



Mobile App Development Class 01, Series 04

Mobile App

HahuJobs



CLASS 01

- 01 Cross-platform Development
- 02 Dart Programming
- 03 Introduction to Flutter
- 04 Setting up Flutter SDK
- 05 Widgets



Cross-platform Development

What is it?

It's a means to develop apps for multiple platforms (operating systems) or devices with one code base, at the same time, instead of building separate apps for each one. This way, one developer can for example develop mobile apps for Android and iOS. There are different methods of building such an app.



Cross-platform Development

Pros and Opportunities

- When compared to native development , it provides a fast paced development and takes a reduced time-to-market.
- Developments cost is halved since only a one team is needed to work on a project.
- The learning & development easier
- Apps made with a cross platform frameworks are easier to maintain.



Cross-platform Development

Disadvantage and Challenges

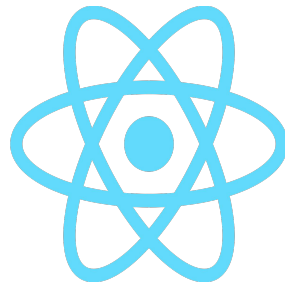
- Native apps have an advantage when it comes to some system level features and performance.
- Highly dependent on libraries
- Opinionated
- Difficulties with for native UI developers, although most frameworks support native components and feel they are a bit behind on bringing the pure native looks.

Cross-platform Development

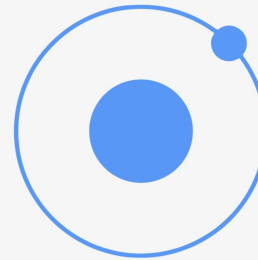
Popular Frameworks



Flutter



React Native



Ionic



Xamarin



Dart Programming

What it is?

- is an high level object oriented programming language developed by google.
- Its is compiled in dart VM.
- is used to write cross platform apps for Android , iOS, Linux and Windows.
- primary language for flutter.



Basics of Dart

The Basics(Practical)

- **Data Types**: the generic data types in dart include **Numbers**, **Strings**, **Lists**, **Booleans** and **Maps**.
- **Operators** : Arithmetic ,Equality and Relational , Type test , Bitwise ,Assignment ,Logical Operators.
- **Control-Flow** : if-else & switch statements, for , do-while and forEach loops.
- **Collections**: Lists, Sets, Maps and HashMaps.
- **Class & Objects**: it supports basic OOP features in addition to that it includes mixins.



Introduction to Flutter

What is it ?

Flutter is a cross-platform UI toolkit that is designed to allow code reuse across operating systems such as iOS and Android, while also allowing applications to interface directly with underlying platform services. The goal is to enable developers to deliver high-performance apps that feel natural on different platforms, embracing differences where they exist while sharing as much code as possible



Introduction to Flutter

Architecture

Flutter's core architecture is founded on three layers to that made it achieve its high performance and reactive status.

- **Framework(Dart)** - this layer handles the basic UI interactions, widgets ,animation and surface level rendering.
- **Engine(C++/C)** - It provides the low-level implementation of Flutter's core API, including graphics (through Skia), text layout, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain.
- **Embedder** - this layer is responsible for handling platform level implementations like plugins, Render surface setup, threading and how the app is packaged.

Introduction to Flutter

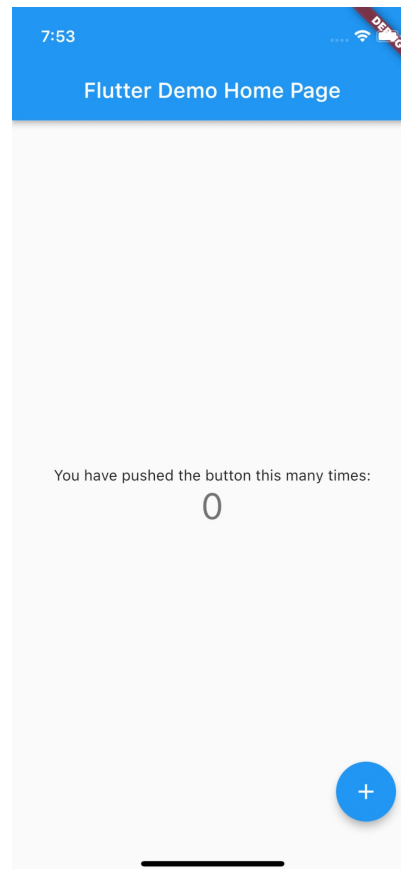
Installation - Follow Instructor





Introduction to Flutter

Flutter Initial Run



Introduction to Widgets

What is a widget?

Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the app. And the structure of the code of an app is a tree of widgets.



a simple button bar in flutter

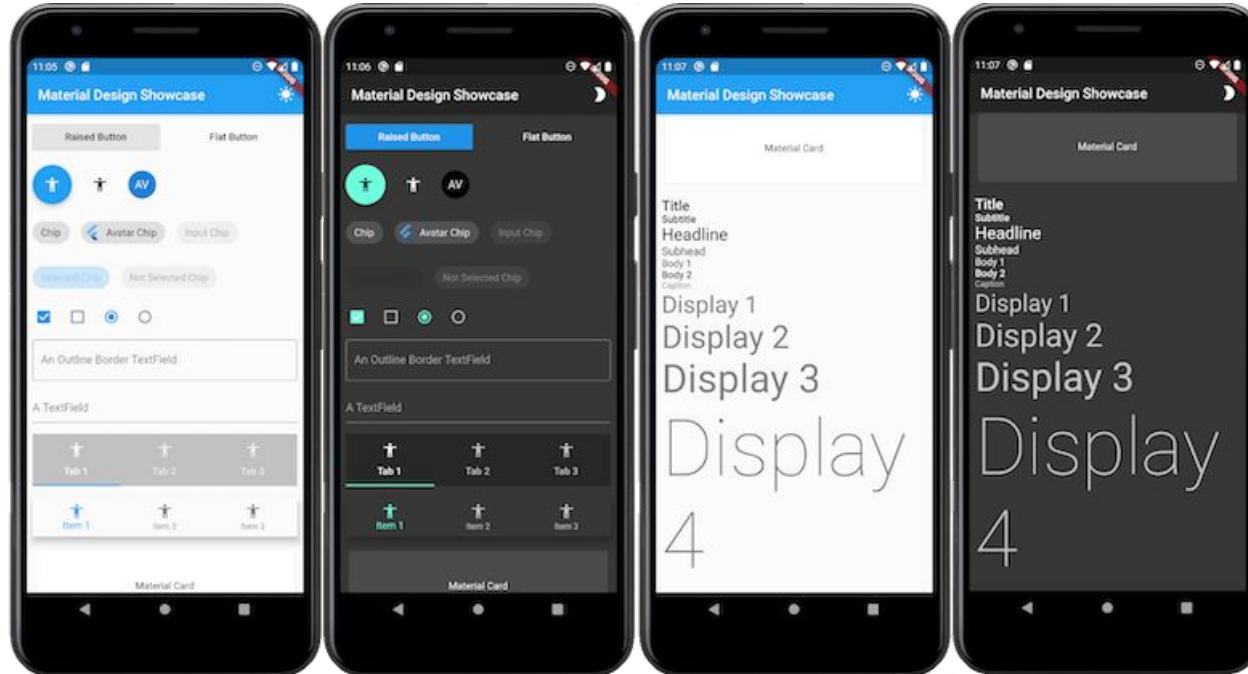


everything is a widget

Introduction to Widgets

Types of Widget - Practical

Platform Specific Widgets: are widgets derived from native looks of ios or android.

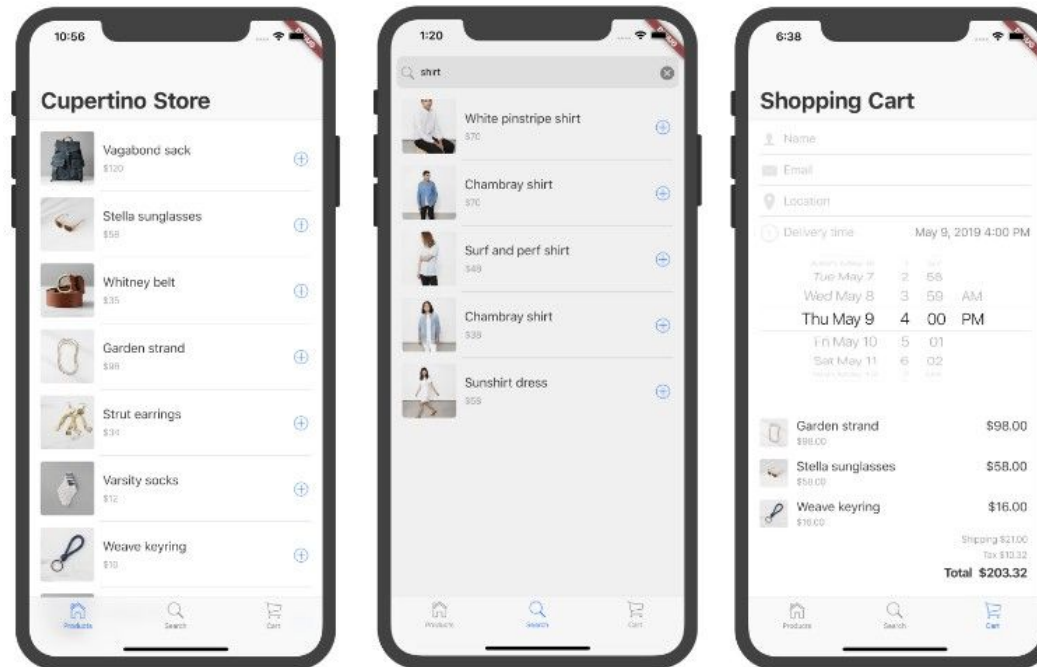


Android material widgets in flutter

Introduction to Widgets

Types of Widget

Platform Specific Widgets: are widgets derived from native looks of ios or android.

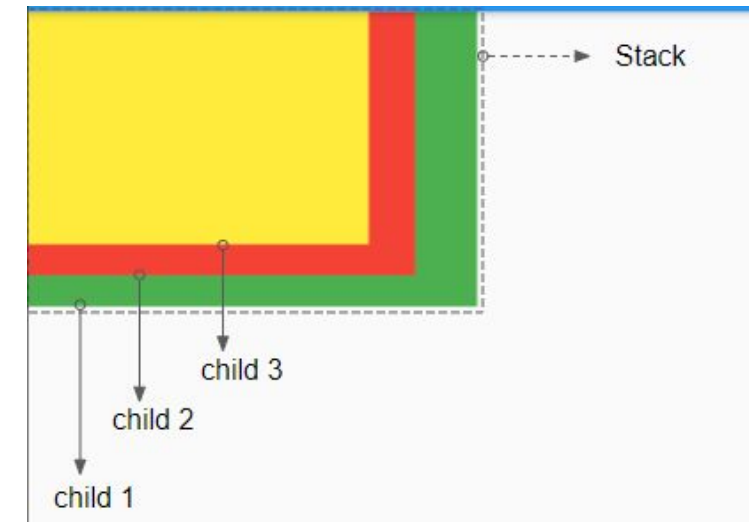
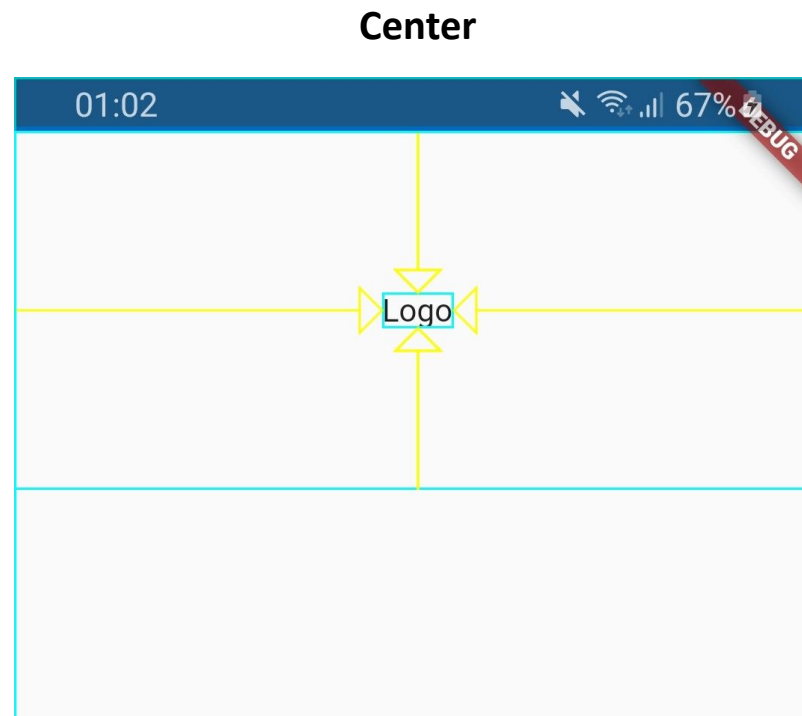
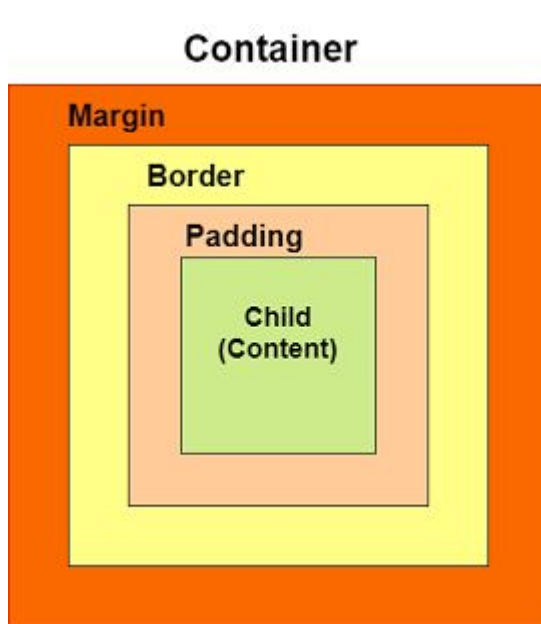


iOS cupertino widgets in flutter

Introduction to Widgets

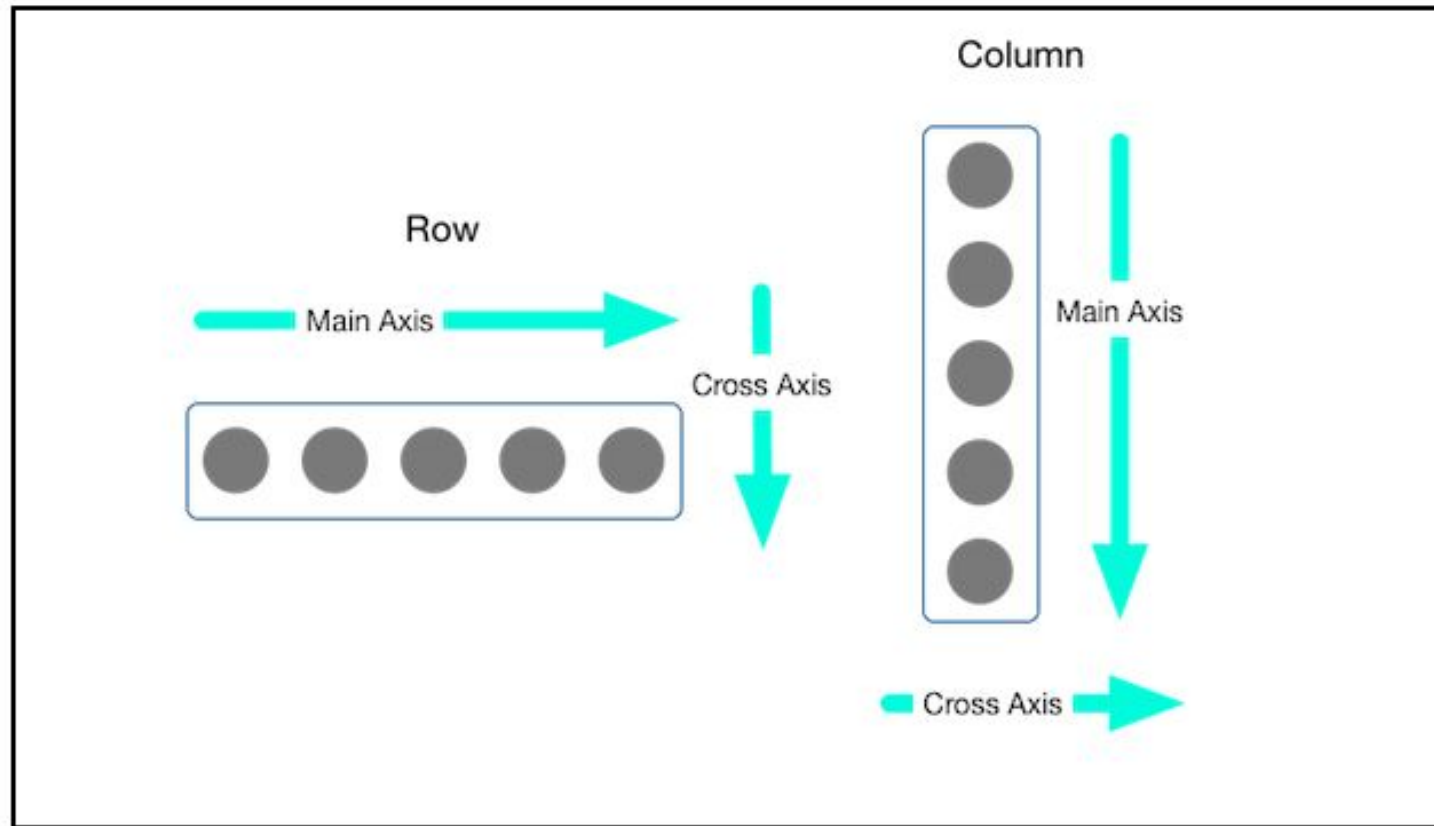
Types of widget

Layout Widgets: are widgets that facilitate how widgets are placed in viewport, this includes alignment ,scroll direction and position .



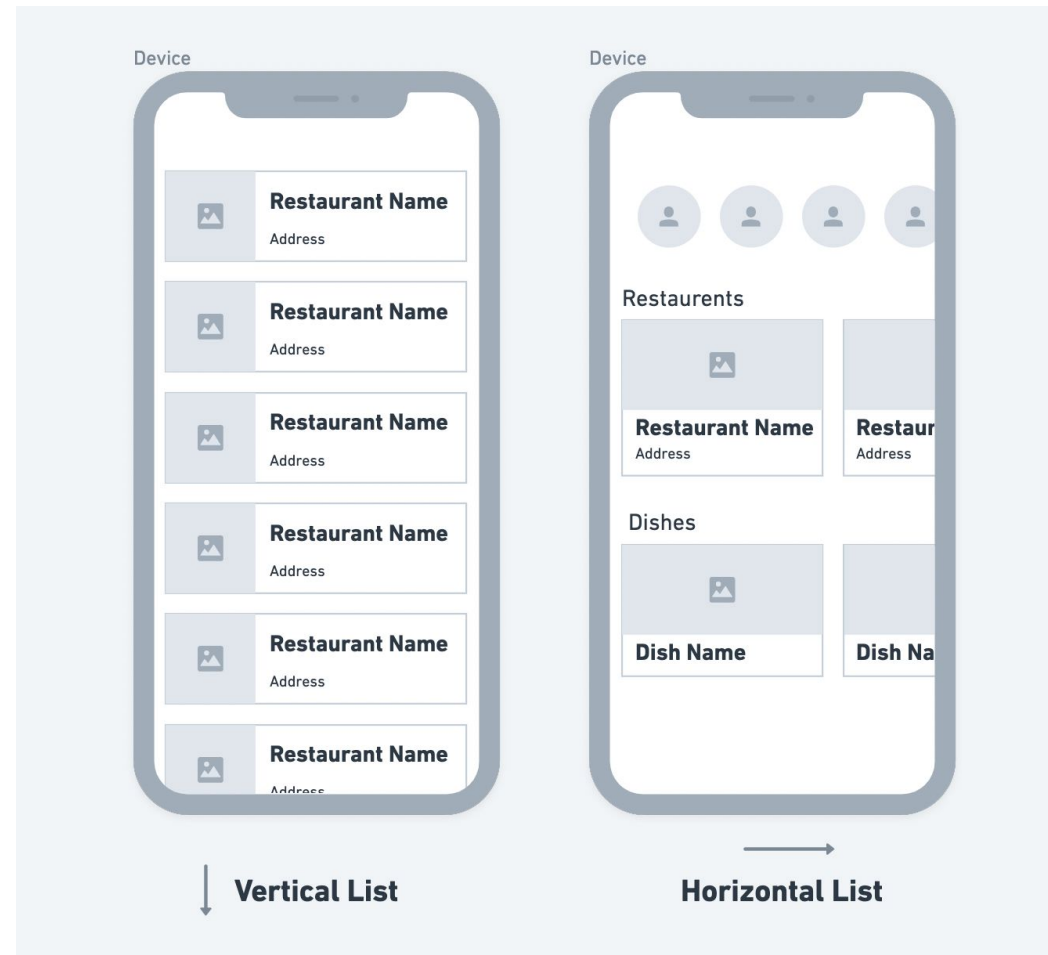
Introduction to Widgets

Types of widget - Row & Column



Introduction to Widgets

Types of widget - Listview





Introduction to Widgets

Types of widget - State Maintenance

- **Stateful Widgets**: stateful widget is dynamic and its state can be mutated within that widget, any event that triggers a state change , refreshes the UI.
- **Stateless Widgets** : is widget that never changes or updates unless its props change, the props in stateless widgets are immutable.



Reading Assignment

- SliverLists
- Custom ScrollView
- Nested Scrollview
- Images
- Dribbble
- Dart Cheat sheet

Thank you!

HahuJobs

ለህገ ልጅ ስህተት ልጅ !

Tesfahun Kebede

tesfahunkebede336@gmail.com

