

# Welcome to the **Flutter** Workshop



# Workshop Plan :

1. Installation
2. Introduction
3. Dart & OOP
4. Flutter in hand
5. Flutter Widgets
6. Lets Code

# Introduction

# What is Flutter ?

Flutter is an open source UI framework by Google for building multi-platform applications from a single codebase.

# What is Flutter ?

Flutter is an open source UI framework by Google for building multi-platform applications from a single codebase.

# FrontEnd



Used technologies:

ReactJs , AnuglarJs....(web)

Java/kotlin (Android)

Flutter (crossPlatform)

....

# BackEnd



Used technologies:

Django (python)

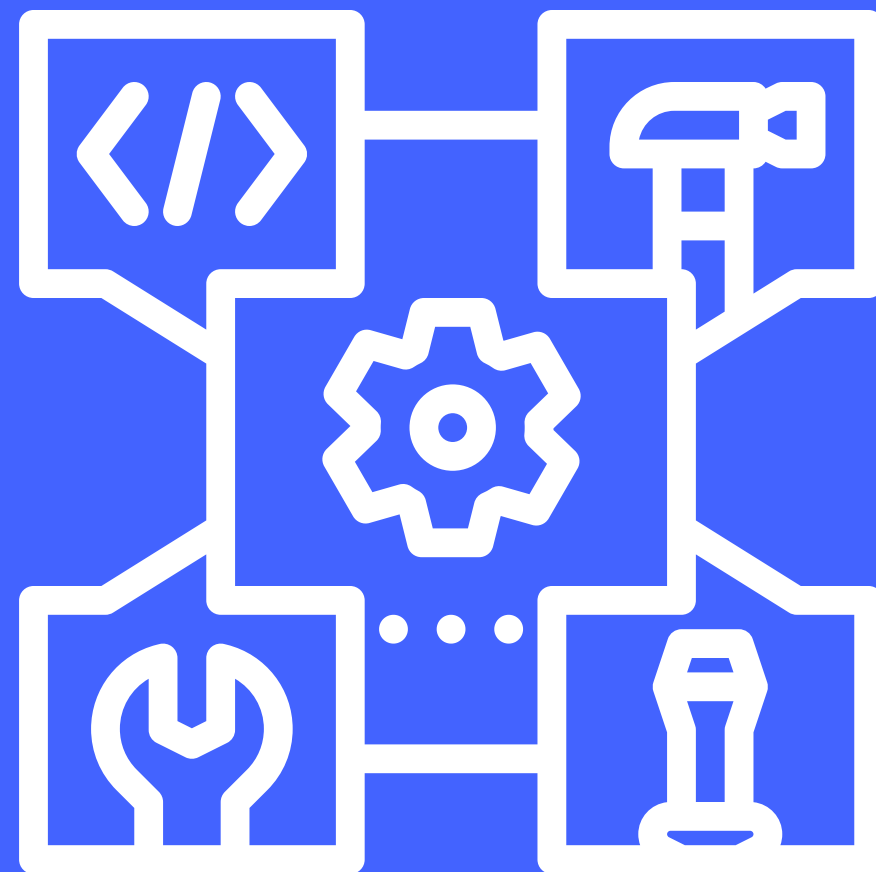
Node js (java script)

Laravel (php)

....

# What is a Framework ?

A framework is a structure that you can build software on. It serves as a foundation, so you're not starting entirely from scratch. Frameworks are typically associated with a specific programming language and are suited to different types of tasks.



# How multi-platform ?

## Native

Android 

java  / kotlin 

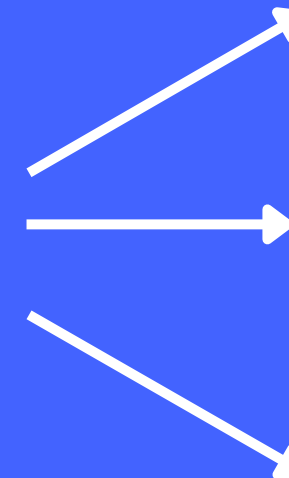
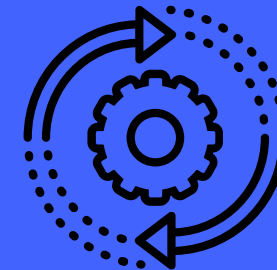
iOS 

swift 

## Multi-Platforme



Dart 



Mobile



Web



Desktop



 Flutter



# What makes Flutter Unique ?

Fast Development

Flexible Toolkit

Native Performance

# What should I know to start learning Flutter?

- Basic programming skills
- Dart Language
- Some understanding of Object-Oriented Programming (oop)

# Introduction to Dart Language



# Variables and Types

```
bool clicked = false; // boolean type can be true or false
const double pi = 3.14159265359; // const keyword
int cpt = 34;
double height = 21.75;
String name = "Yazid";
List<int> numbers = [12, 16, 21, 39];
Map<String, int> ages = {"Yacine": 20, "Mazene": 21}; // stores key-value pairs
```

# Loops and Conditional statements

## //For loop

```
for ( int i = 0 ; i<10 ; i++){  
    print(i);  
}
```

## //If else statement

```
for ( int i = 0 ; i<10 ; i++){  
    if (i % 2 ==0)  
    {  
        print("$i est pair");  
    }  
    else {  
        print("$i est impair");  
    }  
}
```

## //For in loop

```
List nombrePair= [0,2,4,6,8];  
  
for (int i in nombrePair){  
    print(i);  
}
```

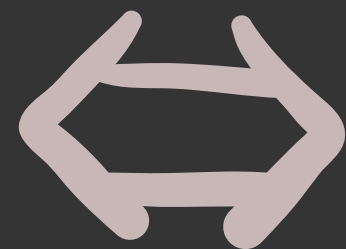
## //While loop

```
int i=0;  
while (i < 10){  
    print(i);  
    i++;  
}
```

# Ternary operator

Condition ? action if true : action if false;

```
for ( int i = 0 ; i<10 ; i++){  
    if (i % 2 ==0)  
    {  
        print("$i est pair");  
    }  
    else {  
        print("$i est impair");  
    }  
}
```



```
for ( int i = 0 ; i<10 ; i++){  
    i % 2 == 0 ? print("$i est pair") : print("$i est impair");  
}
```

# Fucntion and Arrow Syntaxe

```
bool isImpair(int n){  
    if(n % 2 == 0){  
        return false;  
    }  
    else{  
        return true;  
    }  
}
```

```
bool isImpairArrow(int n) => n % 2 != 0 ;
```

# Null Safety

Null Safety in simple words means a variable cannot contain a 'null' value unless you initialized with null to that variable



# OOP

Object Oriented programming (OOP) is a programming paradigm that relies on the concept of classes and objects rather than functions and logic

## **oop principals**

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

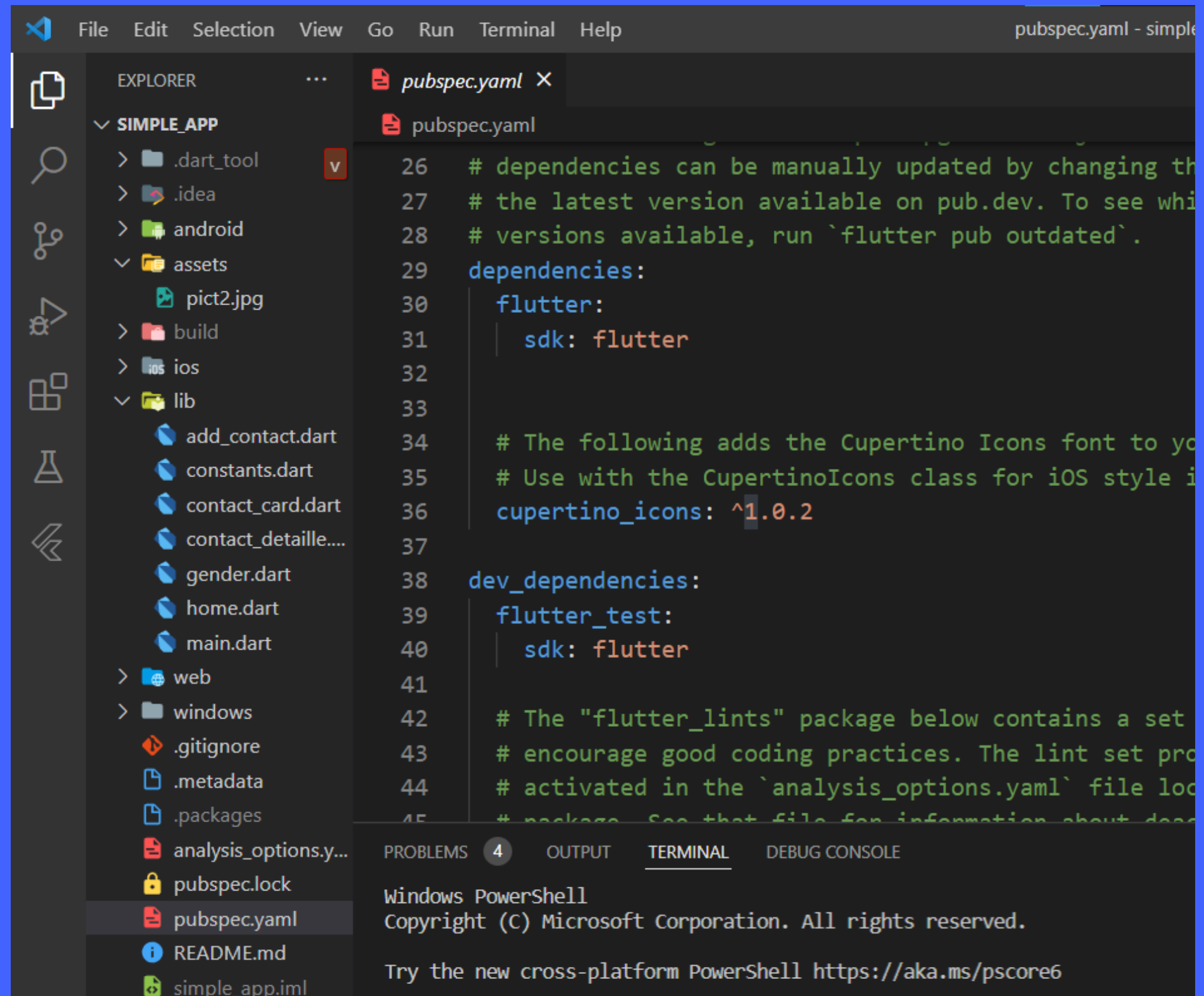
# **Flutter in hand**

```
$ flutter create my_project
```



# pubspec.yaml

contains metadata  
about the project that  
the Dart and Flutter  
tooling needs to  
know.



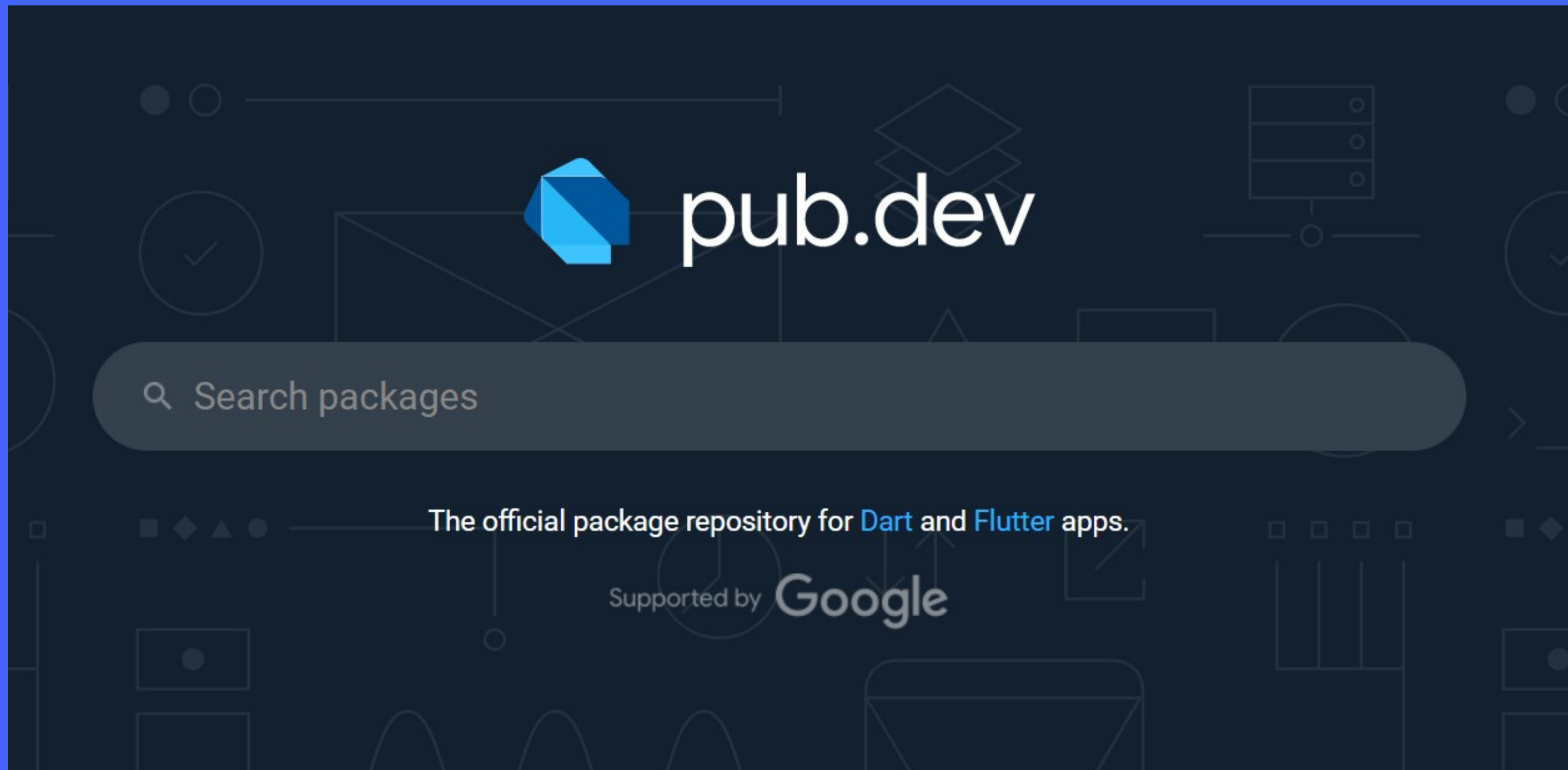
The screenshot shows an IDE window with the following components:

- EXPLORER:** A file tree for a project named `SIMPLE_APP`. It includes folders like `.dart_tool`, `.idea`, `android`, `assets` (containing `pict2.jpg`), `build`, `ios`, `lib` (containing `add_contact.dart`, `constants.dart`, `contact_card.dart`, `contact_detail.dart`, `gender.dart`, `home.dart`, `main.dart`), `web`, and `windows`. It also shows `.gitignore`, `.metadata`, `.packages`, `analysis_options.yaml`, `pubspec.lock`, `pubspec.yaml` (selected), `README.md`, and `simple_app.iml`.
- pubspec.yaml:** The main content of the file is visible, showing dependencies and dev\_dependencies. The visible code is:

```
26 # dependencies can be manually updated by changing the  
27 # the latest version available on pub.dev. To see which  
28 # versions available, run `flutter pub outdated`.  
29 dependencies:  
30   flutter:  
31     sdk: flutter  
32  
33  
34 # The following adds the Cupertino Icons font to your  
35 # Use with the CupertinoIcons class for iOS style icons  
36 cupertino_icons: ^1.0.2  
37  
38 dev_dependencies:  
39   flutter_test:  
40     sdk: flutter  
41  
42 # The "flutter_lints" package below contains a set  
43 # encourage good coding practices. The lint set provided  
44 # activated in the `analysis_options.yaml` file located  
45 # package. See that file for information about the
```
- Terminal:** At the bottom, the terminal shows the Windows PowerShell prompt and copyright information: "Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved. Try the new cross-platform PowerShell https://aka.ms/pscore6".

# Pubdev

It is the package manager for the Dart programming language, containing reusable libraries & packages for Flutter,



# StatelessWidget vs StatefulWidget

```
class Example extends StatefulWidget {  
  const Example({ Key? key }) : super(key: key);  
  
  @override  
  State<Example> createState() => _ExampleState();  
}  
  
class _ExampleState extends State<Example> {  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
  
    );  
  }  
}
```

```
class Exap1me extends StatelessWidget {  
  const Exap1me({ Key? key }) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
  
    );  
  }  
}
```

# Flutter Widgets





# The most important ones



& CSH

# Basic Widgets

- **Text**
- **Icon**
- **Image**
- **Container**
- **Center**
- **Padding**
- **SizedBox**



# Child property

```
return Container(  
  child: Text("Hello world"),  
); // Container
```



# Multi-child layout widgets



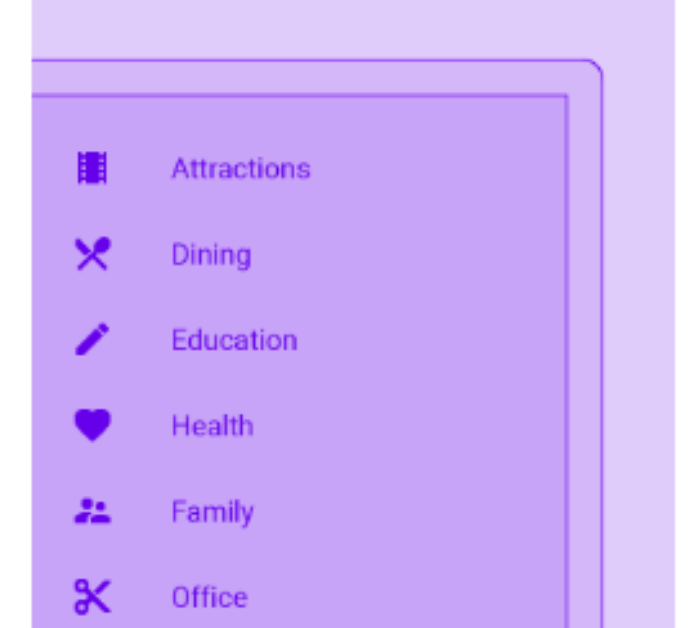
## Column

Layout a list of child widgets in the vertical direction.



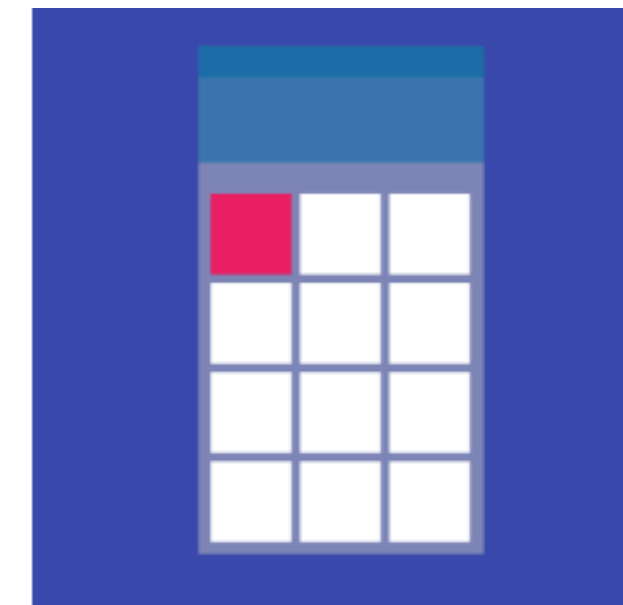
## Row

Layout a list of child widgets in the horizontal direction.



## ListView

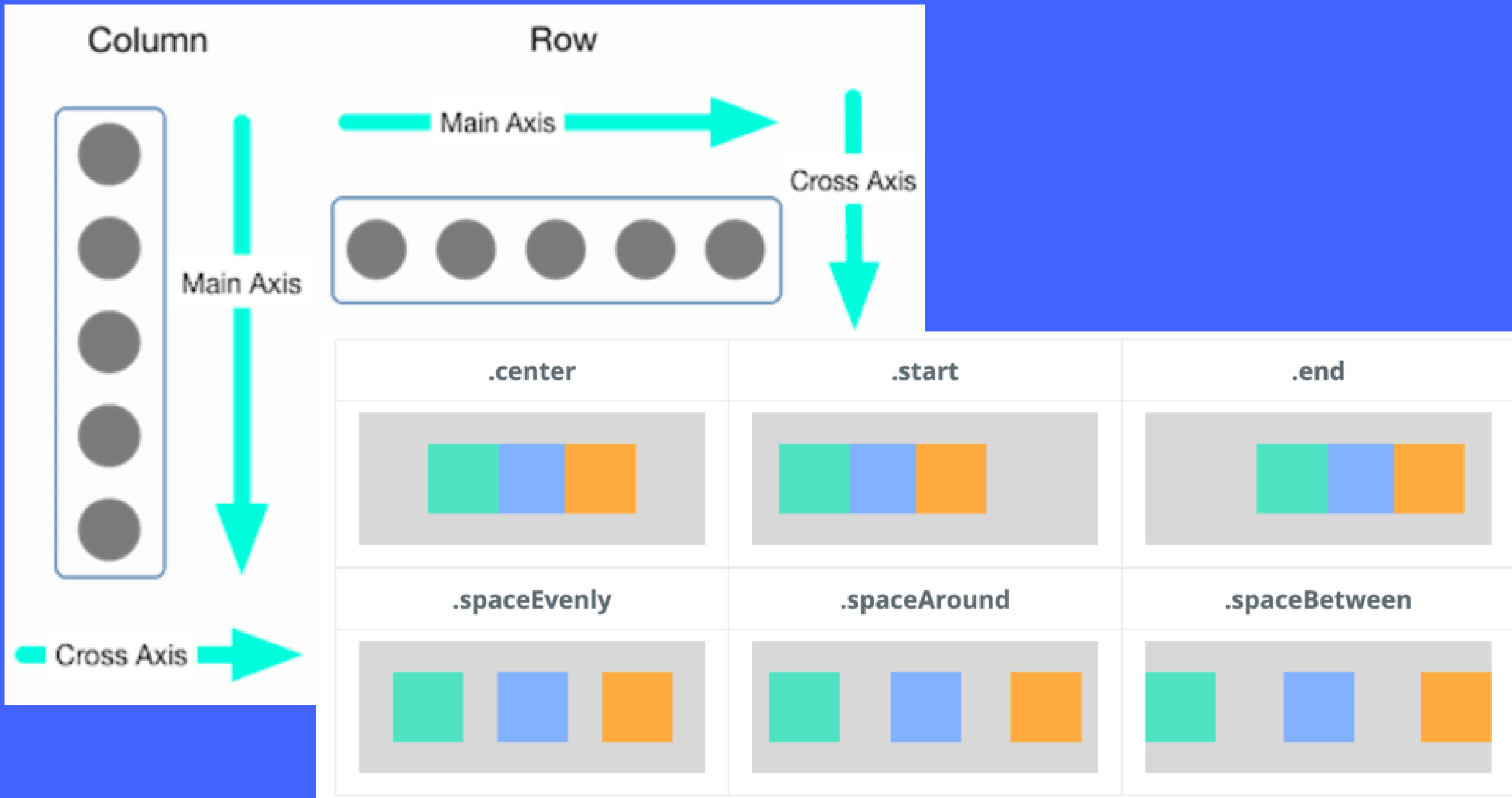
A scrollable, linear list of widgets. ListView is the most commonly used scrolling widget. It displays its children one after another in the scroll direction....



## GridView

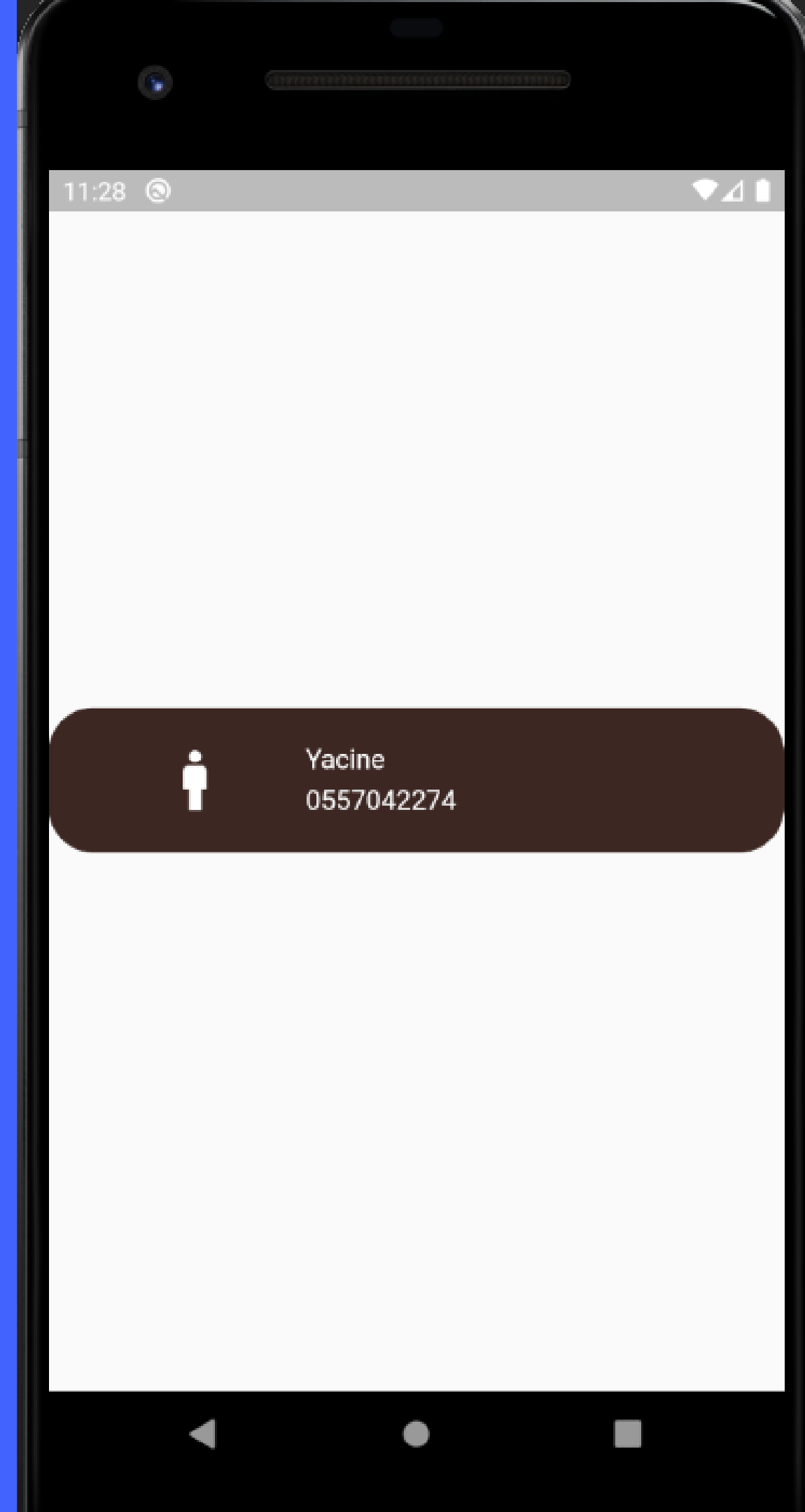
A grid list consists of a repeated pattern of cells arrayed in a vertical and horizontal layout. The GridView widget implements this component.

# Alignment in Column & Row



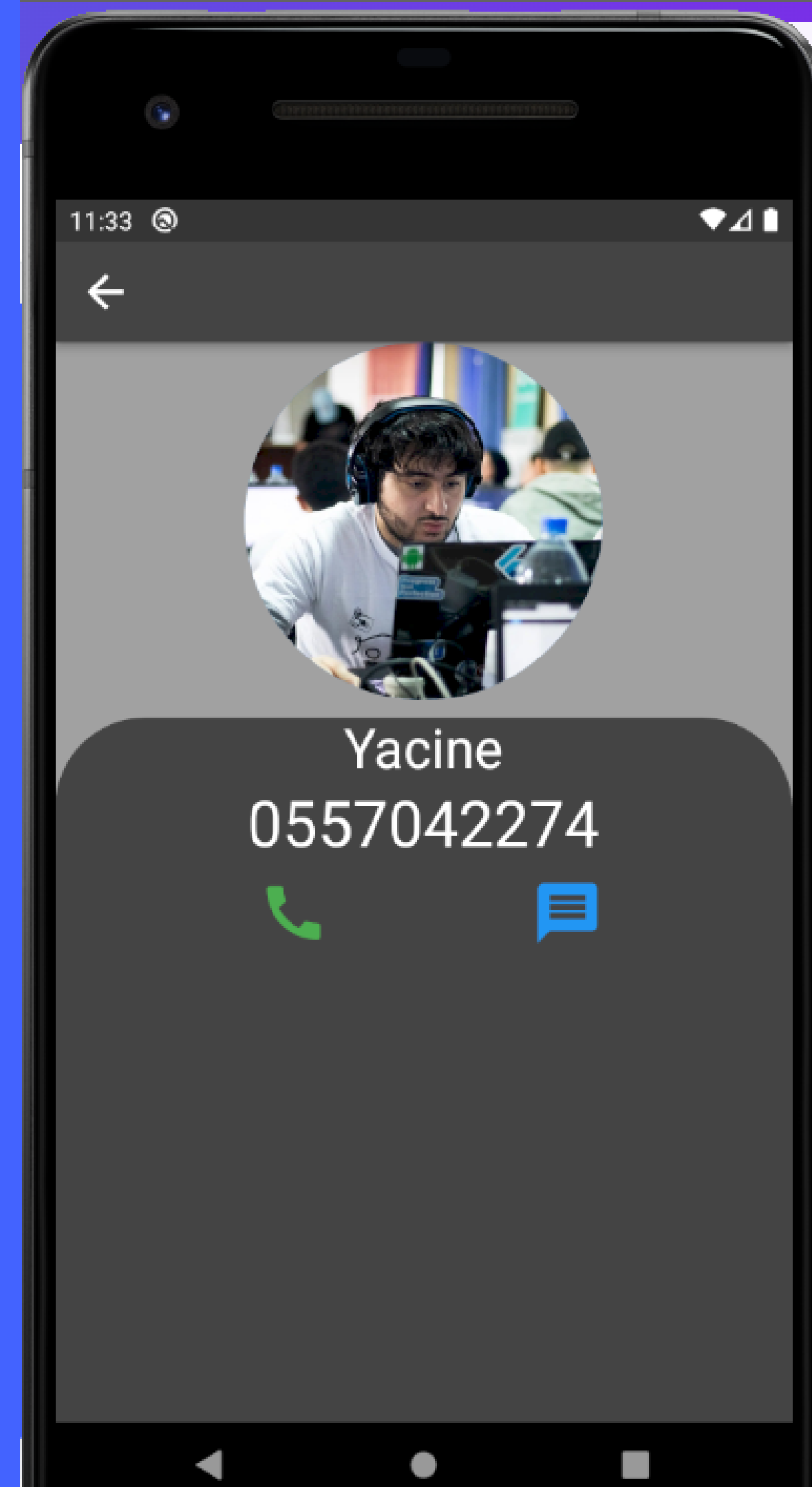
# Exercice 1

1. Which widgets are used in this screen?
2. Draw the widget tree.



# Exercice 2

1. Which widgets are used in this screen?
2. Draw the widget tree.



# Exercice 3

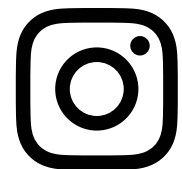
1. Which widgets are used in this screen?
2. Draw the widget tree.



# Resources

- Flutter official documentation
- The NetNinja ( Youtube )
- Reso Coder ( Youtube )
- Flutter Widget of the Week ( Youtube )

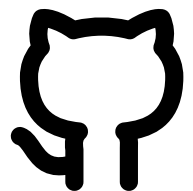
# Thank You



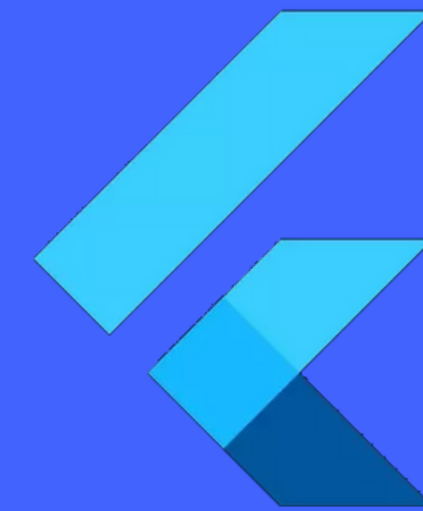
ahmed\_yacine\_bouchouareb



Ahmed Yacine Bouchouareb



chechna9



# Flutter