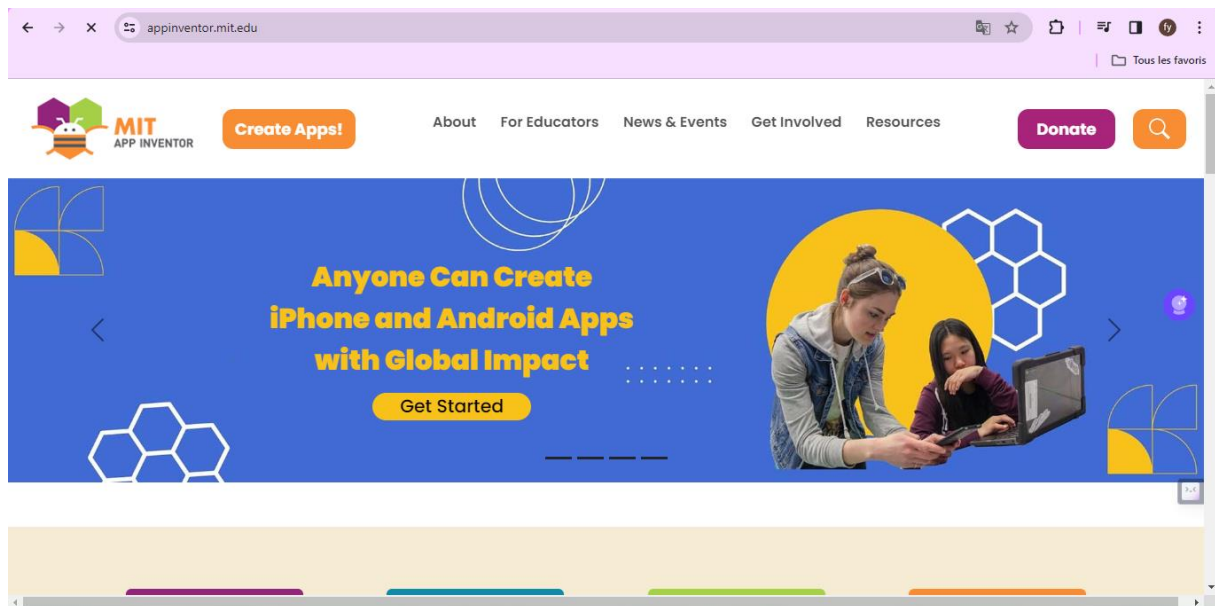


Lab session 1: Flutter Installation and UI Design using Flutter

Task-7 Create your first App using MIT AppInventor

In this documentation, we will walk through the process of creating our first app using MIT App Inventor. MIT App Inventor is a visual development platform that allows us to build mobile applications for Android devices without the need for traditional programming languages. We will explore the various steps involved in creating our app and highlight the key concepts and features of MIT App Inventor.



Setting up the environment

Before getting started with app development, we need to set up MIT App Inventor. This involves creating an account on the MIT App Inventor website and installing the necessary software or utilizing the web-based version.

Get Started with MIT App Inventor

App Inventor is an [online tool](#), meaning you can create apps for phones or tablets right in your web browser. This website offers all the support you'll need to design apps:

- **Setup Instructions for Testing:** Set up your iOS or Android phone or tablet for live testing. (Or, start the emulator if you don't have a mobile device)
- **Designer and Blocks Editor Overview:** Tour the App Inventor environment
- **Beginner Tutorials:** Jump in and get started programming in App Inventor
- **Sharing and Remixing Apps:** Share your app code with friends to load in App Inventor and remix
- **Building and Installing Apps:** Build your app for installing on a phone or tablet (Android only, for now)

Setting Up App Inventor

Set up App Inventor to test your app while you build (also called "Live Testing"). Check out the four options below:

Option 1 - **RECOMMENDED**

Test your apps with an iPhone or Android phone and a Wi-Fi connection: [Instructions](#)

Option 2

Test your apps with a Chromebook: [Instructions](#)

Many Chromebooks are capable of running Android apps. That lets you create, test, and run the finished app on the same device.

To facilitate the live testing, we have downloaded the MIT App Inventor Companion app on our mobile phone and then connected it to the same network as the computer.

Connect your Phone or Tablet over WiFi



Step 1: Download and install the MIT App Inventor Companion app on your Android or iOS device.

Open the Google Play store or Apple App store on your phone or tablet, or use the buttons below to open the corresponding page:

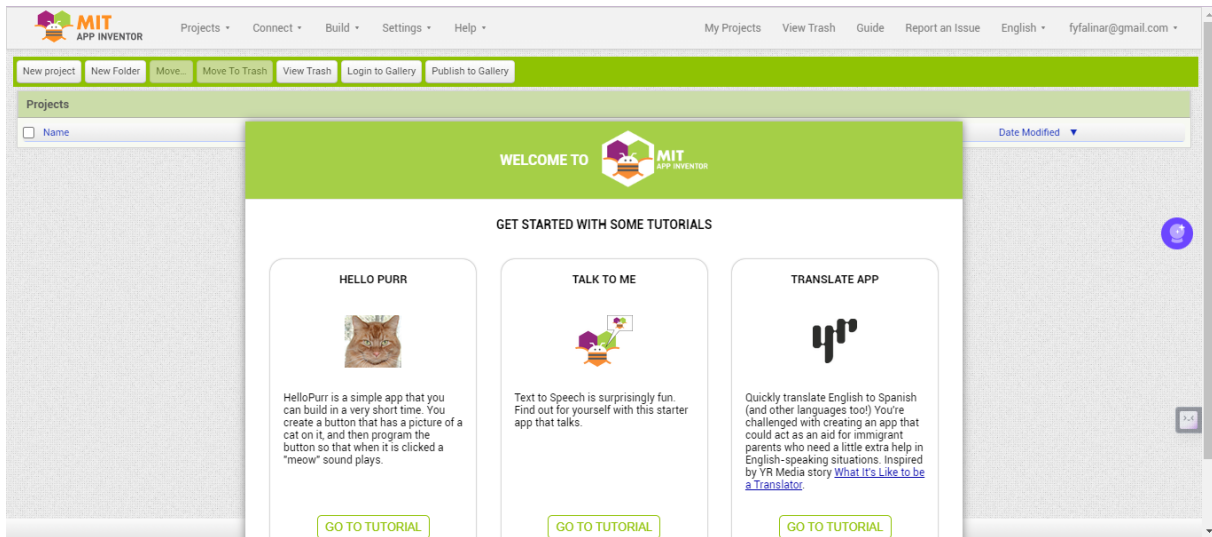


Step 2: Connect both your computer and your device to the **SAME** Wi-Fi network

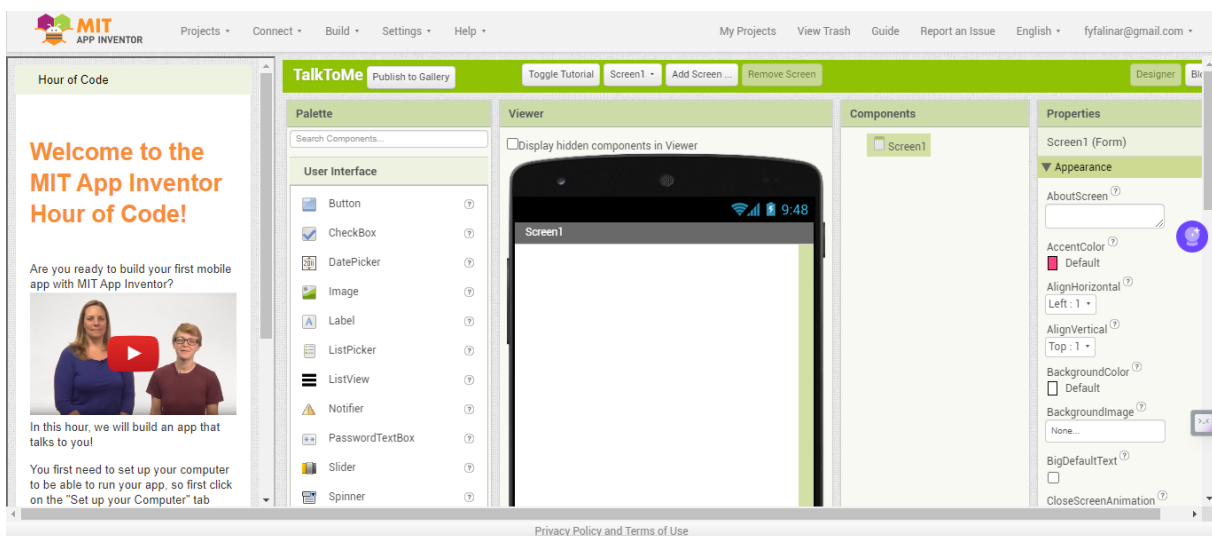
App Inventor will automatically show you the app you are building, but only if your computer (running App Inventor) and your device (running the Companion) are connected to the same Wi-Fi network. [See a more detailed explanation of this here.](#)

Exploring MIT App Inventor

Once the set up done, we started exploring the MIT App Inventor and all the available features.



We have started with the Talk To me tutorial as it is the best way to learn more about this powerful tool.



Tutorials are well explained.

We have started by setting up the computer for the live testing with the mobile phone.

Welcome to the MIT App Inventor Hour of Code!

Are you ready to build your first mobile app with MIT App Inventor?



In this hour, we will build an app that talks to you!

You first need to set up your computer to be able to run your app, so first click on the "Set up your Computer" tab below and follow the instructions. Then you should continue with the "Talk to Me" tab to make the app.

Set up your Computer

1 of 7

[Next](#)

Which way will you connect?

You have three ways to run your app.

1. Use the **emulator**, which runs your app in a window in your browser. Click [here](#) for instructions to set up your computer to run the emulator.
2. If you have a phone or tablet but do not have wifi access, you can **connect your device to your computer with a USB cable**. Instructions to set up your computer to use the USB connection, click [here](#).
3. **Connect your device over wifi**. If you choose this option, click the "Next Step" button below to follow instructions to set up your computer.

Next Step

Set up your Computer

[Previous](#)

2 of 7

[Next](#)

Connect with your device over wifi

These instructions will show you how to connect using the MIT AI2 Companion app, so you can see your app on your mobile device.



Build your project on your computer



Test it in real-time on your device

Next Step

Set up your Computer

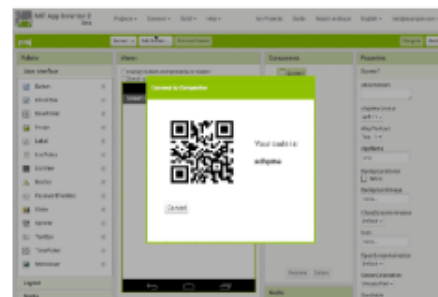
[Previous](#)

5 of 7

[Next](#)

Open your app's QR code

Click Connect, then AI Companion.




Next Step

Set up your Computer

[Previous](#) 6 of 7 [Next](#)

Connect your app with your phone

A QR code similar to this should pop up on your screen.



You can scan the QR code or enter the code in the text box.

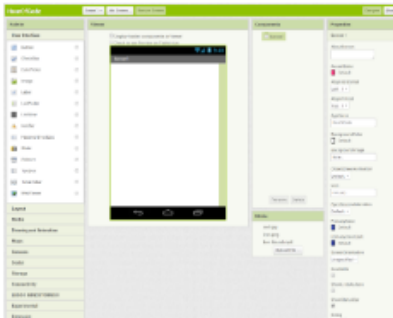
Next, the tutorial introduced us to the environment where we will create the app.

1 of 8 [Next](#)

Introduction to Environment

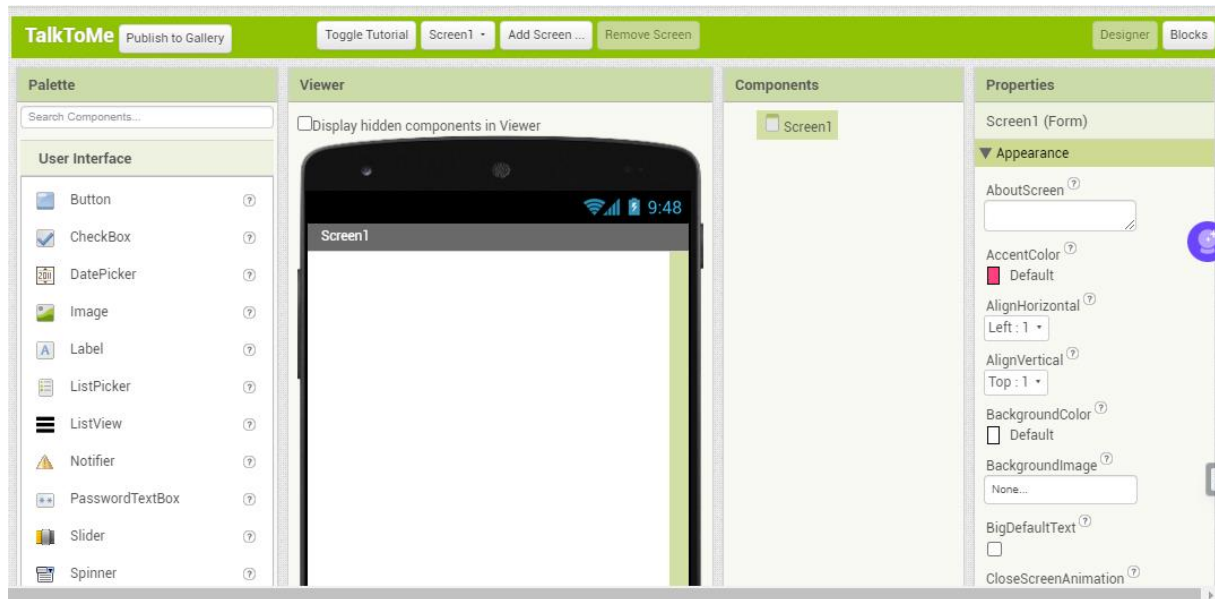
There are two main components to the App Inventor environment: the **Designer** and the **Blocks Editor**.

The **Designer** allows you to add *components* to your app and lay out what it will look like.

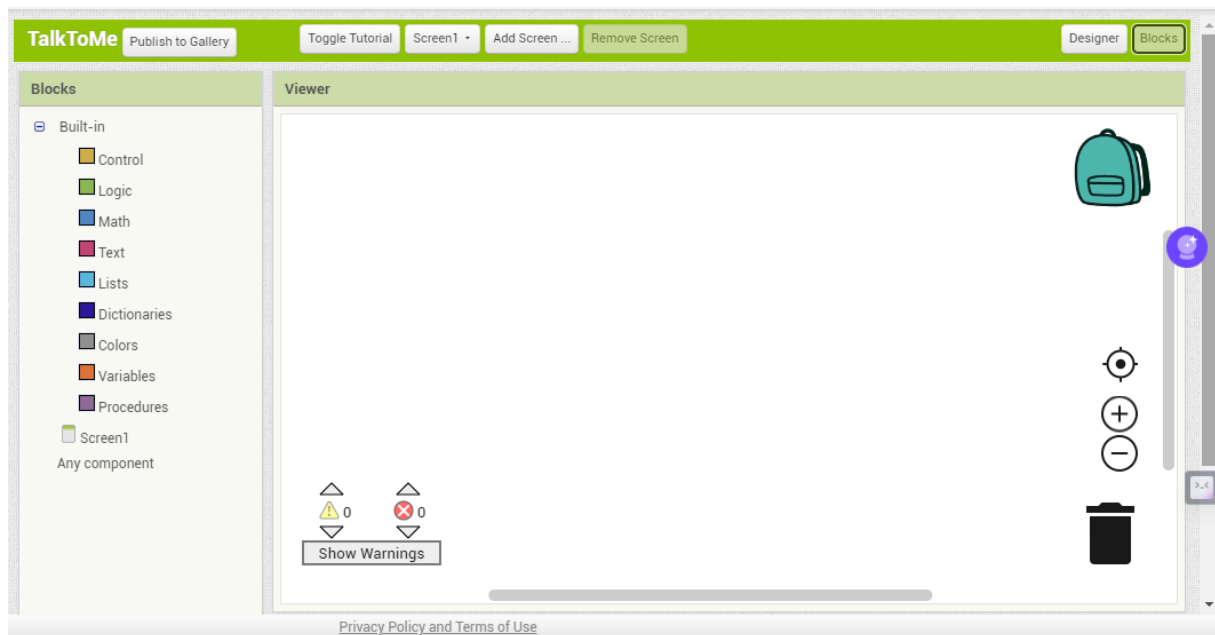


The **Blocks Editor** allows you to *code* your app to give it functionality--that is, make it do stuff!

The designer part will allow us to design our app using different interface already available. The MIT App Inventor is a drag and drop system.



The block part is where we will define the blocks of code that will define the functionalities of our app.



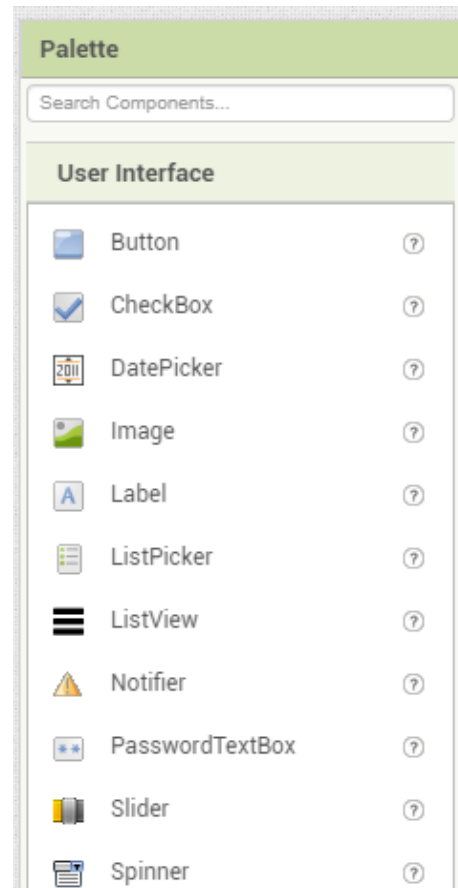
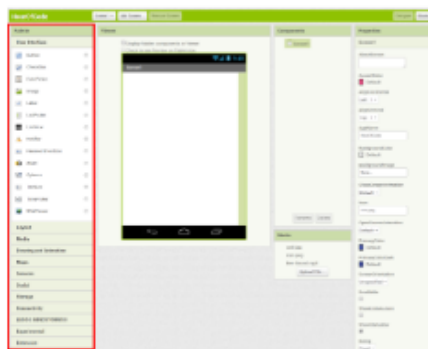
Next, the tutorial explains each component of each part.

[Previous](#) 2 of 8 [Next](#)

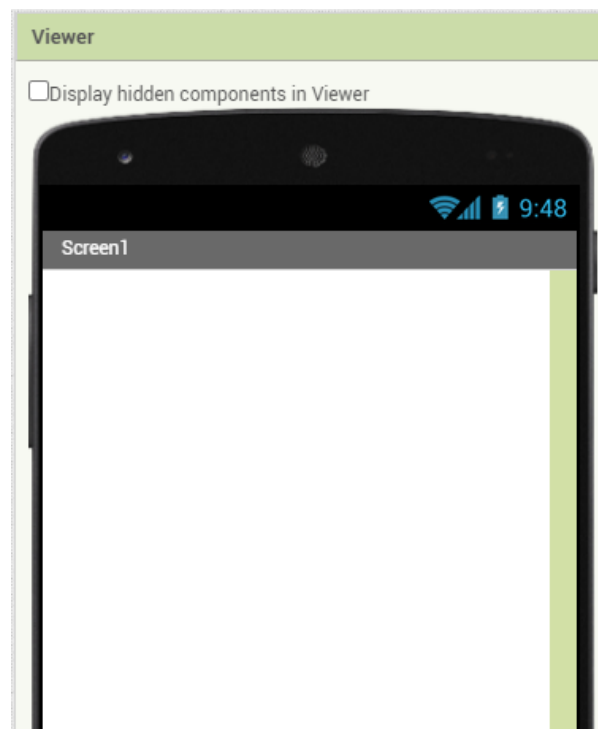
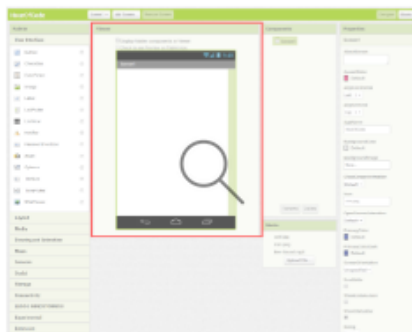
Designer Components

Within the Designer, there are five windows: *Palette*, *Viewer*, *Components*, *Media*, and *Properties*.

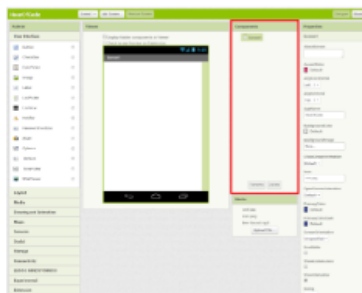
The **Palette** is where all available components are. You can drag and drop any component from the Palette into the *Viewer*.



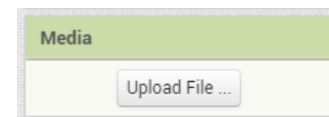
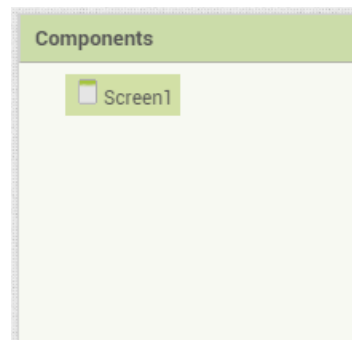
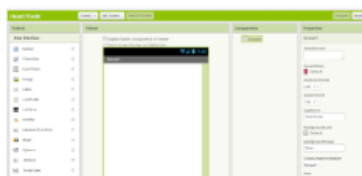
The **Viewer** is where you can see what the app will look like and is where you drag components.



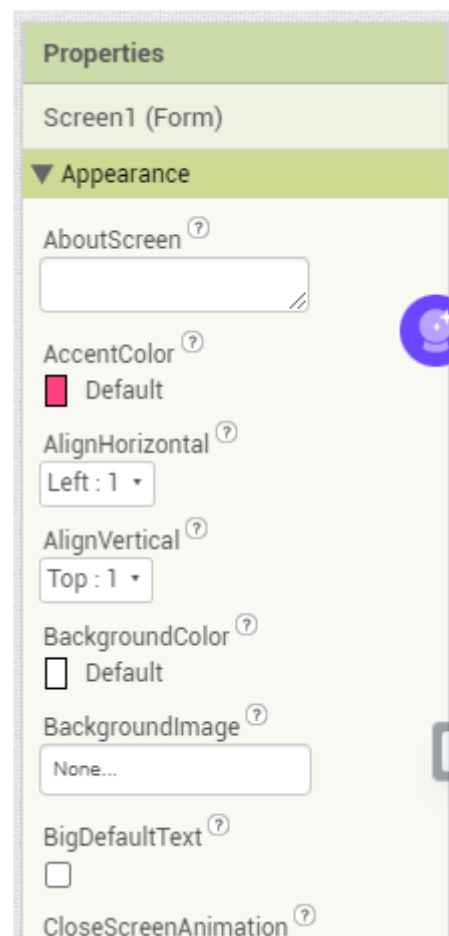
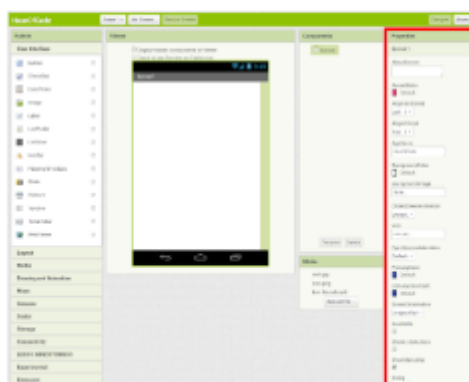
The **Components** window shows all components that have been added to the app. Note that the Screen component is automatically listed as a component.



The **Media** window shows any uploaded media files, like images and sounds.




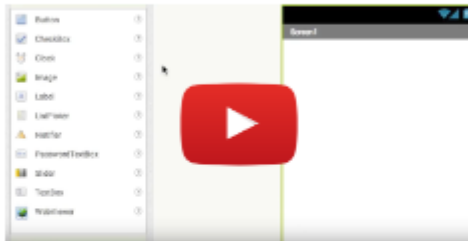
Finally, the **Properties** window allows you to view or change any of the properties (or characteristics) of the currently selected component.



Then, we started adding all the necessary component to create the Talk To Me app.

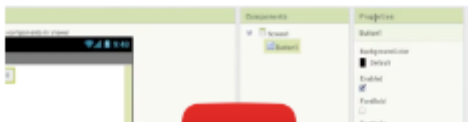
Add a Button

In the designer, add a Button  component.




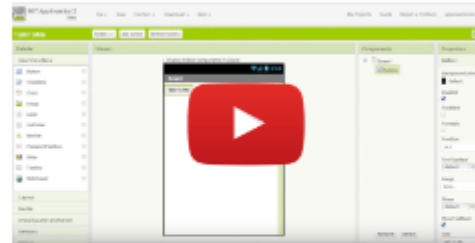
If you have your device connected, or emulator running, you should see your button appear.

Now change the Text property for Button1 to "Talk to Me".

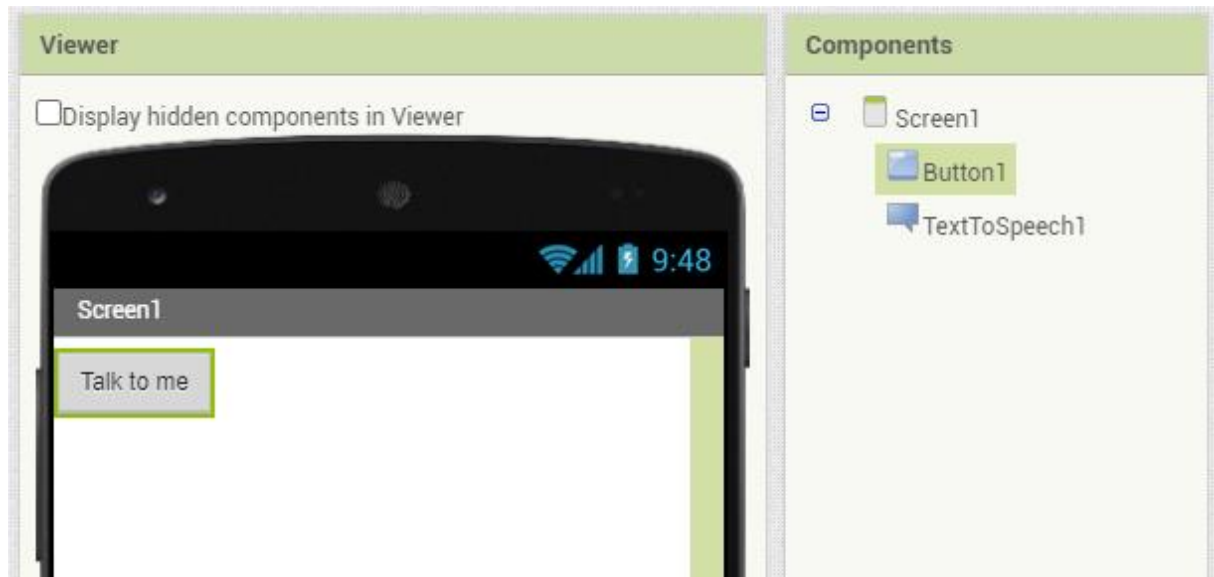


Add a TextToSpeech component

Now add another component, TextToSpeech , which is in the Media drawer.



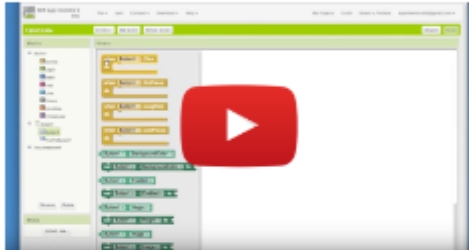
And that's it for the components! Now switch to the Blocks editor.



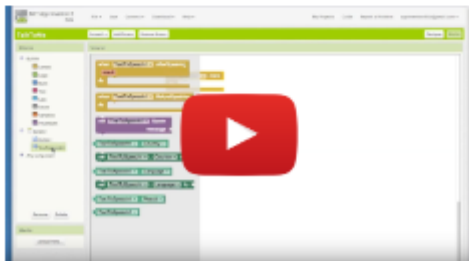
After working on the design part, we shifted to the block part where we arranged the blocks according to what we want the app to do.

Code the Button.Click event

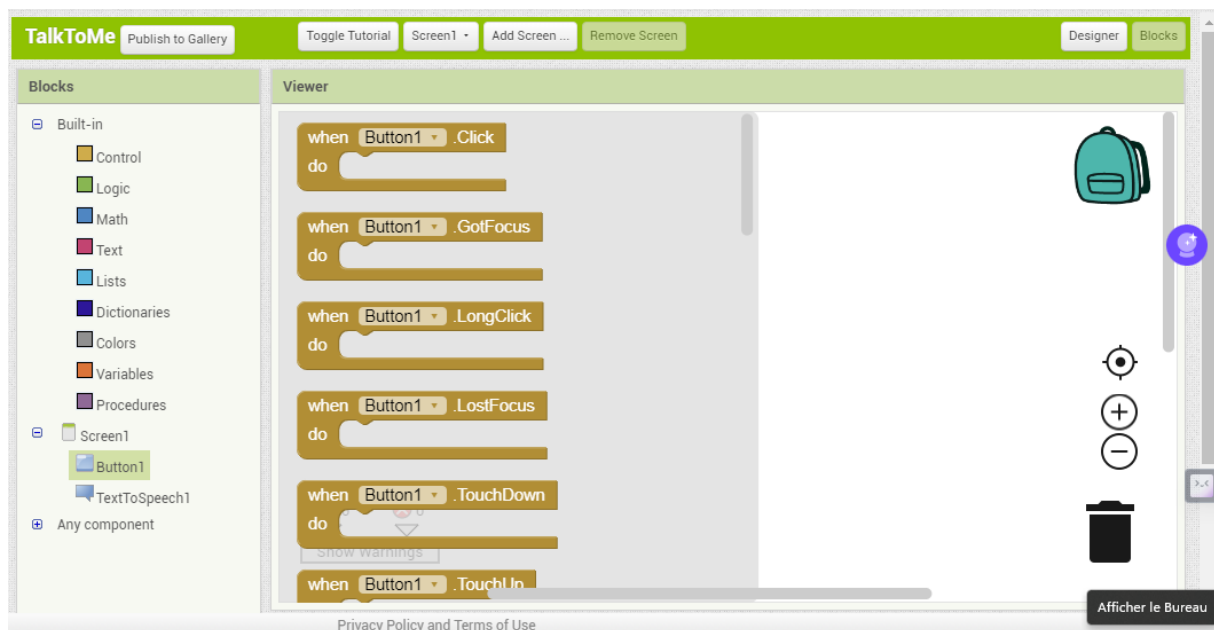
In the Blocks Editor, you will choose Button1, and drag in a Button1.Click event block.



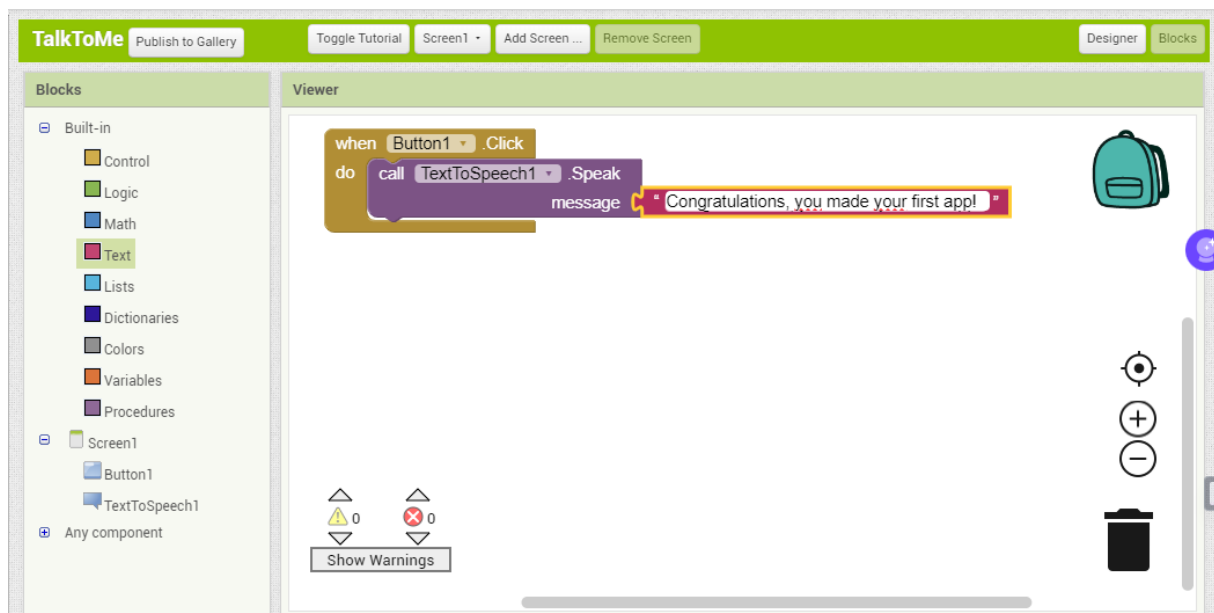
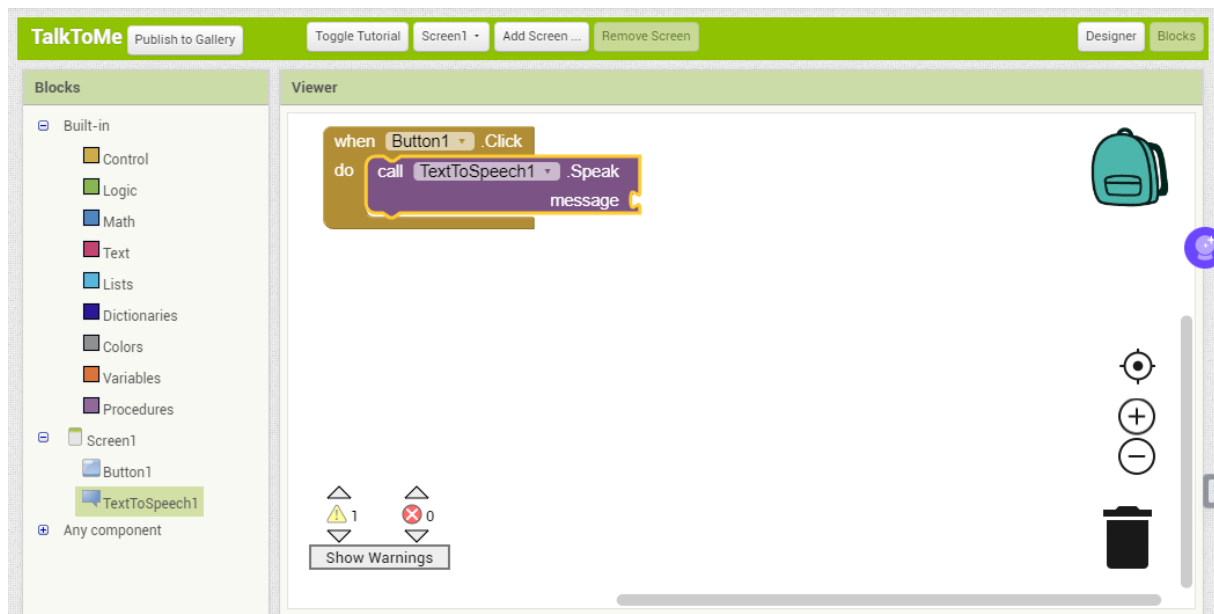
Now drop in a **TextToSpeech.Speak** block into the **Button1.Click** event block.



We can see that the blocks represents basic to complex ready to use algorithms.

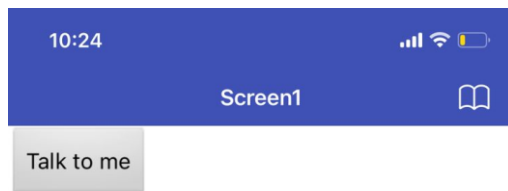


We just need match each blocks to create a function.



Here, we have created an on-click function that will read the text in the pink block every time the button is clicked.

We can see below how it is displayed on the mobile phone we have used for live testing. We did not add complicated design, that's why only the button appears.




The test was succesful, the phone successfully read the text every time I click on the button.
Now, we explore what else we can do with the blocks.

Talk To Me Part 2

1 of 7 [Next](#)

Add new features to Talk to Me

We will add shaking and variable text in Part 2.



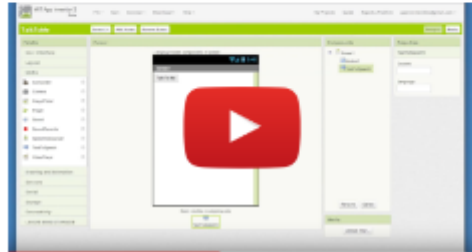
Next Step

Talk To Me Part 2

[Previous](#) 2 of 7 [Next](#)

Add the Accelerometer Sensor

Go back to the Designer window, and drag in an Accelerometer component from the Sensors drawer.

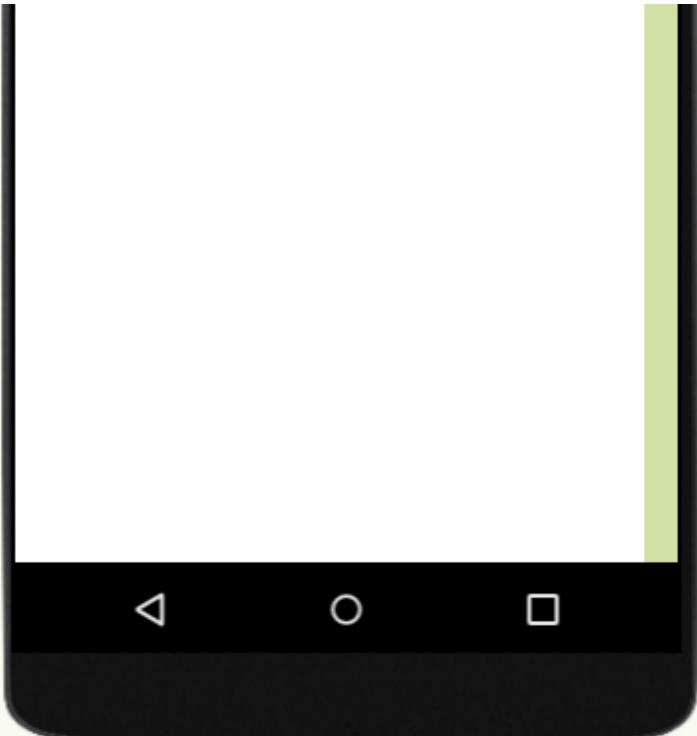


Sensors

- AccelerometerSensor ?
- BarcodeScanner ?
- Barometer ?
- Clock ?
- GyroscopeSensor ?
- Hygrometer ?
- LightSensor ?
- LocationSensor ?
- MagneticFieldSensor ?
- NearField ?
- OrientationSensor ?
- Pedometer ?
- ProximitySensor ?

Non-visible components

TextToSpeech1 AccelerometerSensor1



The accelerometer sensor detects any acceleration on the mobile phone.

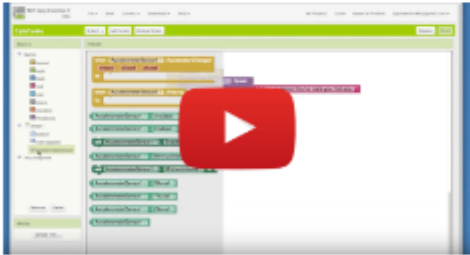
Talk To Me Part 2

[Previous](#) 3 of 7 [Next](#)

Accelerometer.Shaking block

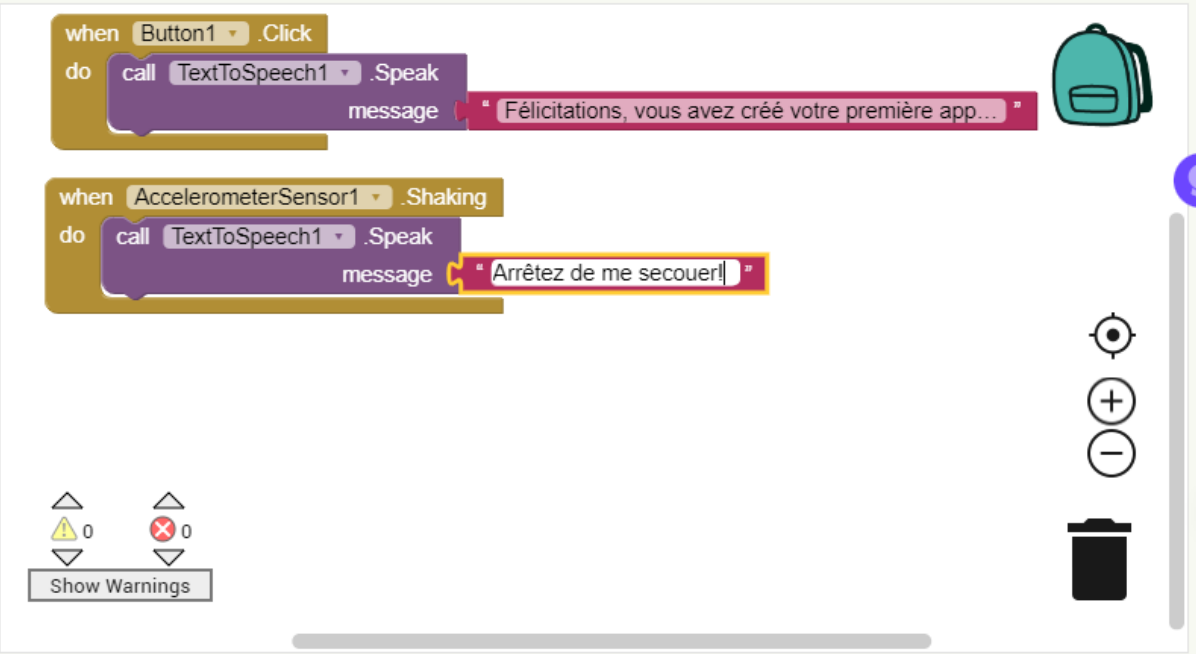
Back in the Blocks Editor, drag out an **Accelerometer1.Shaking** block.

Note: If you are working with the emulator, this code will not be able to be tested, because you can't shake your computer!



Copy and paste the **TextToSpeech.Speak** block from the **Button1.Click** event block, and snap it into the **Accelerometer1.Shaking** event block.

Viewer



```

when Button1.Click
do
  call TextToSpeech1.Speak
  message "Félicitations, vous avez créé votre première app..."

when AccelerometerSensor1.Shaking
do
  call TextToSpeech1.Speak
  message "Arrêtez de me secouer!"
  
```

0 0

Show Warnings

We added another function that will read the text in the pink block every time the accelerometer detects shaking movements.




The testing here was also successful.

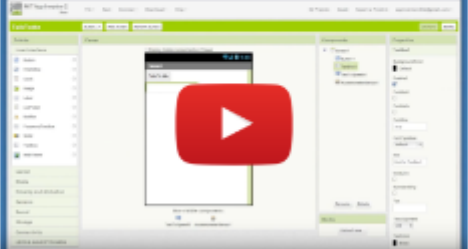
Now, we will try to make the app read everything written in a text box.

Talk To Me Part 2

[Previous](#) 5 of 7 [Next](#)

Add a TextBox

Go back to the Designer, and drag a TextBox  into the Viewer. Then move Button1  below the TextBox .



Next Step

Talk To Me Part 2

[Previous](#) 6 of 7 [Next](#)

Speak what is in the TextBox

Go back to the Blocks Editor. Then drag out a **TextBox1.Text** block and replace the "Congratulations" text block in the **Button1.Click** event block with the new block.



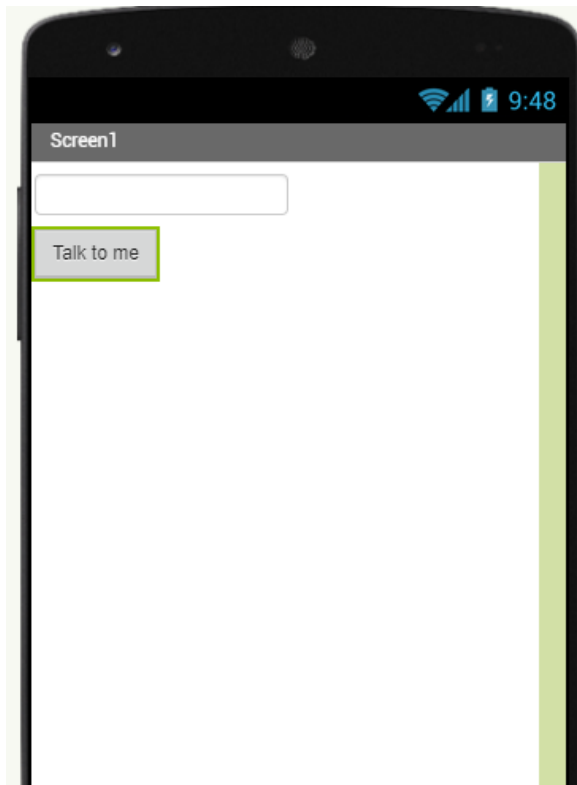
Next Step

Viewer

```

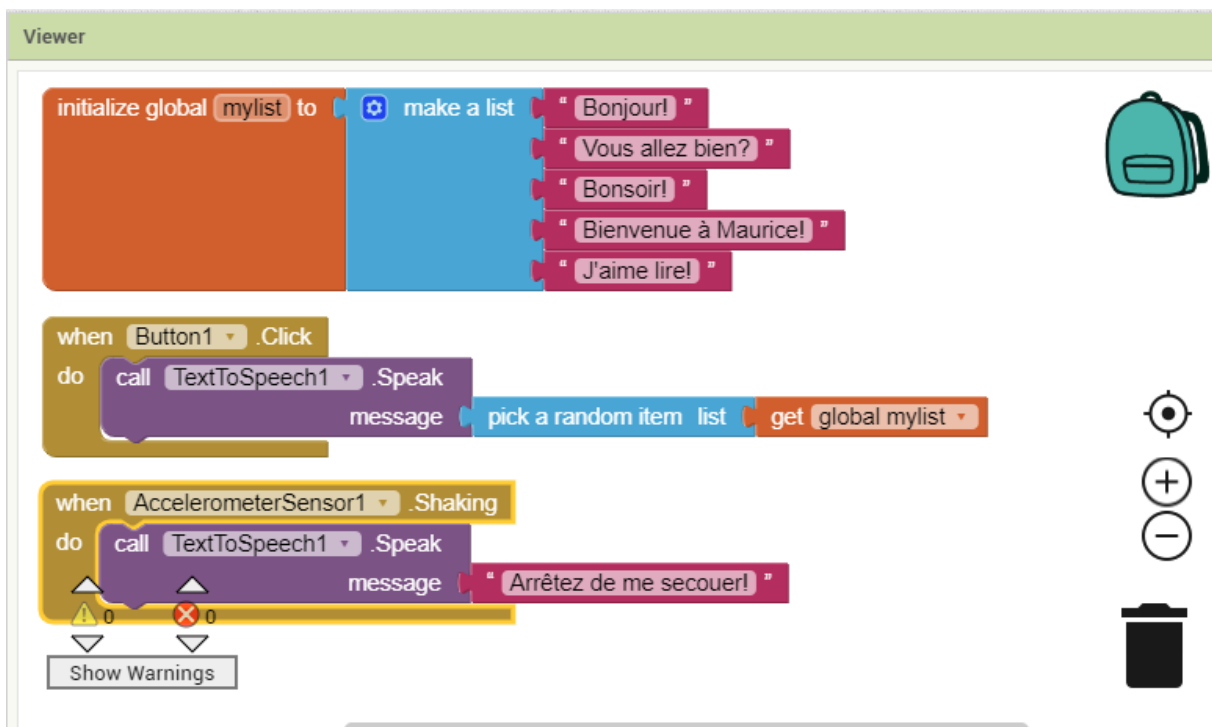
when Button1 .Click
do call TextToSpeech1 .Speak
    message TextBox1 . Text

when AccelerometerSensor1 .Shaking
do call TextToSpeech1 .Speak
    message " Arrêtez de me secouer! "
  
```

This attempt was successful.

Last, we have tried to make the app return random words from a list every time we click on the button and it worked.



After completion of the tutorial, they provided a simple certificate.



In summary, MIT App Inventor is an excellent platform for introducing application programming to beginners. It offers a user-friendly and intuitive interface, making it easy to use and understand. The platform provides well-explained tutorials and resources, which greatly assist beginners in learning the concepts of app development.

MIT App Inventor's visual programming language, known as blocks, simplifies the process of adding functionality to apps. The drag-and-drop interface allows users to visually connect blocks of code, eliminating the need for complex syntax and programming languages. This approach lowers the barrier to entry for beginners and enables them to quickly grasp programming concepts.