Week 9 : Widgets and Layouts

ผศ. ดร. เก็จแก้ว ธเนศวร

kejkaew.tha@mail.kmutt.ac.th

ครั้งที่	วันที่	หัวข้อ
6	21/2/2566	Flutter – Basics of Dart Programming
7	28/2/2566	Article: Methodology + submit background
8	7/3/2566	Flutter – OOP and AppBar Widget
9	14/3/2566	Flutter – More Widgets and Layouts
10	21/3/2566	Article: Experiment and results + submit methodology
12	28/3/2566	Flutter - Animations and Graphics
11	4/4/2566	Article: Conclusions and future work + submit experiment and results
13	11/4/2566	ไม่มีเรียน (GDM443/DMT443)
14	18/4/2566	Flutter - Data Storage and Management
15	25/4/2566	Article: Abstract
16	2/5/2566	Flutter - Data Storage and Management (2)
17	9/5/2566	ส่ง research paper
18	16/5/2566	Flutter - Deploying Flutter Applications
19	23/5/2566	Flutter - ส่งโปรเจค

Last week

OOP

AppBar widget

Today outline

ListView

DrawerHeader widget

UserAccountsDrawerHeader

GestureDetector

Text widget

Layout widgets

Assets and Images

Widgets

MaterialApp class

Flutter has widgets specific to a particular platform - Android or iOS.

Android specific widgets are designed in accordance with Material design guideline by Android OS.

□ Android specific widgets are called as Material widgets.

iOS specific widgets are designed in accordance with Human Interface Guidelines by Apple and they are called as Cupertino widgets.

New Flutter project in VS Code

Open the Command Palette (Ctrl+Shift+P on Windows หรือ Cmd+Shift+P on macOS).

Select the ”Flutter: New Project” command and press Enter.

Select ”Application” and press Enter.

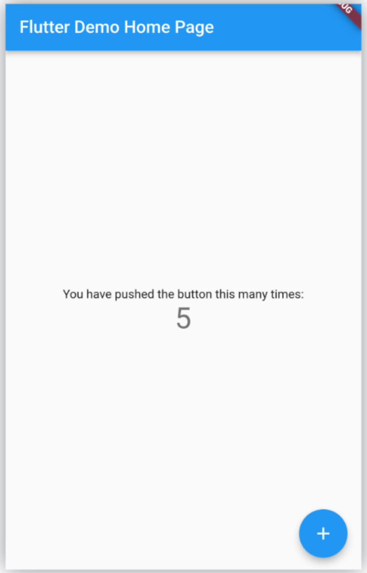
Select a Project location.

Enter your desired Project name.

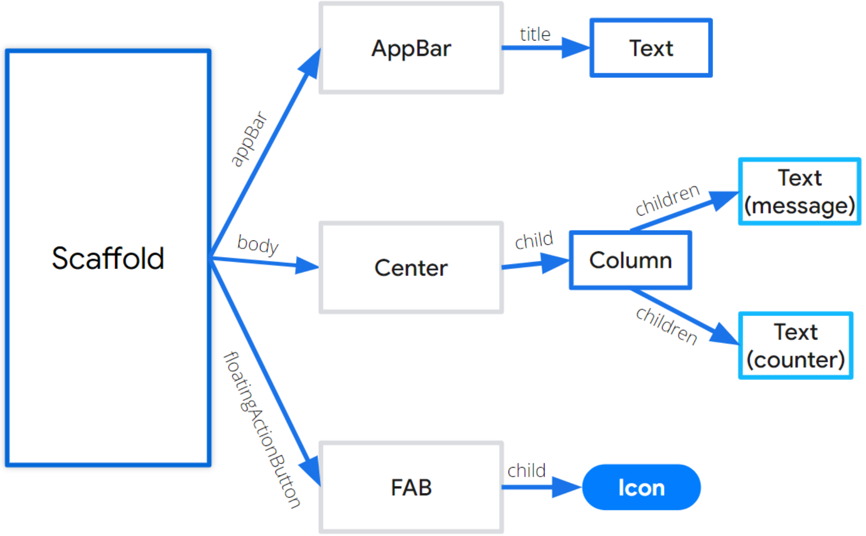
สร้างแล้วจะ setting file ที่จะเป็น main() ใน launch.json file ของ vs code

Example (main.dart)

UI

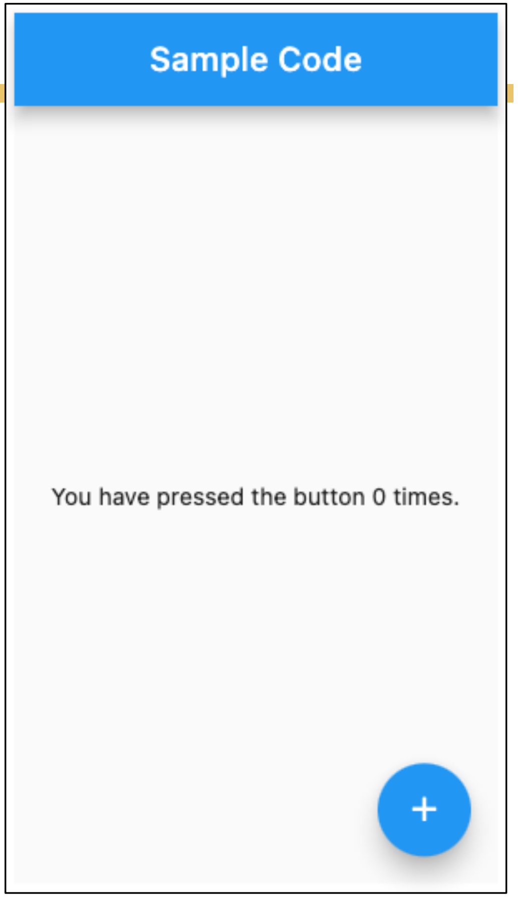


Breakdown



Scaffold

Scaffold is a class in flutter which provides many widgets or we can say APIs like Drawer, Snack-Bar, Bottom-Navigation-Bar, Floating-Action-Button, App-Bar, etc.

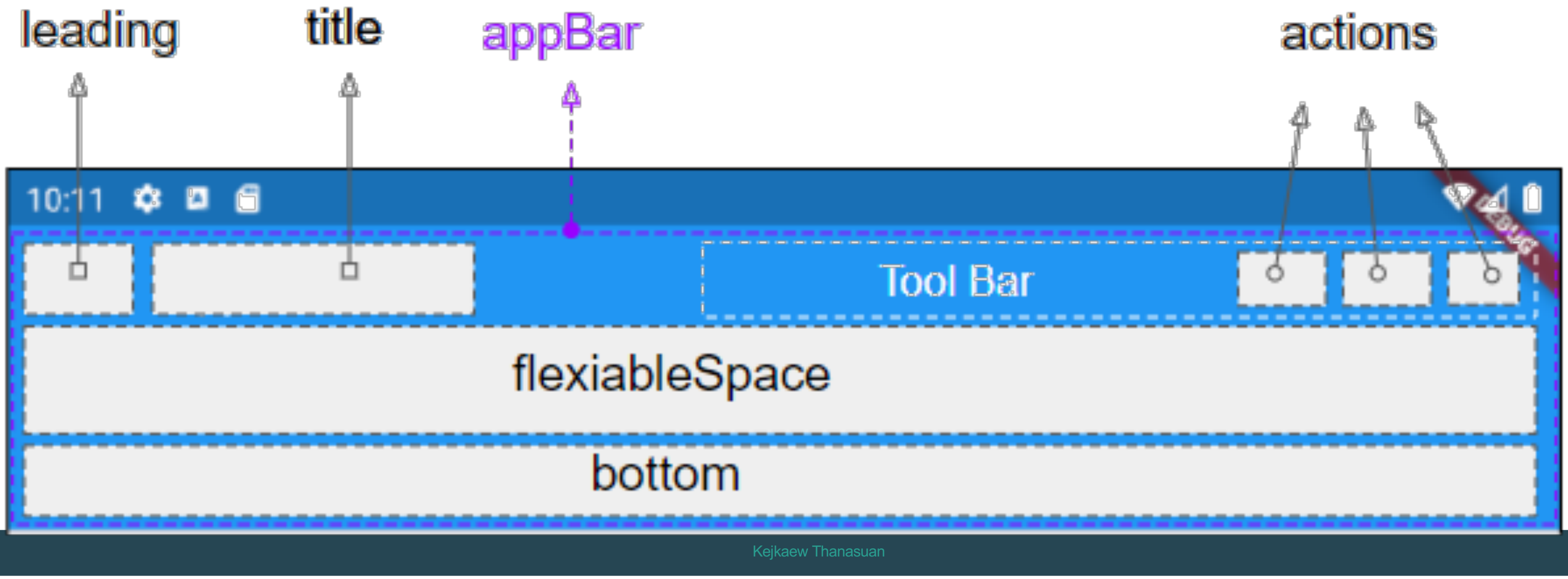


Sample Code

You have pressed the button 0 times.

AppBar: areas

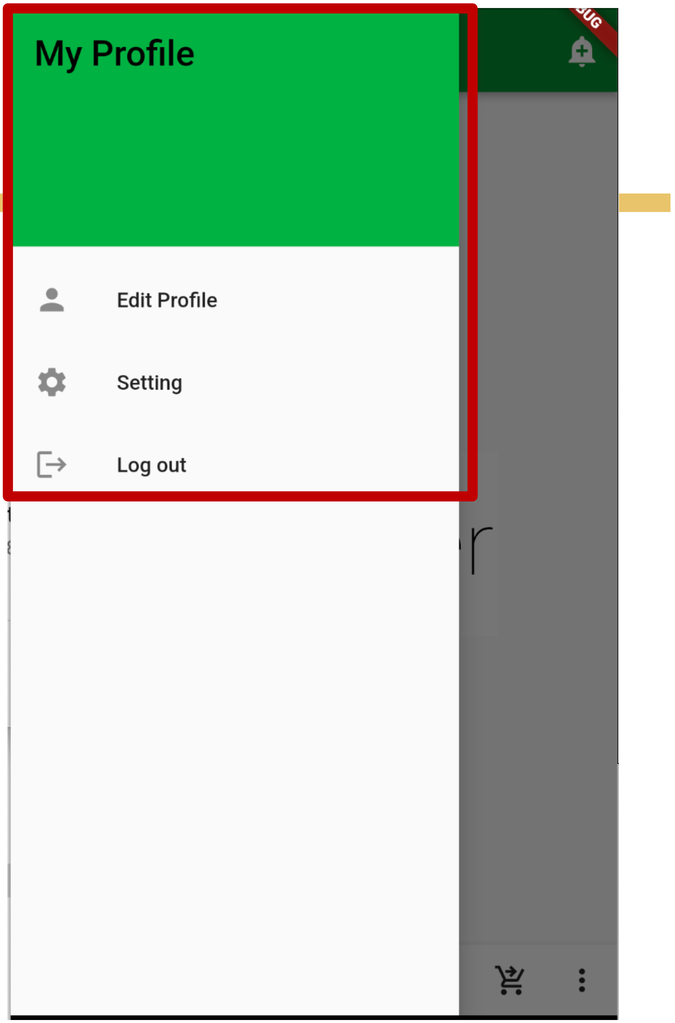
AppBar is divided into five areas, leading, title, Tool Bar (actions), flexiableSpace, and bottom.



ListView

ListView is a scrollable list of widgets arranged linearly.

It displays its children one after another in the scroll direction i.e, vertical or horizontal.



My Profile

Edit Profile

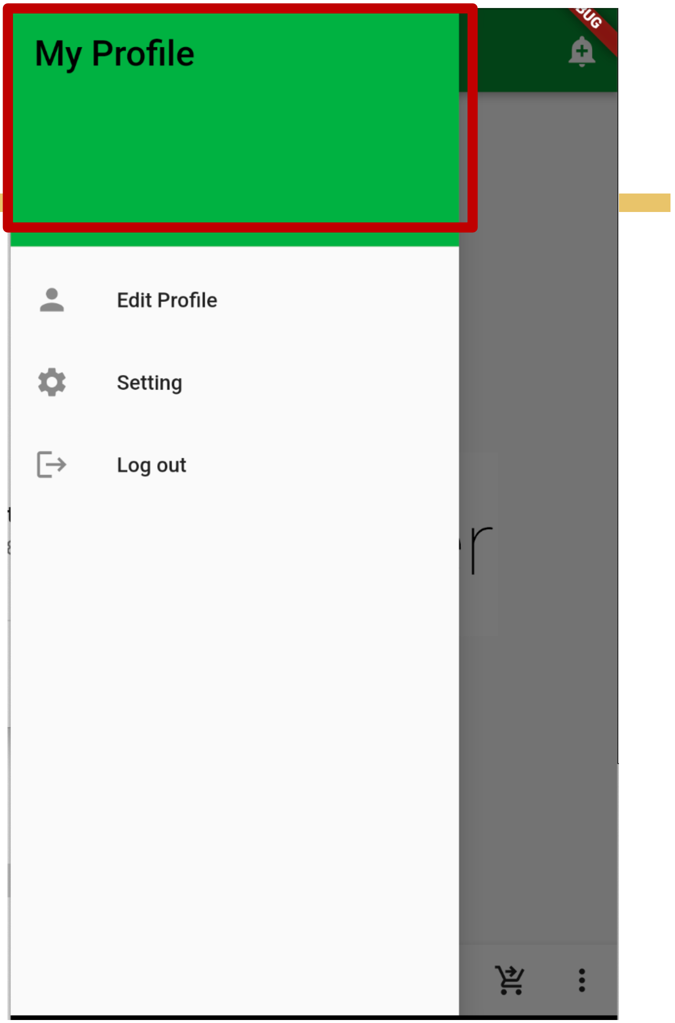
Setting

Log out

DrawerHeader widget

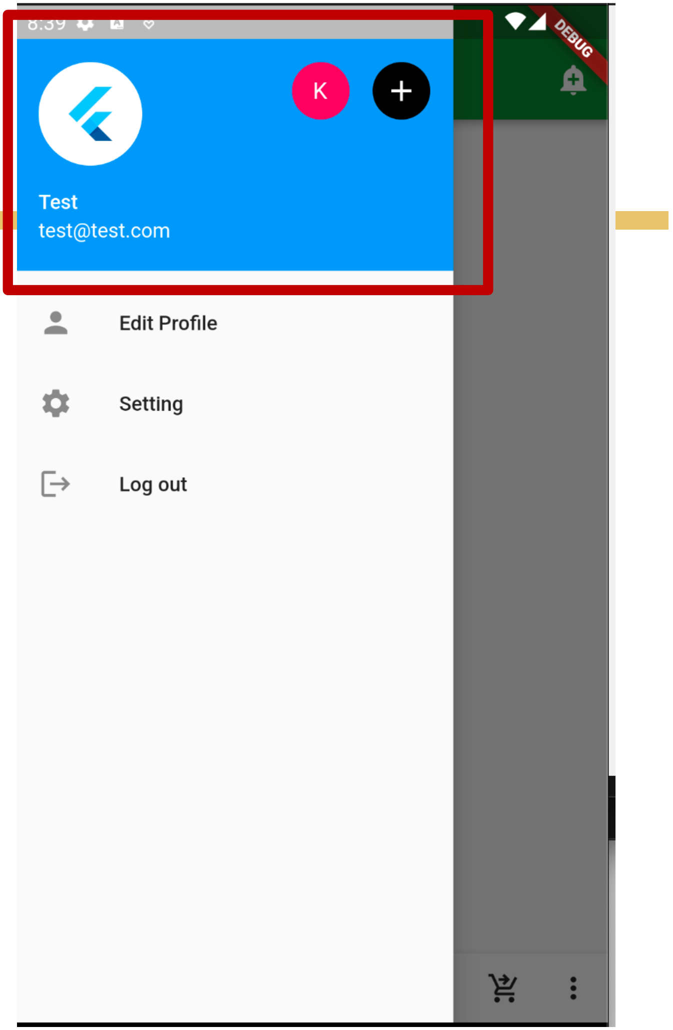
The top-most region of a Material Design drawer.

The header's child widget, if any, is placed inside a Container whose decoration can be passed as an argument, inset by the given padding.



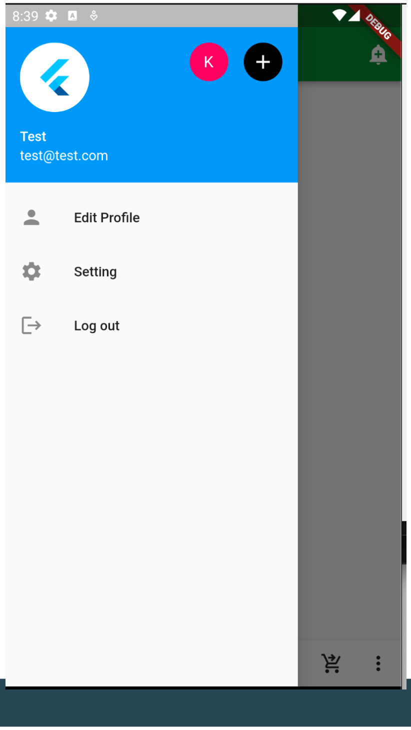
UserAccountsDrawerHeader

A Material Design Drawer header that identifies the app's user.



Activity 1

สร้าง UserAccountsDrawerHeader แทน drawerHeader โดยใช้ตัวอย่างโค้ดที่กำหนดให้

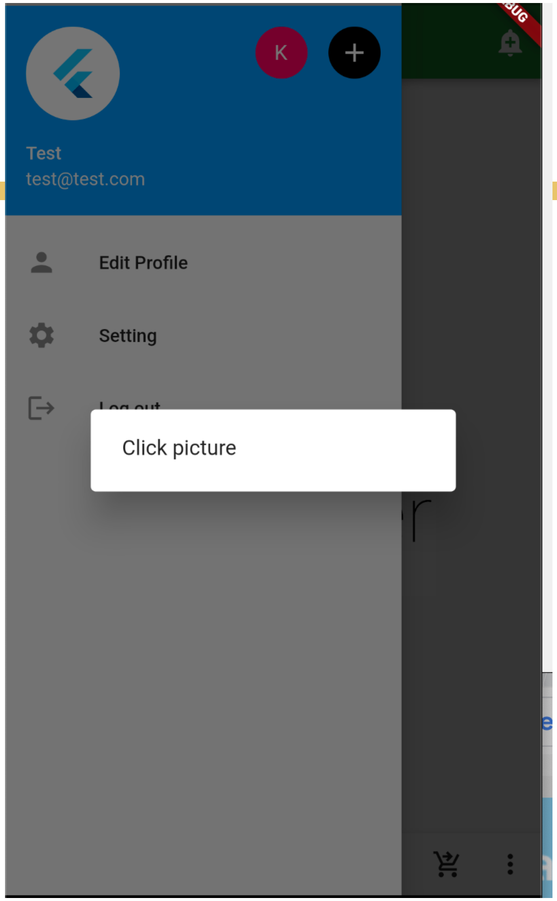


GestureDetector class

A widget that detects gestures.

Attempts to recognize gestures that correspond to its non-null callbacks.

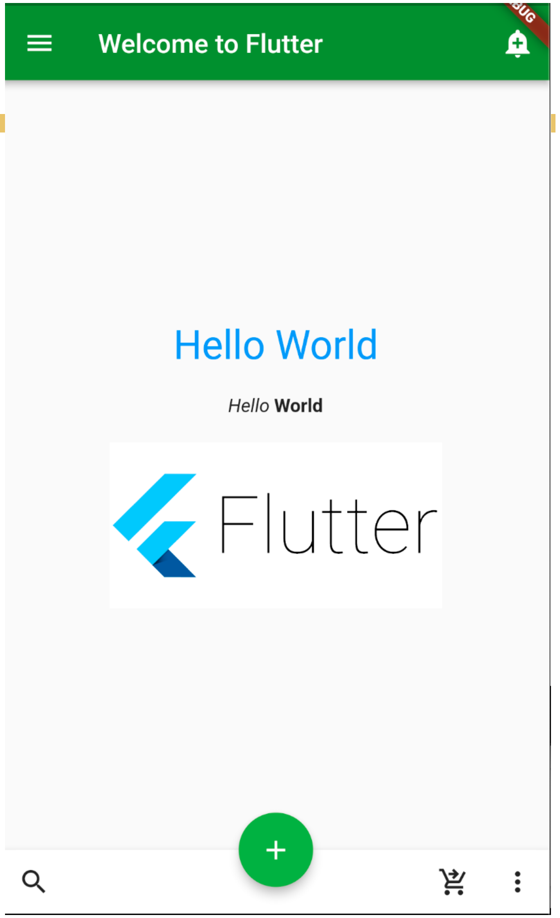
If this widget has a child, it defers to that child for its sizing behavior. If it does not have a child, it grows to fit the parent instead.



Text widget

Text widget is used to display a piece of string. The style of the string can be set by using style property and TextStyle class.

Text widget has a special constructor, Text.rich, which accepts the child of type TextSpan to specify the string with different style.



Text widget properties

style, TextStyle – Specify the style of the string using TextStyle class

textAlign, TextAlign – Alignment of the text like right, left, justify, etc., using TextAlign class

textDirection, TextDirection – Direction of text to flow, either left-to-right or right-to-left

Layout widgets

Layout widgets can be grouped into two distinct category based on its child:

Widget supporting a single child

Widget supporting multiple child

Layout: single child widget

Single child widgets are great options to create high quality widget having single functionality such as button, label, etc.

Center widget just centers it child widget with respect to its parent widget.

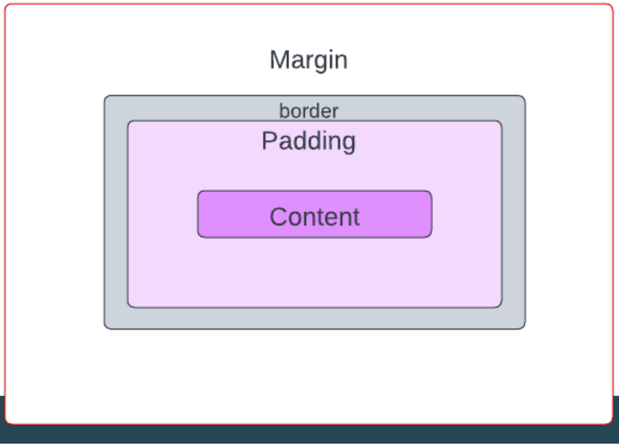
Container widget provides complete flexibility to place it child at any given place inside it using different option like padding, decoration, border, margin etc.,

Layout: single child widget

Padding – Used to arrange its child widget by the given padding. Padding can be provided by EdgeInsets class.

Margin - It adds an empty space around the Container.

Align – Align its child widget within itself using the value of alignment property.



Multiple Child Widgets

Widget will have more than one child widgets and the layout of each widget is unique.

Frequently used widgets

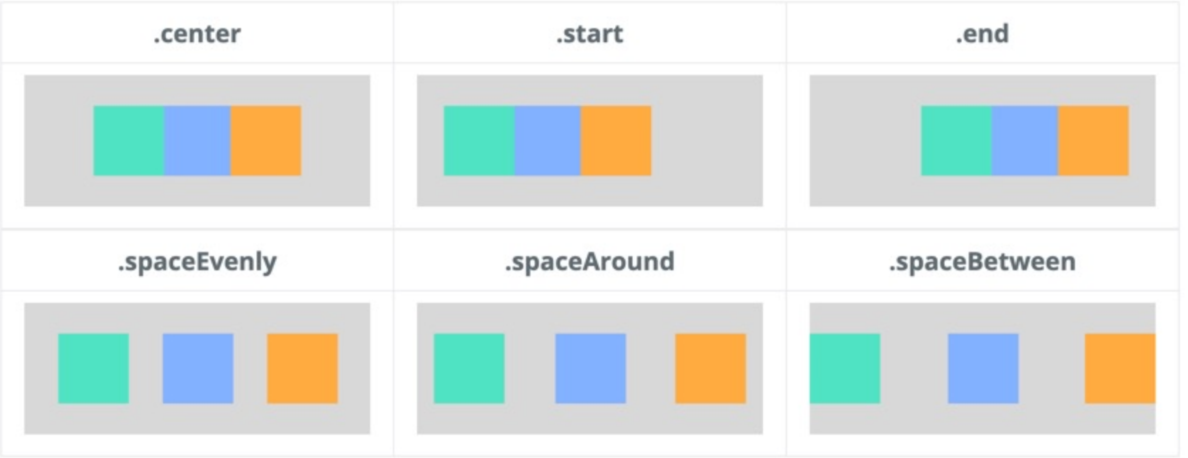
Row – Allows to arrange its children in a horizontal manner.

Column – Allows to arrange its children in a vertical manner.

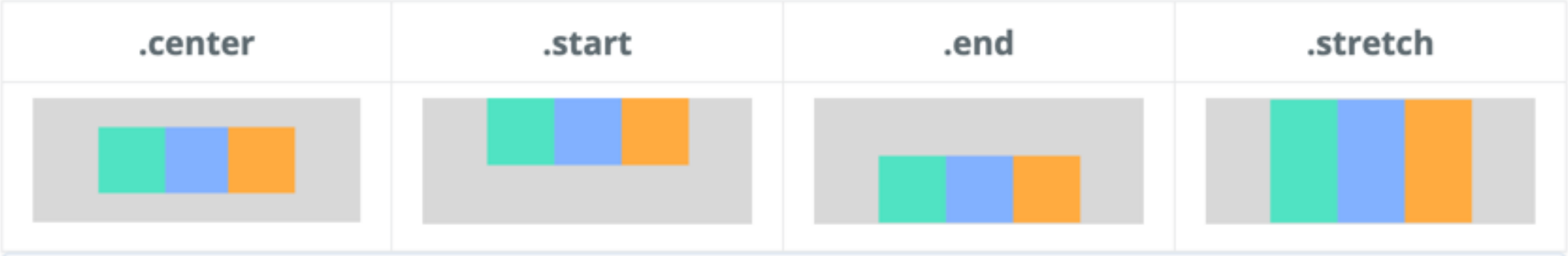
ListView – Allows to arrange its children as list.

Row Properties

MainAxisAlignment

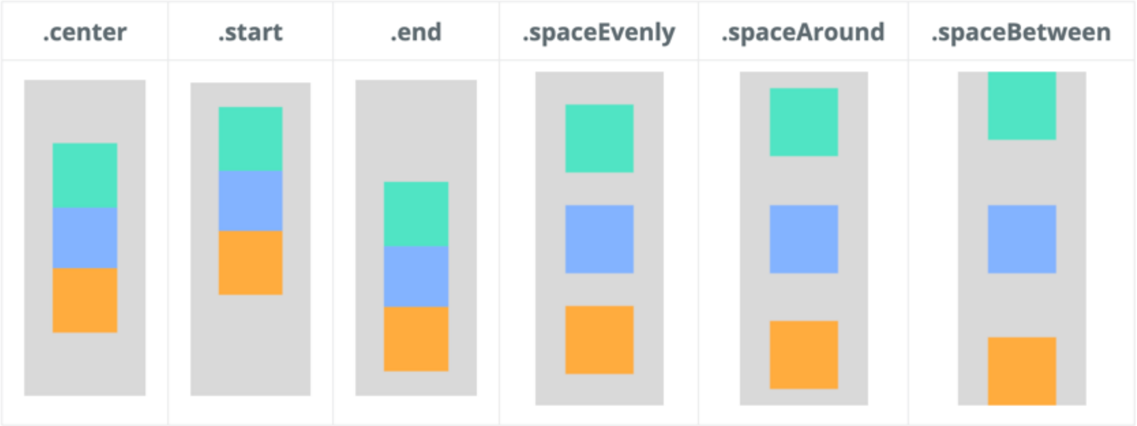


CrossAxisAlignment

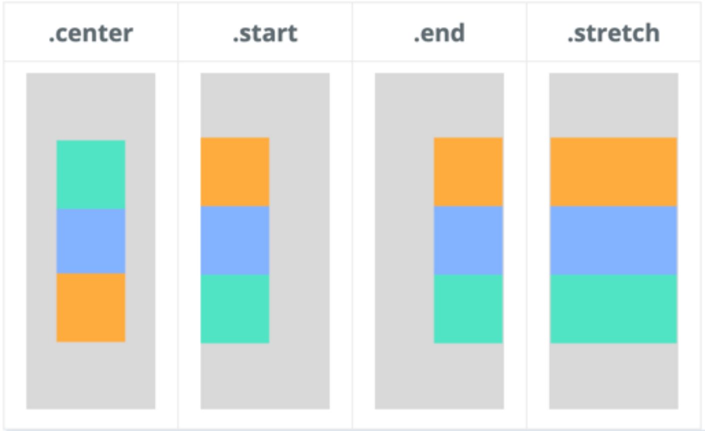


Column Properties

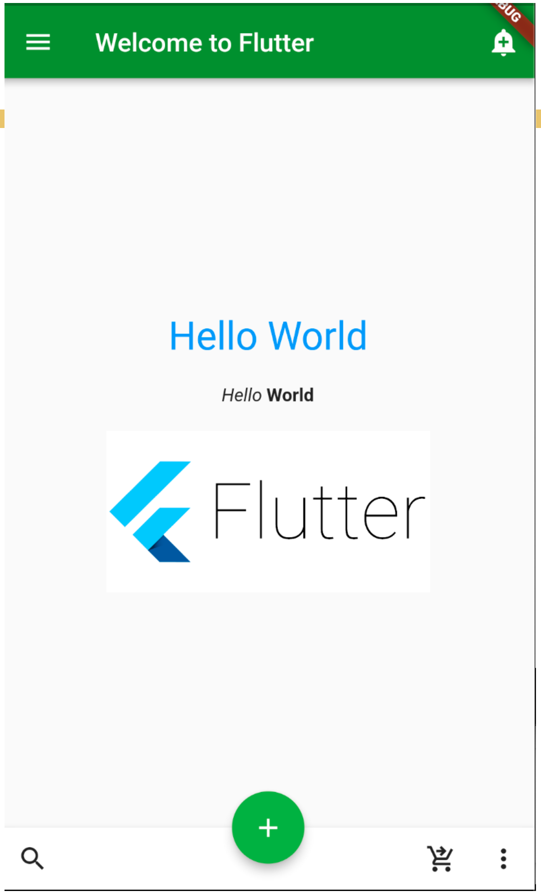
MainAxisAlignment



CrossAxisAlignment



Example



Activity 2

สร้าง Column Widget ที่ประกอบด้วย containers หลายๆ ตัวเป็น children list ในส่วนของ body โดยใช้โค้ดต่อไปนี้ เกิดอะไรขึ้น

ถ้าเปลี่ยนเป็น ListView Widget แทน column จะเกิดอะไรขึ้น

Assets and Images

Flutter apps can include both code and *assets* (sometimes called resources).

An asset is a file that is bundled and deployed with your app, and is accessible at runtime.

Common types of assets include static data (for example, JSON files), configuration files, icons, and images (JPEG, WebP, GIF, animated WebP/GIF, PNG, BMP, and WBMP).

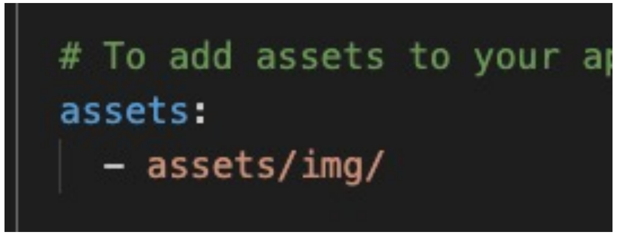
Specifying assets

Flutter uses the pubspec.yaml file, located at the root of your project, to identify assets required by an app.

flutter:

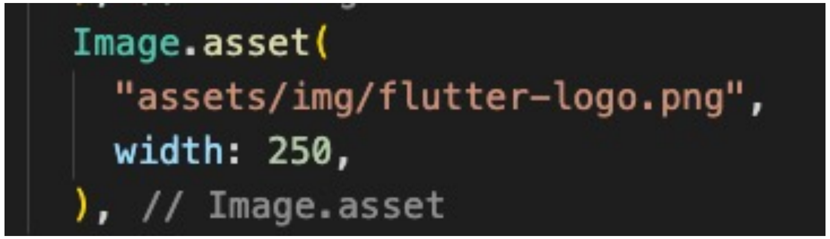
assets:

- directory/
- directory/subdirectory/

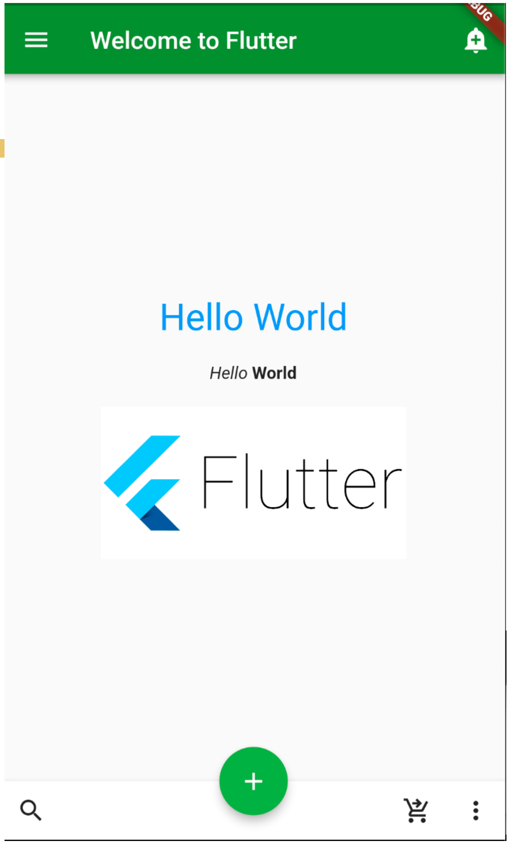


```
# To add assets to your ap
assets:
  - assets/img/
```

Example: Loading image



```
Image.asset(
  "assets/img/flutter-logo.png",
  width: 250,
), // Image.asset
```

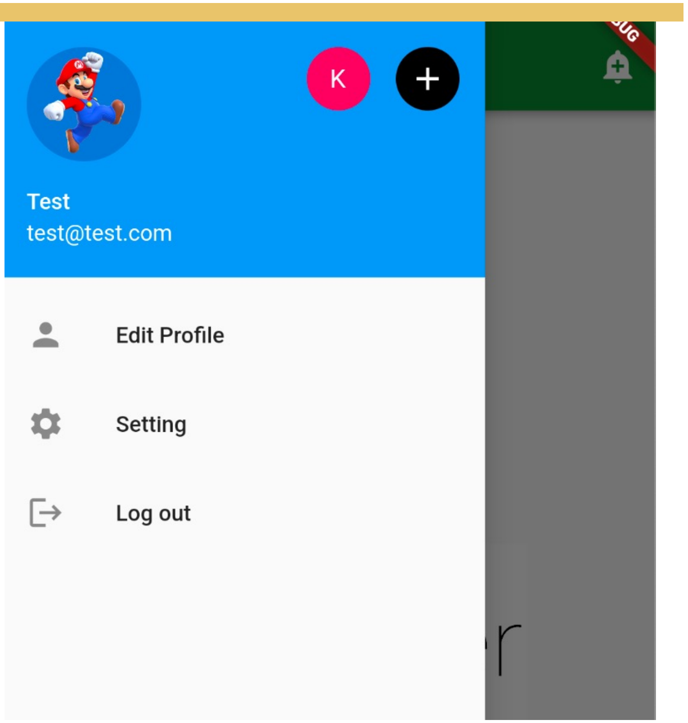


Display images from the internet

Displaying images is fundamental for most mobile apps.

Flutter provides the Image widget to display different types of images.

To work with images from a URL, use the Image.network() constructor.



Activity 3

เพิ่มรูปภาพในส่วน body โดย เอาไว้ด้านบนของคำว่า Hello world ตามตัวอย่าง

โดยใส่ padding ด้วย กำหนดให้

```
Padding(padding:
   EdgeInsets.only(bottom:
   20.0),child: …ใส่รูปนะ..)
```

Week 9: Classroom game

ตอบคำถามตาม link นี้เลย…

https://forms.gle/5mdXrVdgYQ95gzT19

Reference

https://dart-tutorial.com/introduction-and-basics/

https://www.tutorialspoint.com/flutter/flutter_introduction_to_widgets.htm

https://docs.flutter.dev/development/ui/widgets-intro