



## 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

Name

First  Last

Choose your username

@gmail.com

I prefer to use my current email address

Create a password

Confirm your password

Birthday

Month  Day  Year

Gender

I am...

Mobile phone

Your current email address

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

About ▾ News & Stories ▾ Resources ▾ Create apps!

f g+ t Y

App Inventor is now in French and Russian!

Google Custom Search



## 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

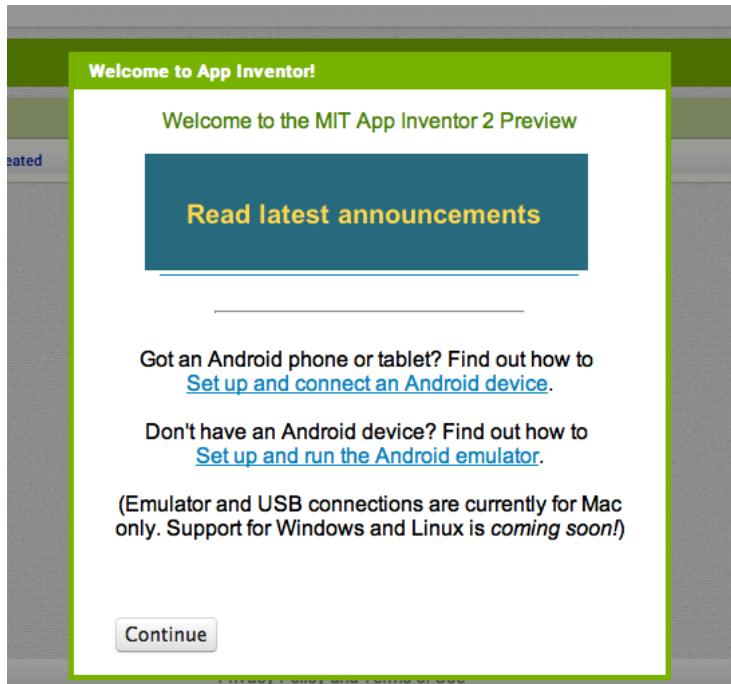
Google is not affiliated with the contents of MIT AppInventor Version 2 or its owners. If you sign in, Google will share your email address with MIT AppInventor Version 2 but not your password or any other personal information.

[Allow](#)   [No thanks](#)

[Sign in to another account](#)



**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.

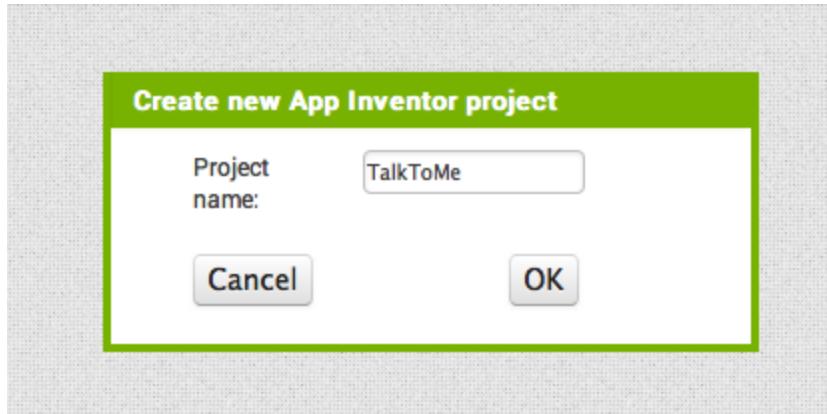
**Start a new project by clicking the "Start new project" button.**

(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.

**Palette** (User Interface section):

- Button
- CheckBox
- Clock
- Image
- Label
- ListPicker
- Notifier
- PasswordTextBox
- Slider
- TextBox
- WebView

**Viewer**: Shows a mobile phone screen with the title "Screen1". A button labeled "Viewer: Arrange components" is visible.

**Components**: Shows a list of components under "Screen1".

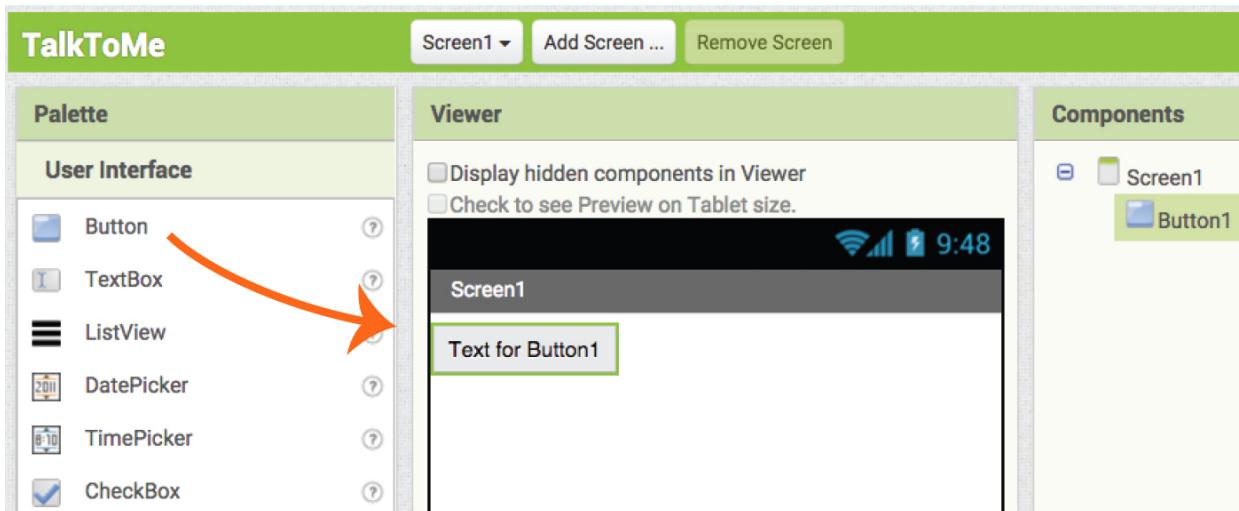
**Properties** (for Screen1):

- Screen1
- AlignHorizontal: Left
- AlignVertical: Top
- BackgroundColor: White
- BackgroundImage: None
- CloseScreenAnimation: Properties: Change component settings

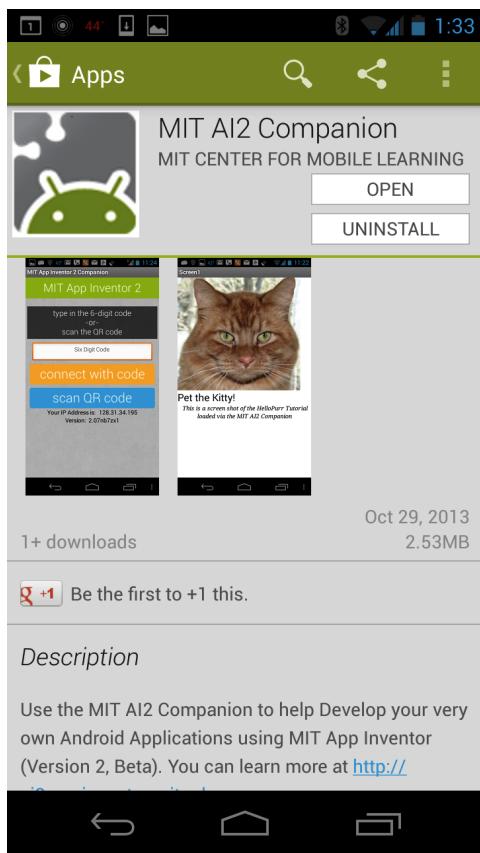


## Add a Button

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. X  
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.)

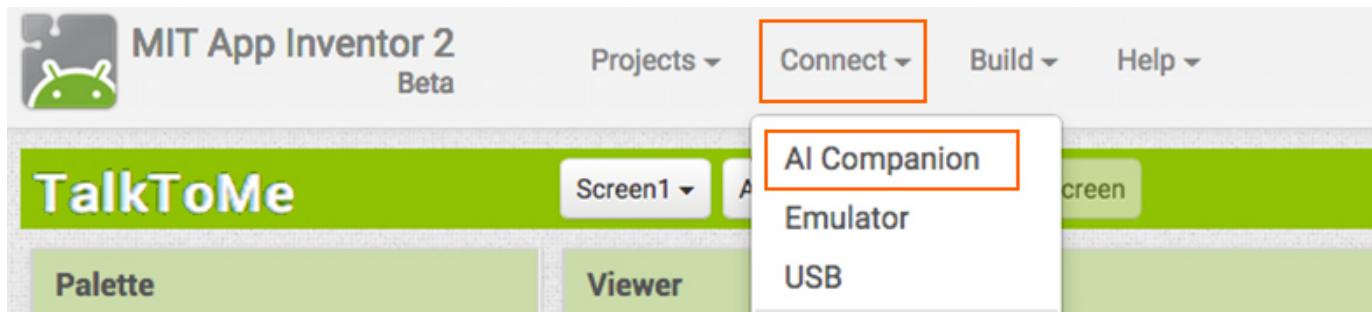
**If you don't have a mobile device, you can still use App Inventor by connecting with the emulator. Visit: <http://appinventor.mit.edu/explore/ai2/setup.html> and follow the instructions under Option 2.**



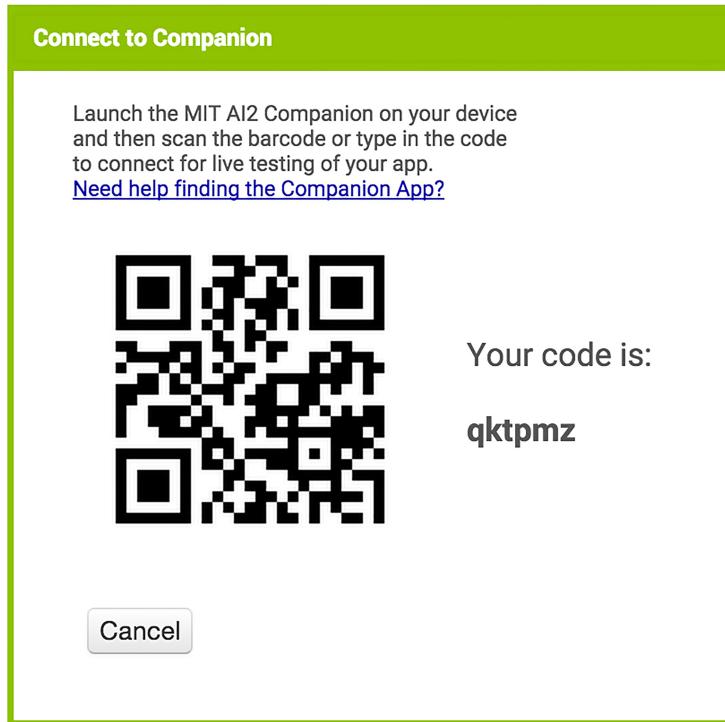
Once you've installed the AI2 Companion app, you can connect your App Inventor project to your phone or tablet for live testing

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.





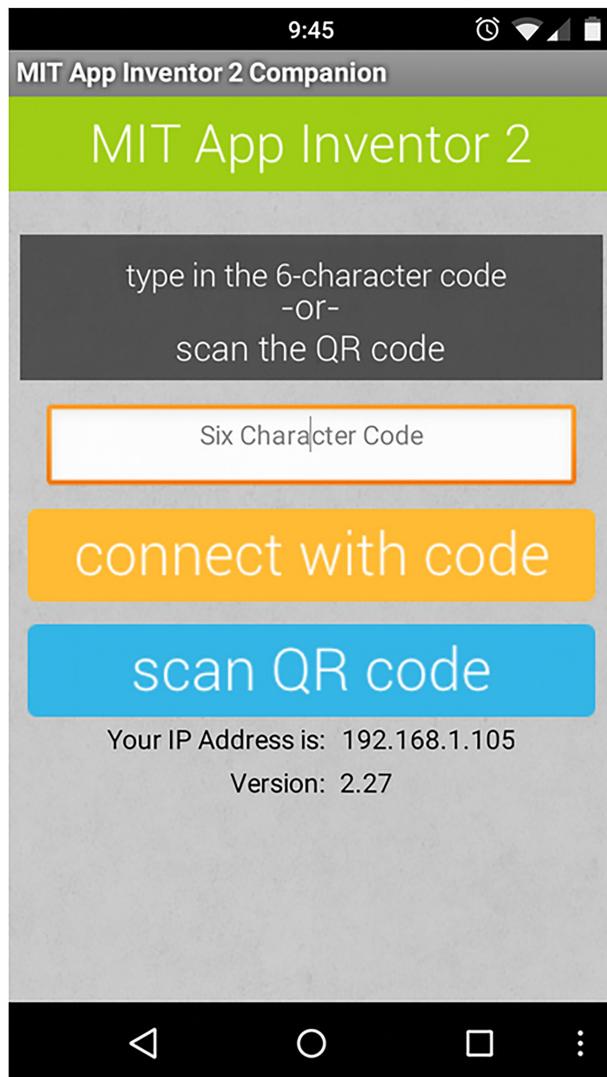
MIT App Inventor  
appinventor.mit.edu

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>

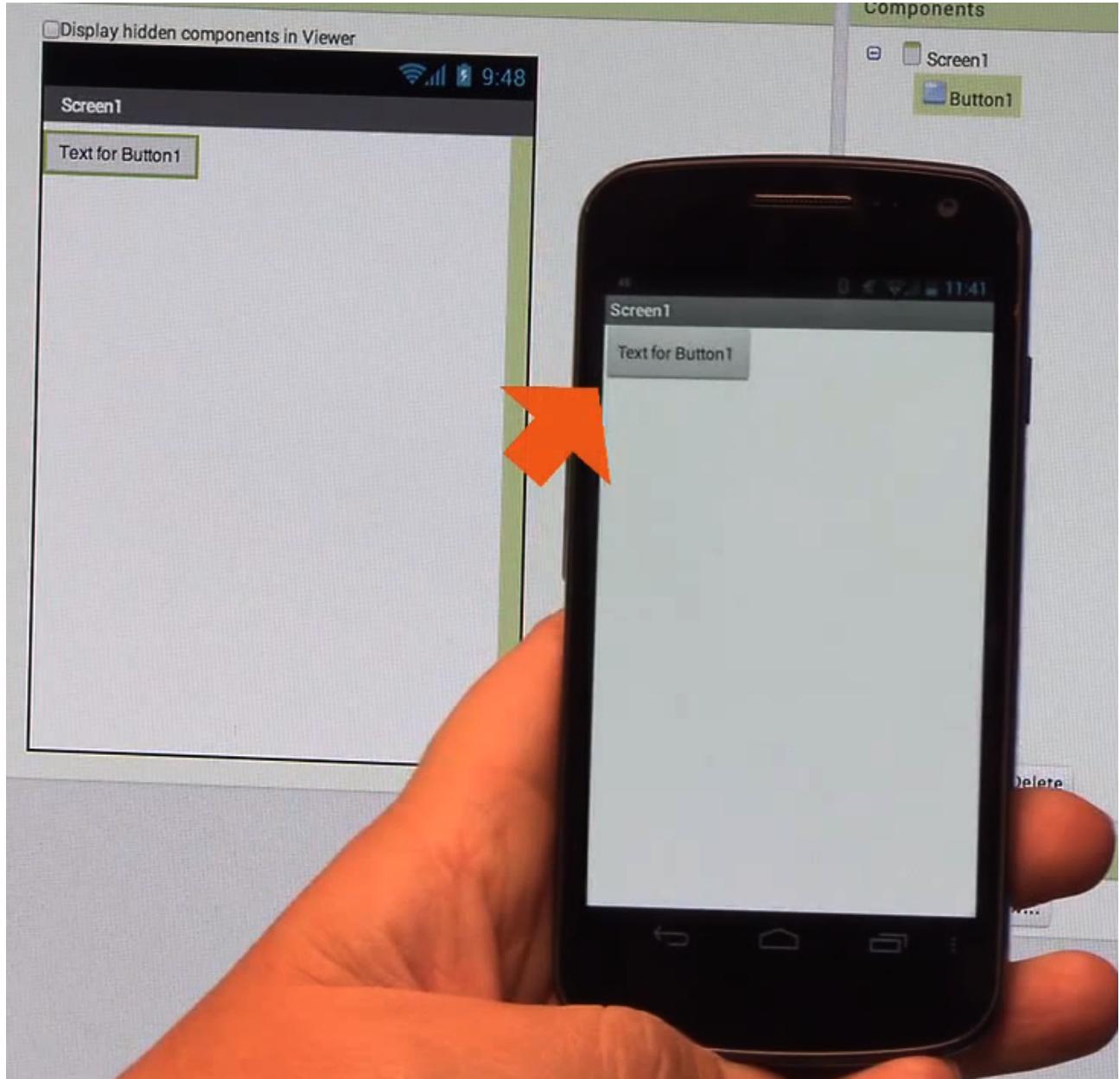


## See your app on your connected device

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.

The screenshot shows the MIT App Inventor interface with the following components:

- Viewer:** Displays the app's screen titled "Screen1" with a button labeled "Talk To Me".
- Components:** Shows a tree structure with "Screen1" expanded, revealing "Button1".
- Properties Panel:** Focuses on the "Button1" component. The "Text" property is highlighted, showing its current value as "Talk To Me".

A cursor is visible in the "Text" input field of the properties panel, indicating it is being edited. The "Text" field is currently empty, with only the first few letters of "Talk To Me" visible as a placeholder.



## 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 shows the MIT App Inventor workspace divided into four main sections:

- Palette:** On the left, a list of component categories. The "Media" category is highlighted with a red oval, and the "TextToSpeech" component within it is also highlighted with a red oval. An orange arrow points from the "TextToSpeech" item in the palette towards the viewer area.
- Viewer:** The central area where app screens are designed. A screen titled "Screen1" contains a button labeled "Talk To Me". Below the screen, a callout box with a grey background and a black border contains the text: "Drop here. Component will automatically show up in Non-visible components area below".
- Components:** A list of components currently in the project. "Screen1", "Button1", and "TextToSpeech1" are listed. "TextToSpeech1" is highlighted with a green oval.
- Properties:** On the far right, there are two dropdown menus: "TextToS" and "Country", both currently empty. Below them is a "Language" dropdown menu which is also empty.



## 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 screenshot shows the MIT App Inventor interface in Designer mode. At the top, there are navigation links: My Projects, Guide, Report an Issue, and an email link (appinventorskilz@gmail.com). Below the header, there are two tabs: 'Designer' and 'Blocks', with 'Blocks' circled in red. The main workspace is divided into three panels: 'Components' (containing a tree view of 'Screen1' and 'Button1'), 'Properties' (listing properties for 'Button1' with checkboxes for BackgroundColor, Enabled, FontBold, and FontItalic), and a preview window on the left showing a smartphone screen with the time 9:48. The 'Components' panel has a red border around its tree view.



## 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 Blocks Editor interface. On the left, the 'Blocks' panel lists categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1 (Screen1, Button1, TextToSpeech1), and Any component. Brackets on the right side of the list group 'Built-in' and 'Screen1'. The 'Viewer' panel contains two sections: 'Built-in Blocks' (describing them as always available for math, text, logic, and control) and 'Component Blocks' (describing them as corresponding to added components). A 'Workspace' area is shown on the right where blocks are assembled. A 'Trash' bin icon is also present. Red arrows point from the explanatory text to their respective targets: one arrow points to the 'Workspace' area, another to the 'Component Blocks' section, and a third to the trash bin icon. At the bottom of the viewer panel, there are warning icons (yellow triangle 0, red triangle 0) and a 'Show Warnings' button.



## 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 interface with the project titled "TalkToMe".

**Blocks Palette:**

- Built-in:
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Colors
  - Variables
  - Procedures
- Screen 1:
  - Button1
  - TextToSpeech1
- Any component

**Viewer:**

- Block 2: **when Button1.Click do** (highlighted with a red oval)
- Block 3: **when Button1.Click do** (being moved from the viewer)
- Other blocks visible in the viewer include:
  - when Button1.GotFocus do
  - when Button1.LongClick do
  - when Button1.LostFocus do
  - Button1.BackgroundColor
  - set Button1.BackgroundColor to [ ]
  - Button1.Enabled
  - 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 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 interface. The top navigation bar includes the logo, 'MIT App Inventor 2 Beta', 'Project', 'Connect', 'Build', 'Help', 'My Projects', 'Guide', and 'Report an Issue'. The main area is titled 'TalkToMe' with tabs for 'Screen1', 'Add Screen ...', and 'Remove Screen'. On the left, the 'Blocks' palette is open, showing categories like 'Built-in' (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), 'Screen1' (Button1, TextToSpeech1), and 'Any component'. A circled '1' is next to the TextToSpeech1 block. In the center, the 'Viewer' shows blocks for the 'when TextToSpeech1.AfterSpeaking' event. A circled '2' is next to a purple 'call TextToSpeech1.Speak message' block, which is highlighted with a red oval. An orange arrow points from this block to a third block labeled '3' in the 'when Button1.Click' event, which also contains a 'call TextToSpeech1.Speak message' block.



## 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 has tabs for "Screen1", "Add Screen ...", and "Remove Screen". The left sidebar is titled "Blocks" and contains a tree view of categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1 (Button1, TextToSpeech1), and a local variable TextToSpeech1. The "Text" category under Built-in is highlighted with a red circle. The main area is titled "Viewer" and shows a script block: "when Button1.Click do call TextToSpeech1.Speak". Inside the "TextToSpeech1.Speak" block, the "message" socket is highlighted with a red circle. A red arrow points from the circled "Text" block in the blocks palette to the circled "message" socket in the viewer.

## 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."

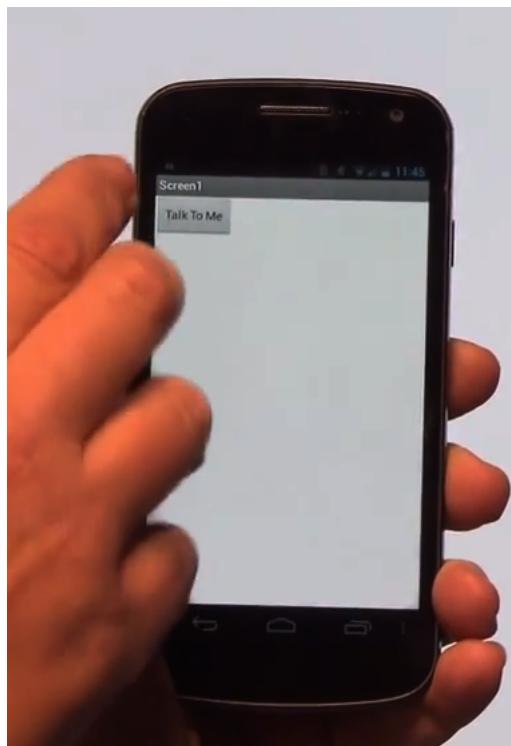
(Feel free to use any phrase you like.)

```
when Button1.Click
do call TextToSpeech1.Speak
    message "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!