

Intro to AI with MIT App Inventor

Diego Barreiro - Open Source Contributor



FOSDEM - Feb 5, 2023

MIT App Inventor in a nutshell

The image displays the MIT App Inventor web interface, split into two panels: Design View (left) and Code View (right).

Design View (Left Panel):

- Challenge:** "How can a machine learn about the world? In some ways computers are like very young babies, always soaking up new examples and trying to put what they sense into different buckets — dog, cat, familiar face, stranger. In this two-part tutorial, you will learn about a type of artificial intelligence (AI) called machine learning (ML), exploring an example called 'image classification' — a way for computers to put what they see into various buckets. You will create an app that sees a human gesture that you make and intelligently responds! If you haven't set up your computer and mobile device for App Inventor, go to the 'Setup Your Computer' tab below. Otherwise, go directly to the 'PICaboo' tab to start the tutorial."
- App Name:** PICaboo_ Starter
- Palettes:** User Interface (Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, WebViewer), Layout, Media, Drawing and Animation, Maps, Charts, Sensors, Social, Storage.
- Viewer:** A mobile device mockup showing a "Loading..." screen with two progress bars at 0% and a globe image. Below the device are "Non-visible components" including PersonalImageClassifier1.
- Components:** Screen1, HorizontalArrangemen, BarLabel1, Percentage1, BarGraph1, BarLabel2, Percentage2, BarGraph2, WebViewer1, StartButton, Space, ToggleCameraButtc, HappyBaby.
- Properties:** Screen1, AboutScreen, AccentColor (Default), AlignHorizontal (Center), AlignVertical (Top), AppName (PICaboo), BackgroundColor (Custom), BackgroundImage (None), BigDefaultText (checkbox), BlocksToolkit (All), CloseScreenAnimation (Default), DefaultFileScope (App), HighContrast (checkbox), Icon (None), OpenScreenAnimation (Default), PrimaryColor (Default), PrimaryColorDark (Default), ScreenOrientation.

Code View (Right Panel):

- Blocks:** Built-in (Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures), Screen1, Media (PICabooModel.mdl, happyBaby.png, sadBaby.png).
- Code Snippets:**
 - when PersonalImageClassifier1 -> Error:** set StatusLabel -> Text to get errorCode ->
 - when PersonalImageClassifier1 -> ClassifierReady:** do set StartButton -> Enabled to true, set ToggleCameraButton -> Enabled to true, set StatusLabel -> Text to Ready, set BarLabel1 -> Text to Me, set BarLabel2 -> Text to NotMe.
 - when ToggleCameraButton -> Click:** do call PersonalImageClassifier1 -> ToggleCameraFacingMode.
 - when StartButton -> Click:** do if PersonalImageClassifier1 -> Running then call PersonalImageClassifier1 -> StopContinuousClassification, set StartButton -> Text to Start, else call PersonalImageClassifier1 -> StartContinuousClassification, set StartButton -> Text to Stop.
 - when PersonalImageClassifier1 -> GoClassification:** do initialize local MeConfidence to, initialize local NotMeConfidence to.

MIT App Inventor by numbers

- Started in 2008 at Google
- Open sourced and transferred to MIT in 2011
- 18 million users since inception
- 84.6 million projects created
- 902 thousand users in the last month
- 164 contributors on GitHub



Visit

fosdem23.appinventor.mit.edu

to access links to all of the
materials in this talk.



Use MIT App Inventor 2

Creating an anonymous account

Use the link to
code.appinventor.mit.edu
to open the PICaboo starter
project.

Click **Continue Without an
Account** to get started.

Welcome to MIT App Inventor!

 **Continue Without An Account**

or

Your Revisit Code: ---

Enter with Revisit Code

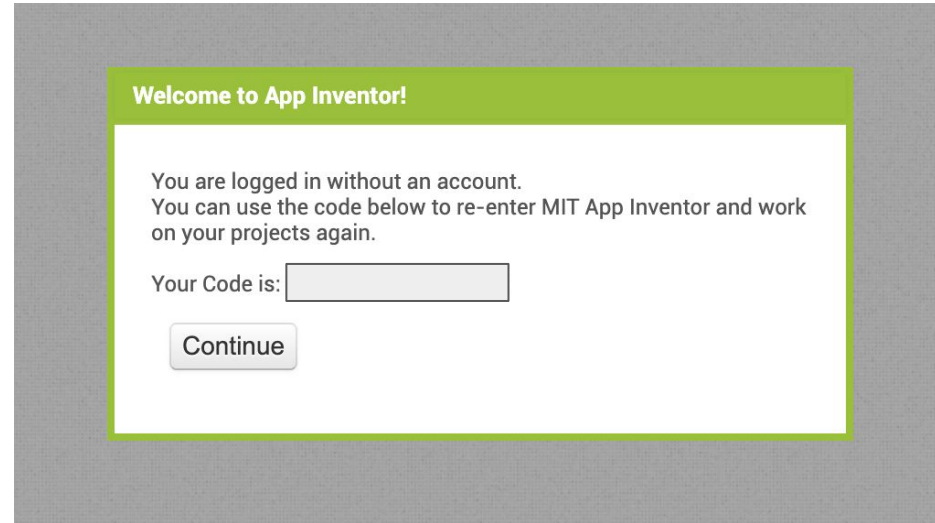


Creating an anonymous account

Write down your code if you want to return.

Click **Continue**

If you get a splash screen,
click **Continue**



Starting the PICaboo Project

The screenshot displays the MIT App Inventor web interface for the PICaboo Starter project. The interface is divided into several panels:

- Challenge:** Contains introductory text about machine learning and a link to a tutorial.
- Setup Your Computer:** A section for configuring the development environment.
- Palette:** A list of UI components categorized into User Interface, Layout, Media, Drawing and Animation, Maps, Charts, Sensors, Social, and Storage.
- Viewer:** A central window showing a mobile app preview. The app displays a "Loading..." screen with two progress bars at 0% and a globe image. Below the preview, a "Non-visible components" section lists "PersonalImageClassifier1".
- Components:** A list of components currently on the screen, including StatusLabel, HorizontalArrangemen, BarLabel1, Percentage1, BarGraph1, BarLabel2, Percentage2, BarGraph2, WebViewer1, StartButton, Space, ToggleCameraButtc, and HappyBaby.
- Properties:** A panel for configuring the selected component (Screen1), including fields for AboutScreen, AccentColor, AlignHorizontal, AlignVertical, AppName, BackgroundColor, BackgroundImage, BigDefaultText, Blocks Toolkit, CloseScreenAnimation, DefaultFileScope, HighContrast, Icon, OpenScreenAnimation, PrimaryColor, and ScreenOrientation.

Train a Classifier

Train a classifier

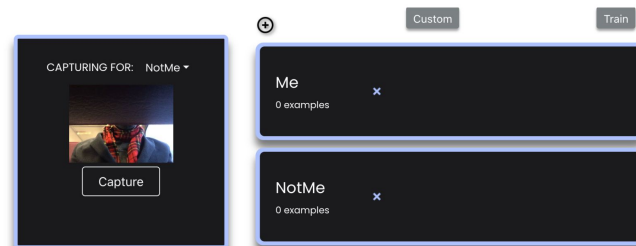
Personal Image Classifier Train Test **Previous Version**

Training Page

To get started, click the plus icon to add a classification and then use the "Capture" button or drag images into the capture box to add images to the selected classification. You can also upload previously generated data and models using the buttons below. When done, hit "Train"

In a new tab, navigate to
classifier.appinventor.mit.edu

This webpage uses your webcam, but no images leave your computer. All training is done in your browser using Tensorflow.js.



Upload Model

Upload Training Data

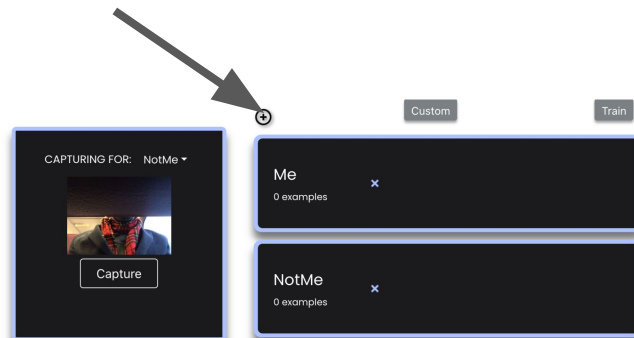
Train a classifier

Personal Image Classifier Train Test [Previous Version](#)

Training Page

To get started, click the plus icon to add a classification and then use the "Capture" button or drag images into the capture box to add images to the selected classification. You can also upload previously generated data and models using the buttons below. When done, hit "Train"

Click the + icon to create two labels: **Me** and **NotMe**



Train a classifier

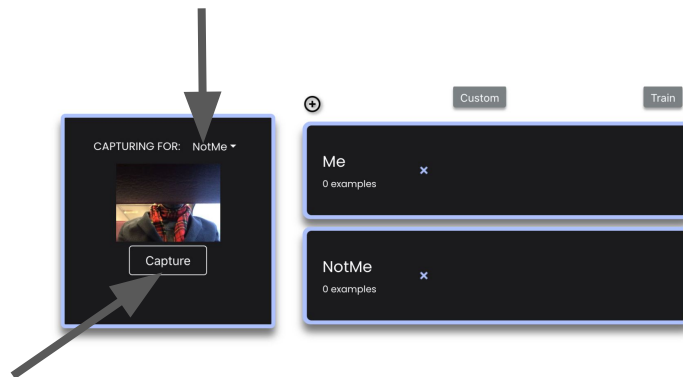
Select the **Me** label and click **Capture** to gather images

Try to get images at slightly different angles and lighting

Personal Image Classifier Train Test [Previous Version](#)

Training Page

To get started, click the plus icon to add a classification and then use the "Capture" button or drag images into the capture box to add images to the selected classification. You can also upload previously generated data and models using the buttons below. When done, hit "Train"



Train a classifier

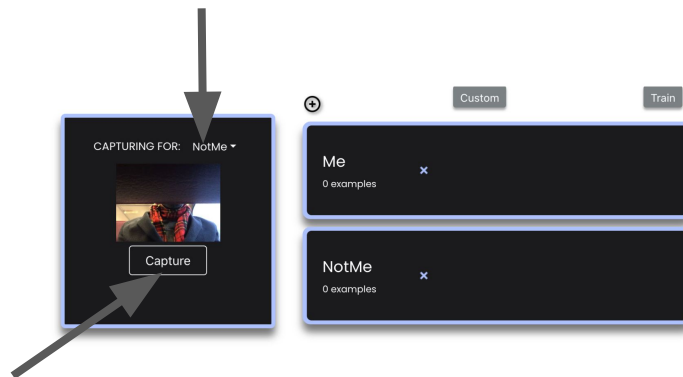
Select the **NotMe** label, cover your face, and click **Capture** to gather images

Try to get images at slightly different angles and lighting

Personal Image Classifier Train Test **Previous Version**

Training Page

To get started, click the plus icon to add a classification and then use the "Capture" button or drag images into the capture box to add images to the selected classification. You can also upload previously generated data and models using the buttons below. When done, hit "Train"



Upload Model

Upload Training Data

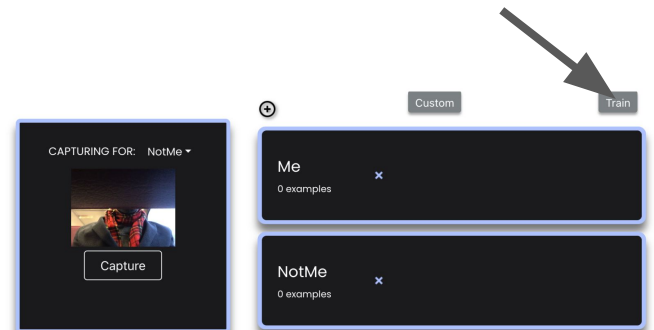
Train a classifier

Personal Image Classifier Train Test **Previous Version**

Training Page

To get started, click the plus icon to add a classification and then use the "Capture" button or drag images into the capture box to add images to the selected classification. You can also upload previously generated data and models using the buttons below. When done, hit "Train"

Click **Train** to start training the model

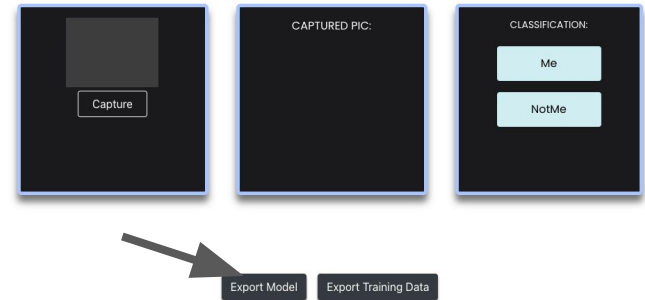


Train a classifier

Testing Page

With a model now generated, you can simply add images as you did in the Training portion to classify them. You can then scroll down to see an overview of the results. When done, y

Click **Export Model** to save the model to your computer

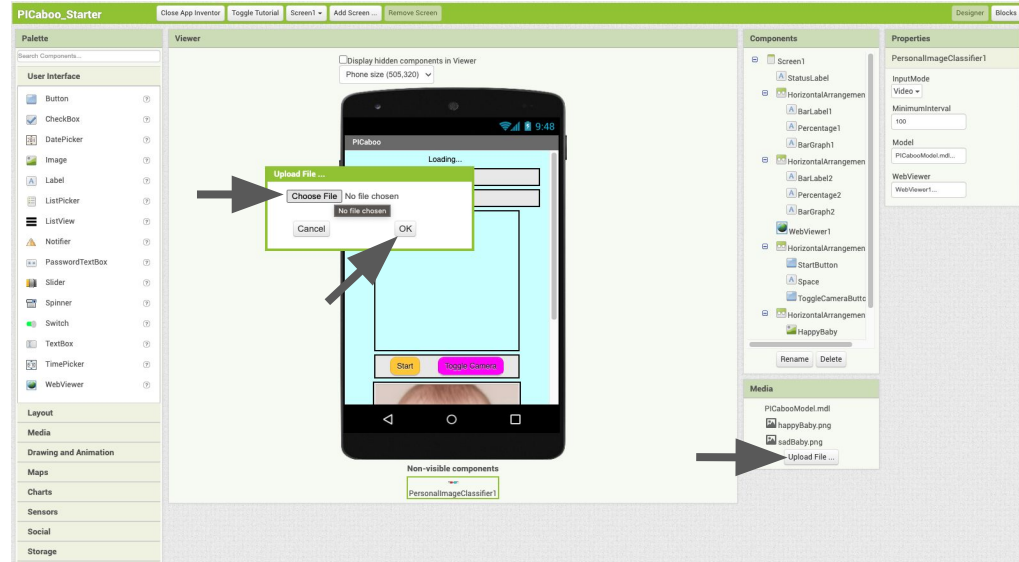


Coding Your App

Uploading your Model

Under **Media**, click **Upload File...**

In the dialog, select your **model.mdl** file and click **OK** to upload



Enabling your Model

Click on
PersonalImageClassifier1 below
the screen.

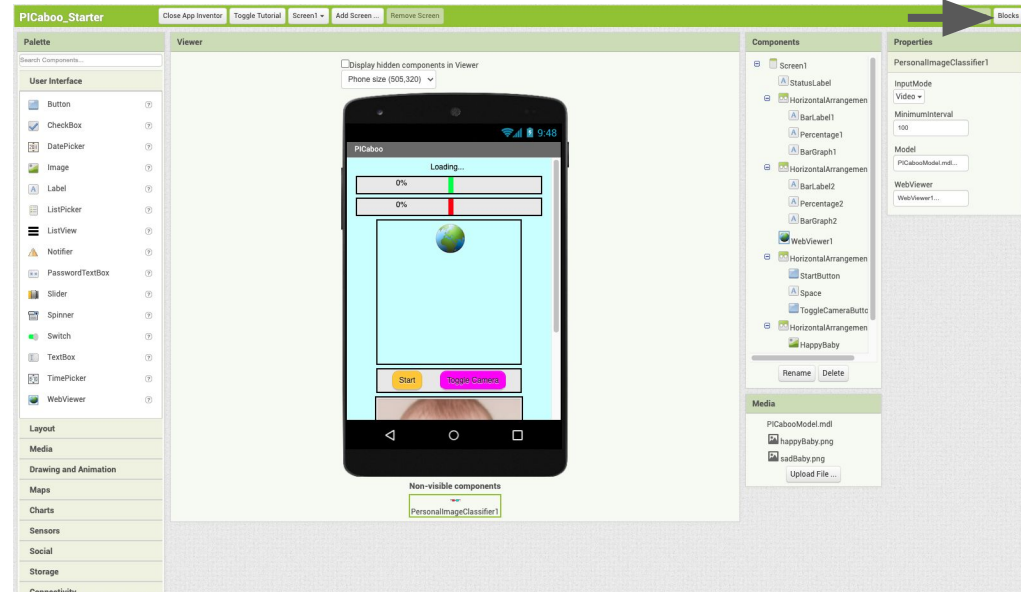
In the **Properties** list, select
Model, choose your file from the
dropdown, and click **OK**.

The screenshot displays the MIT App Inventor interface for a project named 'PICaboo_Starter'. The interface is divided into several panels:

- Palette:** Lists various UI components under 'User Interface' and 'Layout'.
- Viewer:** Shows a mobile device preview with a loading screen. A dropdown arrow at the bottom of the device points to a 'Non-visible components' list containing 'PersonalImageClassifier1'.
- Components:** A tree view of the app's components, including 'PersonalImageClassifier1'. An arrow points to this component.
- Properties:** Shows the properties for 'PersonalImageClassifier1'. The 'Model' property is selected, and a dropdown menu is open, showing 'None', 'PICabooModel.mdl', 'happyBaby.png', and 'sadBaby.png'. An arrow points to the 'Upload File...' button, and another arrow points to the 'OK' button.

Let's Start Coding

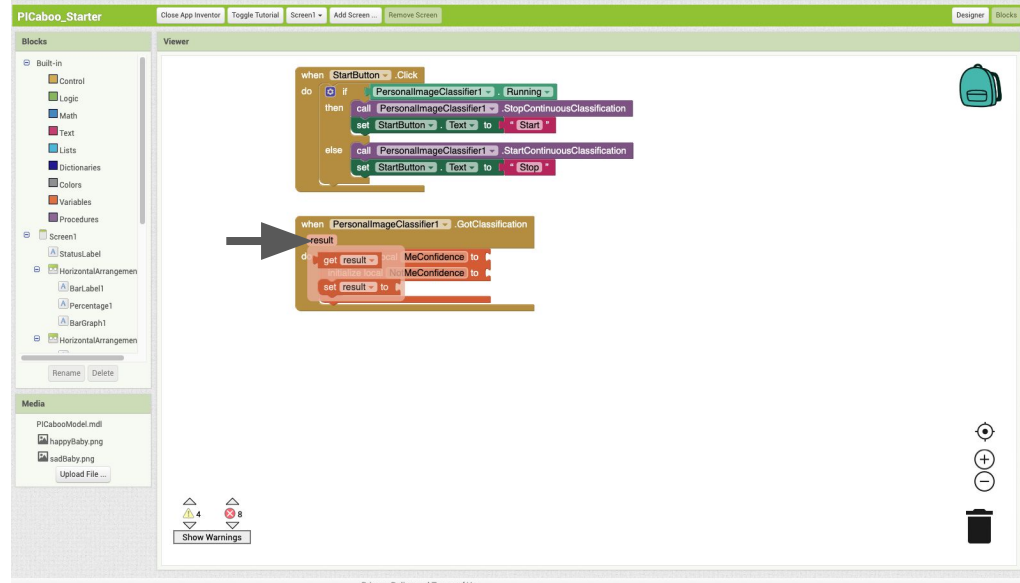
In the top right corner, click **Blocks** to switch to the blocks editor.



Accessing Variables

Hover your mouse over the **result** field to get a dropdown with blocks in it

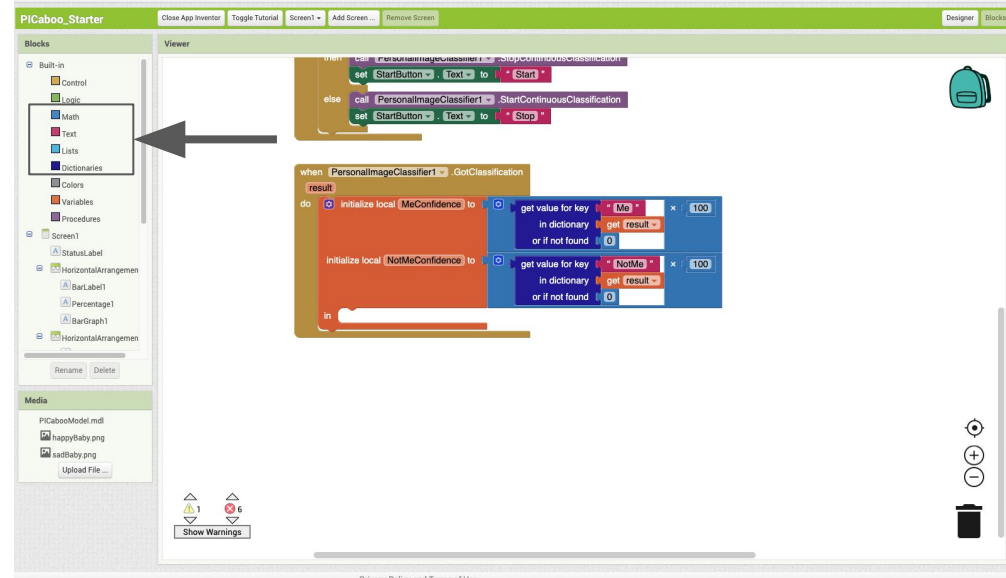
Drag the **get result** block into the workspace



Compute Confidence Levels

Using the **Math**, **Text**, and **Dictionary** blocks, construct the blocks below to compute the confidence of the **Me** class

Right click and select **Duplicate** to copy the blocks to make a **NotMe** version as well

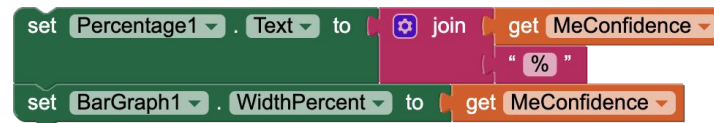
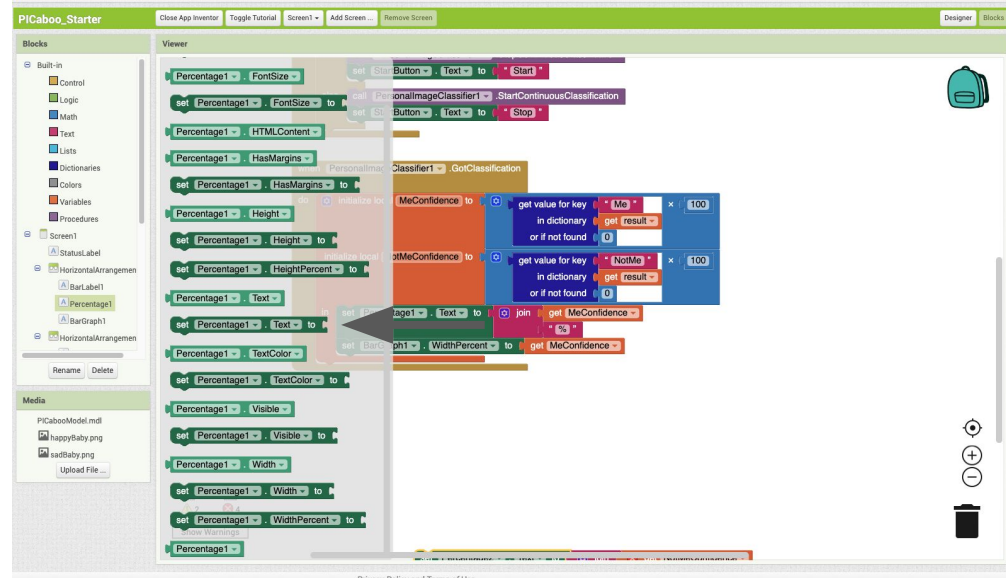


Updating the UI (Part 1)

Select **Percentage1** label in the list and grab its **set Text** block

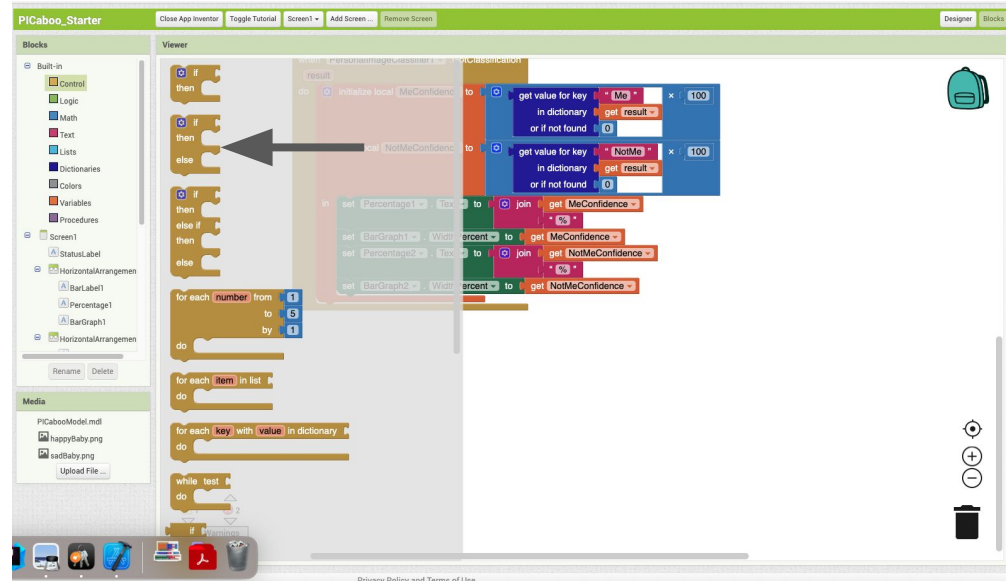
Do the same for **BarGraph1** and its **WidthPercent**

Construct the blocks to the right and then duplicate it for **Percentage2** and **BarGraph2**



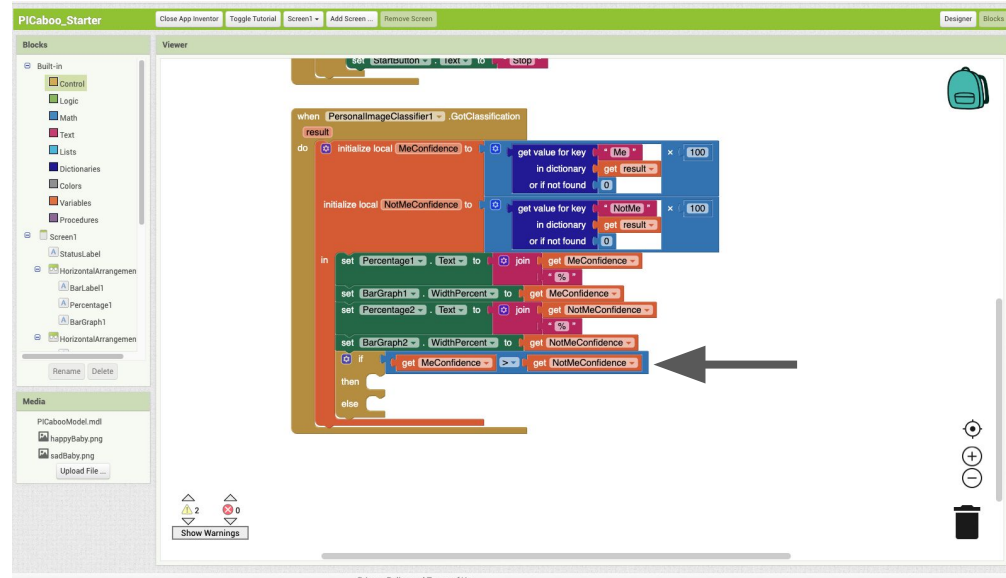
Updating the UI (Part 2)

In the **Control** category, select the **if then else** block so we can update the view based on the confidences



Updating the UI (Part 3)

Grab the = block from the **Math** category and use its dropdown to get the > operator and use it to compare **MeConfidence** to **NotMeConfidence**



The screenshot shows the MIT App Inventor interface for a project named "PICaboo_Starter". The left sidebar contains a "Blocks" palette with categories like Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. The main workspace shows a code block for "when PersonalImageClassifier1.GoClassification" with a "do" loop. Inside the loop, there are two "initialize local" blocks for "MeConfidence" and "NotMeConfidence", each followed by a "get value for key" block with a dropdown menu set to "Me" and "NotMe" respectively, and a multiplication by 100. Below these are "set" blocks for "Percentage1" and "Percentage2" using "Width-Percent" and "join" blocks. At the bottom, an "if" block compares "MeConfidence" and "NotMeConfidence" using the ">" operator, with a "then" block containing a "set" block for "BarGraph2" and a "show warnings" block. A black arrow points to the ">" operator dropdown in the if block.

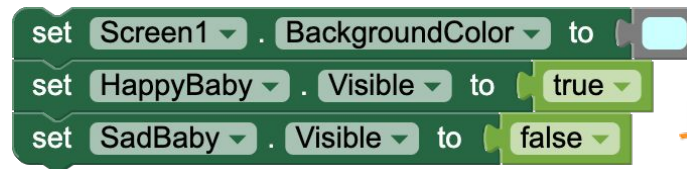
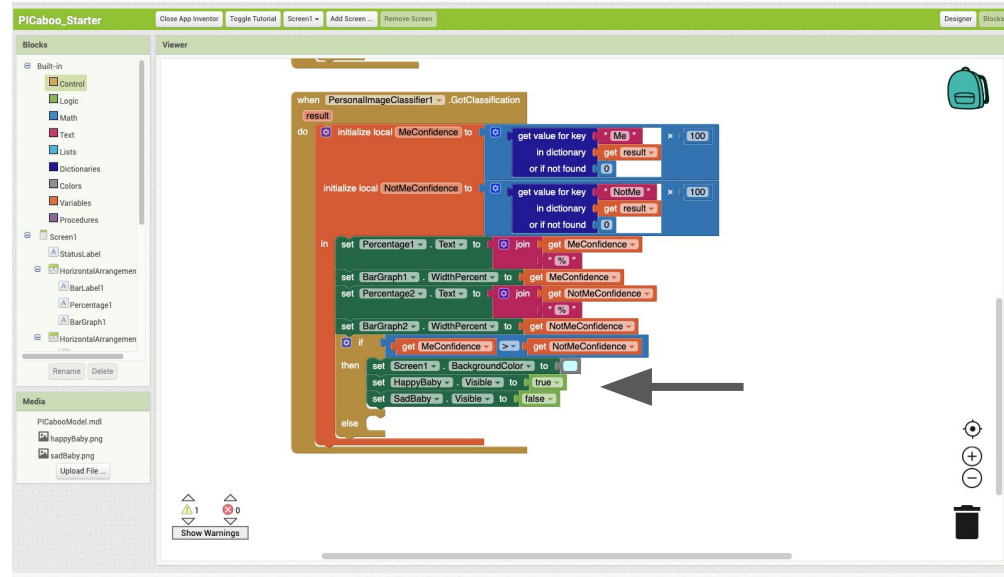
Updating the UI (Part 4)

Select Screen1 from the list and grab the set BackgroundColor block to change it to light blue

Select HappyBaby and select its set Visible block

Duplicate HappyBaby.Visible and use the dropdown to select SadBaby

Build the blocks to the right and plug it into the if block's "then" statement

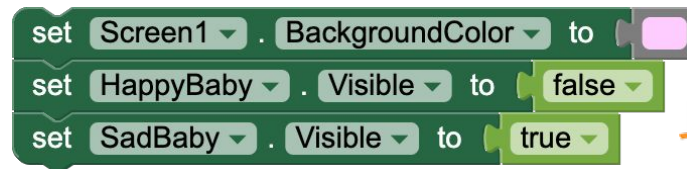
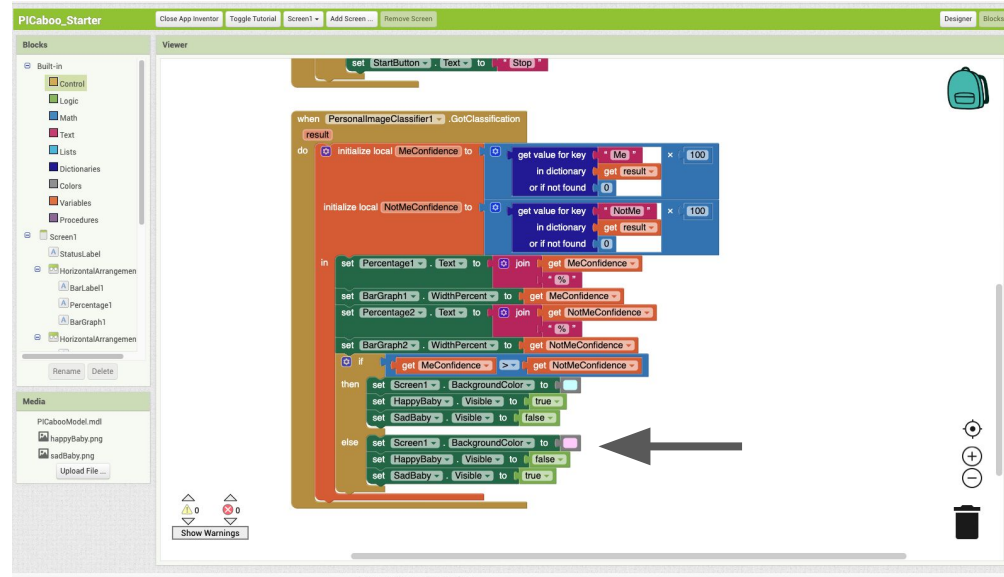


Updating the UI (Part 5)

Duplicate the Screen1, HappyBaby, and SadBaby logic from the previous slide

Change the background color to pink

Swap the true/false blocks to reverse the logic



Test Your App

Install the Companion App

Scan the QR code to install the MIT AI2 Companion app on your Android device

Or install it via the Google Play Store



Connect the Companion

From the **Connect** menu, select
AI Companion

Use the companion app to scan
the QR code that appears

