BECOME AN APP INVENTOR

THE OFFICIAL GUIDE FROM MIT APP INVENTOR

Your Guide to Designing, Building, and Sharing Apps

MiTeenPress

MIT APP INVENTOR

KAREN LANG AND SELIM TEZEL

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY
BECOME AN
APP INVENTOR

THE OFFICIAL GUIDE FROM MIT APP INVENTOR

Your Guide to Designing, Building, and Sharing Apps
BECOME AN
APP INVENTOR

THE OFFICIAL GUIDE FROM MIT APP INVENTOR

Your Guide to Designing, Building, and Sharing Apps

KAREN LANG AND SELIM TEZEL
MIT APP INVENTOR PROJECT

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

MiTeenPress
CONTENTS

Foreword 6
Introduction 8

Chapter 1: Let’s Get Started 12
  Two Sisters 30

Chapter 2: Hello, It’s Me! 34
  Apa Pura 50

Chapter 3: Translation App 54
  Vanessa’s Story 72

Chapter 4: My Piano 74
  Potholes of Malden 102
Developing solutions is fun, especially when the solution involves a mobile application. However, developing mobile applications used to require special programming skills and time. All of that changed for me when I first started using MIT App Inventor. It allowed me to focus on solving a problem rather than spending time on complex code. I wish I’d had this book when I first started playing with App Inventor. With easy instructions that take readers from downloading the MIT AI2 Companion application to creating their own chat app, this book is a beneficial friend for all the young changemakers who have ideas but don’t know where to start or don’t have a teacher or expert to guide them. Recently, I was contacted by a girl who had an idea for creating an app for dyslexic students like her but did not know how to start or code. I recommended that she use MIT App Inventor. This book is an ideal companion for anybody like her who is passionate about developing mobile apps.

The book starts with simple apps, and each chapter is organized to build on the one before and help readers create progressively more interesting apps. My brother struggled to make a weather app as part of a school
assignment. After spending just a few minutes exploring App Inventor and reading through the first chapter, he soon had a prototype ready to go. The book not only allows anybody to build apps relatively quickly, but it also enables teachers and educators to introduce their students to easy mobile app development without having to develop advanced skills themselves. Given that App Inventor allows more time to focus on problem-solving, teachers and educators can now include this in their curriculums for a problem-based learning approach.

Each chapter also features heartwarming true stories of young innovators and changemakers who used App Inventor to solve problems. It was great to read about their journeys from vision to implementation, all using the varied features App Inventor provides.

As a prolific user of MIT App Inventor, I strongly recommend this book for anybody looking to develop mobile apps using MIT App Inventor. I want to thank the authors, Ms. Lang and Mr. Tezel, for providing a step-by-step guide for all young students. This book should be in every library, school, and STEM and coding club, and I hope it will inspire a new generation of inventors and innovators.

Gitanjali Rao
Time magazine's 2020 Kid of the Year
Have you ever played a game on a tablet or watched a video on your phone? If so, you probably used an app to do it. **Mobile applications**, or **apps** for short, are programs designed for use on a device like a phone or tablet. Apps can help you check the weather, look up information online, and share photos with friends.

But where do apps come from? They’re built by people called **programmers**. Programmers create a set of instructions that tells a device what to do. They write the instructions in a special language, called a **programming language**, so a computer can understand it. The instructions are also known as **code**, and another name for programmers is **coders**.

---

**Introduction**

A **computer** is a machine that can follow a set of instructions. Computers come in all shapes and sizes. Desktop computers, laptops, tablets, smartphones, smartwatches, security systems, and many car engines and refrigerators are examples of computers. The instructions they follow are called **programs**, and programs that run on mobile devices like phones and tablets are often called apps.
MIT App Inventor is a kind of programming language that uses blocks, which look like puzzle pieces with words on them. The blocks fit together to make instructions that get translated into computer language so a mobile device can follow them.
If you’re looking for some App Inventor inspiration, meet Gitanjali Rao, a teen from Lone Tree, Colorado, who was named *Time* magazine’s first-ever Kid of the Year in 2020. She’s an amazing scientist and innovator and an accomplished app inventor. In 2017, she demonstrated her invention Tethys, which measures lead levels in water. She was inspired to create Tethys by the Flint, Michigan, water crisis, in which the city’s drinking water was highly contaminated by lead and was poisoning children. Her device uses carbon nanotubes to detect lead levels and report that information through Bluetooth to an app she created with MIT App Inventor. Since then, Gitanjali has continued to learn and invent. Her latest project is Kindly, an app that helps prevent cyberbullying. The app detects words associated with bullying, notifies the user, and gives them the opportunity to edit their language.

Gitanjali believes that innovation happens when you’re inspired and motivated to create something meaningful. Hoping to encourage others, Gitanjali offers innovation workshops to share her knowledge and scientific method with other young people around the globe. She has already mentored thirty thousand kids through her workshops.

What about you? Do you have something that you are passionate about? Does that passion lead you to ideas for new apps? Or maybe you have thoughts about ways to improve an app that already exists. Awesome! This book will teach you how to use MIT App Inventor to take your ideas and turn them into apps for your phone or tablet. Soon you’ll be able to share your apps with friends and family and even publish them for people around the world to download and use. Soon you’ll be an app inventor too.
Let’s Get Started

There are a few steps you’ll need to take before you can dig in and start writing apps. First, you’ll need a computer or laptop on which to design and write the code for your apps. Then you’ll want to try out your apps with a mobile device. To do that, you’ll need to install an app called the MIT AI2 Companion on your phone or tablet so you can load your projects onto your device and test them.

If you’re using an Android device, open the Play Store app and search for MIT AI2 Companion. You can also type this link into a browser (like Google Chrome): http://appinv.us/ai-Android. If you are using an iOS device, open the App Store app and search for MIT AI2 companion. You can also use the link http://appinv.us/ai-iOS. Click the button to install the MIT AI2 Companion on your device. The icon for the app will show the mascot, Codi the Bee.

You’ll need to install the MIT AI2 Companion only once and then leave it on your phone or tablet for whenever you use App Inventor.
If you don’t have access to a mobile device, or don’t have a good Wi-Fi connection, there are other ways to test your apps. If you don’t have a mobile device, you can use the emulator that App Inventor provides, which looks like a phone but runs in your computer’s browser. You can also create and test App Inventor projects with a Chromebook. If Wi-Fi is an issue, you can connect a mobile device to your computer with a USB cable. To get instructions for these other options, open the AI2 Getting Started page (http://bit.ly/ai2-getstarted) in a web browser and click on the instructions for the option you’d like to use.

Once you have set up your mobile device, open a browser like Google Chrome or Mozilla Firefox on your computer.

// NOTE

Android users: If you choose not to go through the Play Store and instead load the app directly by using the website link (aka “side load”), you will need to change your device’s settings to allow it to install apps from “unknown sources.” To find this setting on versions of Android before 4.0, go to Settings > Applications and check the box next to Unknown Sources. For devices running Android 4.0 or above, go to Settings > Security or Settings > Security & Screen Lock and check the box next to Unknown Sources and confirm your choice.
Logging in with a Gmail Account

If you have a Gmail account, type this address into the URL field:

Now all the projects you create will be saved under your account.
Logging in without a Gmail Account

If you don’t have a Gmail account and can’t make one, use this address: http://code.appinventor.mit.edu. Just press the button (Continue Without An Account), and you can log on to the server.

Once you press the button, you’ll be given a revisit code, which will help you save your projects.

Make sure to copy the revisit code and keep it somewhere safe, because without it, you won’t be able to get your projects back! The next time you log in to http://code.appinventor.mit.edu, you will type in your code rather than clicking (Continue Without An Account).
Once You’re Logged In

When you log in, you’ll see a screen with the latest information about MIT App Inventor and a link to setup instructions. Press the **Continue** button.

On the next display, which has links to some starter tutorials, press the button on the bottom left labeled **Start a Blank Project**.
You’ll be asked to name your new project. For your first app, you can name it **HelloItsMe**.

You can’t have spaces in your project names. Removing the spaces and capitalizing the first letter of each word is called “camel case” because each capital letter looks like a hump on a camel! 🌐

Click **OK**. You will see your blank project in the App Inventor IDE. **IDE** stands for interactive development environment, and it’s what you’ll use to make your apps.

What does the **IDE** name mean? You are developing apps in an *interactive* way because you’ll work with buttons and menus. And it’s a special *environment*—in this case a workspace for building things.
Chapter 1—Let’s Get Started

The Designer

Palette
choose and add components

Viewer
arrange components
This IDE window is called the Designer. It’s where you’ll decide how your apps look and how users will interact with them. You’ll do this by adding and arranging components. **Components** include visible things like buttons and text boxes, as well as invisible things like sounds and sensors.

There are four panels in the Designer: **Palette, Viewer, Components**, and **Properties**.
The Palette Panel

On the left in the Designer is the Palette. The Palette lists all the components you can add to your app, organized in drawers according to their type. If you click on a drawer, it will expand so you can see the components inside.
The Viewer Panel

To select a component, click on it, drag it, and drop it onto the phone screen image that appears in the second panel, the Viewer.
Chapter 1—Let’s Get Started

The Components Panel
As you drag components to the Viewer to add them to your app, they also appear in a list in the Components panel.

The Properties Panel
You can select a component by clicking on it in the Viewer or in the Components panel. Once a component is selected, you can change its settings in the Properties panel.
For example, if you select **Button1**, you can make changes to things like its background color, font size, shape, text, and text color!
The Blocks Editor

Once you’ve added your components in the Designer, you can switch to the Blocks Editor by clicking on the (Blocks) button in the upper-right corner.

In the Blocks Editor, instead of dragging components the way you do in the Designer, you’ll drag out various code blocks that fit together like puzzle pieces. These blocks are the instructions that tell your app what to do. The main difference between the Designer and the Blocks Editor is that the Designer controls the look and feel of your app, whereas the Blocks Editor controls the behavior of your app. You’ll need them both to build a successful app.
The Blocks Palette

On the left in the Blocks Editor is the Blocks palette. This is where you can find all the blocks, divided into two sections—Built-in blocks and Component blocks.

Built-in Blocks

Built-in blocks are organized by category. Clicking on the category makes its blocks appear. For example, clicking on Math shows all math-related blocks, like the + block, which lets you add numbers, and the / block, which lets you divide them.
Component Blocks

Component blocks match up with the components that you’ve added. As soon as you add a component in the Designer, it will automatically appear here too, with its own set of blocks that control the component’s behaviors, properties, and actions. Click on a component to make its available blocks appear. For example, clicking on Button1 shows all the blocks you can use that control the behavior of that button, or what should happen when a user clicks or touches the button.
If you click on **Label1**, you’ll see that it doesn’t have any gold event blocks, because you can’t click or touch a label to make it do something the way you can with a button. But you’ll see that labels do have green blocks, which set their properties.
For both Built-in blocks and Component blocks, click and drag them into the Viewer panel to make them part of your app.
By adding blocks and snapping them together, you’ll create a set of instructions. And that’s how you’ll build apps.

Now that you know the basics, are you ready to get started?
**Two Sisters**

**Bethany:** Hi there, coders! We are sisters. I’m Bethany, and I’m fifteen.

**Ice:** I’m Ice, and I’m twelve. When I was small, I loved exploring the machines and devices at home. Sometimes I break stuff when I’m exploring.

**Bethany:** Oh, so you were the one who broke the radio by putting a coin into it.

**Ice:** That’s not the point. I fixed some of the other machines before you or our parents even noticed. Anyway, then we got this really cool machine called the computer! I was overjoyed and immediately I loved it. Whenever I found some new apps or websites, I explored ways to use them without the user manual. When I was about nine, I was introduced to coding. We made a robot car move around by tapping a button! I thought it was seriously cool—and then I joined the CoolThink@JC competition and was introduced to App Inventor 2 (AI2) by my IT teacher in April 2019.

**Bethany:** At that time, Ice got to learn how to code on AI2, and when she got home, she proudly showed us her apps. I got interested in AI2
as well. I asked her to teach me the basics that summer. In a few days, I had become a level 1 beginner AI2 coder.

**Ice**: That summer, we created an app together with five games to play on our vacation. It was awesome! Although our games were very basic and simple, we felt extremely triumphant because we created them ourselves and it was fun to work together.

**Bethany**: After that, whenever we were free, we also experimented with some other functions of AI2 and carried on with creating new games.

**Ice**: Life was all right, but then the COVID-19 pandemic spread across the globe. We couldn’t go to school or go out, so we gradually became bored and unmotivated. “What can we do during this pandemic?” we pondered. One afternoon, I scrolled through the App Inventor website—they had announced a monthly Coronavirus App Challenge! We were like, whoa!

**Bethany**: That same night, we brainstormed some concepts for the challenge, and suddenly a great idea popped into our minds. Why not create an app with games to defeat virtual viruses? That would be fun to make and also raise users’ awareness of hygiene! We squealed in excitement and gave each other high fives.

**Ice**: The next few weeks we started to make our app. It has a checklist of habits to keep clean and healthy, which gets you points; a small game to get more points; and a “Virus Battle”
page to kill virtual viruses with some spray you can buy with your points. It also has pages for seeing all the viruses you defeated and for facts about the pandemic. Plus there’s a detailed user manual.

**Bethany:** We planned to add a lot of colorful illustrations and interesting cartoons in the app, but we were worried that we couldn’t finish all of them in time! We worked out a huge plan where we would draw the illustrations with our phones and code a little after we had finished our online homework.

**Ice:** After drawing all of the pictures, we had already coded nearly a quarter of the app. We worked together in front of our computer, dragging components and blocks across the screen while calculating the required values and testing out the app using the emulator. Finally, after two weeks, our app was completed!

**Bethany:** Our success didn’t come to us overnight. In fact, we faced a lot of problems while making the app. For instance, we were frustrated when we spent almost three hours trying to figure out some calculations for the game! At that point, we stopped working on that part for a while and had a little rest. Then we went back and eventually thought of a solution!

**Ice:** We submitted our app to the MIT App Inventor website and we became the MIT App Inventor Coronavirus App Challenge Young Inventors of the Month! We were absolutely ecstatic!

**Bethany:** Later we did a brief reflection of our app-making process. We concluded that we learned more AI2 techniques and functions, and to never to give up because there is always a way to solve our problems.

**Ice:** More important, we understood how to work as a team. We got past
our differences and respected the other person’s opinions because we were eager to finish our app together. Our sisterly bond has been enhanced greatly! Our minds are practically in sync these days!

**Bethany:** To sum up, we are incredibly grateful to have this chance to express our ideas through coding.

**Ice:** We would like to thank our parents, who gave us encouragement and support, as well as my IT teacher for introducing me to App Inventor, coding, and STEM.

**Bethany:** And we sincerely thank MIT for giving us an App of the Month award. It means a lot to us and it has motivated us to continue coding on App Inventor 2!

**Ice:** If any frustrated coder is flipping through these pages and trying to solve a bug in your app, here is some advice: relax! Don’t let it make you angry or tired. Instead, take a rest, give your brain time to think about the problem, go through your codes calmly, and maybe have a snack or two . . .

**Bethany:** Then ta-da! Inspiration strikes! You snap your fingers and say, “I know what’s wrong now!” and find a solution to your bug. Debugging complete!

**Ice:** Happy coding!
Hello, It’s Me!

Do you have relatives and friends who live far away? They might enjoy seeing your face and hearing your voice when you can’t be with them. In this app, HelloItsMe, your picture will appear, and when the user clicks on it, they’ll hear you greeting them!

Prepare Your Assets

**Assets** are extra files, like images and sounds, that are part of an app. For this app, you will be adding two assets: an image (of yourself) and a sound (your voice). Start by finding a good digital picture of your face and saving it somewhere on your computer where you can find it later, like your desktop. If you don’t have one already, you can take a picture with your computer. If you’re on a Mac computer, use the Photo Booth app. Take a picture, then under the File menu, choose Export and save it to your desktop. When you save the file, name it so you can easily find it again. If you’re on a Windows computer, use the Camera app. When you take a picture, it’s automatically saved in the Camera Roll folder inside the Pictures folder on your computer. Don’t forget to smile for the camera!
The second thing you need is a recording of your voice. Think of a good greeting or message for your friends and family. It can be something simple like “Hi, it’s me! Wanted to say hello and wish you a happy day!” or you can record something a bit longer: “Hello, my name is Codi. I am a big fan of biking, skating, and programming apps. This is my first app. I hope you like it!” Feel free to get creative! If you use a Mac, the QuickTime application is an easy way to record your voice. Just go to the File menu and choose Audio Recording. If you use a Windows computer, you can record your voice using the Sound Recorder app. In both cases, make sure to save your recording on your computer where you can find it later.

If you don’t have MIT App Inventor already open from chapter 1, use a browser on your computer to open it up by going to either http://ai2.appinventor.mit.edu (with your Gmail account) or http://code.appinventor.mit.edu (with your return code).
Open Your Project

1) If your HelloItsMe project isn’t already open, go to the Projects menu and select My projects from the drop-down options.

2) Click on HelloItsMe in the project list to open it.

   Your project should open in the Designer window, but if it doesn’t, just click the Designer button in the upper-right corner.
3) If you have any components left over from following along in chapter 1, you’ll need to delete them by clicking on each component in the Components panel and then clicking the **Delete** button. You’ll be asked to confirm that you really want to delete them.
Add the Components

4) Add a **Button** and a **Label** component to your app by dragging them from the User Interface drawer in the Palette to the Viewer.
5) Now click on the Media tab in the Palette and drag a **Player** component onto the Viewer.

// NOTE

See how the **Player** component drops below the phone image and appears under Non-visible components? That’s because the **Player** doesn’t show up on your app screen. But it’s still an important component in the app because it plays sounds.
Add Image and Voice Files to Your App

Now it’s time to upload your picture and voice recording to App Inventor.

6) Find the Media panel, which is below the Components panel. Click on the **Upload File ...** button. Then click the **Choose File** button, locate the image of yourself on your computer, and click **OK** to upload it. Then repeat those steps for your voice recording. Both of your files will then appear in the Media panel as assets.
Change the Component Properties

Let’s change some of the properties of the components to make the app look better.

7) Click on Button1 in the Components panel and change the Button1 properties in the Properties panel like this:
   Height: 300 pixels
   Width: Fill parent

Remember, properties are different characteristics of a component that you can change to alter how the component looks. Properties include things like width, height, color, and visibility.

You can tell which component is the parent by looking at the Components panel. See how Button1, Label1, and Player1 are all indented under Screen1? That means that Screen1 is the parent, and Button1, Label1, and Player1 are Screen1’s children. You can think of a container as a parent and any components inside the container as children.

Fill parent means to make the property, in this case the width of Button1, the same as its parent. Button1’s parent is Screen1, because Button1 is contained within Screen1. So you are setting the width of Button1 to the same width as Screen1.
8) The next step is to find **Image** in the Properties panel and click None…. Choose the image file that you uploaded earlier, then click **OK** to use your image as the background for **Button1**. Once you choose the image file, the image will appear on the button in the Viewer.

9) Choose **Player1** from the Components list and set its **Source** property to the voice recording file that you uploaded earlier. Then click **OK** to use your sound file as the source when it plays. This will attach the sound file to the Player, so when the Player is started in the app, it knows which sound to play.
10) Click on **Label1** in the Components list and change the Label properties:

**BackgroundColor**: *Choose a color*

**FontSize**: 30

**Text**: *Press Me*

**TextColor**: *Choose a color*

---

// NOTE

For background and text color, choose whatever colors you like!

---

You’ve completed the Designer portion of your app! Now that you’ve added your components and set their properties so the app looks good, it’s time to make the app work.
Code the Blocks

Your app has all its components, but it doesn’t do anything yet. Now is your chance to code the app so it works when the user runs it.

11) Click on the Blocks button in the upper-right corner of the screen to switch to the Blocks Editor.

In the app, you want your voice recording to play when the user presses on your picture, which is displayed on Button1. Clicking or pressing a button is called an event. When an event happens, it should trigger an action. In this case, the event of clicking on your picture should make your voice recording play. In the Button1 blocks, the gold ones with an empty area in the middle are called event handlers. Event handlers do just that—they handle events. When an event happens in an app, the code blocks inside the event handler tell the app what actions should happen.

For example, in the following code blocks, when the user clicks on Button1, the text for Label1 is set to Hello World! You can add as many blocks as you want inside the when Button1.Click event block. All the blocks inside will run whenever the user clicks on the button.