



Coding for Young People

Translate It!

Train the Trainer Webinar

Introduction to Building Smartphone Apps in Android
Using the MIT App Inventor Platform

Trainer: Rosanna Kurrer
with Cheryl Miller and Jil Theis

Digital Leadership Institute International



History of the App Inventor Platform

About Us

What is MIT App Inventor?

MIT App Inventor is an innovative beginner's introduction to programming and app creation that transforms the complex language of text-based coding into visual, drag-and-drop building blocks. The simple graphical interface grants even an inexperienced novice the ability to create a basic, fully functional app within an hour or less.

Mission

The MIT App Inventor project seeks to democratize software development by empowering all people, especially young people, to transition from being consumers of technology to becoming creators of it.

History

Google's Mark Friedman and MIT Professor Hal Abelson co-led the development of App Inventor while Hal was on sabbatical at Google in 2009. Other early Google engineer contributors were Sharon Perl, Liz Looney, and Ellen Spertus. App Inventor runs as a web service administered by staff at MIT's Center for Mobile Learning - a collaboration of MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) and the MIT Media Lab.



Taken from MIT App Inventor website: <http://appinventor.mit.edu/explore/about-us.html>





Trainer Goals

- To introduce the App Inventor User interface, and guide the participants in understanding and using the Designer editor and Blocks editor;
- To guide the participants through basic tutorials and facilitate their understanding of block programming;
- To point out the resources available online to encourage them to deepen their skill after the workshop by building more apps and using other block components;
- To introduce the concept of pair programming and collaboration in problem solving (https://en.wikipedia.org/wiki/Pair_programming).





Learning Objectives:

The participant should be able to:

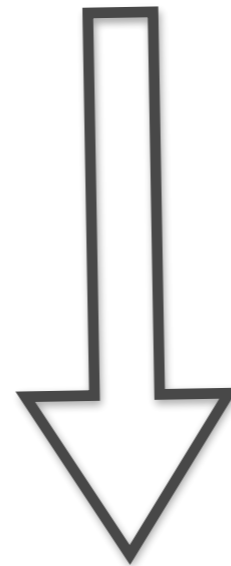
- effectively use the Designer editor and Blocks editor;
- appreciate the concept of blocks programming and build simple apps;
- understand the potential of the App Inventor platform in teaching computer science principles;
- appreciate the advantages of pair programming and collaboration in solving problems and creating digital solutions.





Mindset change required!

Source of all Knowledge



Facilitator for Learning



Pre-workshop preparation:

- Inform the participants beforehand that they will need a Gmail account;
- Make sure there is a good WiFi connection on-site, you test the connection between your laptop and mobile device onsite before the class/workshop;
- Install the MIT App Inventor 2 Companion, a free App available at the Google Play store, on the Android device and test (a device can be shared among up to three pairs of participants);
- Prepare slide deck and print out handouts;
- Prepare feedback forms.

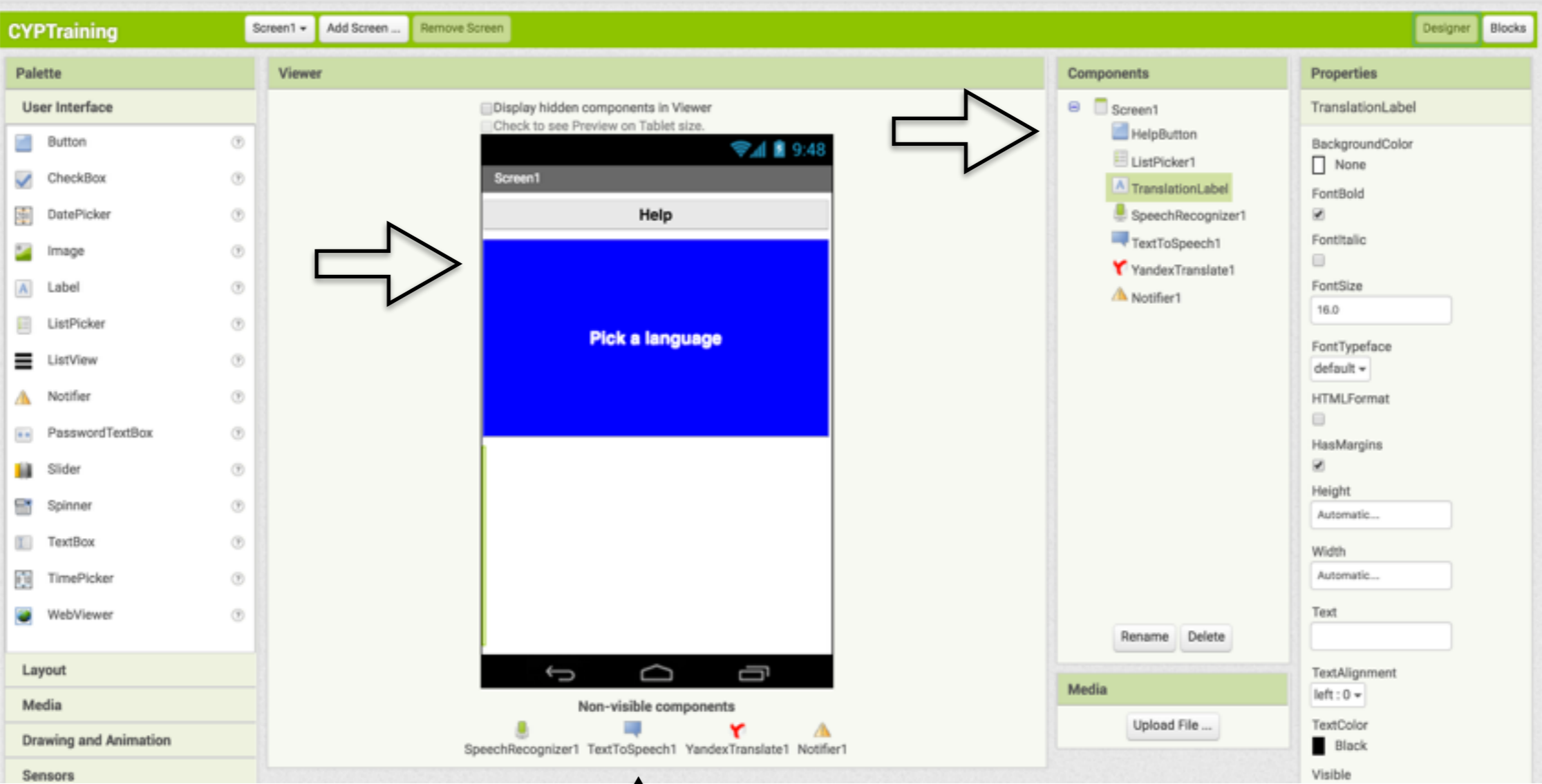


Time	Activity	Focus
±15 minutes	What is App Inventor?	<ul style="list-style-type: none"> -Introduce App Inventor, the history and the mission of the AI team (http://appinventor.mit.edu/explore/about-us.html); -Introduce block programming; -Introduce the designer and blocks editors; -Introduce the concept of pair-programming (https://en.wikipedia.org/wiki/Pair_programming).
±20 minutes	Tutorial 1: Talk to Me (http://appinventor.mit.edu/explore/sites/all/files/hourofcode/TalkToMePart1.pdf)	<ul style="list-style-type: none"> -Event-driven programming (https://en.wikipedia.org/wiki/Event-driven_programming); -Designing User Interface Components; -Adding behaviour to the components.
±5 minutes (optional)	Add shake feature to Talk to Me App (http://appinventor.mit.edu/explore/sites/all/files/hourofcode/TalkToMePart2.pdf)	<ul style="list-style-type: none"> -Using the sensors of a mobile device (http://appinventor.mit.edu/explore/support/glossary.html).
±20 minutes	Tutorial 2: Translate It! App	Tutorial follows



Step 1: Set up the User Interface

 (Designer Editor)



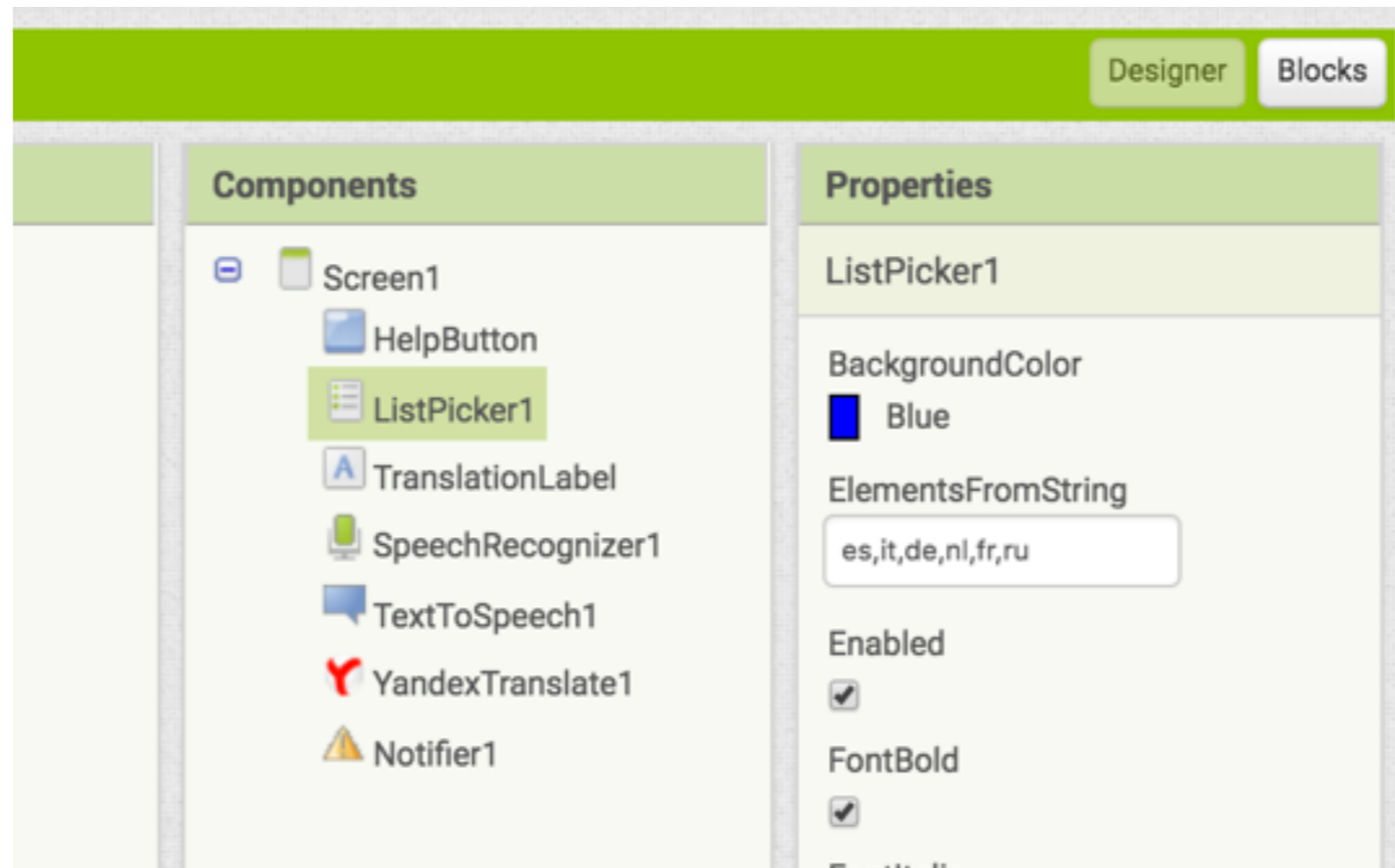
The screenshot displays the CYPTraining Designer Editor interface. The top bar shows the project name "CYPTraining" and navigation options like "Screen1", "Add Screen...", and "Remove Screen". The interface is divided into several panels:

- Palette:** A list of UI components such as Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, TextBox, TimePicker, and WebViewer.
- Viewer:** A central area showing a mobile app preview. The app has a blue header with "Help" and a large blue button labeled "Pick a language".
- Components:** A list of components currently on the screen, including HelpButton, ListPicker1, TranslationLabel, SpeechRecognizer1, TextToSpeech1, YandexTranslate1, and Notifier1.
- Properties:** A panel for configuring the selected component (TranslationLabel), showing properties like BackgroundColor, FontBold, FontItalic, FontSize (16.0), FontTypeface, HTMLFormat, HasMargins, Height, Width, Text, TextAlignment, TextColor, and Visible.
- Non-visible components:** A row of icons for components not currently visible on the screen, including SpeechRecognizer1, TextToSpeech1, YandexTranslate1, and Notifier1.

Four large white arrows point to the Palette, Viewer, Components, and Properties panels, highlighting their roles in the design process.

Step 2: Specify languages: List picker component

 (Designer Editor)



Step 3: Enable Speech Recogniser

List picker block



(Blocks Editor)

The screenshot shows the CYPTraining Blocks Editor interface. The top bar includes the title "CYPTraining" and navigation buttons for "Screen1", "Add Screen ...", and "Remove Screen". The left sidebar, titled "Blocks", lists various categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1 (HelpButton, ListPicker1, TranslationLabel, SpeechRecognizer1, TextToSpeech1, YandexTranslate1, Notifier1), and Any component. The main "Viewer" area displays a script with the following blocks:

- Initialize global language to ""
- when ListPicker1 .AfterPicking
- do
 - set global language to ListPicker1 . Selection
 - call SpeechRecognizer1 .GetText

Step 4: Use Yandex Translate SpeechRecognizer block



CYPTraining Screen1 Add Screen ... Remove Screen

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - HelpButton
 - ListPicker1
 - TranslationLabel
 - SpeechRecognizer1
 - TextToSpeech1
 - YandexTranslate1
 - Notifier1
- Any component

Viewer

```
when SpeechRecognizer1 .AfterGettingText
  result
do
  call YandexTranslate1 .RequestTranslation
    languageToTranslateTo join "en-"
    textToTranslate get global language
  get result
```

Step 5: Use TextToSpeech

Yandex block



(Blocks Editor)

CYPTraining Screen1 Add Screen ... Remove Screen

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - HelpButton
 - ListPicker1
 - TranslationLabel
 - SpeechRecognizer1
 - TextToSpeech1
 - YandexTranslate1
 - Notifier1
- Any component

Viewer

```
when YandexTranslate1 GotTranslation  
  responseCode translation  
do  
  set TextToSpeech1 Language to get global language  
  set TranslationLabel Text to get translation  
  call TextToSpeech1 Speak  
    message get translation
```

Step 6: Help Button using Notifier

HelpButton block



(Blocks Editor)

CYPTraining Screen1 Add Screen ... Remove Screen Des

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - HelpButton
 - ListPicker1
 - TranslationLabel
 - SpeechRecognizer1
 - TextToSpeech1
 - YandexTranslate1
 - Notifier1
- Any component

Viewer

```
when HelpButton.Click
do
  call Notifier1.ShowDialog
    message join
      " This Translation App translates short sentences in English to Spanish (es), "
      " Italian (it), German (de), Dutch (nl), French (fr) and Russian (ru). "
      " Pick a language, and then speak into the microphone. It will automatically be translated. "
    title " What does this translation app do? "
    buttonText " Ok "
```

Resources:

<http://appinventor.mit.edu/explore/resources.html>

<http://www.appinventor.org/>

