

Introduction to Flutter

Developing a simple mobile app
By Wan Muzaffar Wan Hashim

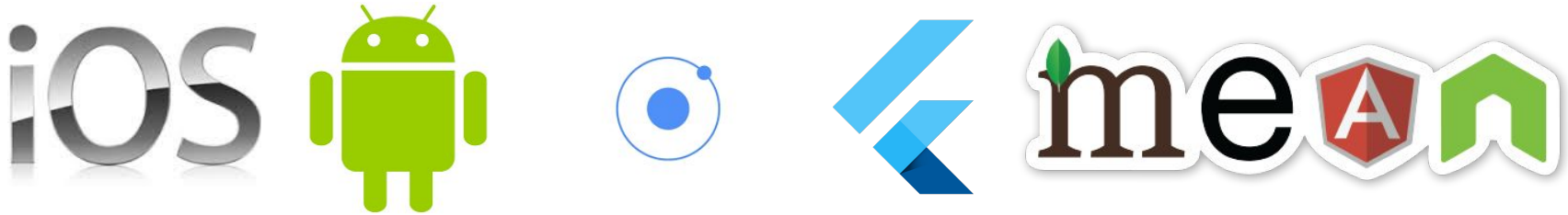


Muzaffar

Founder of MDR-Tech, Co-founder of Anak2U

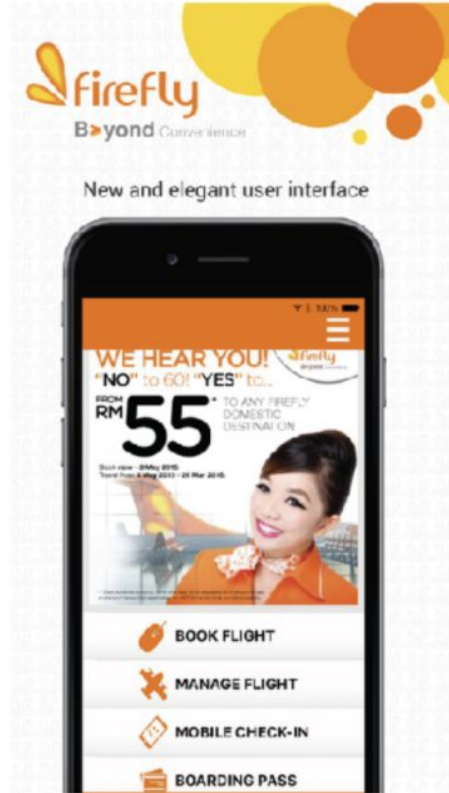
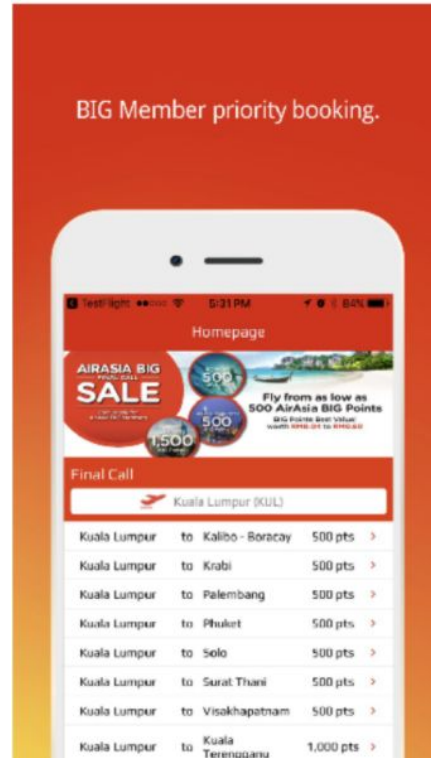
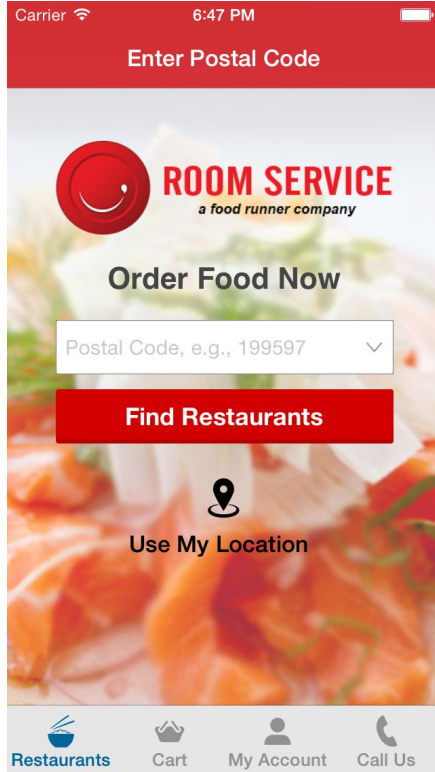
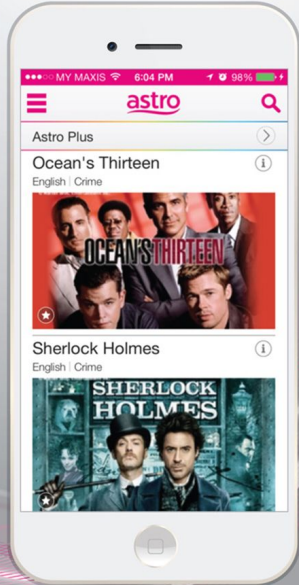
Worked with mobile industry since 2011

Different industry: M-Commerce, Newsfeed, Media Broadcasting, Food Delivery ,
Airline,Loyalty, Education.




Sign up for FREE!

For more features, link your Astro account.
You can also buy a subscription/
pay-per-view.




9:35

← YOUR HOUSEMATE



Zenos Schmickrath



STATUS

I'm bowling right now! 🎳 sometimes I like to add a lot of text to fields

18 Feb

ABOUT

hello! my name is Zenos and this is my about section. I like to write lots of text and it overflows. But this is great so that you can see what a reaaaaaaaaaaaaaaaaaaaaaaaaaaaaaally long status looks like

EMAIL


TEXT

CALL

Home Flatmates Events Services You

9:41

Hello Steven



Lazada Challenge

You saved **RM 300**

You have completed your challenge!

View your reward >







Withdraw your money >

Start new challenge >

Home Rewards Expenses

Anak2u

Kelas Matahari

	Aisyah 5 Year Old		
	Rayyan 5 Year Old		
	Amira 5 Year Old		
	David 5 Year Old		

Home



Mobile app development

- React Native (3 and half week)

- Flutter (3 and half week) -> Emphasize..

Module 1 (Javascript)

- HTML, CSS Javascript -> Building a static website
- Introduction to Front End Framework - Vue JS

Module Mobile

- React Native - JS (More difficult)
- Flutter - Dart (Simpler)

Module backend

- API - JS , ExpressJS
- Database - MongoDB MySQL

Stack...

	JS	Dart	Python	Java	C++	R	PHP	Kotlin	Ruby	Swift
Frontend website	VueJS, ReactJS, Angular Vanilla	Flutter	Flask/Django	JSF, Angular			Pure PHP, Laravel, CakePHP, CodeIgniter		Ruby on Rails	
Backend / API	Express		Flask/Django	Spring / Spring Boot			Pure PHP, Laravel, CakePHP, CodeIgniter		Ruby on Rails	Can

Mobile App Development

- A mobile application is a software application designed to run on smartphones, tablet computers and other mobile devices.
- Users on smartphones typically check the news, weather, email or their social networks. They have a choice between the mobile web version or a specially-created mobile app.

Mobile App Dev: Current State

Native Development	Crossplatform Development
<ul style="list-style-type: none">● Android - Kotlin or Java● iOS - Swift or Objective C (Mac)	<ul style="list-style-type: none">● Flutter - Dart● React Native - JS● Ionic - JS● Xamarin - .NET

Ionic 1 - 2016

Ionic 2 - 2017

Ionic 3 - 2018

Ionic 4 - 2019

Ionic 5 - 2020

Flutter - 2019

Flutter 2 - 2020

Mobile App Types

- **Native**
 - Programmed using Swift/Objective C on the iPhone or using Java/Kotlin on Android devices.
- **Crossplatform**
 - Mix between these two types of mobile applications.
 - Normally based on web programming language, eg: HTML, CSS, Javascript, Dart
 - Built once to be run on Android and iOS.
- **Web Apps / Progressive Web Apps.**
 - Web based.
Runs in the phone's browser.
 - Can have native features based on HTML5

When to choose what?

Native	Crossplatform
I need to have 2 developers Or I need to have 1 person that knows two languages	1 developer only..
4 months	$\frac{2}{3}$ to $\frac{3}{4}$ of Native - 3 months
Stable, spend less time to look for support	Still new, less resources, you might spend more time to look for support
RM 60 000 (Initial cost)	RM 45 000 (Initial cost)
Maintenance cost is lower	Maintenance cost is higher
Long term project...	Short term project
Project that I can take time to do it	Project that I need to do fast
You have a business you want a new	You don't have a business you want to

Quiz

AirAsia - Native.. / It's a channel

Astro Go - Native , security..

Astro Awani - Crossplatform / simple no need security

Maybank2U - Cross platform ? React Native / [I think they are wrong]

MySejahtera - Crossplatform / UI Macam sama.., short term, they need it fast

Selangkah - PWA

Setel - Crossplatform / New business

GSC/TGV - Native , It's a channel

Supply Demand

HTML / CSS / JS -> Almost everyone can do it, from certificate to degree

PHP Developer -> Can hire from diploma level

Ionic Developer, PWA -> Some Diploma Mostly Degree , UITM, UTEM

(Laravel)

Android -> Some Diploma Mostly Degree ,UTM, UTEM

VueJS -> Easy to learn..

Flutter -> Not in school official syllabus, only those who self learn will know, got demand , can self learn.. Might have some self learned student

Cost

Developer Account...

Google Play - 25USD per lifetime..

Phone / tablet

Apple iOS Developer Account

- 99 USD per year

- Macbook

- iphone/ ipad/ ipad mini

What is Flutter

Open source UI Framework by Google

Able to create iOS, Android and web application using Dart

High performance, high fidelity, low latency, as it renders the Native UI.

Use DART as main programming language

Open source / github.



What is Flutter

Open source UI Framework by Google

Able to create iOS, Android and web application using Dart

High performance, high fidelity, low latency, as it renders the Native UI.

Use DART as main programming language

Open source / github.



About Dart

Dart is a programming language developed by Google

Learning it isn't hard if you have experience with Java or JavaScript. You will quickly get it.

You can use `dartpad` as an online compiler of Dart

<https://dartpad.dev/>



Dart

Who uses Flutter



Google



Alibaba.com



Baidu 百度



GROUPON



eBay



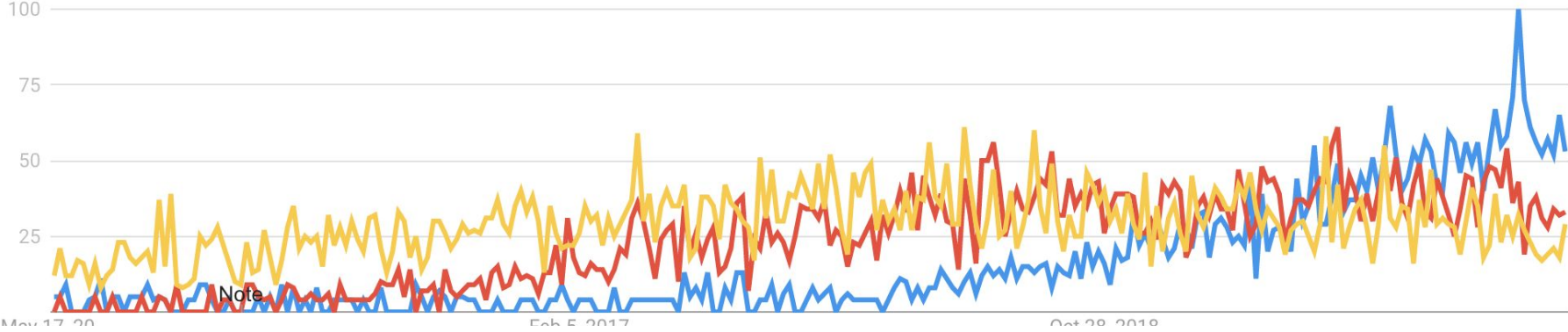
Grab

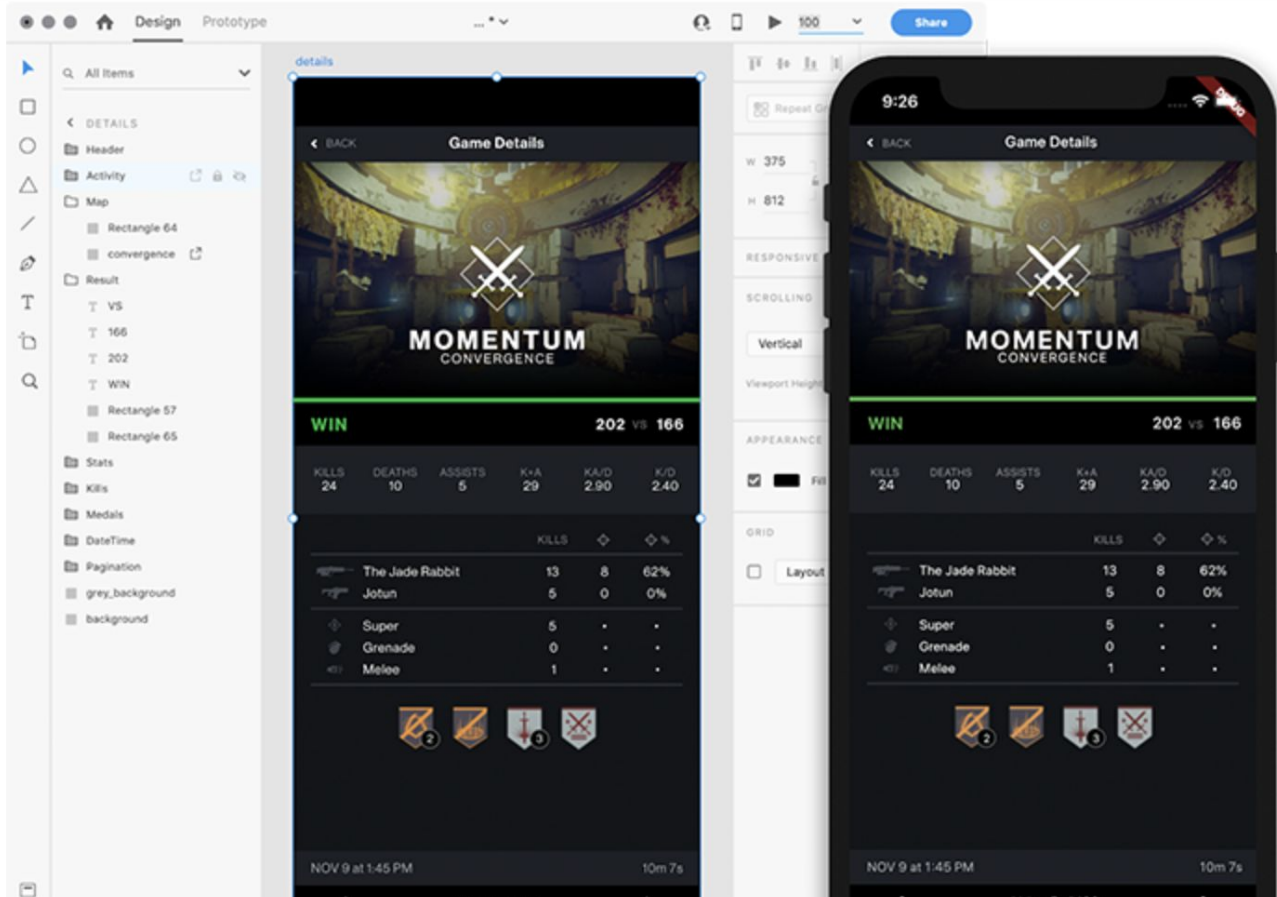


rym
mobile

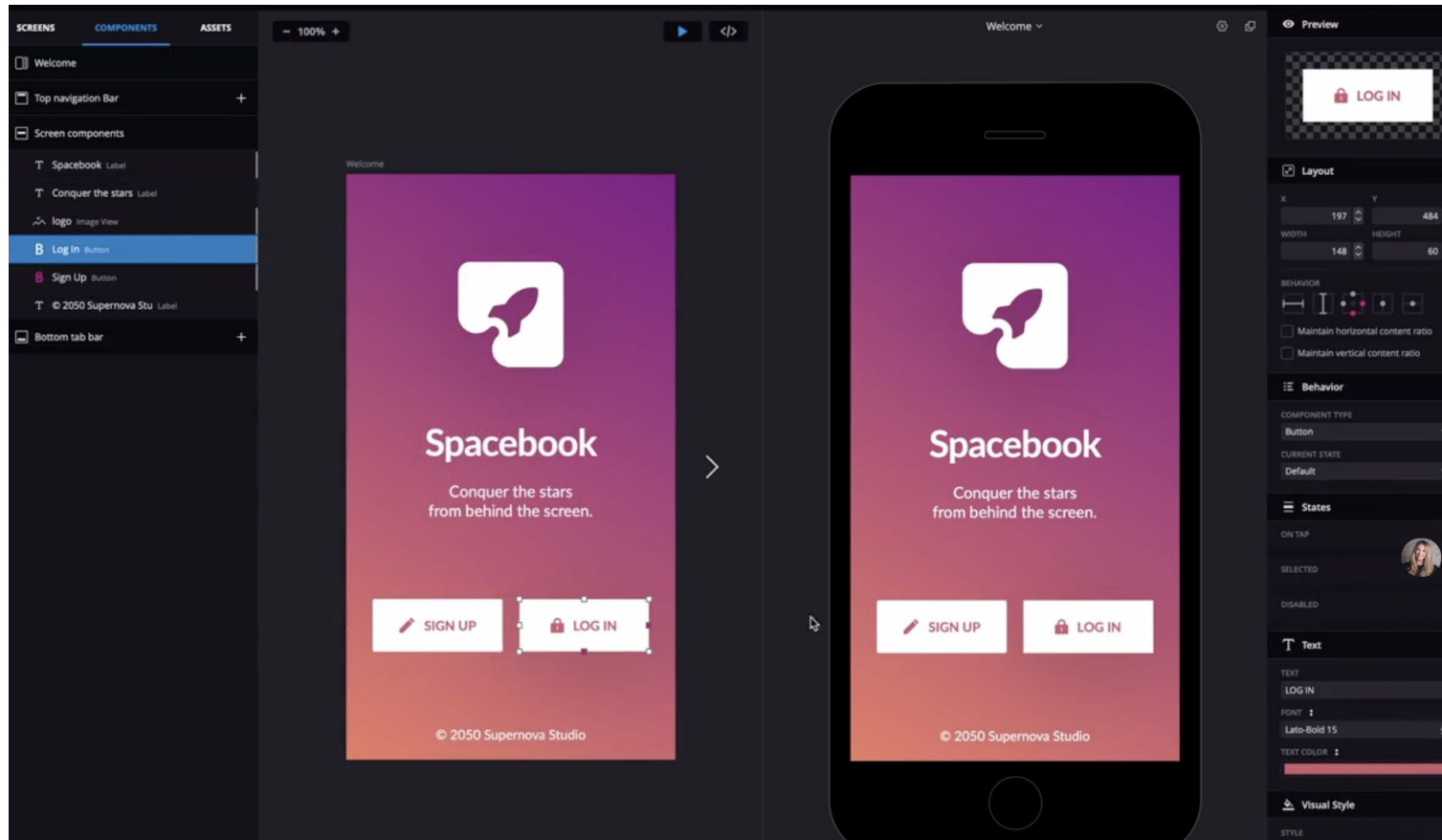


Malaysia Google Trend (over 5 years)





Bridge gap between designer and developer - XD Flutter integration



Bridge gap between designer and developer - Supernova io

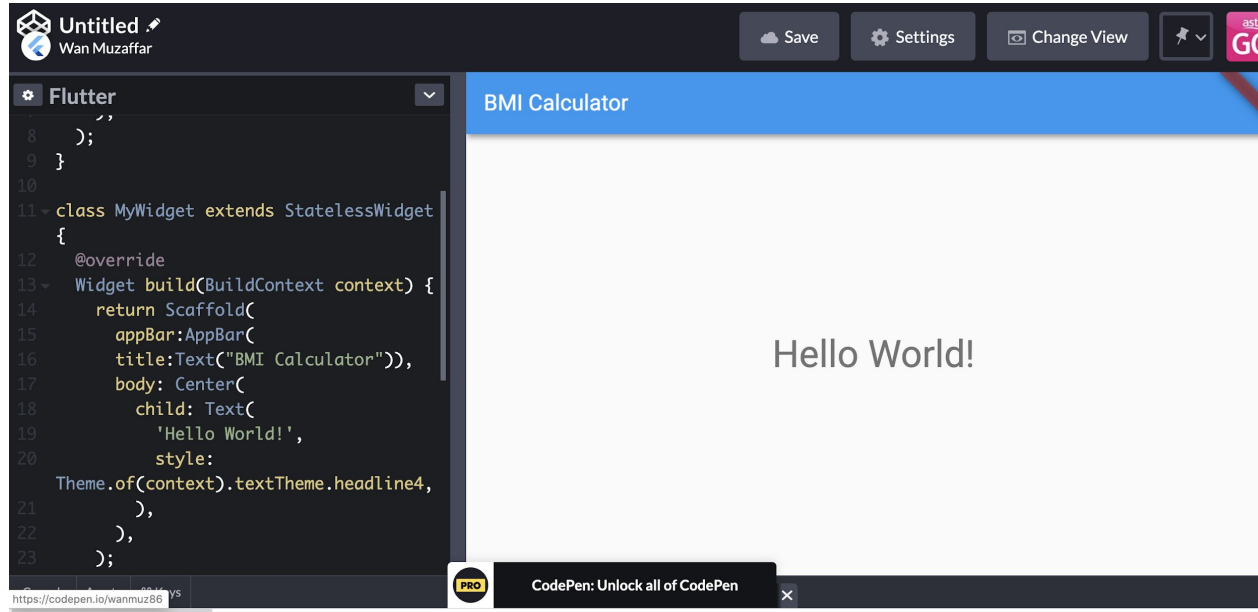
Setup your Editor



<https://flutter.dev/docs/get-started/editor>

You will need to configure an emulator after setting up the SDK.

Online Editor (Demo purposes - no setup)



```
8 );
9 }
10
11 class MyWidget extends StatelessWidget
12 {
13   @override
14   Widget build(BuildContext context) {
15     return Scaffold(
16       appBar: AppBar(
17         title: Text("BMI Calculator")),
18       body: Center(
19         child: Text(
20           'Hello World!',
21           style:
22             Theme.of(context).textTheme.headline4,
23         ),
24       ),
25     );
26   }
27 }
```

BMI Calculator

Hello World!

CodePen: Unlock all of CodePen

https://codepen.io/wanmuz86 vs

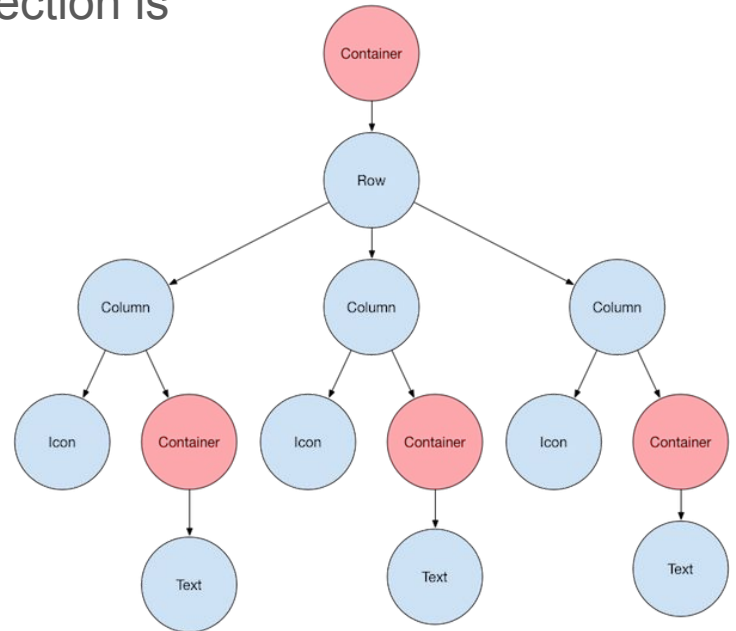
<https://codepen.io/pen/editor/flutter>

Everything is a widget

You build widget upon widget.

Your screen, a section in a screen, a tiny little section is also a Widget.

You create and customize your own widget.



Widget catalog

[Docs](#)[Showcase](#)[Community](#)[Get started](#)[Get started](#) ▾[Samples & tutorials](#) ▾[Development](#) ▴

User interface

[Introduction to widgets](#)[Building layouts](#)[Adding interactivity](#)[Assets and images](#)[Navigation & routing](#)[Animations](#)[Advanced UI](#)[Widget catalog](#)[Data & backend](#)[Accessibility & internationalization](#)[Platform integration](#)[Packages & plugins](#)[Add Flutter to existing app](#)[Tools & techniques](#)

Widget catalog

[Docs](#) > [Development](#) > [UI](#) > [Widgets](#)

Create beautiful apps faster with Flutter's collection of visual, structural, platform, and interactive widgets. In addition to browsing widgets by category, you can also see all the widgets in the [widget index](#).

Accessibility

Make your app accessible.

[Visit](#)

Animation and Motion

Bring animations to your app.

[Visit](#)

Assets, Images, and Icons

Manage assets, display images, and show icons.

[Visit](#)

Async

Async patterns to your Flutter application.

Basics

Widgets you absolutely need to know before building your first Flutter app.

Cupertino (iOS-style widgets)

Beautiful and high-fidelity widgets for current iOS design language.

<https://flutter.dev/docs/development/ui/widgets>

The boilerplate code of an app - Scaffold

```
Scaffold(  
  
  appBar: AppBar(  
  
    title: const Text('Sample Code'),  
  
  ),  
  
  body: Center(child: Text('Hello World')),  
  
  floatingActionButton: FloatingActionButton(  
  
    onPressed: () => {},  
  
    tooltip: 'Increment Counter',  
  
    child: const Icon(Icons.add),  
  
  ),  
  
);
```

Scaffold

A scaffold is a basic structure of an application having the following property by default:

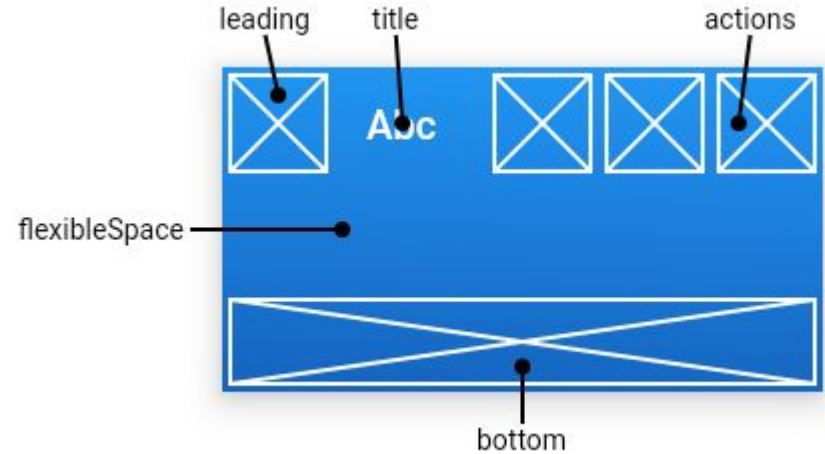
- appBar
- body
- floatingActionButton
- bottomNavigationBar
- drawer

AppBar

An app bar consists of a toolbar and potentially other widgets,

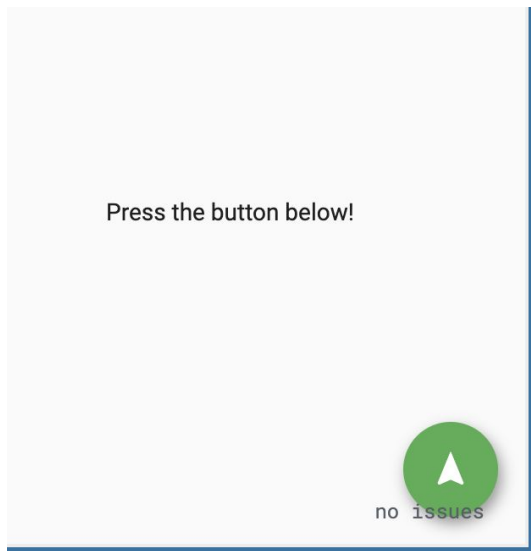
For example, if you would like to add a button on the left side you use `leading` and `actions` on the right side.

You may change the property `backgroundColor` to change the background color of the AppBar.



Floating Action Button

A floating action button is a circular icon button that hovers over content to promote a primary action in the application



```
floatingActionButton:  
FloatingActionButton(  
    onPressed: () {  
        // Add your onPressed code  
        here!  
    },  
    child: Icon(Icons.navigation),  
    backgroundColor:  
    Colors.green,
```

Body

This is where you build the content of your application.

Widgets for layouting / positioning

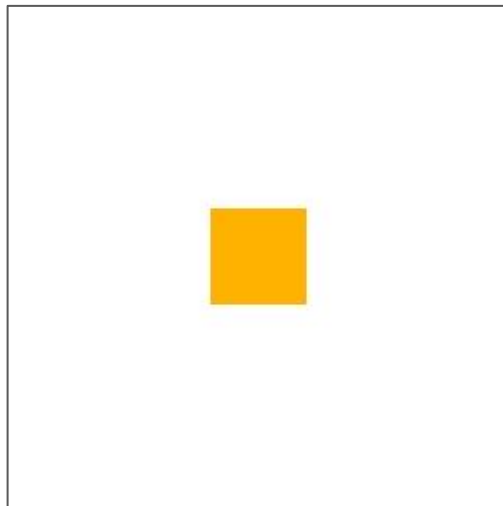
We will discover the widgets that are used to position items within a page. Here are some important/main widgets:

- Container
- Center
- Column
- Row
- SingleChildScrollView

Container

A container is a box! You can specify the width, height, color, padding and margin. In the below example, `EdgeInsets.all` means all direction (top, bottom, left, right)

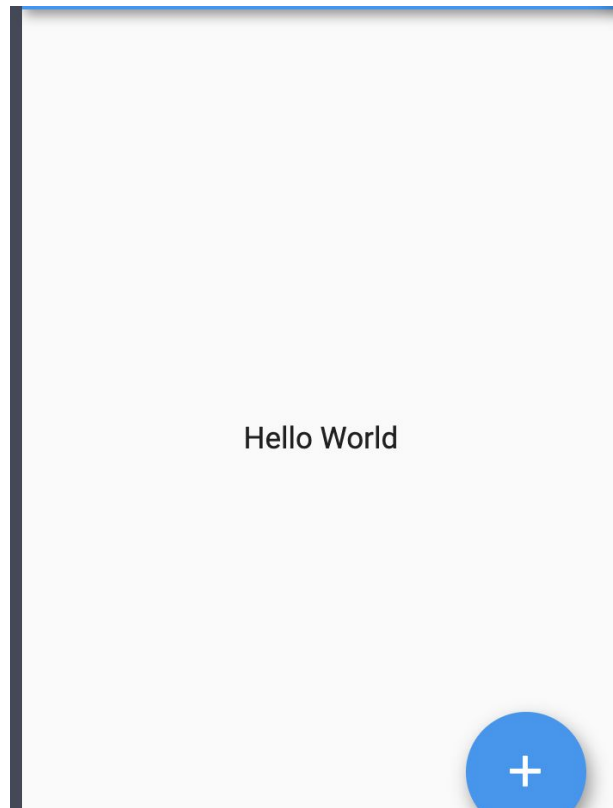
```
Center(  
  
  child: Container(  
  
    margin: EdgeInsets.all(10.0),  
  
    color: Colors.amber[600],  
  
    width: 48.0,  
  
    height: 48.0,  
  
    padding:EdgeInsets.all(10.0)  
  
  ),  
  
),
```



Center

A widget that centers its child within itself.

```
Center(child: Text('Hello World')),
```



Row

A widget that displays its children in a horizontal array.

```
Row(  
  children: <Widget>[  
    Expanded(  
      child: Text('Deliver features faster', textAlign:  
TextAlign.center),  
    ),  
    Expanded(  
      child: Text('Craft beautiful UIs', textAlign:  
TextAlign.center),  
    ),  
    Expanded(  
      child: FittedBox(  
        fit: BoxFit.contain,  
        child: const FlutterLogo(),  
      ),  
    ),  
  ],  
)
```

Deliver features
faster

Craft beautiful UIs



Column

A widget that displays its children in a vertical array.

```
Column(  
  children: <Widget>[  
    Text('Deliver features faster'),  
    Text('Craft beautiful UIs'),  
    Expanded(  
      child: FittedBox(  
        fit: BoxFit.contain,  
        child: const FlutterLogo(),  
      ),  
    ),  
  ],  
)
```

Deliver features faster
Craft beautiful UIs



SingleChildScrollView

A box which allows a single widget to be scrolled.

You will use this when you have a single box that will normally be entirely visible, for example a clock face in a time picker, but you need to make sure it can be scrolled if the container gets too small in one axis

Visible widget in Flutter

Once you know how to position items on a page, we will see some of the widgets that you can use in your application. Here are some important/main widgets:

- Text
- Image
- Button
- Icon
- Slider

Text

This widget is used to displays a text with single style.

You might need to use TextStyle widget as well with this widget to add styling to the text, for example to add color, set to bold

```
Text(  
  'Hello World',  
  textAlign: TextAlign.center,  
  style: TextStyle(fontWeight: FontWeight.bold,  
    color:Colors.red),  
),
```

Image

To show an image. You may show an image from:

- Downloaded from a URL (Image.network)
- Stored locally in assets folder

```
Image.network(  
  
  'https://flutter.github.io/assets-for-api-docs/assets/widgets/owl.jpg'  
  
)
```

Icon

As per its name, an icon is a widget that is predefined, and can be used directly within your application.

You may refer to Icon documentation, to see all available icon ready to be used in your application

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

```
Icon(  
  Icons.audiotrack,  
  color: Colors.green,  
  size: 30.0,  
),
```


TextButton

A TextButton, follows Material design principle is a button that raises slightly, configurable via elevation property.

You will need to declare what should happen when the button is pressed via it's onPress property.

Other type of button includes FlatButton

```
TextButton (  
  child: Text("Press Me"),  
  onPressed: () {  
    print("Hello World")  
  },  
)
```

Slider

A slider can be used to select from either a continuous or a discrete set of values.

We will use `onChanged` property to update the value of item, once the value of slider changed.

```
Slider(  
  value: _value.toDouble(),  
  min: 1.0,  
  max: 10.0,  
  onChanged: (double newValue) {  
    setState(() {  
      _value = newValue.round();  
    });  
  },  
);
```

Stateless Widget

Stateless Widget is a widget that is immutable.

Stateless widgets cannot change their state during the runtime of the app, which means the widgets cannot be redrawn while the app is in action.

```
class MyApp extends StatelessWidget {  
  
  @override  
  
  Widget build(BuildContext context) {  
  
    return Container()  
  
  }  
  
}
```

The starter code of a Flutter project / Boilerplate code

Once your flutter project is created, remove all the codes from line 30 to the end (class MyHome ...)

Remove all the comments from line 14 - 22

stless + tab ... Type MyHomePage

Remove the error on line 17

Stateful Widget

Stateful Widget is a widget that stores variable (state).

This widget will rebuild itself whenever there is a change of its state (variable)

For example when user interact with a button, you might change the state/variable within the widget => Widget will be refreshed.

```
class MyWidget extends StatefulWidget {  
  
  @override  
  
  _MyWidget createState() => _MyWidget();  
  
}  
  
class _MyWidget extends State<MyWidget>{  
  
  @override  
  
  Widget build(BuildContext context){  
  
    return Container();  
  
  }  
  
}
```

Demo - BMI Calculator


BMI Calculator DEBUG

BMI Calculator

We care about your health

Height 170.00 (cm)

Weight 50.00 (kg)

 Calculate

Demo

We will create a simple BMI calculator app that will calculate BMI based on height and weight entered by user.

- An application using stateful widget since we are storing height, weight and bmi
- Create the structure using scaffold
- Add Scrollview
- The a Column with:
 - Image (logo of our app)
 - App title and subtitle
- Two containers containing slider for user to choose height and weight
- Button when the button is pressed you will do the BMI calculation

Path to learn to build mobile app

- How to create UI element (focus on one page first) - button, text, imageview...
- Navigation, multiple page = Stack, Tab, Drawer
- Passing data from one page to another page (forward pass, backward pass)
- Showing data on a List
- Retrieving data from Internet (**GET**, POST, API with Header)
- Storing data in local storage/Shared Preference
- Use device features : Camera, Geolocation, Social Sharing, Photo Library
- Improve architecture (Redux - React Native, Provider- Flutter)
- Finetune - Localization
- App submission

Create a weather app using API from Open Weather that will show weather based on Geolocation

Contact me

The Moose Academy

Common Room Bangi

Wan Muzaffar Wan Hashim (LinkedIn)

Notes

During the QnA Session I mistakenly quote price for Apple is 99USD per month, it is actually 99USD per year.

iOS : 99 USD per year - Need a MAC or Rent macbook on cloud ->
Apple Developer Account

Android - 25 USD per life time -> PC or MAC -> Google Developer
Account