FLUTTER WORKSHOP

Agenda

- Installation and set-up
- What is Flutter?
- Why Flutter?
- Overview of Flutter architecture
- Quick review of Functions, Classes, and Objects - *Demo*

Installation

- ❑ Download Android Studio https://developer.android.com/studio
- ❑ Download Git https://git-scm.com/downloads
- ❑ We will provide you with a demo on how to set up your code editor

☺

What is Flutter?

- Mobile UI for creating apps for both iOS and Android systems
- One code-base for both operating systems
- Uses Dart as a programming language

Why use Flutter?

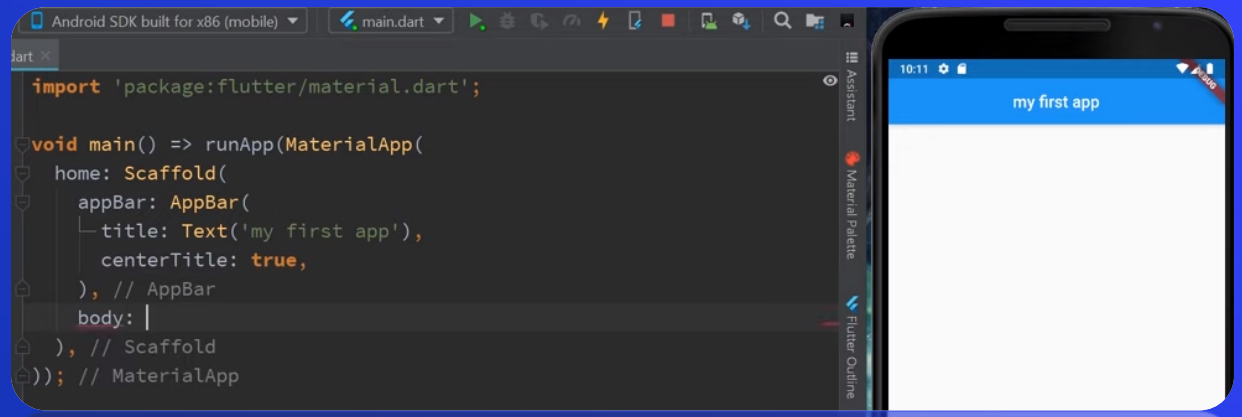
- ❑Easy to use with only 1 code base
- ❑Has great documentation and guides
- ❑Integrates with Firebase (Online Database often used as a backend service)
- ❑It is an open-source framework
- ❑Flexible to allow user to customize design

Apps currently made with Flutter

- ❑ Pairing
- ❑ Meeve
- ❑ n8
- ❑ Topline
- ❑ Music Tutor
- ❑ JD Finance
- ❑ Google Ads
- ❑ SG Bus Tracker

Scaffold Widget

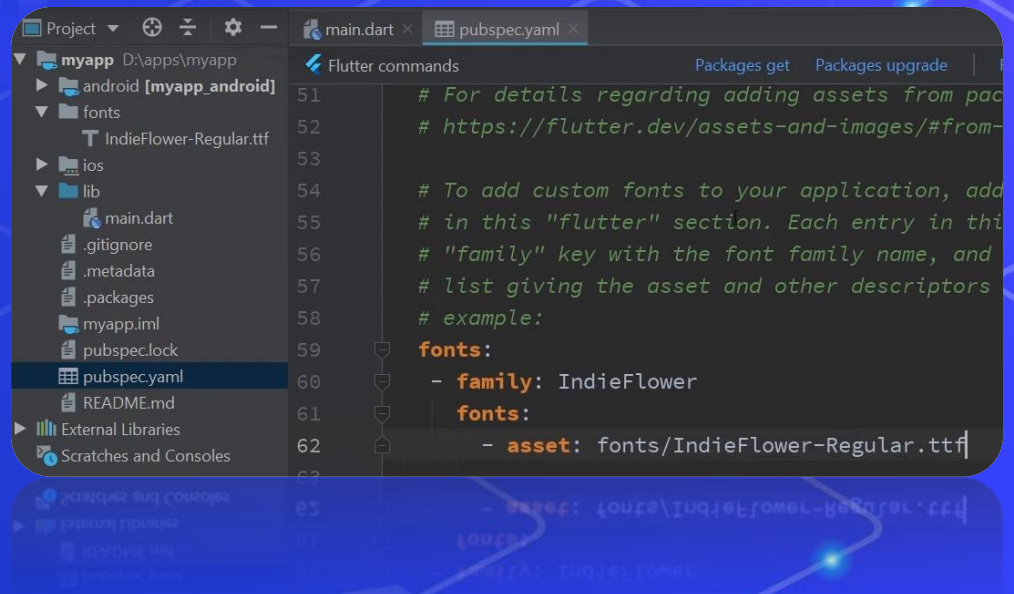
- A **Scaffold Widget** is basically a wrapper to a few different layout widgets.
- For example, the app bar, the body, and a floating action button.
- Provides a framework which implements the basic material



Text Style widget

- Allows you to customize text
- Allows you to change text:
  - Font size
  - Ability to bold text
  - Letter spacing
  - Color
  - Font

```
style: TextStyle(
  fontSize: 20.0,
  fontWeight: FontWeight.bold,
  letterSpacing: 2.0,
  color: Colors.grey[600],
```



Hot Reload

- Hot reload feature helps you quickly
  - Easily experiment
  - Build User Interfaces’
  - Add features
  - Fix bugs
  - Injecting updated source code files into the running Dart Virtual Machine (VM). After the VM updates classes with the new versions of fields and functions, the Flutter framework automatically rebuilds the widget tree, allowing you to quickly view the effects of your changes.

Stateless Widgets Vs Stateful Widgets

Stateless Widgets

the state of the widget **cannot** change over time

Stateful Widgets

the state of the widget **can** change over time

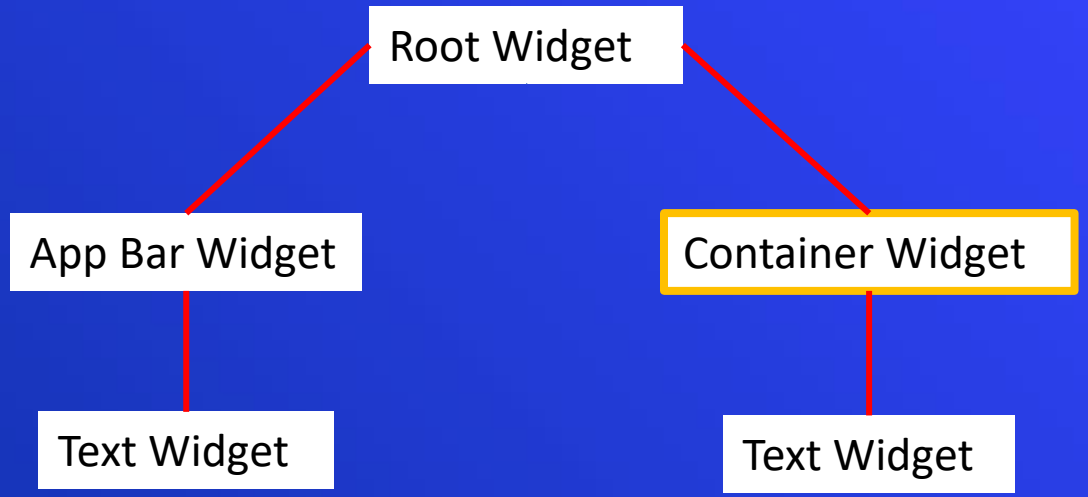
Image widget

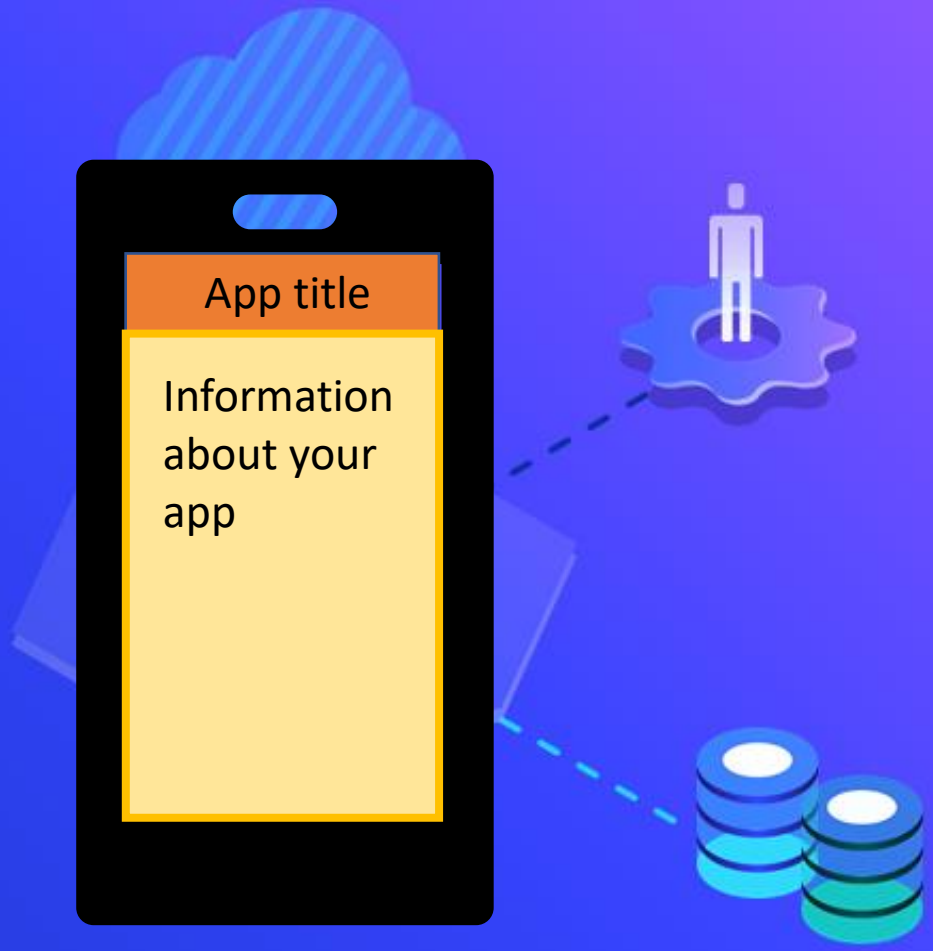
- Images can be added via online links or files from your desktop.
- NetworkImage()
  - displays images from the internet
  - You can copy a url by right clicking an image online and clicking “copy image url”
- Assetimage()
  - Inputs local image on your personal computer.

```
body: Center(
  child: Image(
    image: NetworkImage('https://images.unsplash.com/photo-1501549
```

```
body: Center(
  child: Image(
    image: AssetImage('assets/space-3.jpg')
```

How does flutter operate?





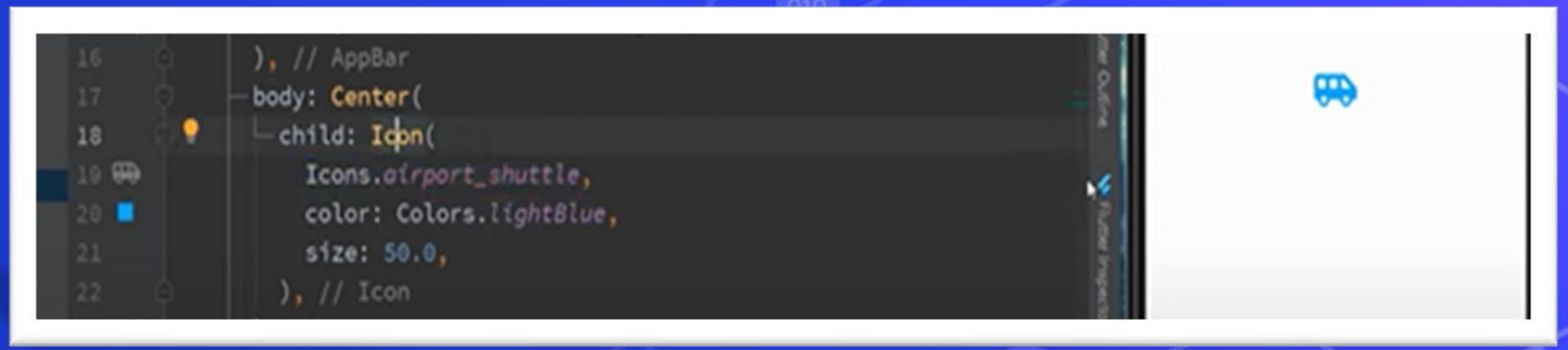
How does Flutter operate?

- A combination of widgets
- Common widgets that are used include:
  - Text widget
  - Button widget
  - Row widget
  - Column widget
  - Image widget

Icons

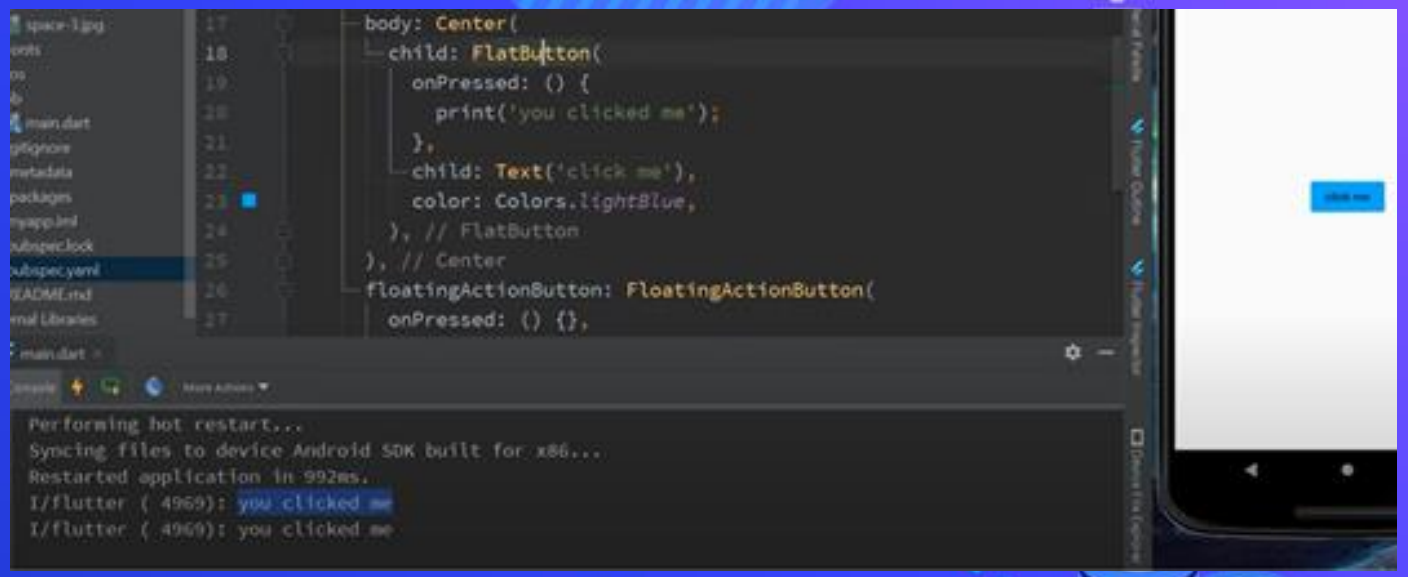
Icon is a widget that can be added to the app to show symbols. To do this set Icon as a child class

Within the child class of icon, you can specify which icon you want, the color, size etc.



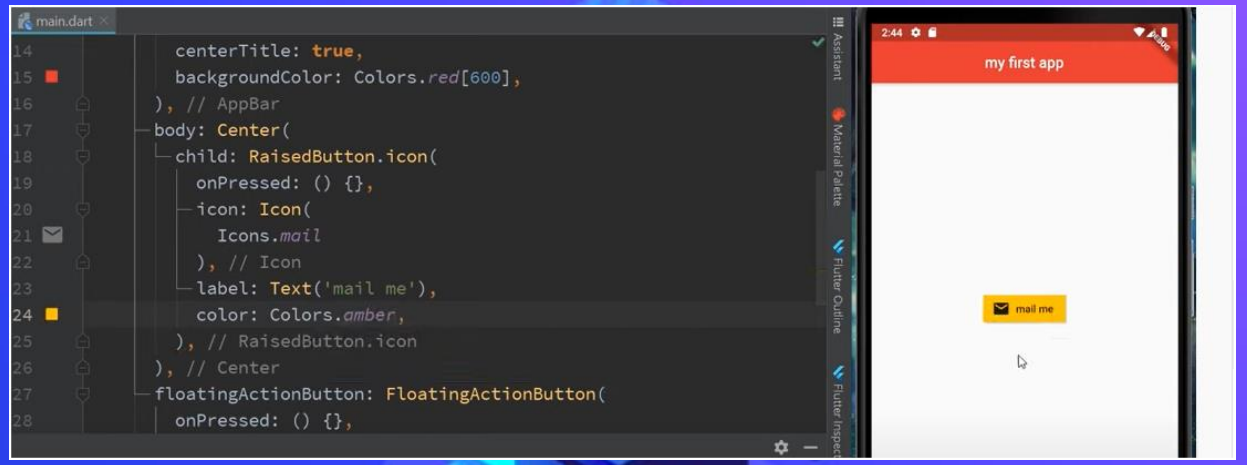
Button

- Create a child class and specify what type of button you want. It can be a flat button or a raised button
- *onPressed* : When the button is pressed a functions will be executed. This case it will state you clicked me
- Child: text (*will appear in the button)*
- Color: can use any color you want
- If you press Control +Q it will list all the other attributes that can be apart of the button class

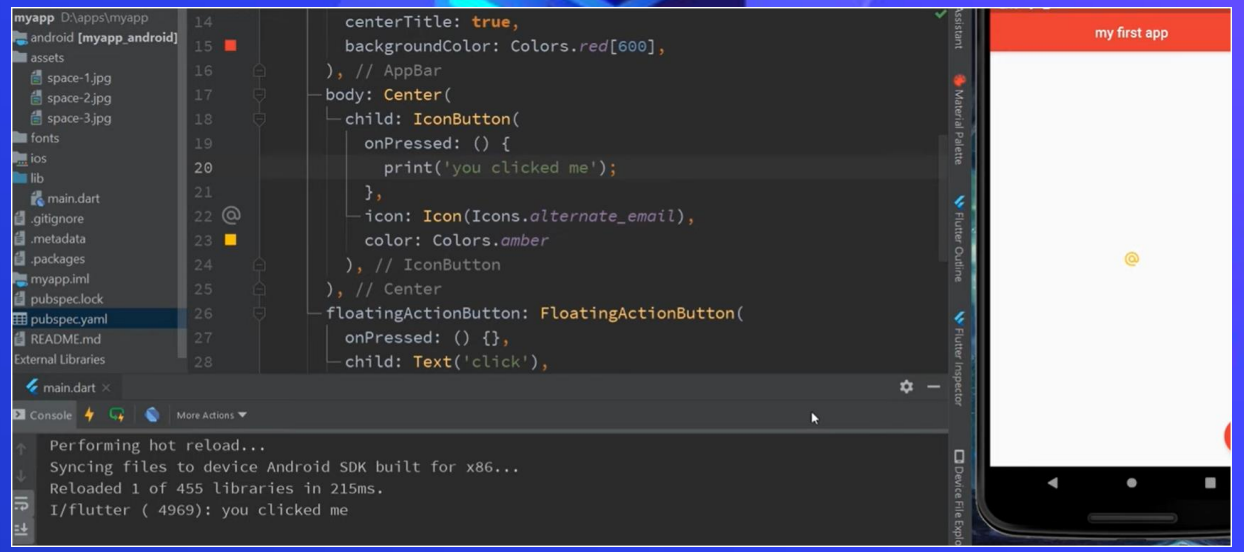


Adding an Icon to the Button

- Child: Raisedbutton.icon *(this means want an icon inside the button)*
- Same rules apply as earlier shown earlier
- Or you can make the icon as a button shown in the bottom figure



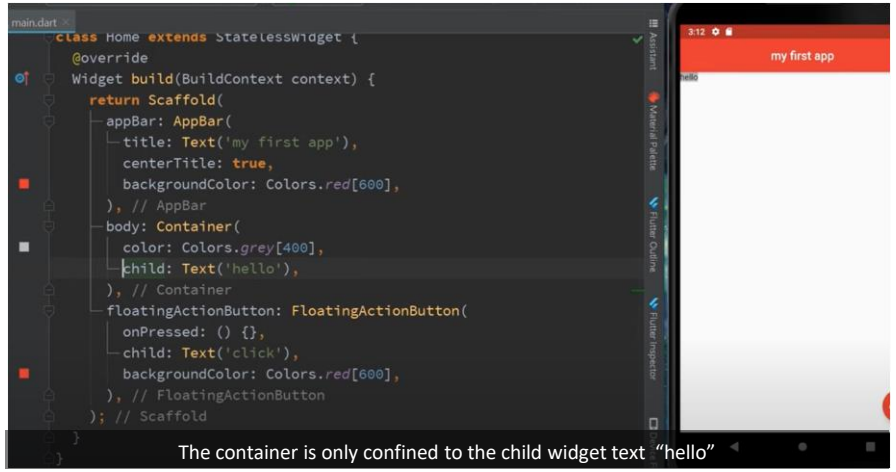
Making the icon appear in the button



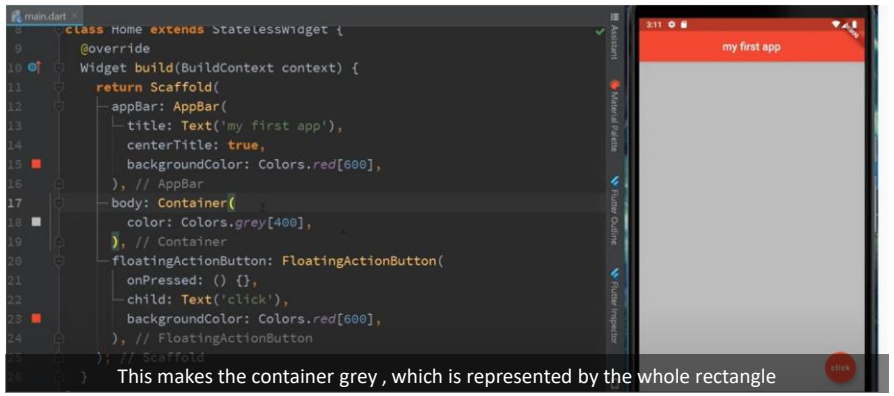
You can have an Icon Button

Container

- Container widget: it helps contain all of the widgets and it comes with properties like padding, margin and color
- Body: Container (container for the widgets
- Color: Color. grey[400] (this is the color of the container)



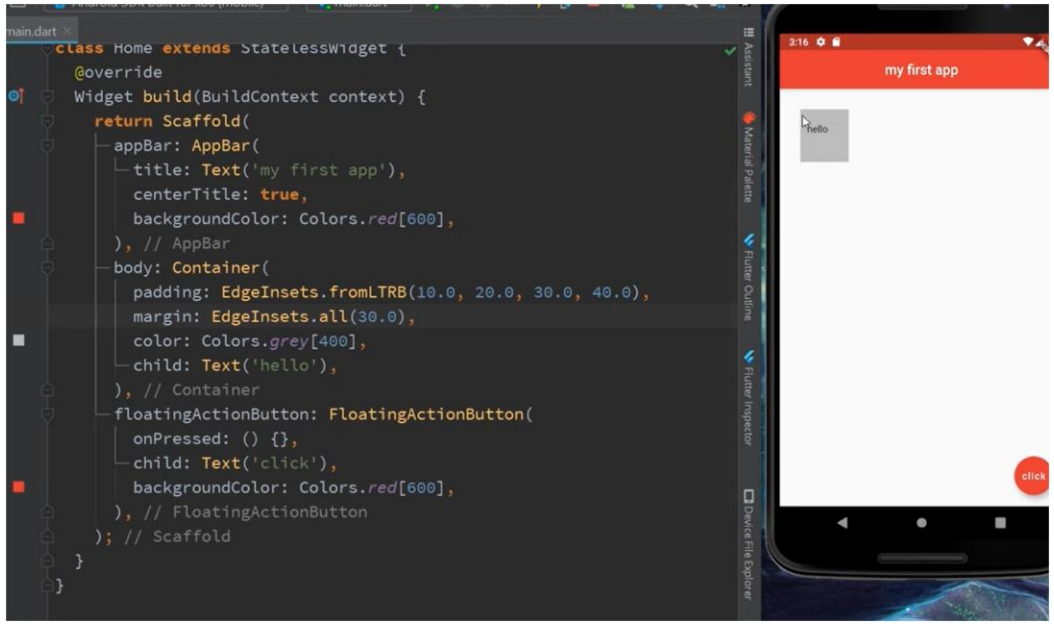
The container is only confined to the child widget text "hello"



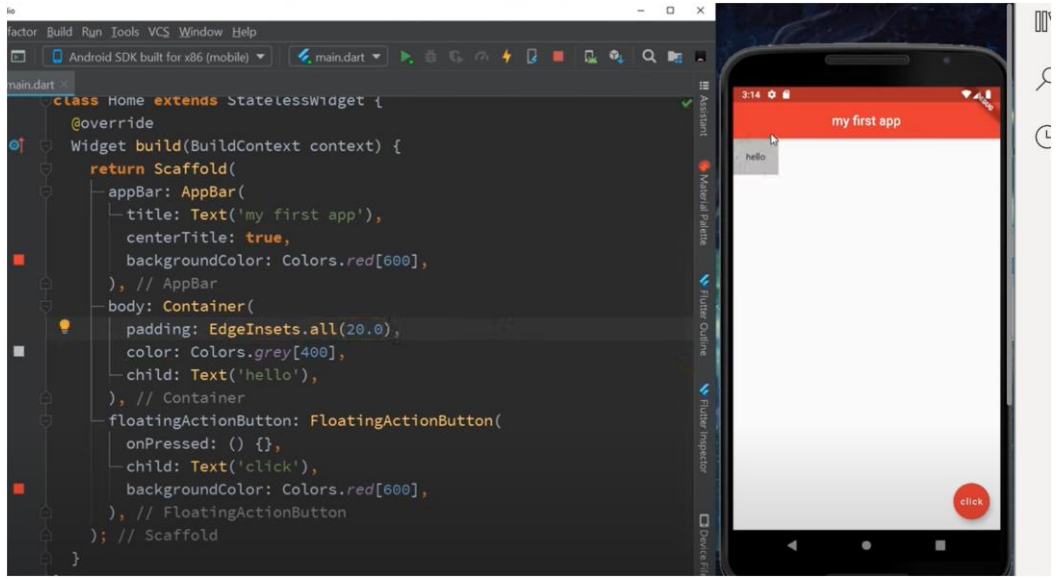
This makes the container grey , which is represented by the whole rectangle

Padding

- Can be used to change the container to whatever shape and size
- **Function:** Edgeinsets()
  - this is how you control padding and margin inside flutter



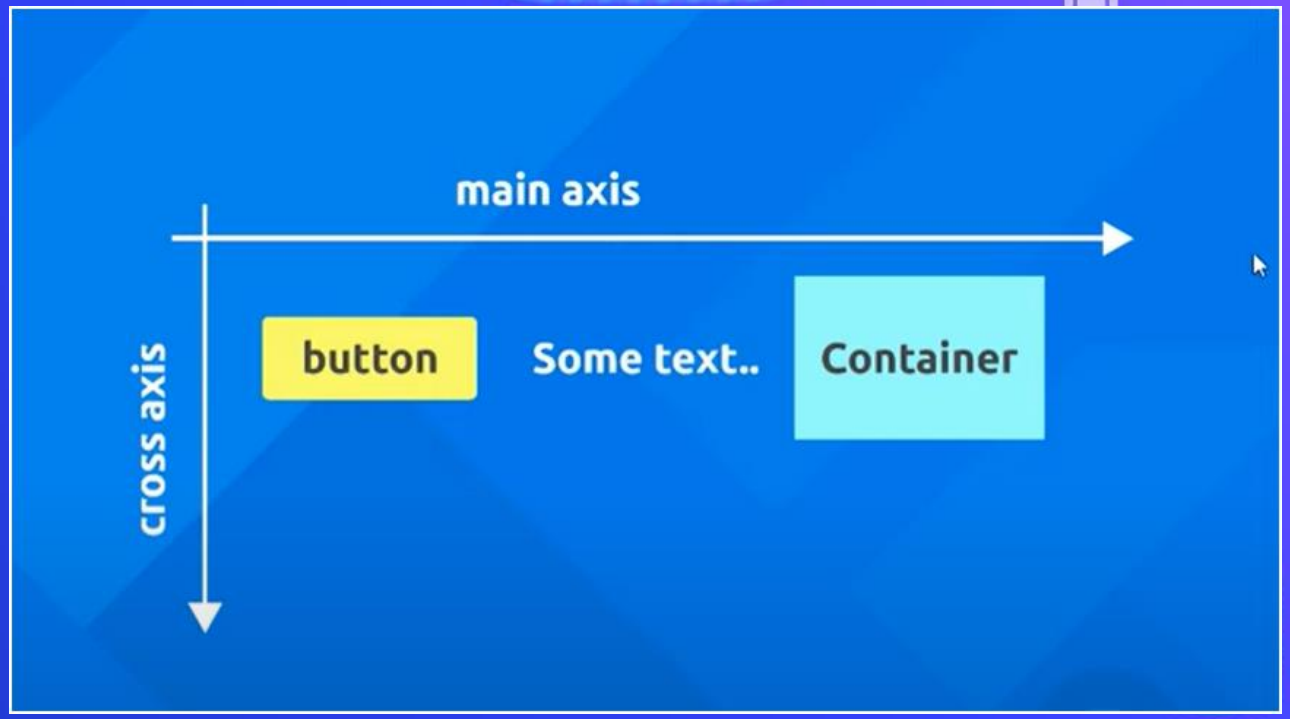
Margin: is around the container



Padding : EdgeInset.all(20) will add 20 pixels of padding around all edges

Rows

- Within your children widget you can have multiple different rows
- Main axis
  - Allows for a horizontal shift
  - Function : MainAxisAlignement.()
- Cross Axis
  - Allows for a vertical shift
  - Function: CrossAxisAlignment.()



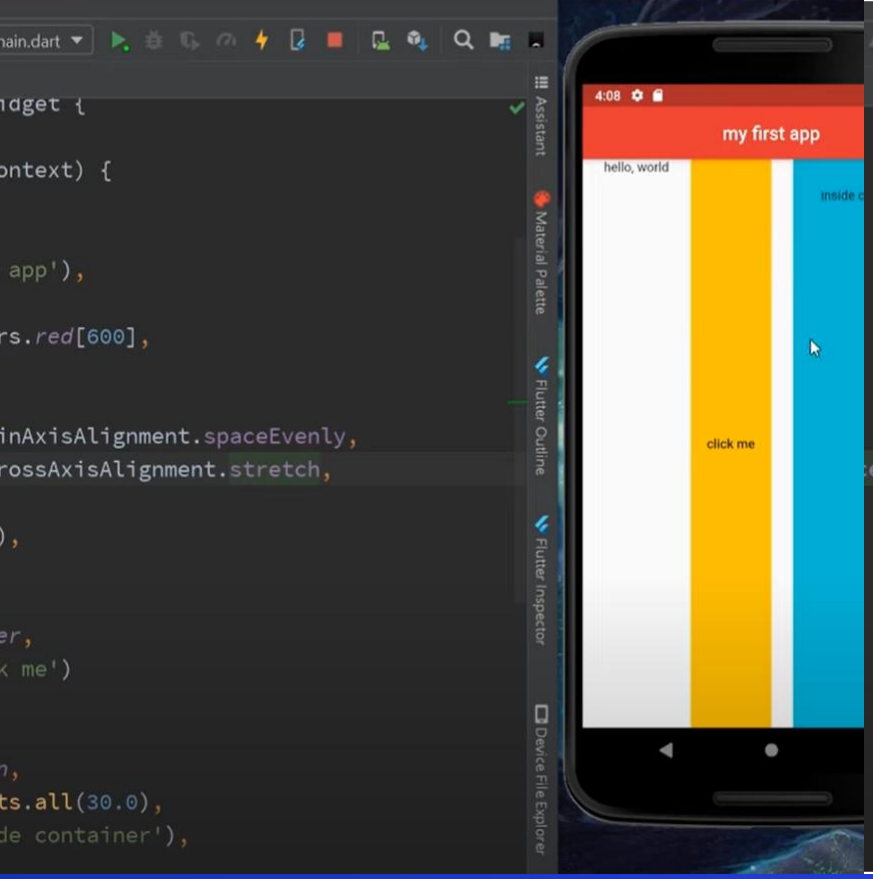


Figure:CrossAxisAlignment.stretch stretches the widget across the vertical axis

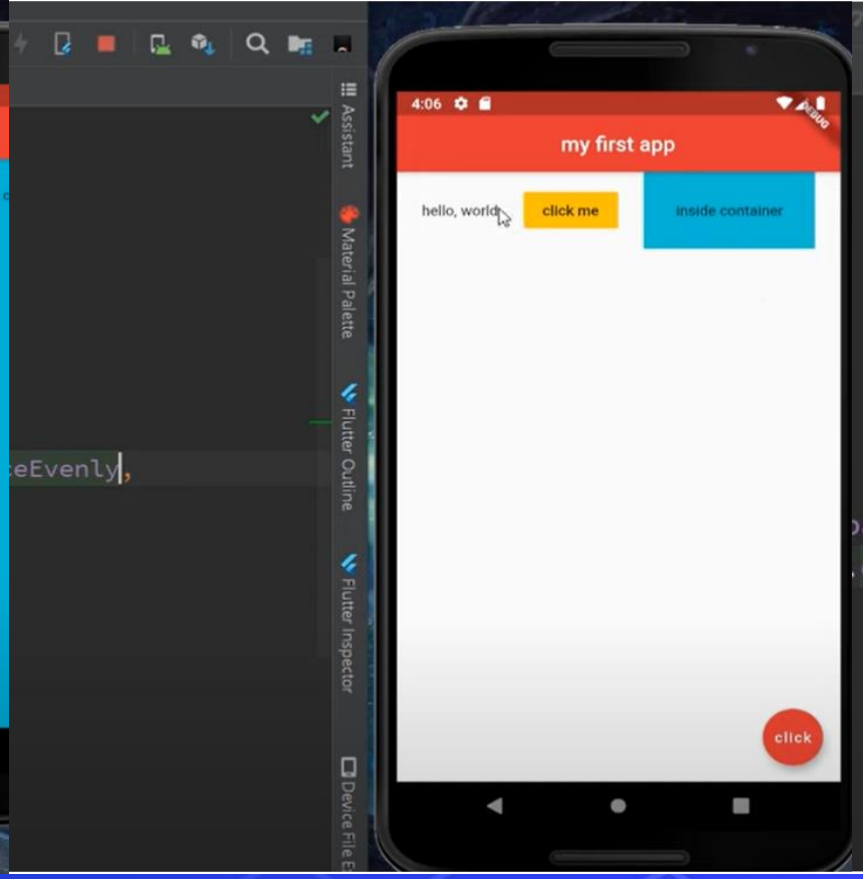


Figure:MainAxisAlignment.spaceEvenly places the widgets evenly along the horizontal axis

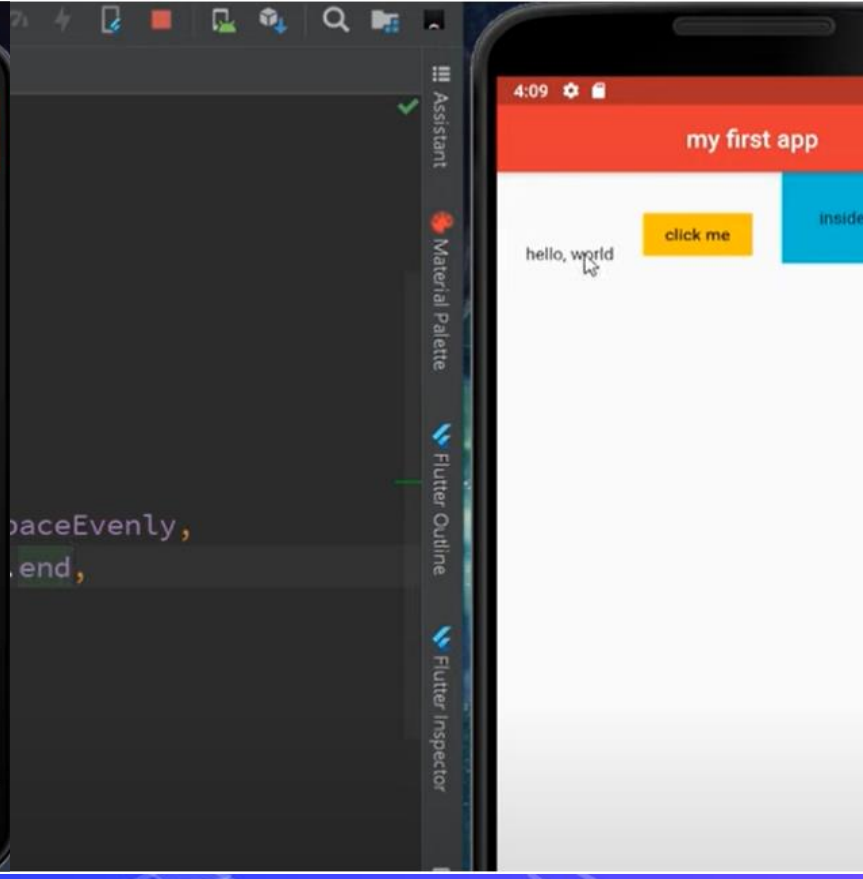


Figure:CrossAxisAlignemnt.end allows the smallest widget to be at the bottom of the highest widget along the vertical axis

Rows

Columns

- Elements stacking on top of the other
- Main axis
  - Allows for a vertical shift
  - Function: CrossAxisAlignment.()
- Cross Axis
  - Allows for a horizontal shift
  - Function : MainAxisAlignement.()