



Summer Course on Flutter

DAY 1

Dr. Sujatha R Upadhyaya

Department of Computer Science and Engineering

Day 1 - Agenda

- ❑ Introduction to Flutter
- ❑ Scope of this Course
- ❑ Hardware and Software Requirements
- ❑ Installation of Flutter and other software required
- ❑ Getting my first working!

Why Flutter?

- ❑ Mobile development is split into two world



- ❑ One Customer base
- ❑ Double the effort of development!
- ❑ Double the effort of maintenance!

Why Flutter?

- ❑ Native vs Hybrid approach
 - ❑ Native approach
 - ❑ Requires two different codebases
 - ❑ Hybrid approach
 - ❑ One codebase but performance is compromised
- ❑ Different SDKs used for Android & iOS app development
 - ❑ Therefore, Different frameworks!
- ❑ Need **ONE FRAMEWORK** that works for both

What is Flutter?

- ❑ Google's answer for App Development Challenges



- ❑ An open-source UI development-kit
- ❑ A Cross-Platform app development framework
- ❑ A 2D app development framework (SDK + Dart)
- ❑ Official Flutter Documentation is available at

<https://flutter.dev>

Scope of this Course

- ❑ What will we learn?
- ❑ Flutter Framework
- ❑ Dart
- ❑ Building and Compiling Flutter Code for Android Development
- ❑ Testing
- ❑ Why only Android Development?
- ❑ Limitations we have

Hardware Requirement

- ❑ Microsoft® Windows® 7/8/10 (64-bit)
 - ❑ 4 GB RAM minimum, 8 GB RAM recommended.
 - ❑ 2 GB of available disk space minimum,
 - ❑ 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
 - ❑ 1280 x 800 minimum screen resolution.
- ❑ macOS 10.15 (Catalina) or newer.
 - ❑ CPU: Intel Core i5-8400 or better.
 - ❑ Memory: 16 GB RAM.
 - ❑ Free storage: 20 GB SSD.

Software Requirement

- ❏ Flutter + Dart Dart is a language framework
- ❏ VS Code (a lightweight framework)
- ❏ Android Studio (IDE)

Software Requirement

- ❏ Flutter + Dart
 - ❏ Dart is a language framework
- ❏ Android Studio
- ❏ VS Code

Agenda for the Rest of the day

- ❏ Installation of Flutter
- ❏ Installation Android Studio
- ❏ Installation of VS Code
- ❏ Flutter Architecture

Install Flutter

- ❏ Go to <https://docs.flutter.dev/get-started/install>
- ❏ Choose installation for your OS and Complete Installation
- ❏ Remember to:
 - ❏ Set Path
 - ❏ Test with Flutter Doctor

Install Android Studio

- ❏ <https://developer.android.com/studio>
- ❏ Download the version appropriate for your OS
- ❏ Complete Installation
- ❏ Install Flutter
 - ❏ Preferences - Plugin - search for Flutter - yes to install
dart plugin - Restart when prompted
- ❏ Add a device to emulator

Install VS Code

- ❏ Installation of Flutter
- ❏ Installation Android Studio
- ❏ Installation of VS Code
- ❏ Flutter Architecture

Creating Your First App

- ❏ Open Android Studio
- ❏ Create New Flutter Project
 - ❏ File - New Flutter Project
 - ❏ You may asked for the following: Fill in appropriately
 - ❏ Project name:
 - ❏ Flutter SDK Path:
 - ❏ Project Location:
 - ❏ Description:

Creating Your First App

- ❏ Add package configuration
 - ❏ Add organization's domain
 - ❏ Default domain(android studio)
 - ❏ example.com
 - ❏ Default Package name
 - ❏ com.example.projectname

Project Structure

- **android** - Auto generated source code to create android application
- **ios** - Auto generated source code to create ios application
- **lib** - Main folder containing Dart code written using flutter framework
- **lib/main.dart** - Entry point of the Flutter application
- **test** - Folder containing Dart code to test the flutter application
- **test/widget_test.dart** - Sample code
- **.gitignore** - Git version control file
- **.metadata** - auto generated by the flutter tools
- **.packages** - auto generated to track the flutter packages
- **.iml** - project file used by Android studio
- **pubspec.yaml** - Used by **Pub**, Flutter package manager
- **pubspec.lock** - Auto generated by the Flutter package manager, **Pub**
- **README.md** - Project description file written in Markdown format



Summer Course on Flutter

DAY 2

Dr. Sujatha R Upadhyaya

Department of Computer Science and Engineering

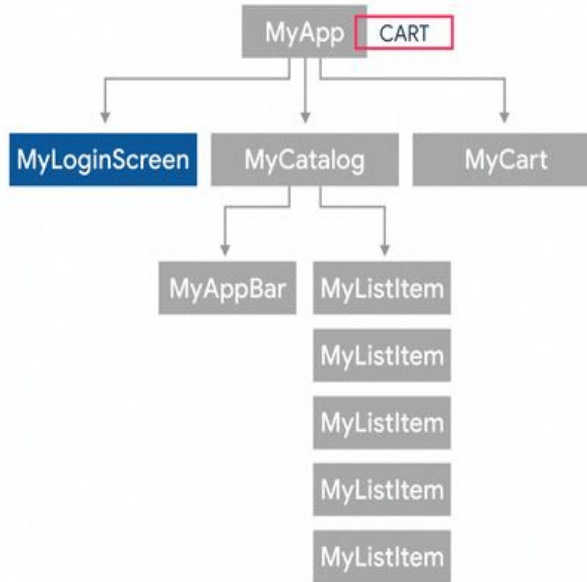
Agenda - Day 2

- ❑ Understanding Flutter Architecture
 - ❑ Widgets / Widget Tree / Widget Types
- ❑ Write a simple code depicting a widget tree
- ❑ Improve on that to learn to use more types of widgets
- ❑ Learn more about using VS code
- ❑ Improve your code quality

Flutter Architecture

- ❏ Everything in a Flutter App is Widget
 - ❏ Menu Section
 - ❏ Buttons
 - ❏ Dropdown Menus
- ❏ Whole of the app is a Widget Tree
 - ❏ Main App - the root of a widget tree

Flutter Architecture



Flutter Architecture ...

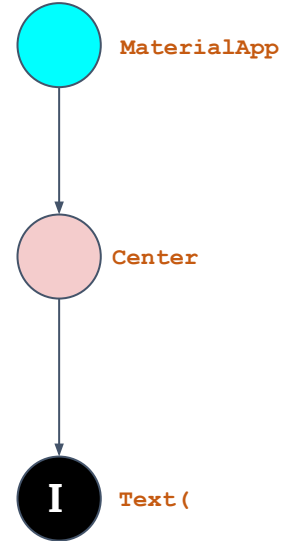
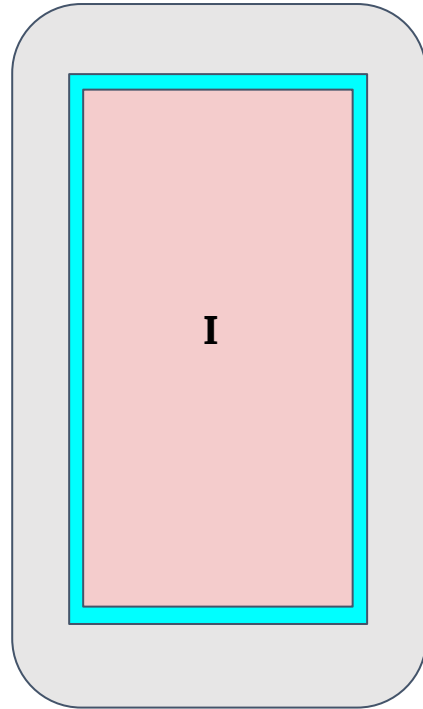
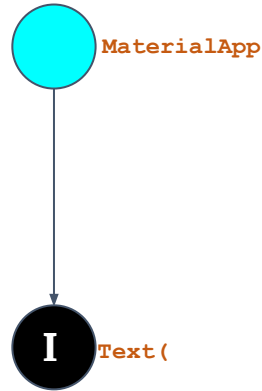
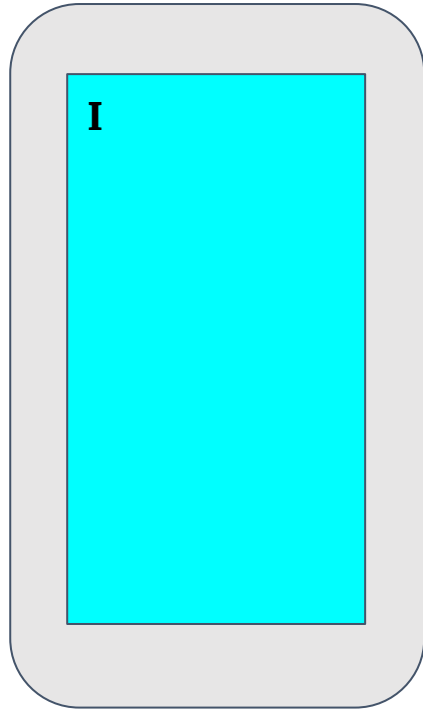
- ❑ Apps perform alike on both platform
- ❑ The way the interact with hardware devices are different
 - ❑ Example - bluetooth
- ❑ Embraces platform differences

Types of Widgets

- ❑ Platform specific widgets
 - ❑ Material Widgets and Cupertino Widgets
- ❑ Layout widgets
- ❑ State maintenance widgets
 - ❑ Stateless and Stateful
- ❑ Platform independent / basic widgets
 - ❑ Text, Row, Column, Container, Center, Stack

Creating an App from Scratch

'I am Happy' App - Build a Widget Tree



```
void main() {
  runApp(
    MaterialApp(
      home: Center(

child:Text("I"),
      ),
    ),
  );
}
```

While coding with DART...short cuts

- ❏ ctrl + k - reformat code
- ❏ ctrl + shift + p - Command Palette
- ❏ ctrl + p - Search Bar
- ❏ stless - Insert a StatelessWidget
- ❏ stful - Insert a StatefulWidget
- ❏ stanim - Insert a StatefulWidget with an
- ❏ ctrl + s - Hot Reload
- ❏ Edit → Preference → Appearance
 - ❏ - Make changes to how your edit looks
 - ❏ Enable / disable comments
- ❏ **Notice that hot reload does not work with previous code!**

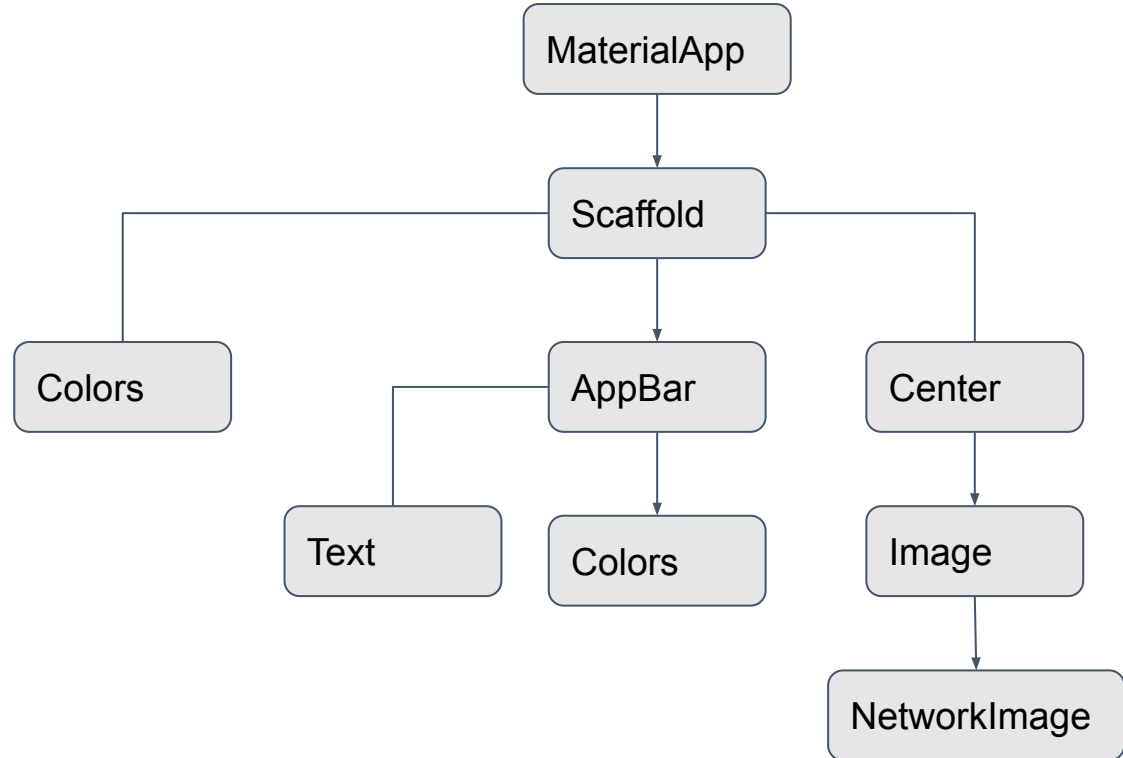
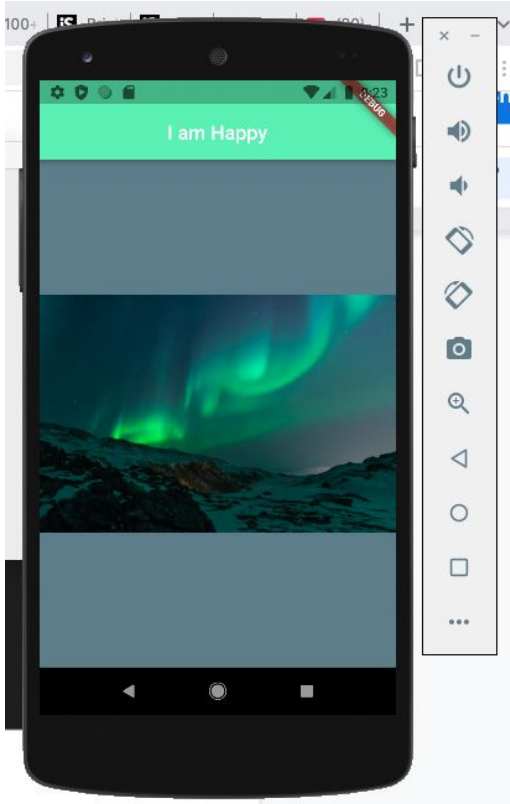
Let's improve...

- ❑ This does not even look like a typical app!
- ❑ Use widgets that give you better control
 - ❑ Scaffold
 - ❑ appBar, body
 - ❑ AppBar
 - ❑ backgroundColor, title

Widgets



Let's improve...this way



https://w3schools.com/w3css/img_lights.jpg

Resources

 [Widget catalog | Flutter](#)

 <https://api.flutter.dev/flutter/material/Colors-class.html>

