

Thunkable Cheatsheet

What is Thunkable?

Thunkable is a mobile app-building platform that allows users to develop an app without coding using a visual interface to design and build an app.

Key terms and sections

Component: visual elements (eg. buttons, labels, text fields, images) or non-visible elements (eg. timers, sensors) that can be added to the app's screens. Each component has its own set of properties and events customized to fit its looks and functionality.

Design section: where the user visually creates the user interface of the app by dragging and dropping components onto the screen

Component Tree (Screens): provides a hierarchical view of all components in app, used to manage and navigate structure of app

Components: lists all UI elements and features you can add to your app

Screen Properties: adjust background color, name, and visibility of screen

Blocks section: code blocks

Block sections

UI components: used to visually design the user interface by adding various elements by using the blocks to define behaviour and interacts within the app (eg. define what happens when the screen is opened, set the screen's background color or picture)

Core:

- **Control:** manage the flow of the app's logic by managing order of code execution using loops and conditions (if statements)
- **Logic:** used to create conditional statements using logical operators, often paired with control blocks (<, >, =, and, or)
- **Math:** used to perform calculations and work with numerical data with mathematical functions and arithmetic operators (+, -, round, absolute)
- **Text:** used to handle and manipulate text information and work with strings of words or sentences by joining text, extracting portion of text, and finding text lengths (eg. typing in a chat, filling out forms)
- **Lists:** used to work with lists and arrays of elements by adding or removing items from a list, sorting items, and getting a specific item based on it's position
- **Color:** used to specify and change colors used within your app's interface (design screen)
- **Device:** used to access device-specific capabilities like sensors, time and date, camera, functionalities, or location services
- **Objects:** used to create objects and add properties to it

- **Variables:** make your own variable (eg. number, text string, boolean TRUE or FALSE value)
- **Functions:** reusable block of code that performs a specific task or set of tasks, used to organize code

App features: used when working data from app features like camera, location, files, sound, sign in, etc.

Variable (in Thinkable)

A variable is a storage location that holds information like numbers or text. Each variable has a name, type, value, and scope.

- Name: name of variable is unique
- Type (*static* - fixed name but value may change, *dynamic* - name is created while app runs so not fixed and value does not need to be initialized)
 - App: used when storing data within an app session and are reset when the app is closed and re-opened
 - Stored: used when storing data on the device (data is saved between app sessions)
 - Cloud: used when storing data in the cloud in a Firebase database (data can be accessed from multiple devices)

Loops

What are Repeat loops?

- Allows you to repeat a set of actions a certain number of times

What are Forever loops?

- A loop that repeats the action inside it forever
- Useful for making something happen continuously

How to create a loop

- Using Repeat block
 - Go to Control and drag and drop the “repeat” block
 - Set the number of times you want the action to repeat
 - Place code blocks you want repeated inside the “repeat” block
- Using Forever block
 - Go to Control and drag and drop the “forever” block
 - Place the code blocks you want repeating indefinitely forever inside the “forever” block

If statements

What is an If statement?

- Checks if a certain condition is true
 - If condition is met, the code inside the “if” block will run

- Otherwise, the code inside the “if” block does NOT run and the program moves to the next set of code blocks after the “if” statement

What is an If-Else statement?

- Extends the “if” statement by providing an alternative set of code for the program to follow the condition is false
 - If the initial condition is true, the code inside the “if” block will run
 - Otherwise, the code in the “else” block runs

How to create an If statement

- Creating an If statement
 - Go to Control and drag and drop the “if” block
 - Place the condition you want to check inside the hexagonal space in the “if” block
 - Add the code blocks you want to run when the condition is true inside the “if” block
- Creating an If-Else statement
 - Go to Control and drag and drop the “if-else” block
 - Place the condition you want to check inside the hexagonal space in the “if” block
 - Add the code blocks you want to run when the condition is true inside first section of the “if” block
 - Add the code blocks you want to run when the condition is false inside second section of the “if” block (under “else”)

Lists

What is a List?

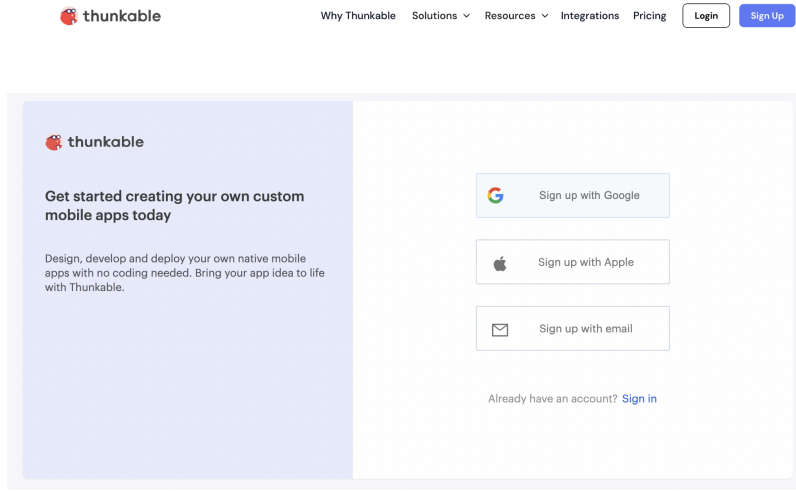
- A list is a data structure that allows you to store and organize multiple values of the same or different types
- Each value is called an element (item) and each element is associated with an index (position in a list)

How to make a List?

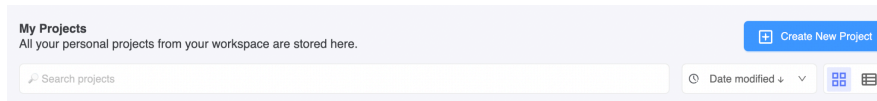
- Using the “list” block
 - Go to Core and then Lists, and drag and drop the “list” block
 - Change the values of the elements in the list
- Create a list of numbers from A to B by X using the “list of numbers from” block
 - Go to Core and then Lists, and drag and drop the “list of numbers from” block
- Create a list from text or make text from a list using the “make list from text/ text from list” block
 - Go to Core and then Lists, and drag and drop the “make list from text/ text from list” block
 - Define the delimiter (eg. ,) that separated the list values

Getting started

1. Create an account or sign in: <https://thinkable.com/>



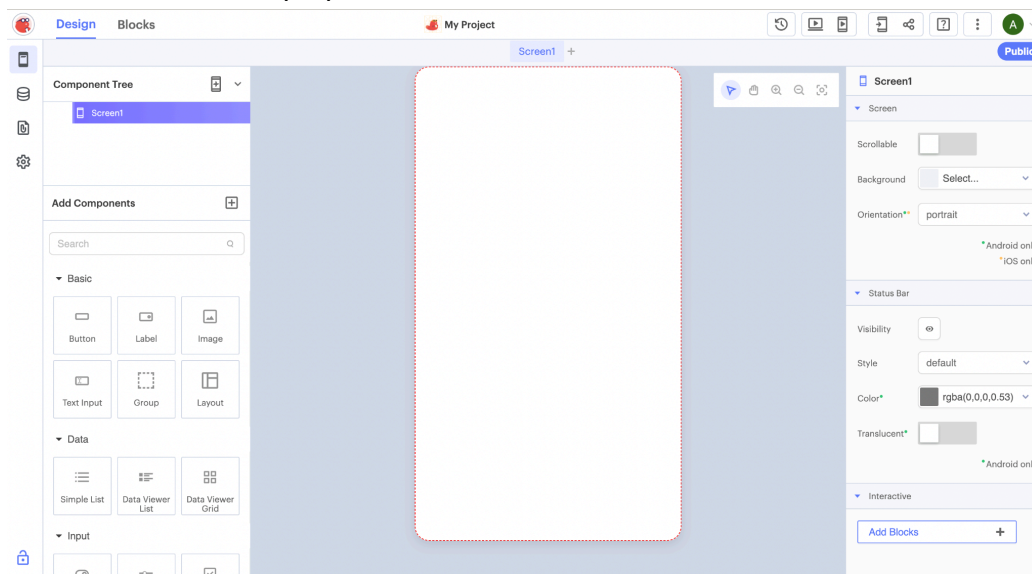
2. Click the “Create New Project” button to create a new project



3. Choose the platform (Android/IOS) and name your project

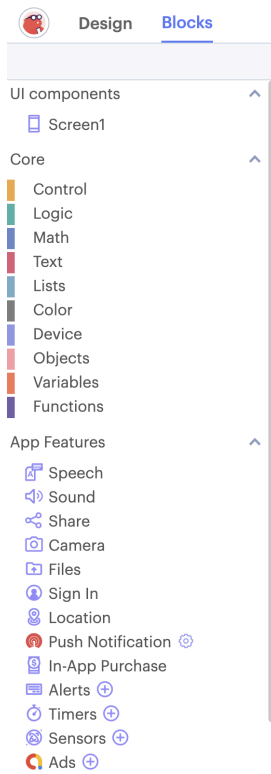
Creating a basic app

1. User interface (Design section)
 - a. Use components to design your app
 - b. Customize its properties like text, color, and size



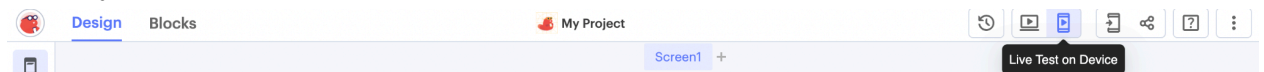
2. Adding functionality (Blocks section)

a. Use blocks to add functionality to components



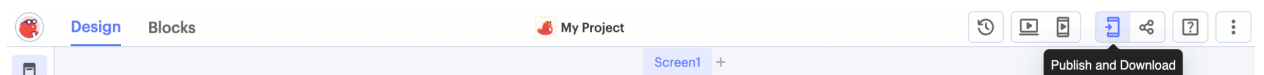
3. Testing

- a. Click the “Live Test on Device” button on the top right to preview your app on your device



4. Publishing

- a. Click the “Publish” button on the top right to publish on your app on the Thinkable app



Additional Resources (YouTube videos and document provided by Thinkable)

- Welcome to Thinkable:
https://www.youtube.com/watch?v=yUE-dtBNqRk&t=31s&ab_channel=Thinkable
- How to build your first Thinkable App:
<https://www.youtube.com/playlist?list=PLB89L9PPGIrwfBqT6T6XHm3rqtAxUny-5>
- Overview of core block categories: <https://docs.thunkable.com/blocks/blocks>
- Connect a firebase realtime database:
<https://docs.thunkable.com/blocks/blocks/variables-overview/connect-a-firebase-realtime-database>

- Create user sign up/ sign in with Firebase:
<https://docs.thunkable.com/blocks/app-features/sign-in/sign-in>