Tour Guide

Make an app to showcase local landmarks in your community with a map

Essential Questions

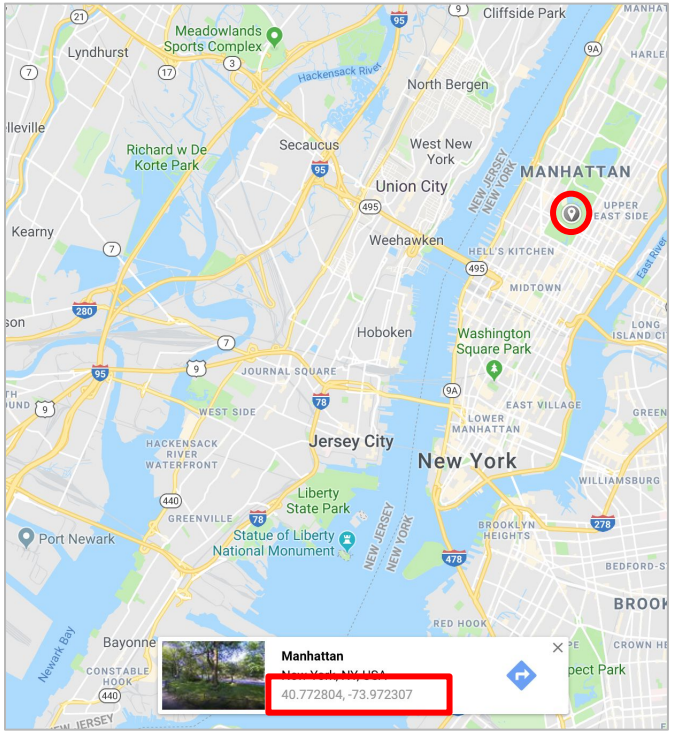
- How does Google Maps work?
- How can users interact with a map app?

Objectives

1. Design and code a location-based app, using the Maps component in App Inventor.
2. Create an app that uses multiple screens and passes values.
3. Demonstrate understanding of lists and indexes, utilizing multiple lists to manage data.
4. Use the TinyDB component to store data persistently on a device.

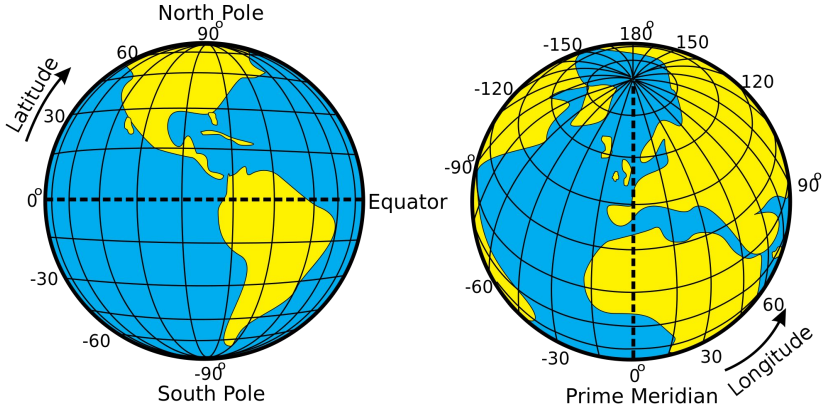
Lesson 1: Latitude and Longitude

- Clicking on a location in Google Maps, you get a location, and the latitude and longitude for that location.
- All locations around the globe can be labeled by their latitude and longitude.



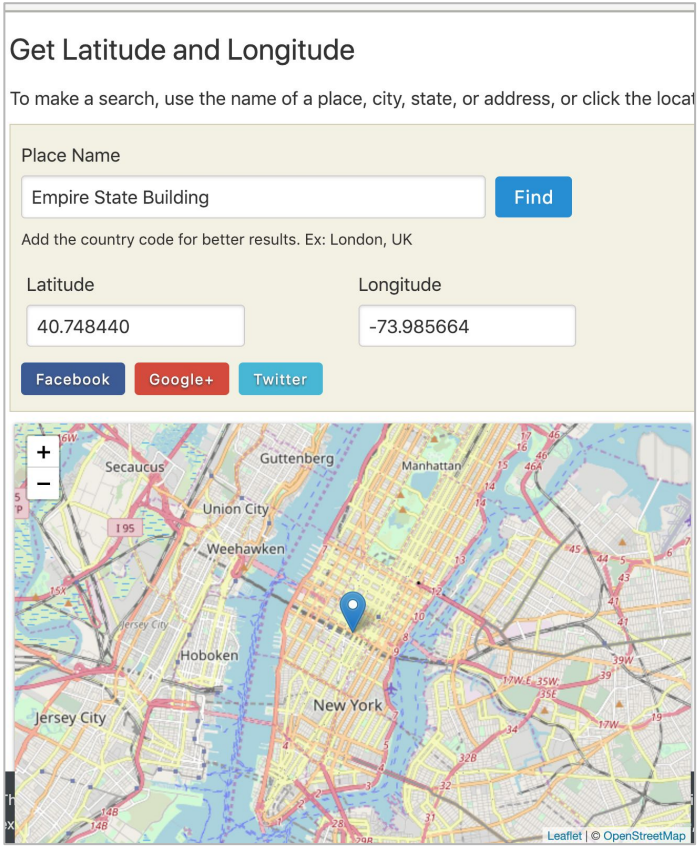
Latitude and Longitude

- **Latitude** is measured with 0º at the equator, the North Pole 90º, and the South Pole -90º.
- **Longitude** is measured with 0º at Prime Meridian, which is located in Greenwich, England. Clockwise around the globe is 0º to positive 180º, and counter-clockwise around the globe is 0º to negative 180º.



Latitude and Longitude

- www.latlong.net is a great website to find the latitude/longitude for a location.
- You can enter the name of a place, or the address to get the results.



Lesson 1: Landmarks Worksheet

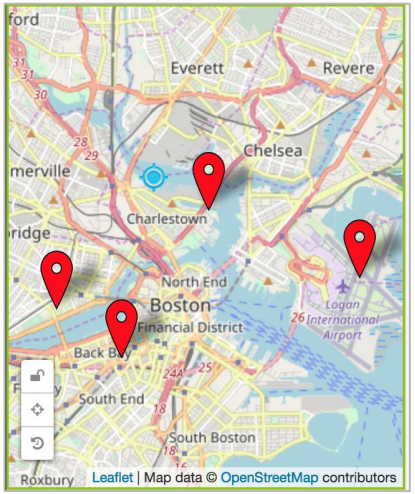
With your partner, decide on four landmarks you want to showcase in your app.

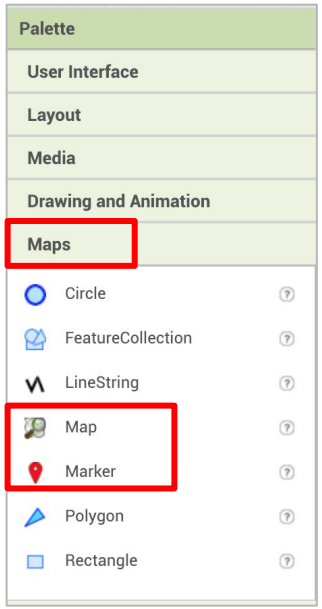
Fill out the ***Landmarks Worksheet*** with the following information on each landmark

- Location name
- Short description
- Longer paragraph (2-3 sentences) of information about the site
- Picture (digital picture) - this should be accessible online or on your computer; list the exact filename on the worksheet
- Latitude and longitude for each landmark, using www.latlong.net

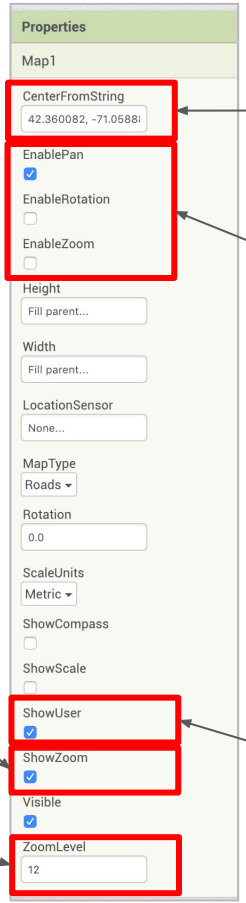
Lesson 2

- Map Components
  - Map and Markers





Map Properties



Lat/long of center of map. Can drag map to center.

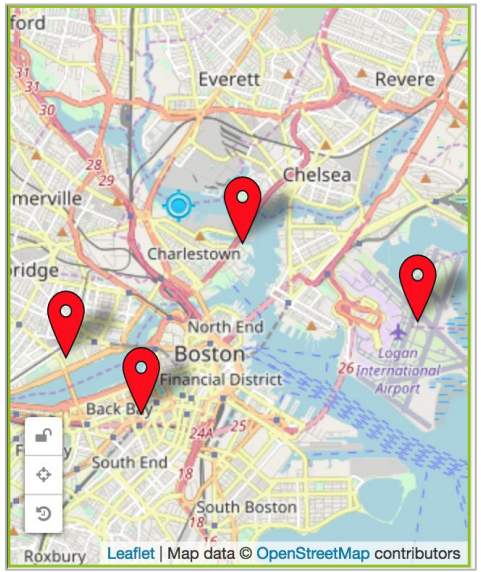
Enable/disable user interactions with map.

Show the zoom +/- icons on map.

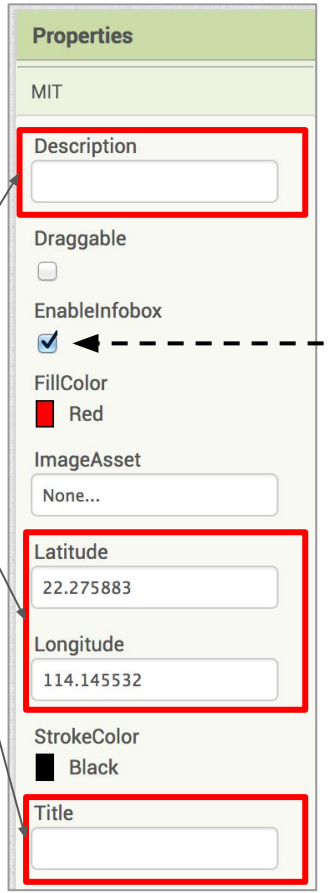
ZoomLevel 1-19, increasing number zooms in.

Show the user's location on map.

Lesson 2: Markers



Marker Properties

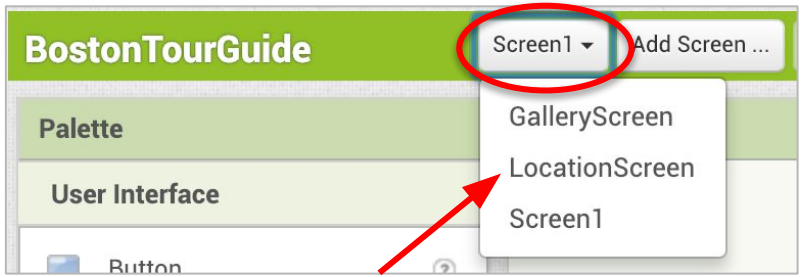


Fill out these properties based on your **Landmarks Worksheet.**

Make sure *EnableInfoBox* is checked! When user clicks on a Marker, the InfoBox appears.

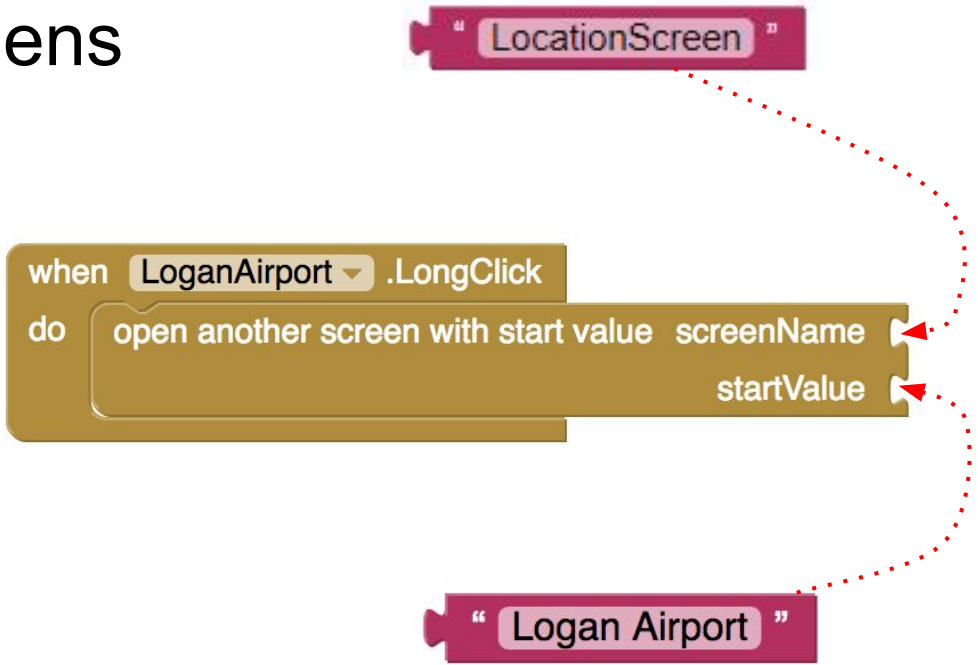
Lesson 2: Multiple Screens

- This app has 3 screens.
- You can choose which screen you are working on with the dropdown menu.
- Each screen has its own Designer and Blocks.
- **open another screen** Block is in the Control Drawer.
- You'll use **open another screen with start value**.



Lesson 2: Multiple Screens

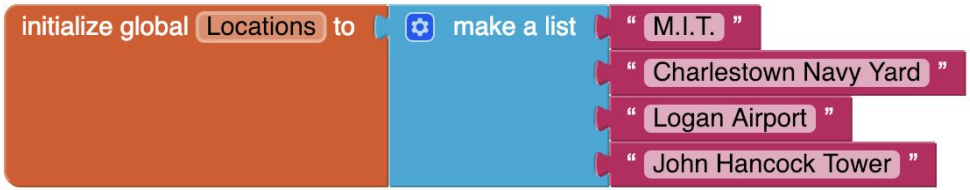
- You specify the ***screenName*** (exactly as you named it)
- ***startValue*** is a value you want to send to the screen that is being opened.
- For this app, you will send the value of the location the user clicks on.



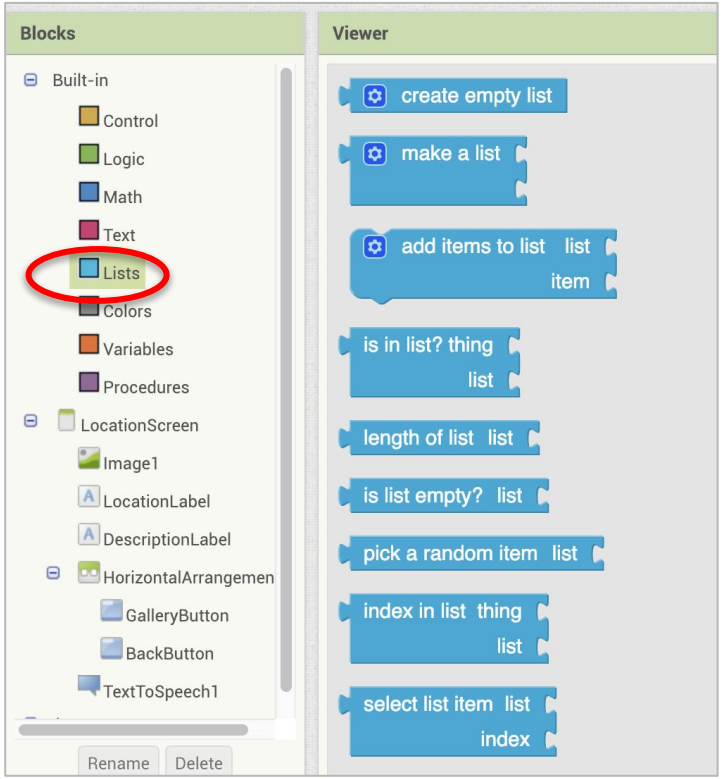
Lesson 2:
Complete Student Guide Part 1:

Lesson 3: Lists

- **Lists** allow you to store multiple values using one name.
- Simpler than having many many variables that all store similar information.
- You access the values by the index.
- For example, M.I.T. is index 1 in the Locations list.
- Logan Airport is index 3.



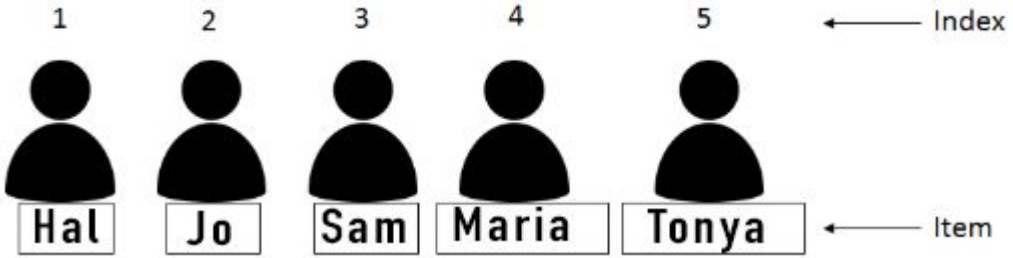
Lesson 3: Lists



Lesson 3: Lists

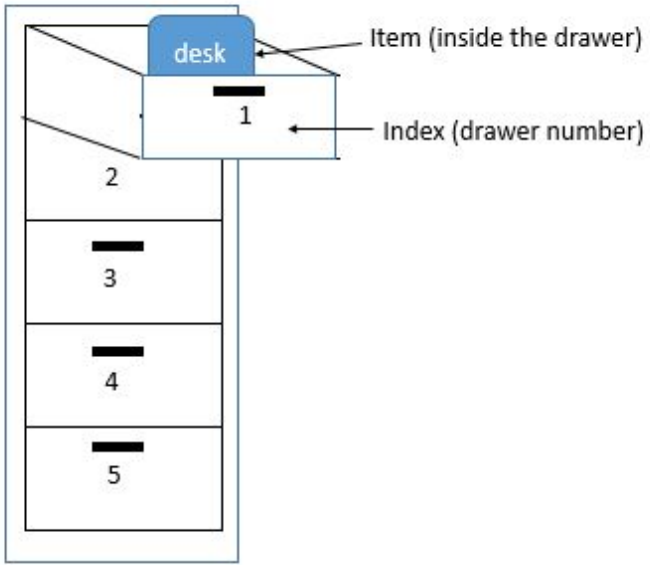
Unplugged Activity:

Selected students stand at front of classroom. Students are called by their index.



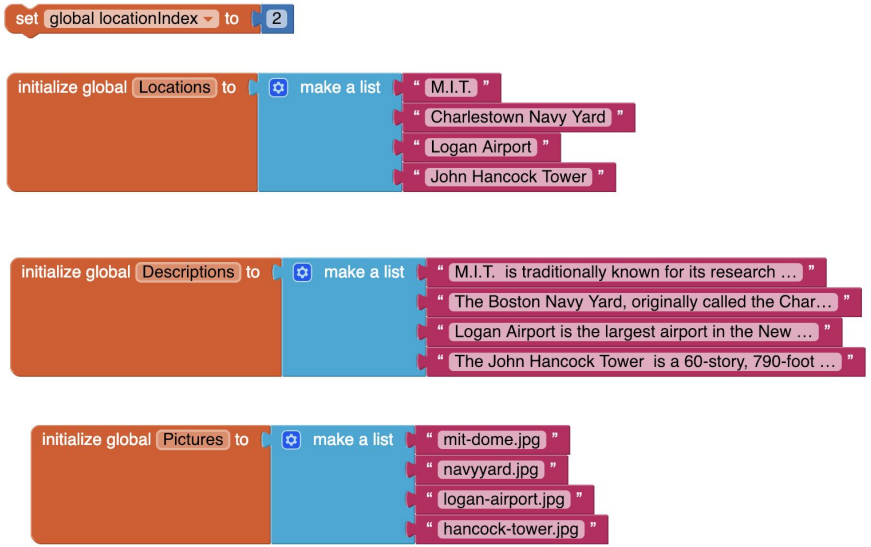
Lesson 3: Lists

- A list is analogous to a file cabinet.
- Each drawer is numbered (index).
- The information inside the drawer is the value (item) at that index.



Lesson 3: Parallel Lists

- In this app, you will set the *locationIndex* based on the *start value*.
- You will use *locationIndex* to extract the correct **Location**, **Description**, and **Picture** from the corresponding list.
- For example, if *locationIndex* = 2,
  - Location is “Logan Airport”.
  - Description is “Logan Airport is the largest airport in the New…”
  - Picture is “logan-airport.jpg”



Lesson 3:
Complete Student Guide Part 2:

Lesson 4: Camera

- Take a picture with the device's camera.

- After the picture is taken, AfterPicture event is triggered. In this app, you will display the *image* on your screen.

Lesson 4: TinyDB

- **TinyDB** lets you store data on your device.
- The data is *persistent* because it doesn't go away when the app closes.
- Variables do go away when the app closes.
- You will store the image taken with the camera in TinyDB so the app knows that is the image to display next time it opens.

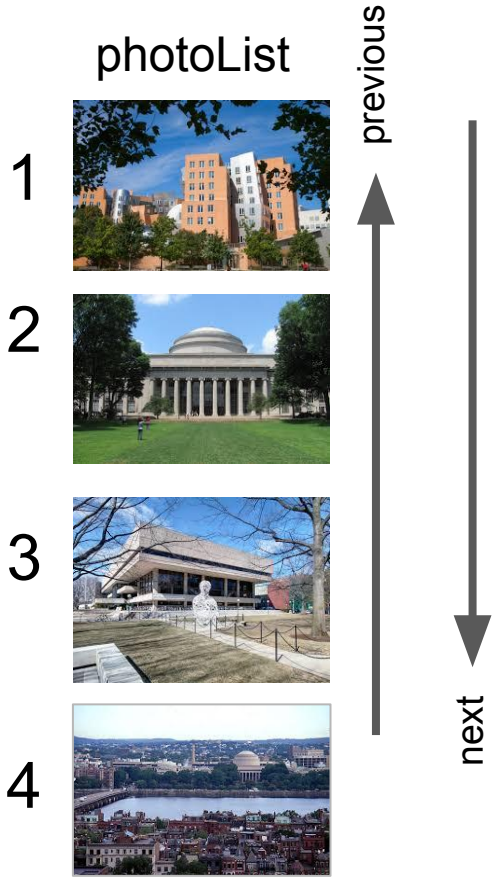
	Variable	**TinyDB**
Persistence	Anything stored in a variable is erased when the app closes	Anything stored in TinyDB with a tag can be retrieved at any point, even after the app is closed and reopened
Storing data	**set name** to value	**TinyDB.StoreValue** (tag, value)
Retrieving stored data	**get name**	**TinyDB.GetValue**(tag, valueIfTagNotThere)

Lesson 4:
Complete Student Guide Part 3:

Lesson 5:
Scrolling through a List

- Previous and Next buttons
- Previous - subtract 1 from index
  - Stop at 1 (lowest index is 1)
- Next - add 1 to index
  - Stop at … 

  - Length of list is biggest index



Lesson 5:
Complete Student Guide Part 4:

Vocabulary Words

Map

Latitude

Longitude

Marker

Multiple screens

Start value

List

Index

Parallel Lists

TinyDB

Scrolling through a list