



## 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](http://ai2.appinventor.mit.edu), or click the orange "Create" button from the App Inventor website.

The screenshot shows the MIT App Inventor website interface. At the top left is the MIT App Inventor logo. Navigation links for Home, Blog, and Support are visible. The 'Create' button is highlighted with an orange circle and an arrow. Below the navigation is a 'Follow Us' section with social media icons and a Google Custom Search bar. The main banner features a smartphone displaying a 'Talk To Me' app, with code blocks visible in the background. The banner text reads: 'Your ideas. Your designs. Your apps. Invent Now'. Below the banner are three main sections: 'Get Started' (with a flag icon and a 'Get Started' button), 'Create' (with a smartphone icon and a 'Create' button), and 'Tutorials' (with a lightbulb icon and a 'Tutorials' button).



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](https://accounts.google.com/SignUp)



One account. All of Google.

Sign in with your Google Account

appinventorskilz@gmail.com

•••••••

Sign in

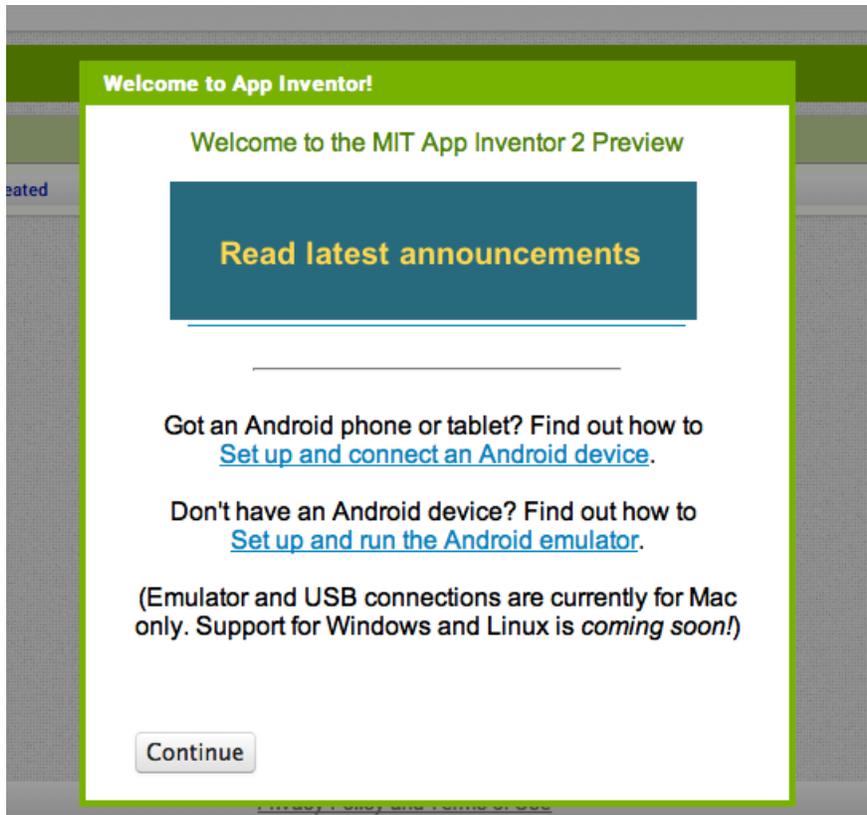
Stay signed in [Need help?](#)

[Create an account](#)

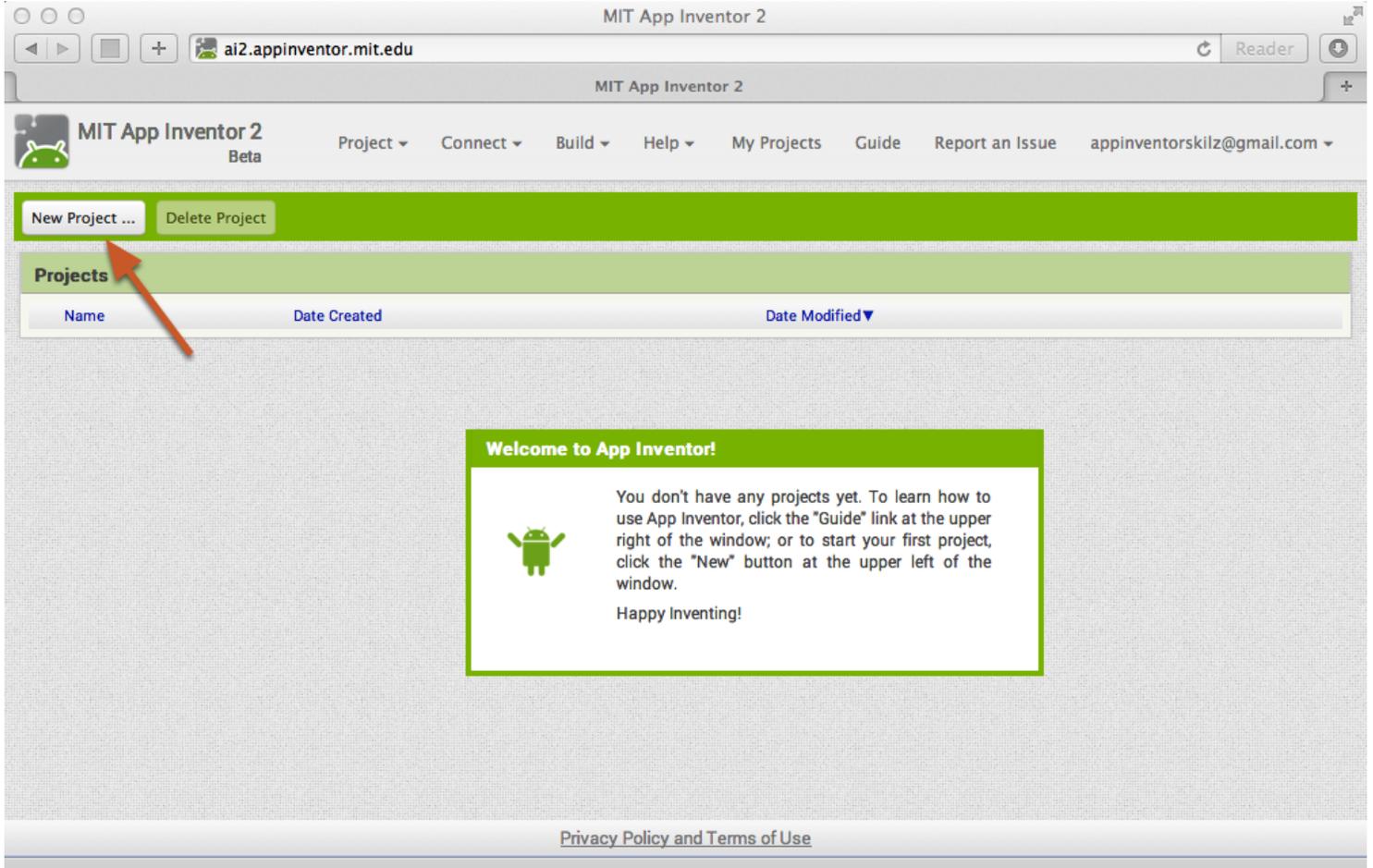


MIT App Inventor  
appinventor.mit.edu

Click "Continue" to dismiss the splash screen.

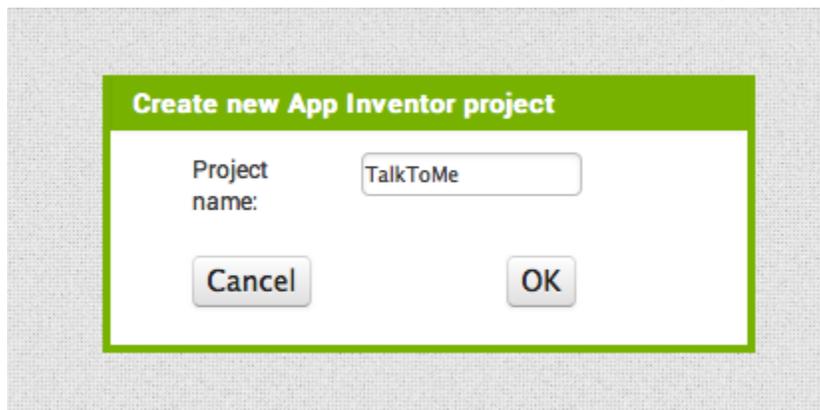


## Start a new project.



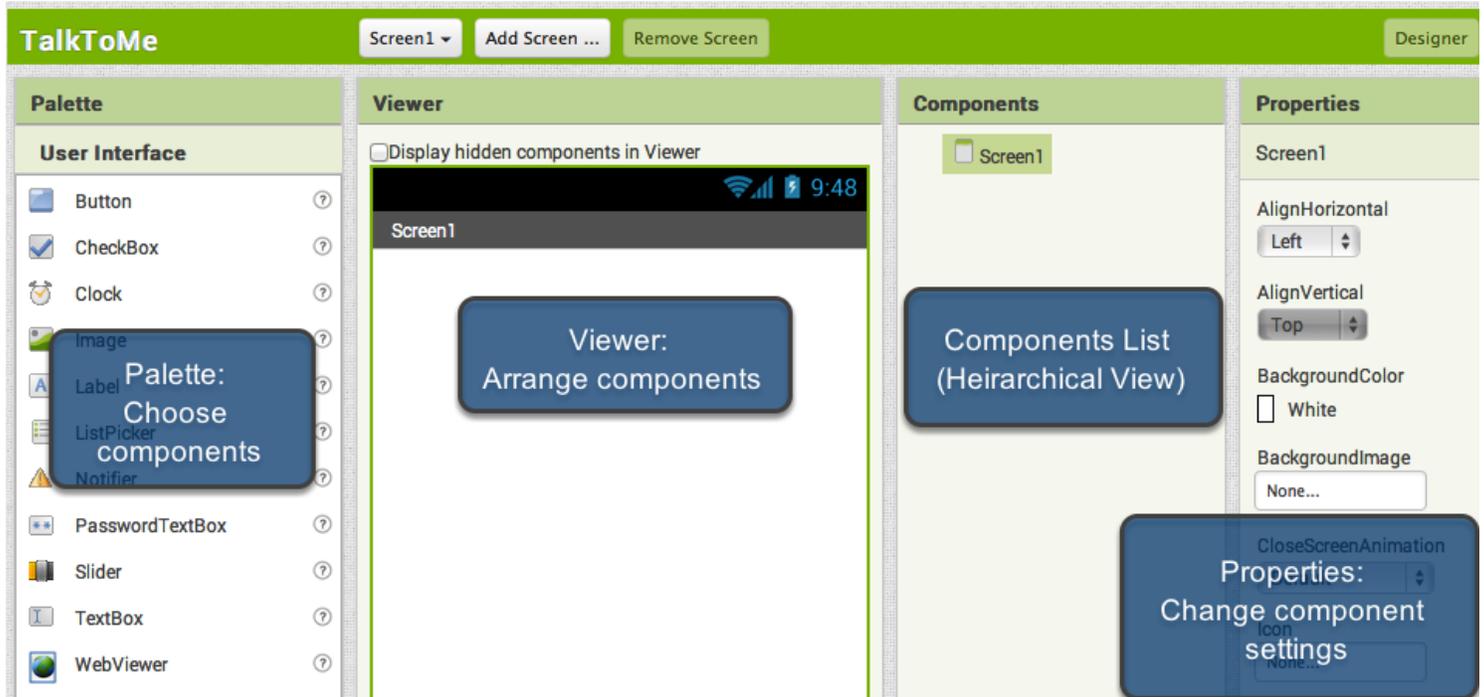
## Name the project "TalkToMe" (no spaces!)

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



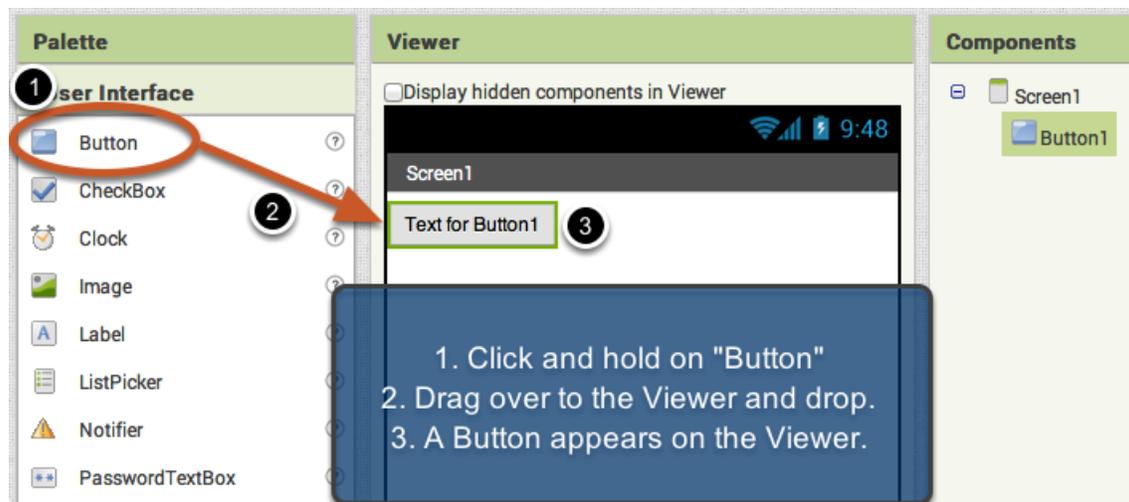
## You are now in the Designer, where you lay out the "user interface" of your app.

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.



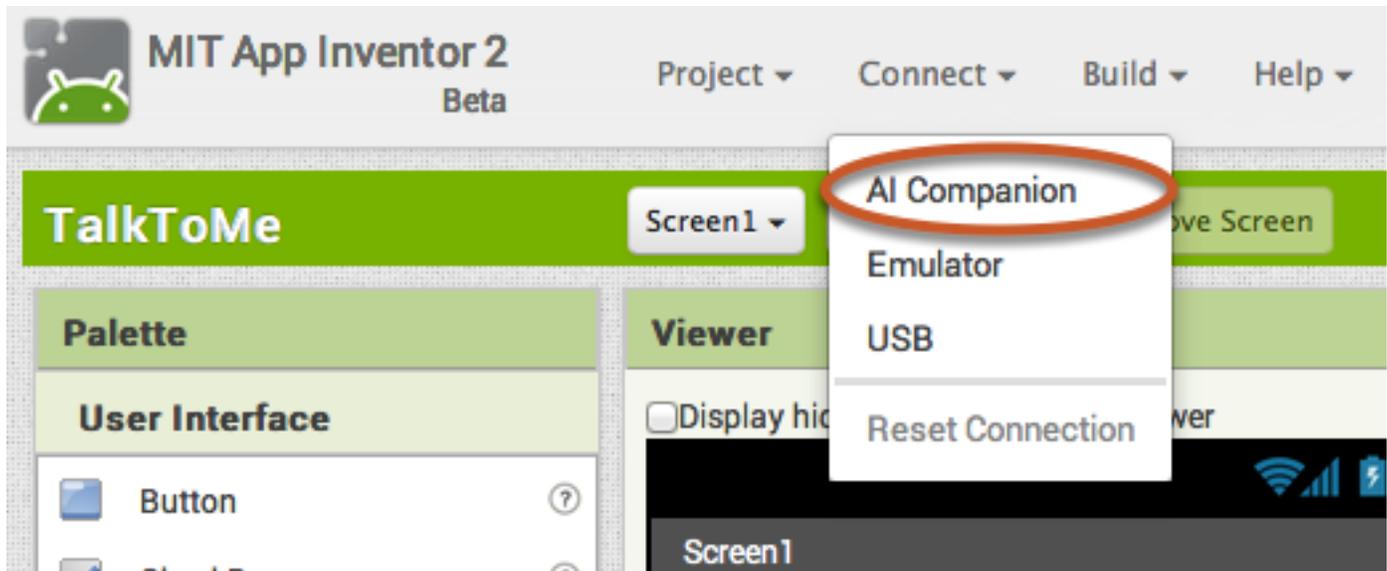
## Add a Button

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.

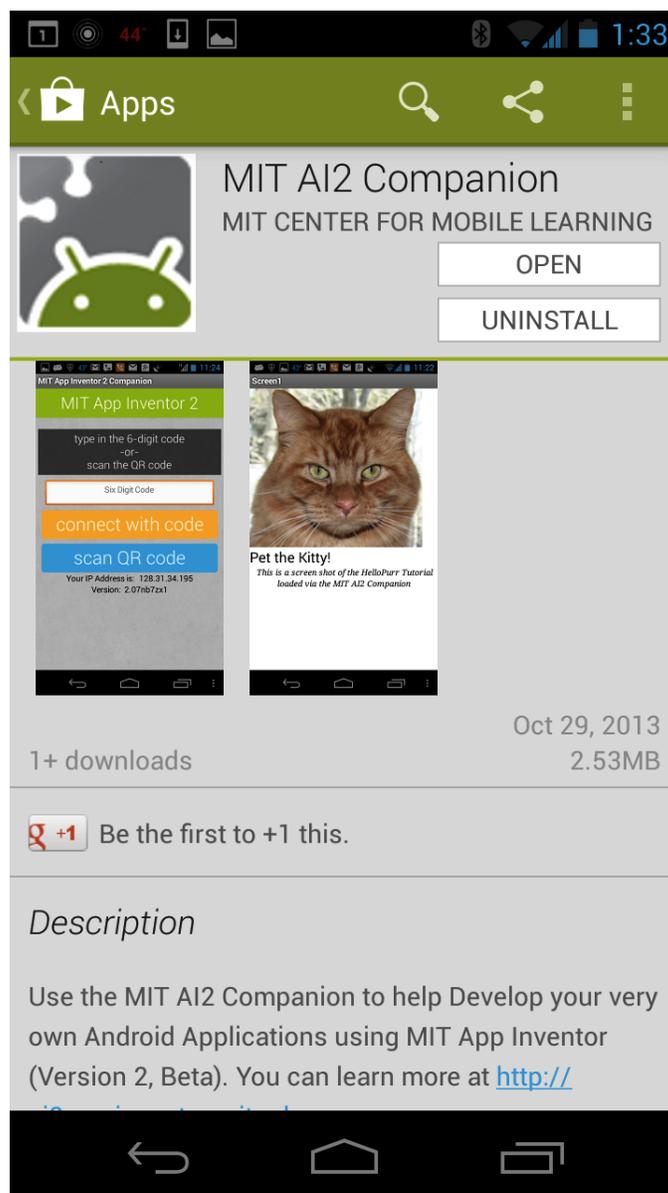




MIT App Inventor  
appinventor.mit.edu

**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.

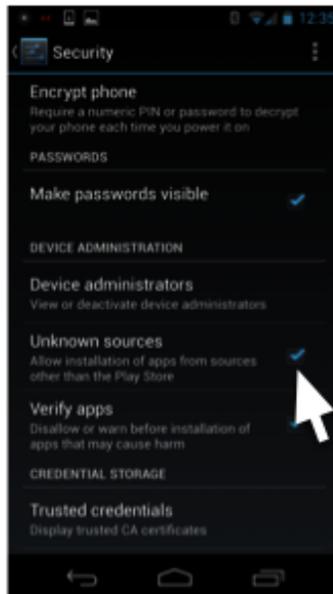
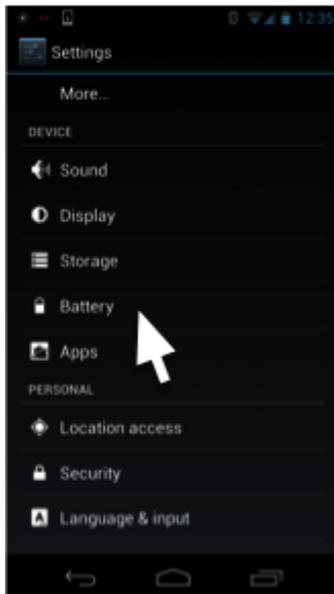
First, you will need to go into your phone's settings (#1), choose "Security", then scroll down to allow "Unknown Sources", which allows apps that are not from the Play Store to be installed on the phone.

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.



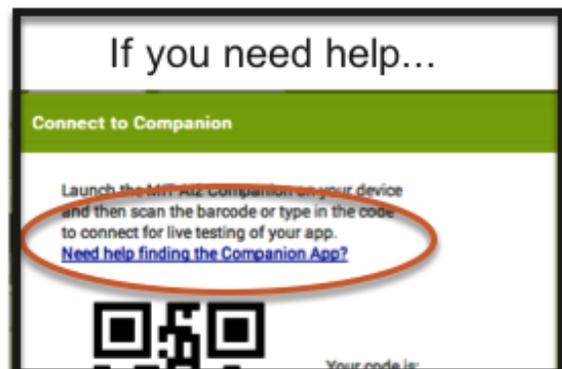
**SKIP THIS STEP if you already got the AI2 Companion from the Play Store**

1

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

2

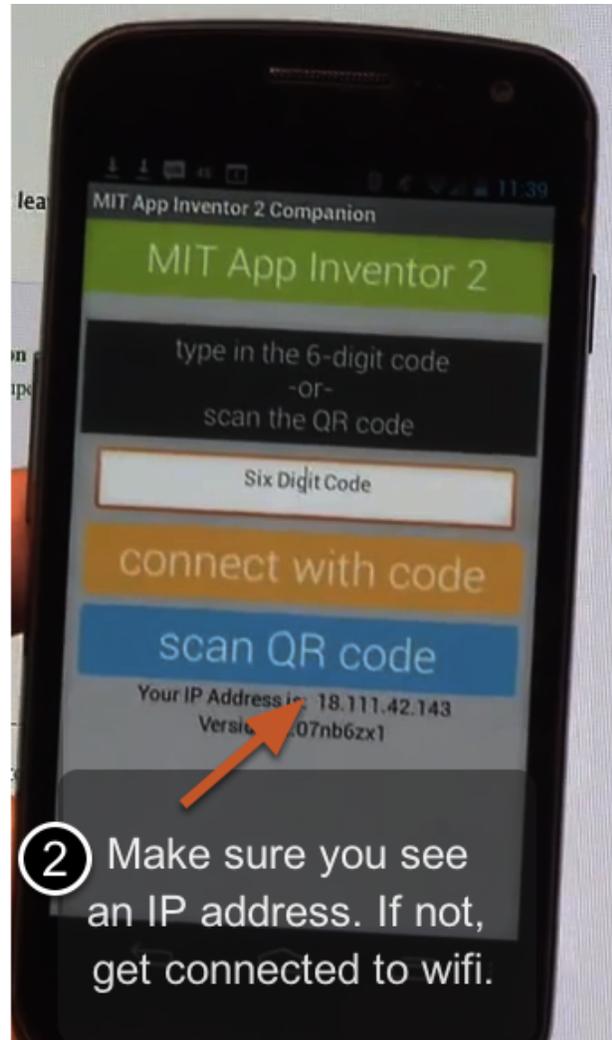
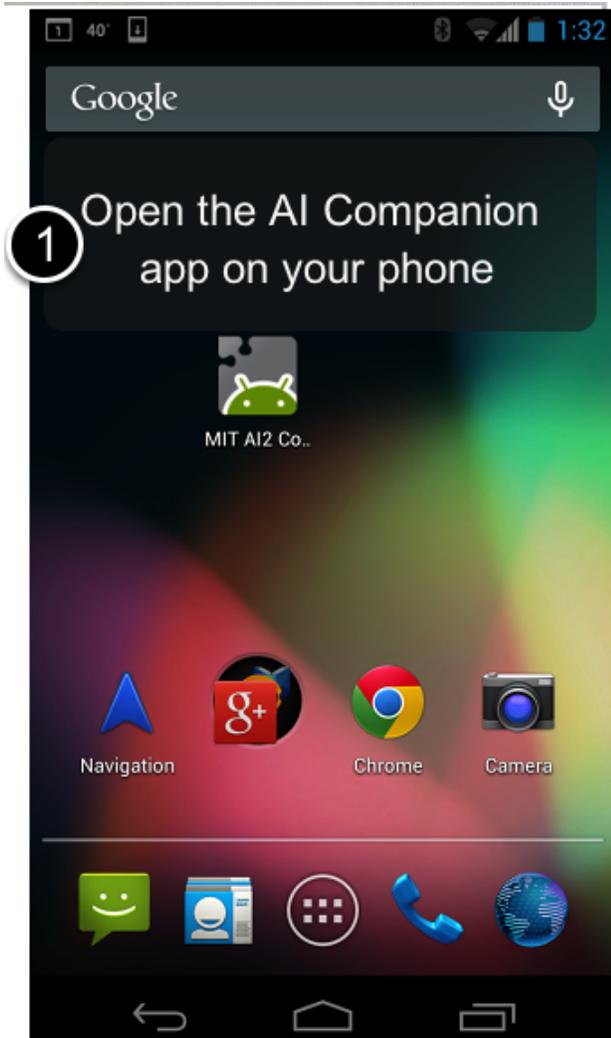
Scan to download MIT AI2 Companion directly to phone





## Start the AI Companion 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).

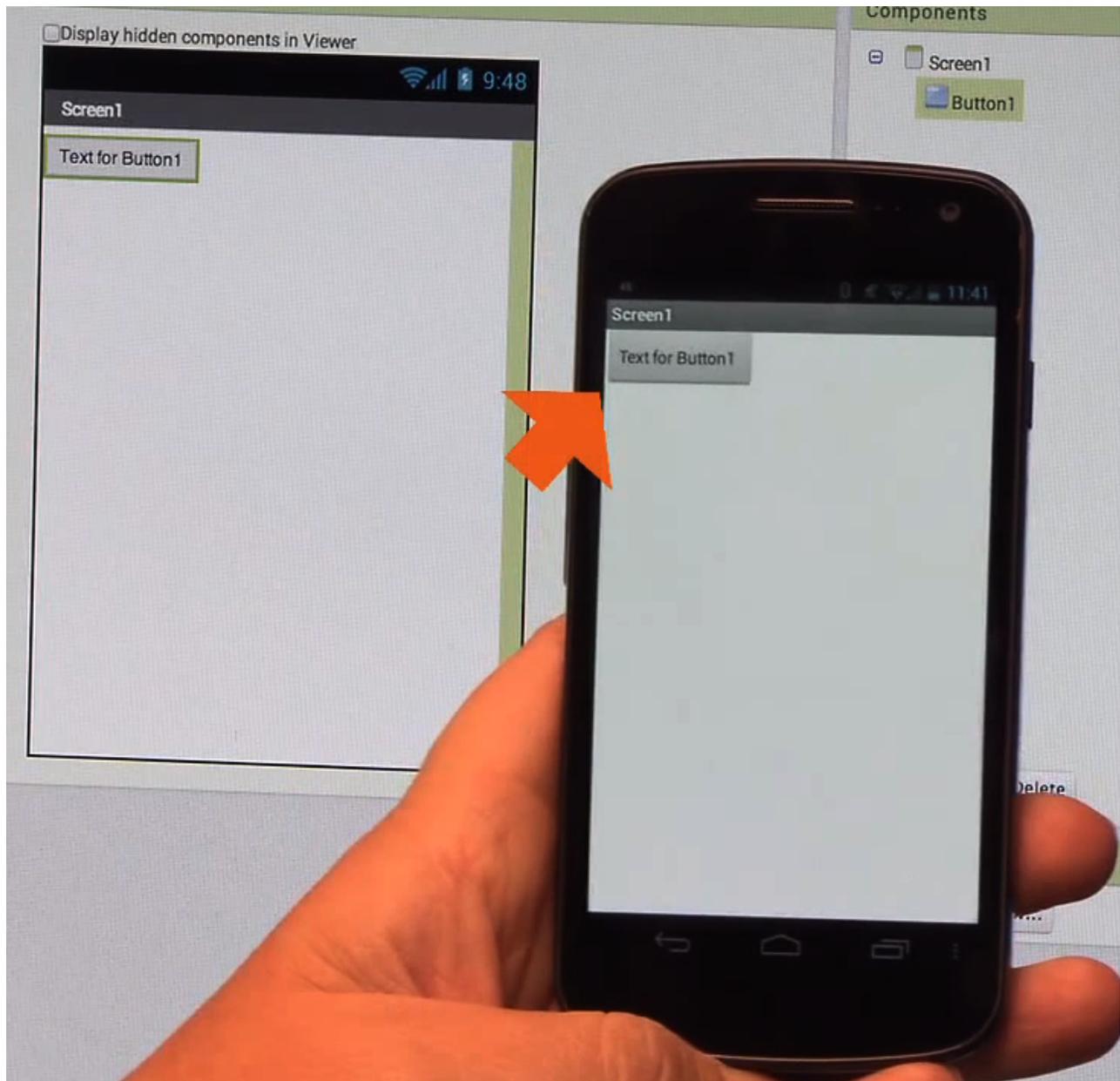
or

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.

The screenshot displays the MIT App Inventor interface with three main panels: Viewer, Components, and Properties. In the Viewer panel, a mobile app preview shows a button with the text "Talk To Me". The Components panel shows a hierarchy with "Screen1" containing "Button1". The Properties panel is open for "Button1", showing various settings. A context menu is overlaid on the "Button1" component, with the "Text" field highlighted and containing "Talk To Me". The Properties panel also has its "Text" field highlighted and containing "Talk T".



## 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 interface with four main panels: Palette, Viewer, Components, and Properties.

- Palette:** The 'Media' category is circled in red. The 'TextToSpeech' component is also circled in red, with an orange arrow pointing from it to the Viewer.
- Viewer:** Shows a mobile app preview with a button labeled 'Talk To Me'. A 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 'Non-visible components' area contains a 'TextToSpeech1' component.
- Components:** The 'Screen1' container holds 'Button1' and 'TextToSpeech1'. The 'TextToSpeech1' component is highlighted with a green box.
- Properties:** The 'TextToSpeech1' component's properties are visible, including 'TextToS', 'Country', and 'Language'.



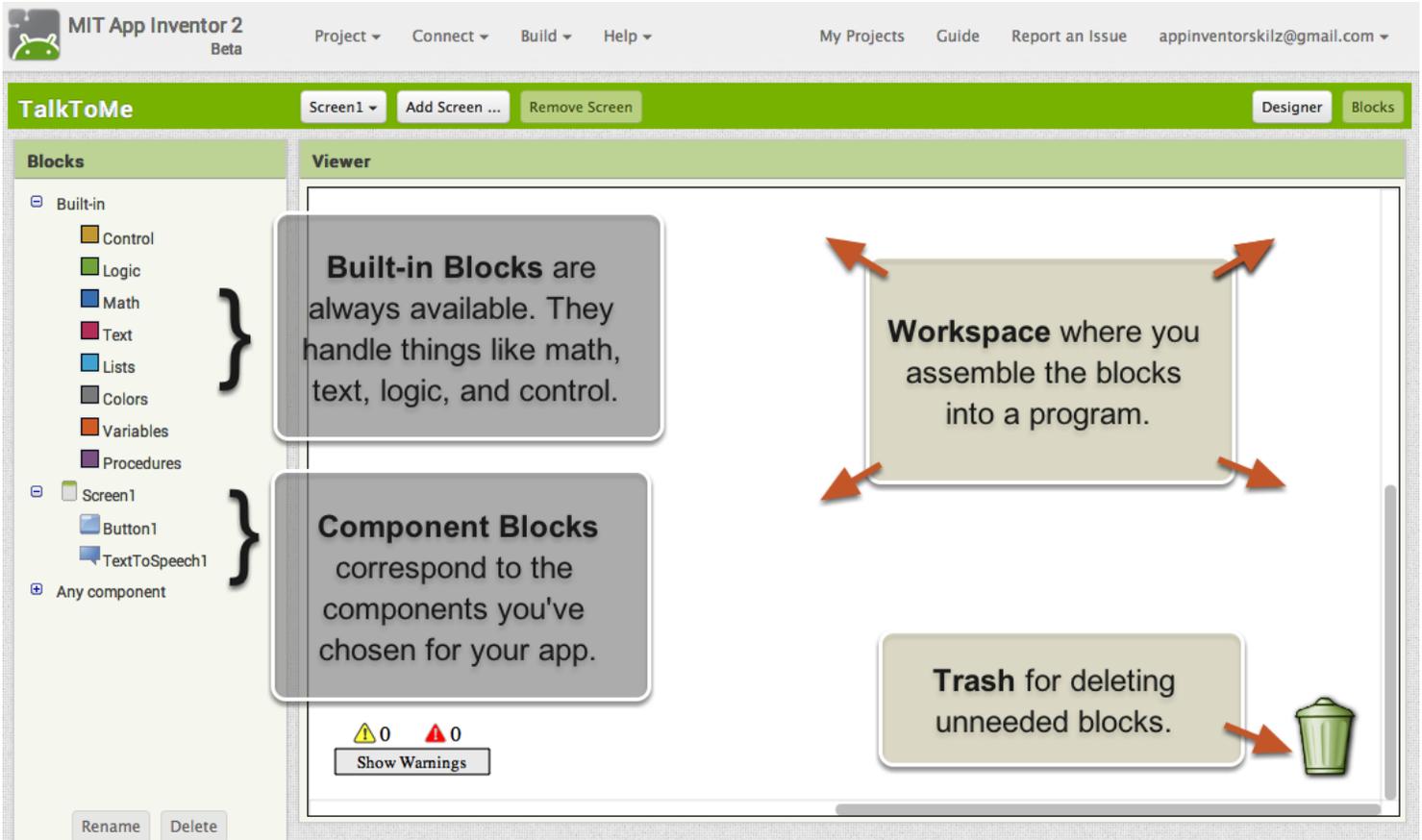
## Switch over to the Blocks Editor

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 screenshot shows the MIT App Inventor web interface. At the top, there is a navigation bar with links for "My Projects", "Guide", "Report an Issue", and a user email "appinventorskilz@gmail.com". Below this is a green header bar with two tabs: "Designer" and "Blocks". The "Blocks" tab is highlighted with a red circle. The main workspace is divided into three panels: a preview area on the left showing a mobile app interface with a status bar at 9:48; a "Components" panel in the middle showing a hierarchy with "Screen1" containing "Button1"; and a "Properties" panel on the right showing settings for "Button1", including "BackgroundColor" (Default), "Enabled" (checked), "FontBold" (unchecked), and "FontItalic" (unchecked).

## 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.*



MIT App Inventor 2 Beta

Project ▾ Connect ▾ Build ▾ Help ▾

My Projects Guide Report an Issue appinventorskilz@gmail.com ▾

TalkToMe Screen1 ▾ Add Screen ... Remove Screen Designer Blocks

**Blocks**

- Built-in
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Colors
  - Variables
  - Procedures
- Screen1
  - Button1
  - TextToSpeech1
- Any component

**Viewer**

**Built-in Blocks** are always available. They handle things like math, text, logic, and control.

**Component Blocks** correspond to the components you've chosen for your app.

**Workspace** where you assemble the blocks into a program.

**Trash** for deleting unneeded blocks.

0 0  
Show Warnings

Rename Delete



## 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 navigation bar includes "MIT App Inventor 2 Beta", "Project", "Connect", "Build", and "Help" menus, along with "My Projects", "Guide", and "Rep" links. Below the navigation bar is a green header for the project "TalkToMe", with "Screen1" selected and buttons for "Add Screen ..." and "Remove Screen".

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

**Blocks Panel:** A list of built-in components is shown, categorized by type: Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under the "Screen 1" category, a "Button1" component is circled in red and labeled with a circled "1".

**Viewer Panel:** A list of event handler blocks for "Button1" is shown. The top block, "when Button1 .Click do", is circled in red and labeled with a circled "2". An orange arrow points from this block to a "when Button1 .Click do" block that is being dragged from the blocks panel into the workspace, labeled with a circled "3". Below the event handlers, several property blocks for "Button1" are visible, including "BackgroundColor", "Enabled", and "set Button1 . BackgroundColor 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 interface is divided into three main sections: Blocks, Viewer, and a top navigation bar.

- Top Bar:** MIT App Inventor 2 Beta, Project, Connect, Build, Help, My Projects, Guide, Report an Issue.
- Project Bar:** Screen1, Add Screen ..., Remove Screen.
- Blocks Panel:** Contains a "Built-in" category with sub-categories: Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under "Screen1", there are "Button1" and "TextToSpeech1" components. The "TextToSpeech1" component is circled in red and labeled with a "1".
- Viewer Panel:** Shows the workspace with several blocks:
  - A "when TextToSpeech1 .AfterSpeaking" block with a "result" block and a "do" block.
  - A "when TextToSpeech1 .BeforeSpeaking" block with a "do" block.
  - A "call TextToSpeech1 .Speak" block with a "message" input, circled in red and labeled with a "2". An arrow points from this block to the "do" block of the "when Button1 .Click" block.
  - A "TextToSpeech1 . Country" block.
  - A "set TextToSpeech1 . Country to" block.
  - A "TextToSpeech1 . Language" block.
- Event Listeners:** A "when Button1 .Click" block is shown on the right, labeled with a "3". Its "do" block contains the "call TextToSpeech1 .Speak" 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 MIT App Inventor interface for an app named "TalkToMe". The interface is divided into a "Blocks" panel on the left and a "Viewer" panel on the right. In the "Blocks" panel, the "Text" block category is highlighted with an orange circle. In the "Viewer" panel, a script is shown starting with "when Button1 Click" followed by "do call TextToSpeech1 .Speak". The "message" socket of the ".Speak" block is highlighted with an orange circle. An orange arrow points from a "Text" block in the "Blocks" panel to the "message" socket. Other blocks visible in the "Viewer" panel include "join", "length", "is empty", "compare texts", and "trim".

## Specify what the app should say when the button is clicked

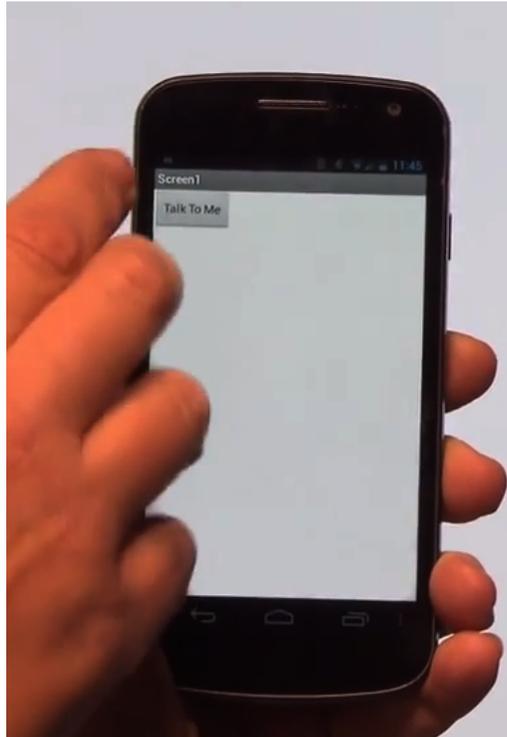
Click on the text block and type in "Congratulations! You've made your first app." (Feel free to use any phrase you like, this is just a suggestion.)

The screenshot shows a close-up of the MIT App Inventor script editor. It displays a "when Button1 Click" block followed by a "do call TextToSpeech1 .Speak" block. The "message" socket of the ".Speak" block is 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.)



## 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.