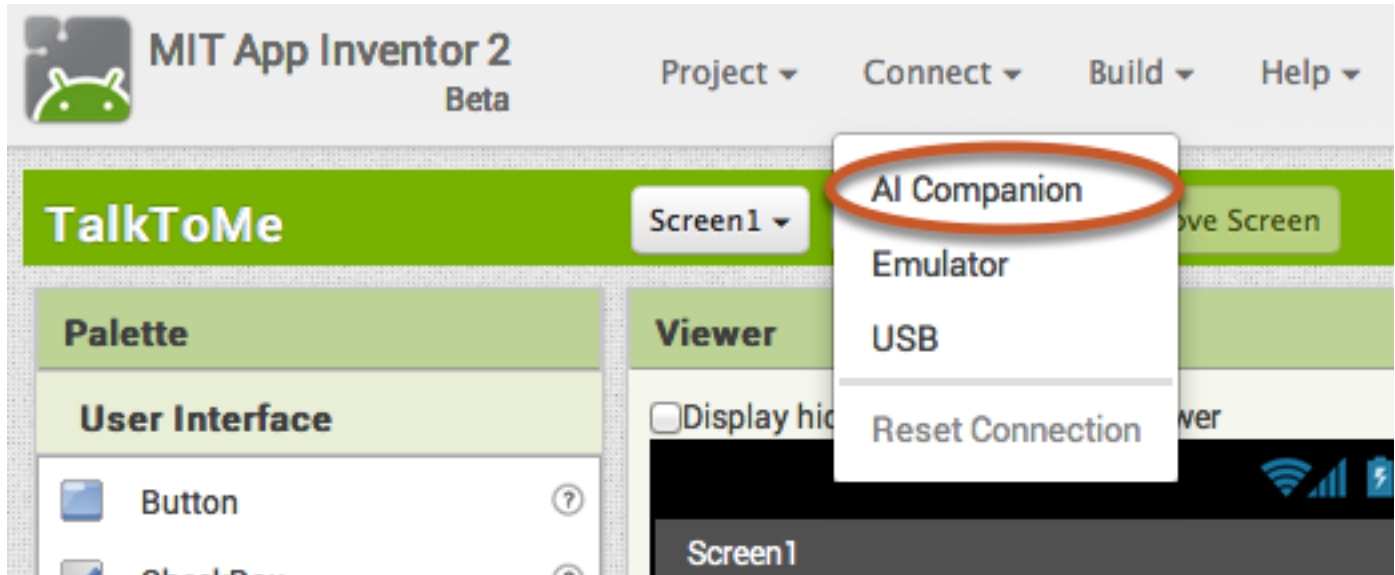
Our project needs a button. **Click and hold** on the word "Button" in the palette. **Drag** your mouse over to the Viewer. **Drop** the button and a new button will appear on the Viewer.





Connect App Inventor to your phone for live testing

One of the neatest things about App Inventor is that you can see and test your app while you're building it, on a connected device. If you have an **Android phone or tablet**, follow the steps below. *If you do not have a device*, then follow the instructions for [setting up the on-screen emulator](#) (opens a new page) and then come back to this tutorial once you've gotten the emulator connected to App Inventor.

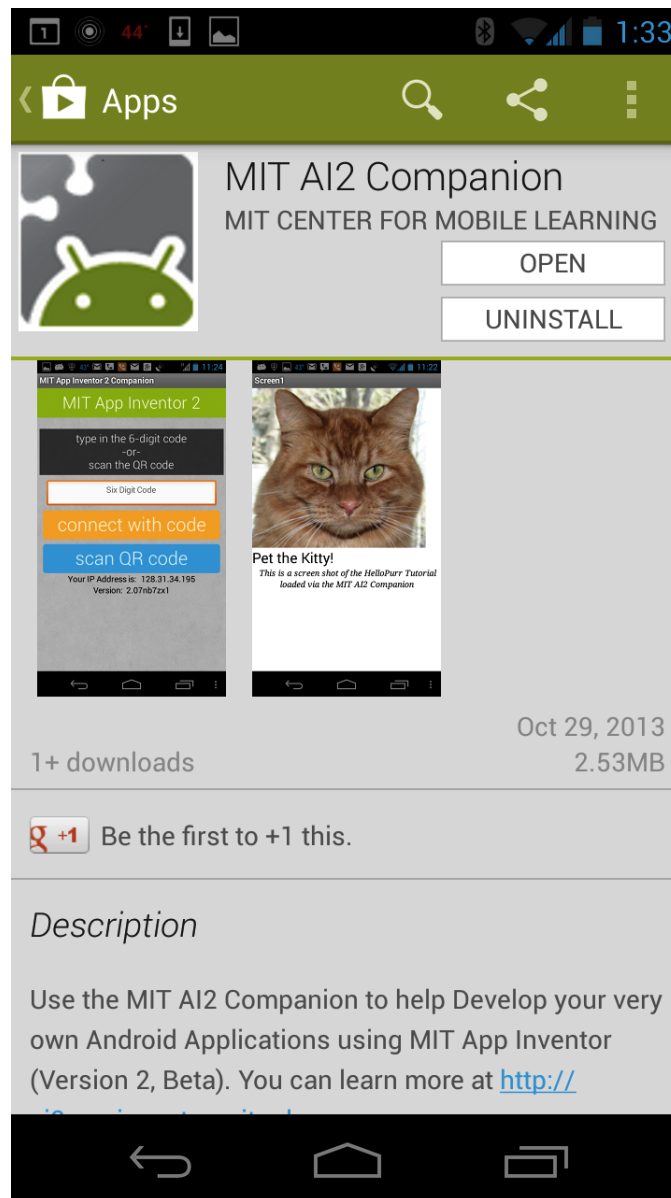




MIT App Inventor
appinventor.mit.edu

Get the MIT AI2 Companion from the Play Store and install it on your phone or tablet.

The preferred method for getting the AI2 Companion App is to **download the app from the Play Store by searching for "MIT AI2 Companion"**.





To download the AI2 Companion App to your device directly (SKIP THIS STEP IF YOU already got the app from Play Store)

If for some reason you can not connect to the Google Play store, you can download the AI2 Companion as described here.

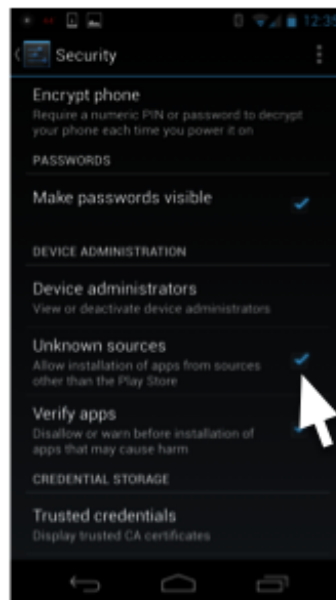
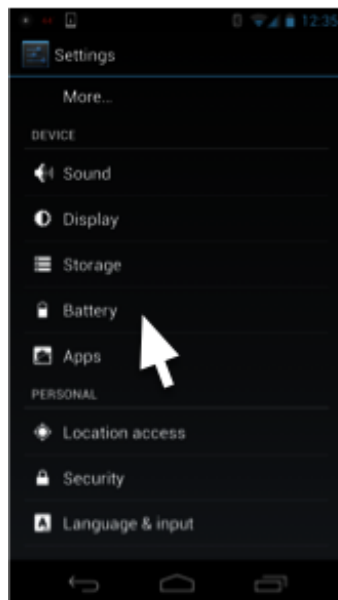
First, you will need to go into your phone's settings (#1), choose "Security", then scroll down to allow "Unknown Sources", which allows apps that are not from the Play Store to be installed on the phone.

Second, do one of the following:

A) **Scan the QR code above (#2)**

or

B) Click the "Need help finding..." link and you'll be taken to the download page. From there you can download the MITAI2Companion.apk file to your computer and then move it over to your device to install it.



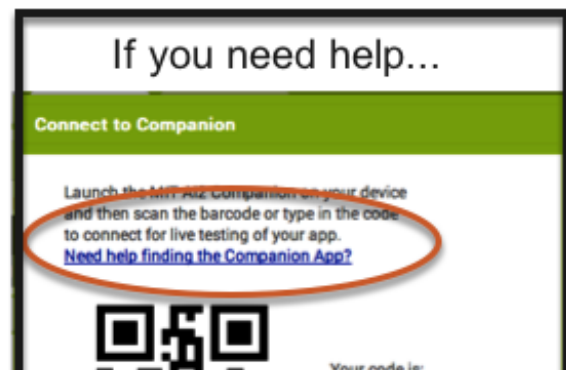
SKIP THIS STEP if you already got the AI2 Companion from the Play Store

1

1. Open your phone's settings and click "Security".
2. CHECK the box for "Unknown sources"

2

Scan to download
MIT AI2 Companion
directly to phone

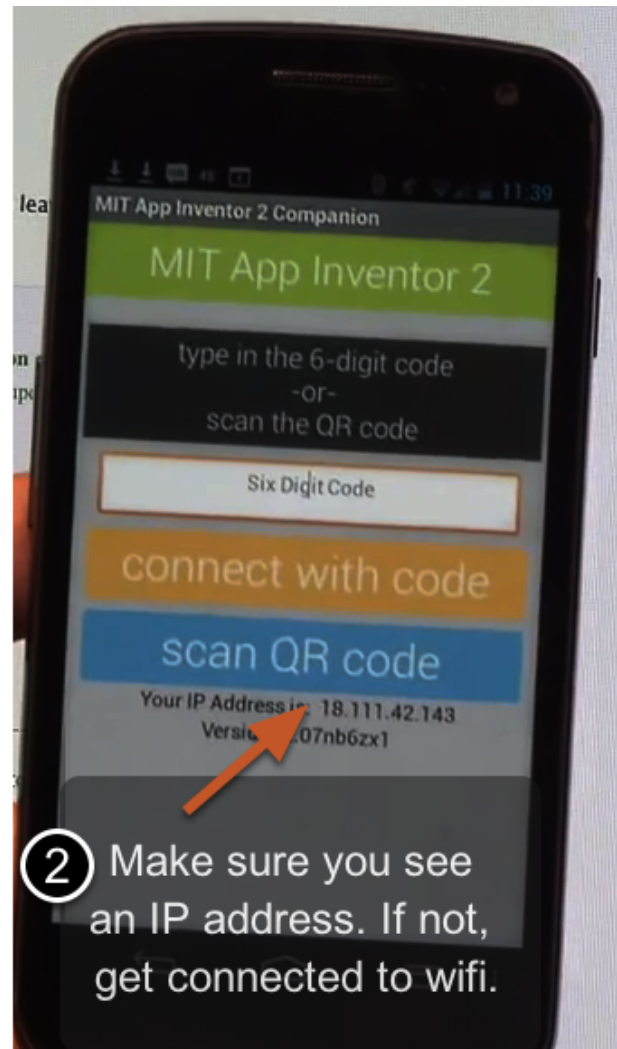
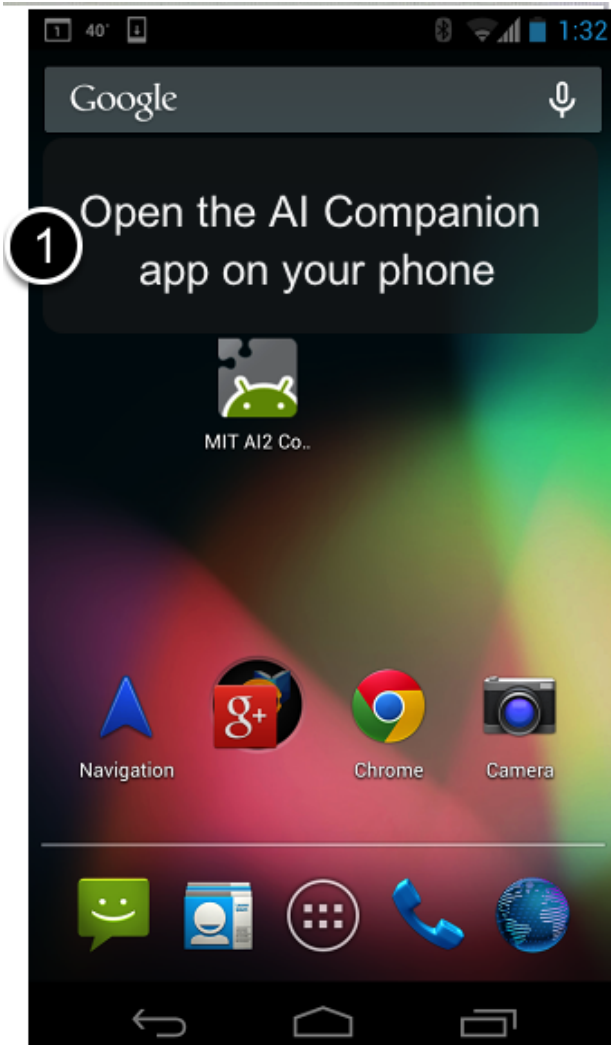




MIT App Inventor
appinventor.mit.edu

Start the AICompanion on your device

On your phone or tablet, click the icon for the MIT AI Companion to start the app. NOTE: Your **phone and computer must both be on the same wireless network**. Make sure your phone's wifi is on and that you are connected to the local wireless network. 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.





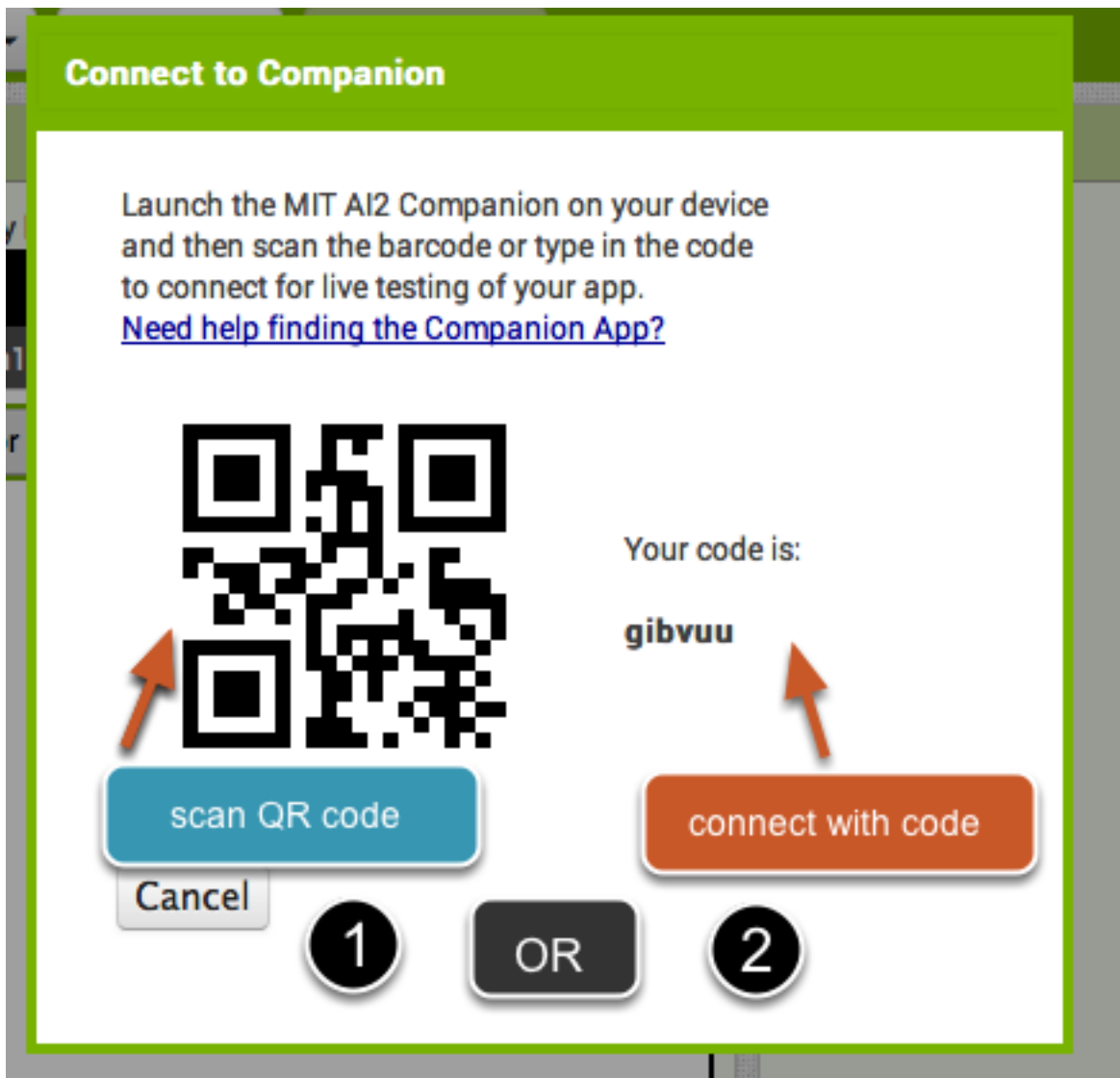
Get the Connection Code from App Inventor and scan or type it into your Companion app

On the Connect menu, choose "AI Companion". You can connect by:

1 - Scanning the QR code by clicking "Scan QR code" (#1).

or

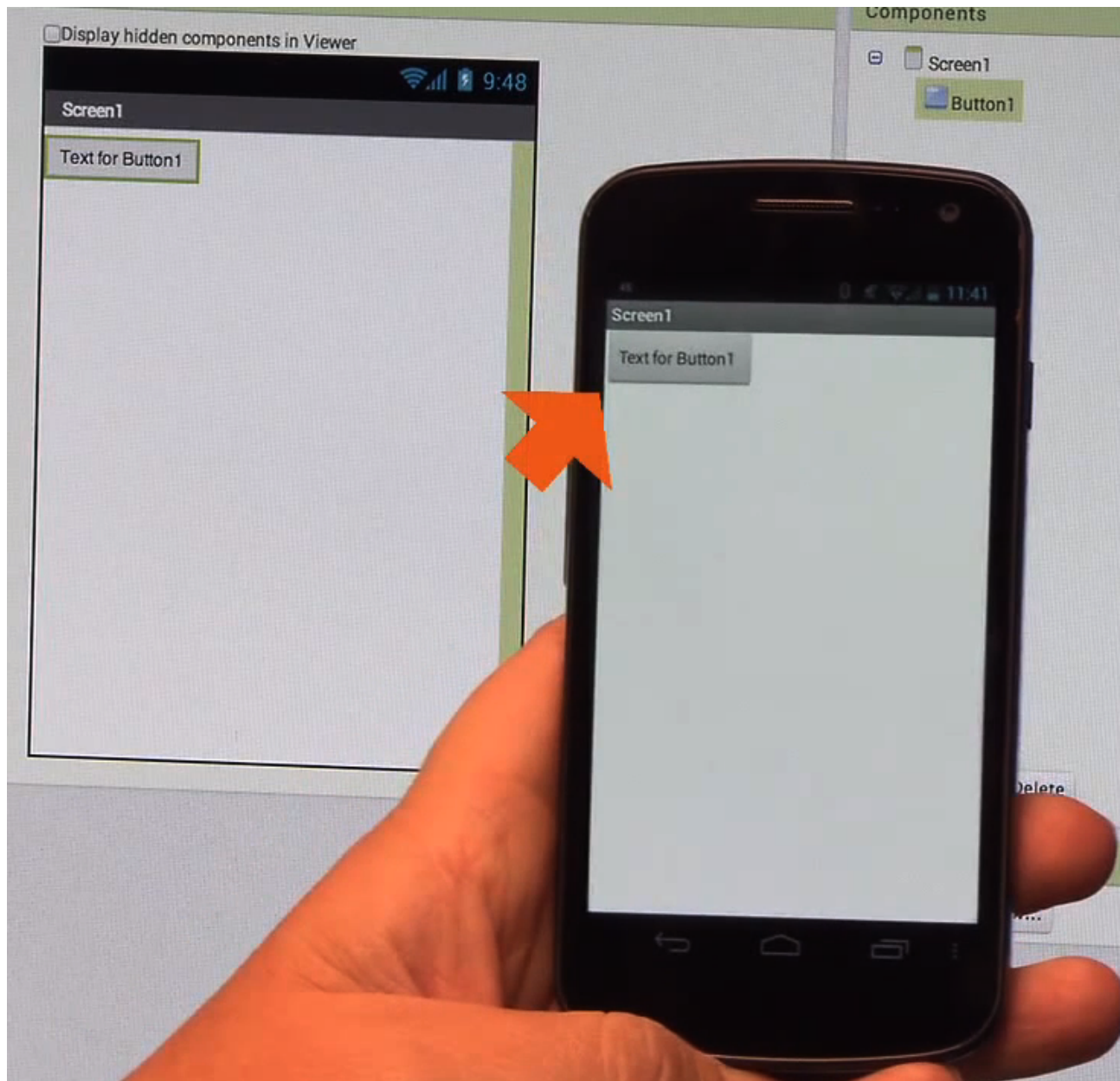
2 - Typing the code into the text window and click "Connect with code" (#2).





See your app on the connected device

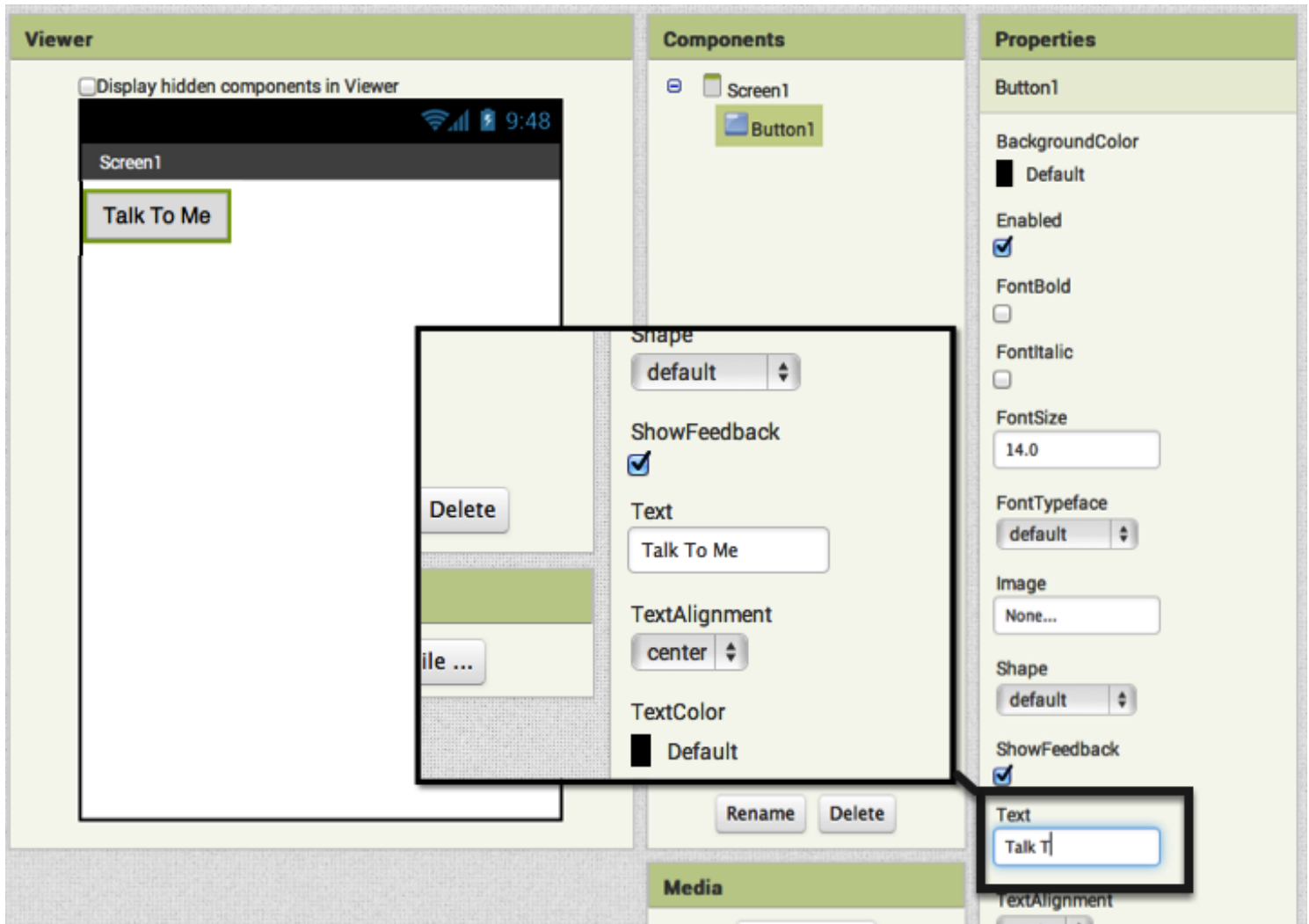
You will know that your connection is successful when you see your app on the connected device. So far our app only has a button, so that is what you will see. As you add more to the project, you will see your app change on your phone.





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.

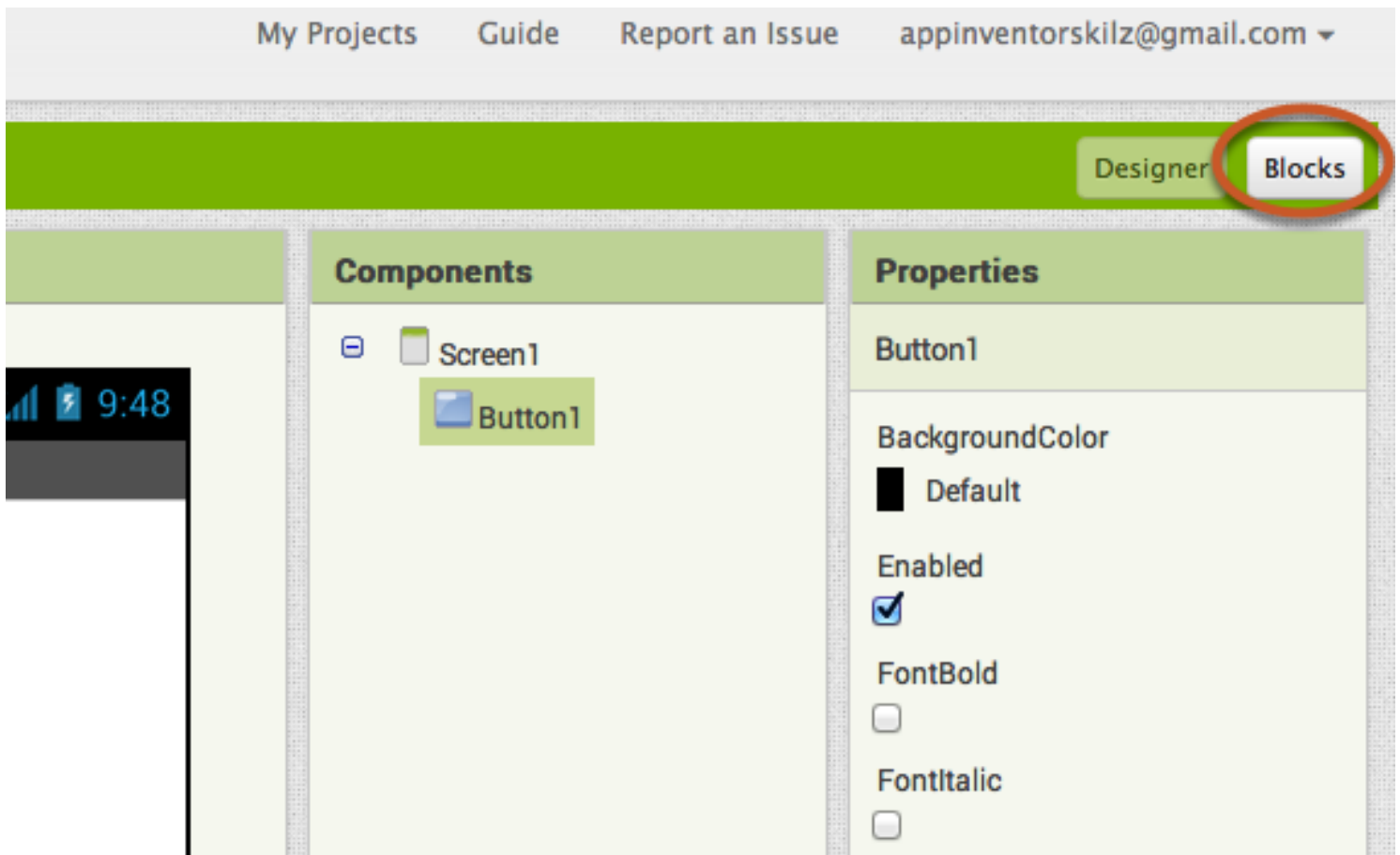
The screenshot displays the MIT App Inventor web interface with four main panels: Palette, Viewer, Components, and Properties.

- Palette:** The 'Media' category is selected and circled in red. Within the Media list, the 'TextToSpeech' component is also circled in red. An orange arrow points from this component to the Viewer panel.
- Viewer:** The central workspace showing a mobile app preview. At the top, there's a status bar with '9:48' and icons for Wi-Fi, signal, and battery. Below it, a button labeled 'Talk To Me' is visible. A large grey box in the center contains the text: 'Drop here. Component will automatically show up in Non-visible components area below'. At the bottom of the Viewer, under the heading 'Non-visible components', the 'TextToSpeech1' component is shown in a green box.
- Components:** The right panel lists the components on the screen: 'Screen1', 'Button1', and 'TextToSpeech1'.
- Properties:** The far right panel shows the properties for the selected 'TextToSpeech1' component, including 'TextToS', '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 Beta interface. The top bar includes the MIT App Inventor logo, the text 'MIT App Inventor 2 Beta', and navigation links: Project, Connect, Build, Help, My Projects, Guide, Report an Issue, and a user email dropdown. Below this is a green header bar for the 'TalkToMe' app, with buttons for 'Screen1', 'Add Screen ...', and 'Remove Screen', and tabs for 'Designer' and 'Blocks'.

The main workspace is divided into two panels:

- Blocks Panel (Left):** Contains a list of block categories. 'Built-in' blocks include Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. 'Screen1' blocks include Button1 and TextToSpeech1. There is also an 'Any component' category. Brackets group the 'Built-in' and 'Screen1' categories.
- Viewer Panel (Right):** Shows a visual representation of the app's workspace. It includes:
 - A grey box stating: **Built-in Blocks** are always available. They handle things like math, text, logic, and control.
 - A larger tan box with four red arrows pointing outwards, stating: **Workspace** where you assemble the blocks into a program.
 - A tan box with a green trash can icon, stating: **Trash** for deleting unneeded blocks.
 - A status bar at the bottom left of the viewer showing two warning icons (yellow and red) with a '0' next to each, and 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** 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 shows the MIT App Inventor 2 Beta interface. The top bar includes the MIT App Inventor logo, the text "MIT App Inventor 2 Beta", and navigation links: "Project", "Connect", "Build", "Help", "My Projects", "Guide", and "Rep". Below this is a green bar with the project name "TalkToMe" and buttons for "Screen1", "Add Screen ...", and "Remove Screen".

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

Blocks Panel: The "Built-in" category is expanded, showing various block types: Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. The "Screen1" category is also expanded, showing "Button1" and "TextToSpeech1". The "Button1" block is circled in red and labeled with a circled "1".

Viewer Panel: The "when Button1.Click do" block is circled in red and labeled with a circled "2". An orange arrow points from this block to a separate "when Button1.Click do" block on the right, which is labeled with a circled "3".

The "Viewer" panel also displays several other blocks for "Button1": "when Button1.GotFocus do", "when Button1.LongClick do", "when Button1.LostFocus do", "Button1.BackgroundColor", "set Button1.BackgroundColor to", "Button1.Enabled", and "set Button1.Enabled to".



Program the TextToSpeech action

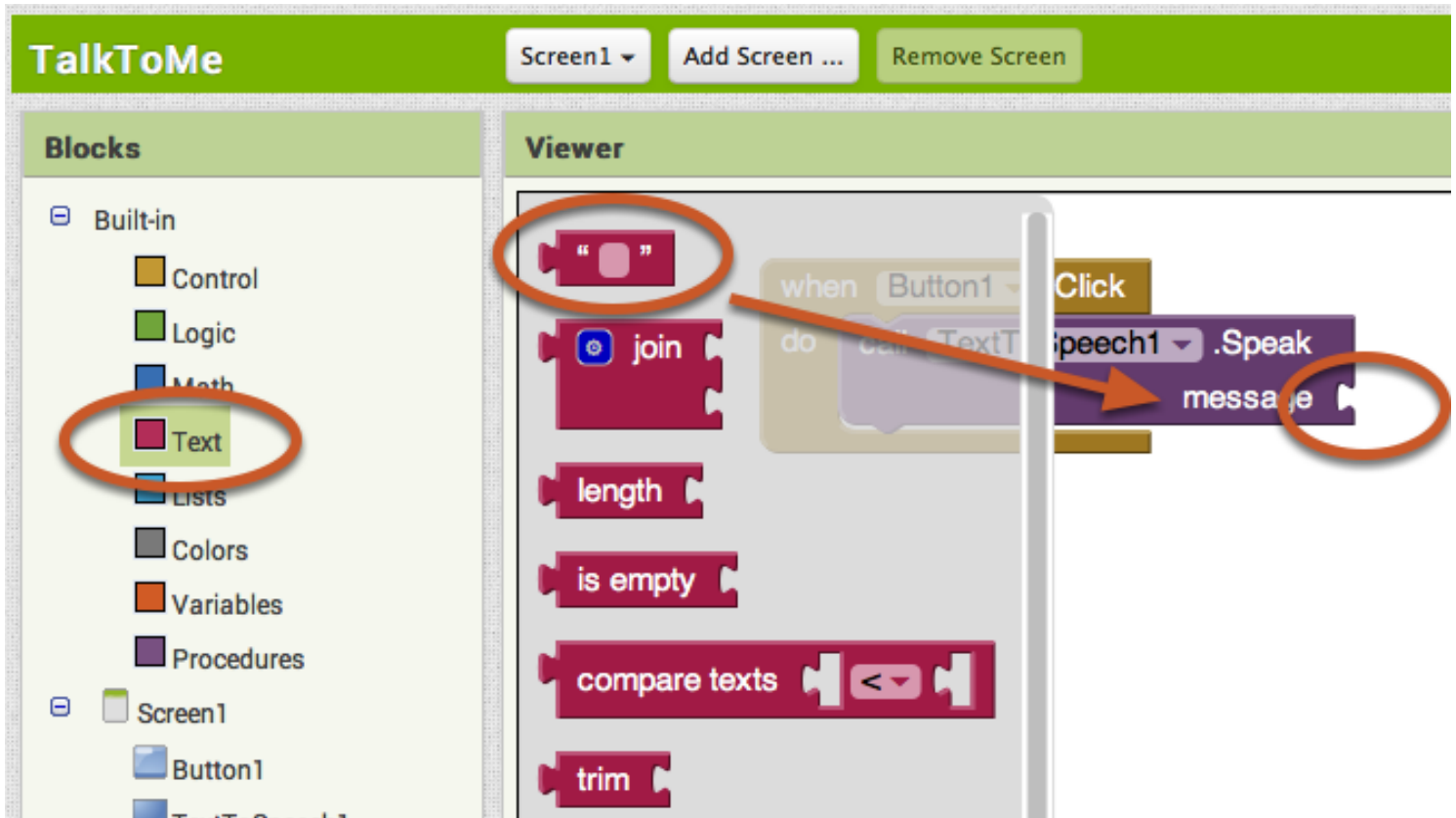
Click on the TextToSpeech drawer. Click and hold the **call TextToSpeech1.Speak** block. Drag it over to the workspace and drop it there. This is the block that will make the phone speak. Because it is inside the Button.Click, it will run when the button on your app is clicked.

The screenshot shows the MIT App Inventor 2 Beta interface for a project named 'TalkToMe'. The 'Blocks' panel on the left lists various built-in blocks, with 'TextToSpeech1' circled in red and labeled with a '1'. The 'Viewer' panel on the right displays the code for the app. A block 'call TextToSpeech1.Speak message' is circled in red and labeled with a '2'. An arrow points from this block to a 'when Button1.Click' block, which is also circled in red and labeled with a '3'. The 'Viewer' panel also shows other blocks for 'TextToSpeech1' including 'Country' and 'Language'.



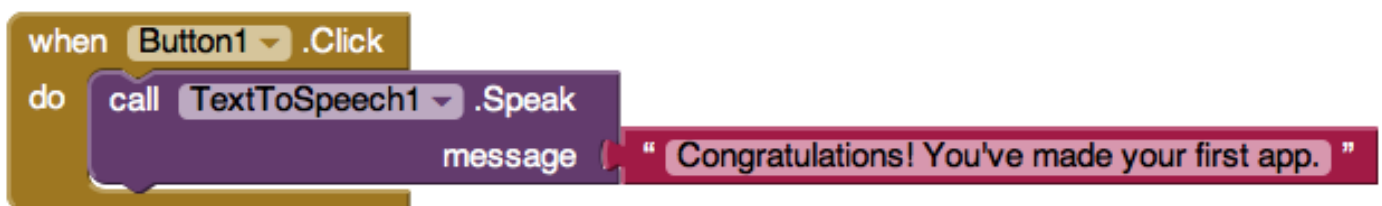
Fill in the message socket on TextToSpeech.Speak Block

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".



Specify what the app should say when the button is clicked

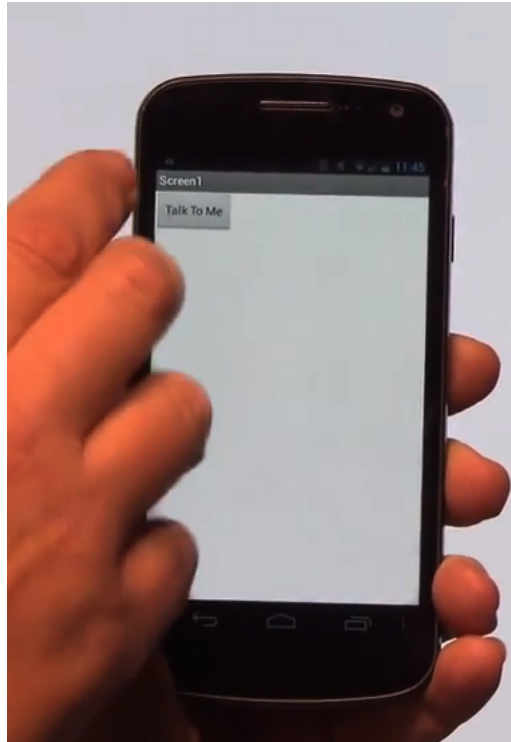
Click on the text block and type in "Congratulations! You've made your first app." (Feel free to use any phrase you like, this is just a suggestion.)





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.)



Great job!

Now move on to TalkToMe Part 2 to make the app respond to shaking and to let users put in whatever phrase they want.