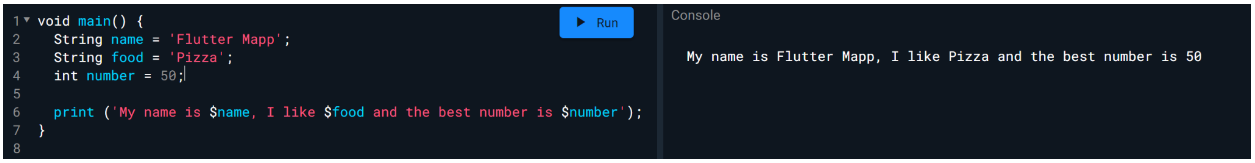
Learn Flutter with an unfair advantage

"If I had to learn Flutter from scratch once again, then this is what I would do." - Flutter Mapp

Level I: Dart basics:

- Go on DartPad (https://dartpad.dev/?) to practice.
- What you have to practice:
    1. Data Types
        a. Strings
        b. Int
        c. Double
        d. Bool
        e. Lists
        f. Maps
    2. Functions
        a. Looping
        b. Arguments (You could also say parameters)
        c. Conditions
    3. Classes
    4. Null Safety

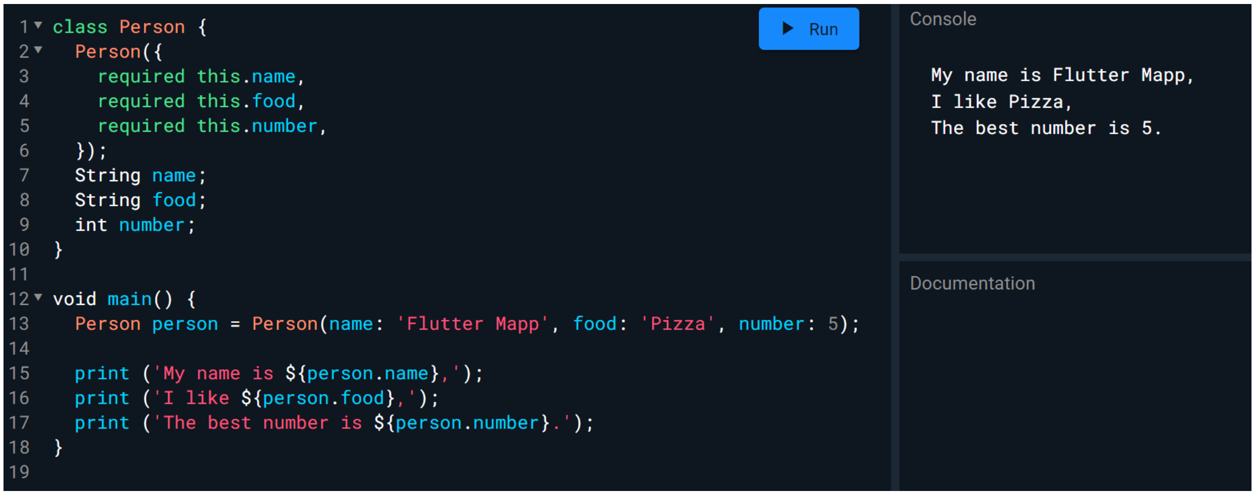
This is some examples:
(Data types)



```
void main() {
  String name = 'Flutter Mapp';
  String food = 'Pizza';
  int number = 50;

  print ('My name is $name, I like $food and the best number is $number');
}
```

```
My name is Flutter Mapp, I like Pizza and the best number is 50
```

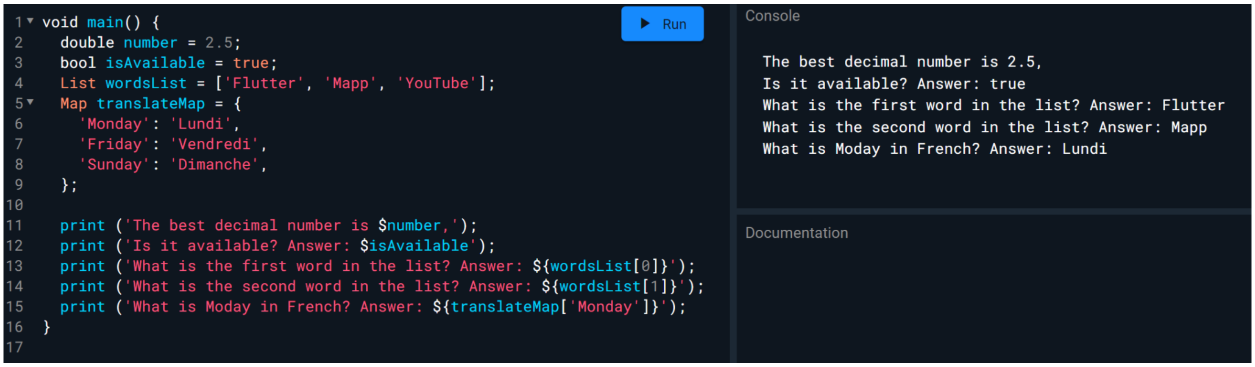
(Classes, arguments, data types)



```
class Person {
  Person({
    required this.name,
    required this.food,
    required this.number,
  });
  String name;
  String food;
  int number;
}

void main() {
  Person person = Person(name: 'Flutter Mapp', food: 'Pizza', number: 5);

  print ('My name is ${person.name},');
  print ('I like ${person.food},');
  print ('The best number is ${person.number}.');
}
```

```
My name is Flutter Mapp,
I like Pizza,
The best number is 5.
```

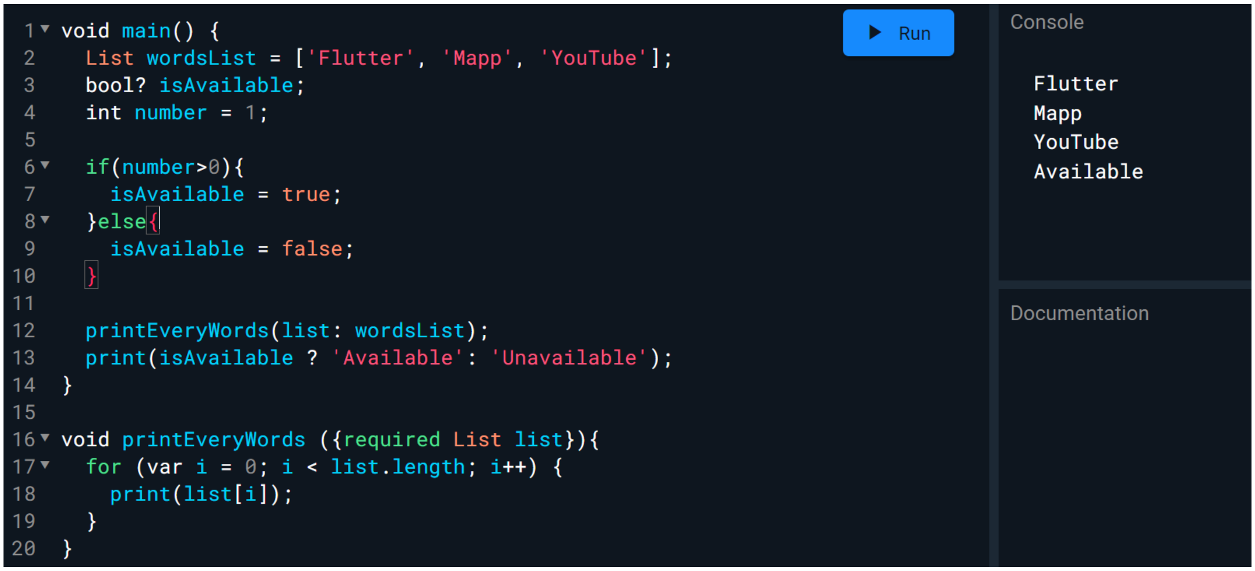
(Data types)



```
void main() {
  double number = 2.5;
  bool isAvailable = true;
  List wordsList = ['Flutter', 'Mapp', 'YouTube'];
  Map translateMap = {
    'Monday': 'Lundi',
    'Friday': 'Vendredi',
    'Sunday': 'Dimanche',
  };

  print ('The best decimal number is $number,');
  print ('Is it available? Answer: $isAvailable');
  print ('What is the first word in the list? Answer: ${wordsList[0]}');
  print ('What is the second word in the list? Answer: ${wordsList[1]}');
  print ('What is Moday in French? Answer: ${translateMap['Monday']}');
}

```

```
The best decimal number is 2.5,
Is it available? Answer: true
What is the first word in the list? Answer: Flutter
What is the second word in the list? Answer: Mapp
What is Moday in French? Answer: Lundi
```

(Function, looping, conditions, data types)



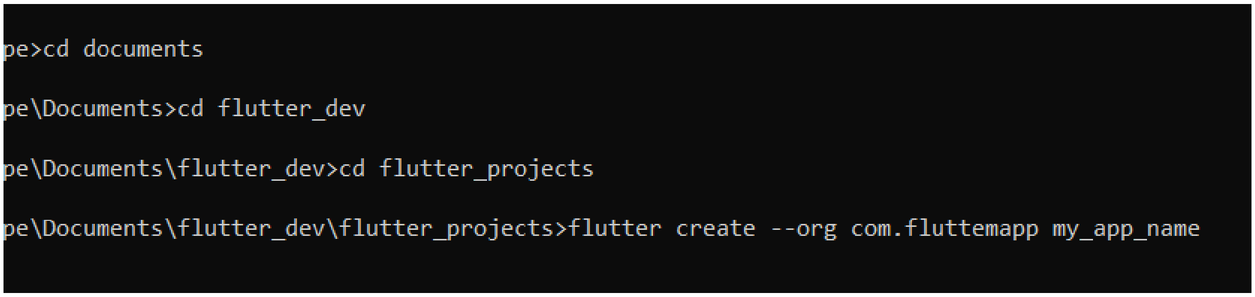
```
void main() {
  List wordsList = ['Flutter', 'Mapp', 'YouTube'];
  bool? isAvailable;
  int number = 1;

  if(number>0){
    isAvailable = true;
  }else{
    isAvailable = false;
  }

  printEveryWords(list: wordsList);
  print(isAvailable ? 'Available': 'Unavailable');
}

void printEveryWords ({required List list}){
  for (var i = 0; i < list.length; i++) {
    print(list[i]);
  }
}
```

```
Flutter
Mapp
YouTube
Available
```

Level II: Flutter basics:

- Install Flutter (This will be hard if it’s your first time)
    - Here is how to install (With a Flutter tutorial for beginners): https://www.youtube.com/watch?v=3XipSv2CWiQ

- Best way to create your Flutter Apps (Because this will set your application ID the right way):

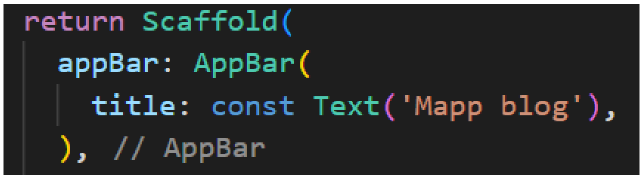


```
pe>cd documents

pe\Documents>cd flutter_dev

pe\Documents\flutter_dev>cd flutter_projects

pe\Documents\flutter_dev\flutter_projects>flutter create --org com.fluttemapp my_app_name
```

- Key concepts about Flutter
    - Flutter is Widget inside Widget
        - Add a Widget
        - Add an argument in the Widget (Ctrl+space) (Command+space for mac)
        - Add another Widget in the argument
        - Do this forever (Until your app is done)

    - Widgets **always** start with a **capital letter.**
    - Arguments **always** start with a **lowercase letter.**

    - In this example (Read carefully the capital and lowercase letters):
        - **S**caffold is a Widget
        - **a**ppBar is an argument of the **S**caffold Widget
        - **A**ppBar is a Widget placed inside the **a**ppBar argument.
        - **t**itle is an argument of the **A**ppBar Widget
        - **T**ext is a Widget placed inside the **t**itle argument.



```
return Scaffold(
  appBar: AppBar(
    title: const Text('Mapp blog'),
  ), // AppBar
```

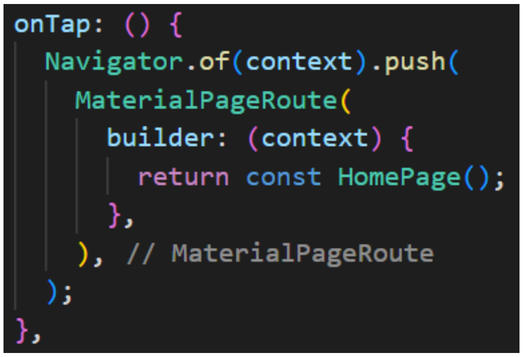
            - Remember, it’s Widget inside Widget!
            - Use this to see all the possible arguments available (Ctrl+space) (Command+space for mac)

- Learn about the Widgets
  - Basic Widgets:
    - StatelessWidget
    - StatefulWidget
    - MaterialApp
    - Scaffold
    - Container
    - Center
    - Appbar
    - Image (Asset or Network)
    - Column
    - Row
    - ElevatedButton
    - SizedBox
    - SingleChildScrollView
    - Icon(Icons.profile)
    - IconButton
    - Colors.blue
    - ListTile
    - Text
    - BottomNavigationBar
    - GestureDetector
  - Other Widgets you should know:
    - Padding
    - TextField
    - Wrap
    - List.generate
    - Drawer
    - DrawerHeader
    - SafeArea
    - CircleAvatar
    - CheckBoxListTile
    - MaterialBanner
    - TextButton
    - OutlinedButton
    - ListView
    - Divider
    - TextStyle
    - Visibility
    - TextFormField
    - SnackBar
  - Widgets you will eventually need (But it's okay if you don't get it now)
    - FutureBuilder
    - ValueListenableBuilder
    - ValueNotifier
    - StreamBuilder
    - Builder
    - LayoutBuilder
- States
  - StatelessWidget = Screen doesn't refresh
  - StatefulWidget = Screen can refresh
  - setState need to be used in the StatefulWidget in order to refresh the screen:

```
setState(() {});
```

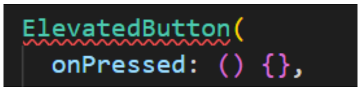
- Change Pages (Navigate in your app)
  - Your app will most likely have multiple pages. Use this to change pages:
    - push (add a page over this one)
    - pushReplacement (replace the current page with this new one)
    - pop (remove the current page)

- This is an example:



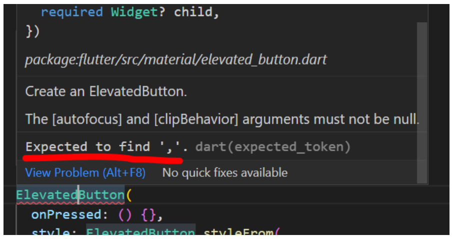
```
onTap: () {
  Navigator.of(context).push(
    MaterialPageRoute(
      builder: (context) {
        return const HomePage();
      },
    ), // MaterialPageRoute
  );
},
```

- How to solve bugs/errors
    - If you have a error like this (With red curly lines):



```
ElevatedButton(
  onPressed: () {},
```

        - Put your mouse over the Widget with the red curly line.
        - A box will appear, scroll down at the bottom of the box
        - Read the error
        - This is what you should see:



```
    required Widget? child,
  })
```

package:flutter/src/material/elevated_button.dart

Create an ElevatedButton.

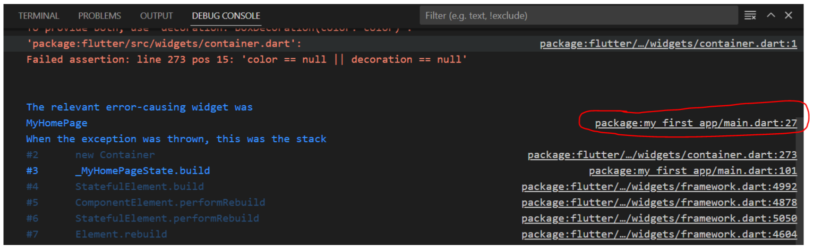
The [autofocus] and [clipBehavior] arguments must not be null.

```
Expected to find ','. dart(expected_token)
```

View Problem (Alt+F8)   No quick fixes available

```
ElevatedButton(
  onPressed: () {},
  style: ElevatedButton.styleFrom(
```

    - You can also have hidden errors (Sometime the screen will be red)



```
To provide both, use "decoration: BoxDecoration(color: color)".
'package:flutter/src/widgets/container.dart':                 package:flutter/…/widgets/container.dart:1
Failed assertion: line 273 pos 15: 'color == null || decoration == null'


The relevant error-causing widget was
MyHomePage                                                        package:my_first_app/main.dart:27
When the exception was thrown, this was the stack
#2      new Container                                     package:flutter/…/widgets/container.dart:273
#3      _MyHomePageState.build                                package:my_first_app/main.dart:101
#4      StatefulElement.build                            package:flutter/…/widgets/framework.dart:4992
#5      ComponentElement.performRebuild                  package:flutter/…/widgets/framework.dart:4878
#6      StatefulElement.performRebuild                   package:flutter/…/widgets/framework.dart:5050
#7      Element.rebuild                                  package:flutter/…/widgets/framework.dart:4604
```

        - Open the Debug Console in the terminal
        - Click the link on the top and it will bring you to your error in the code
        - If you can't understand, google search the error

- You should be able to build an UI with Flutter at this point! Congrats! 🎉

Level III: Practice Dart and Flutter basics stuff:

- Try to build any UI (User interface) app from scratch. You have all the knowledge at this point, what you need is practice.

- Example, you could make a simple Todo app

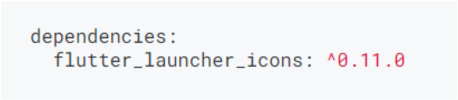
- Use Pinterest to get app design ideas (https://www.pinterest.ca/)

- It’s okay if the data is erased every time you close the app. We will fix this later. For now we just care about you being able to make a UI by yourself.

Level IV: Flutter advanced

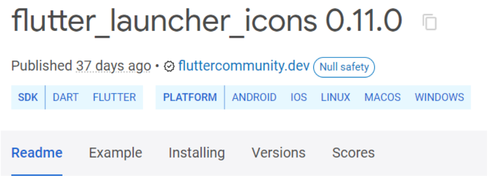
- We will focus ONLY on what you need to make your apps from scratch to published on the stores (App store/Play store)

- This is what you need:
  - **How to use packages**
    - Pub.dev (https://pub.dev/) is where you can find the Flutter packages
      - What is a package?
        - Code already written for you so you same a lot of time
        - Don’t really to much on this, but some are required
      - How to use it?
        - Add the dependencies in your pubspec.yalm file from your Flutter project.



```
dependencies:
  flutter_launcher_icons: ^0.11.0
```

        - More information available in the following tabs
          - Readme
          - Example
          - Installing

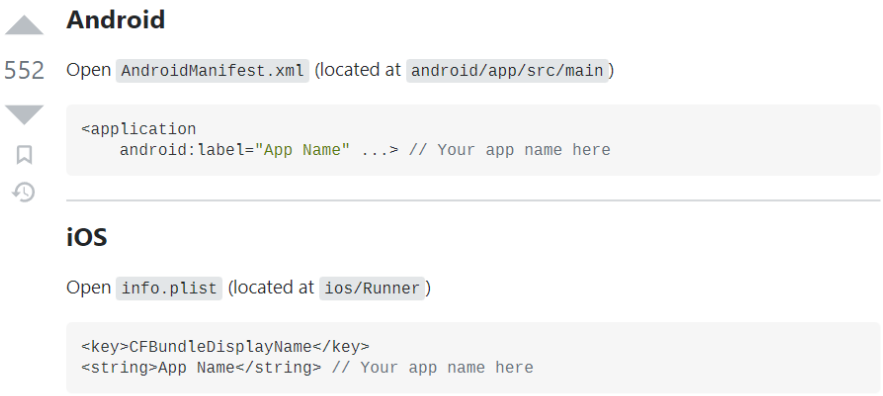


flutter_launcher_icons 0.11.0

Published 37 days ago • flutter community.dev Null safety

  - **Change the app logo**
    - Use this package: https://pub.dev/packages/flutter_launcher_icons

  - **Change the app name**
    - Change the following lines of code in your Flutter project



Android

Open `AndroidManifest.xml` (located at `android/app/src/main`)

```
<application
    android:label="App Name" ...> // Your app name here
```

iOS

Open `info.plist` (located at `ios/Runner`)

```
<key>CFBundleDisplayName</key>
<string>App Name</string> // Your app name here
```

- **Authenticate your users**
    - Use Firebase when you start
        - Firebase is made by Google just like Flutter. Both work really well together.
    - Use this package: https://pub.dev/packages/firebase_auth

- **Create a database** (Data that you can have with an internet connection)
    - Use Firebase also
    - Use this package: https://pub.dev/packages/cloud_firestore

- **Save data on the user's phone** (You can call this a Local database)
    - Use this package: https://pub.dev/packages/shared_preferences

- **State managements**
    - You don't need this
    - It's really useful when you want to manage data within your app, but as a beginner, don't waste time on this.
    - You can use the two following widgets and it will work to manage data within your app:
        - ValueListenableBuilder
        - ValueNotifier
    - One day, when you feel ready, you can check for state managements

- **Publish on Stores** (App store/Play store)
    - For Apple: https://appstoreconnect.apple.com/
    - For Android: https://play.google.com/console

- You should be able to build ANY Flutter apps now! Big congrats! 🎉🎉🎉

Level V: Practice Flutter advanced stuff

- First time you will use the following you won’t really understand:
    - How to use packages
    - Change the app logo
    - Change the app name
    - Authenticate your users
    - Create a database
    - Save data on the user’s phone
    - State managements
    - Publish on Stores

- So create more and more apps and eventually this will get easier.

- You don’t have to understand everything to start posting your first app.

Bonus: What you must know!

- There is not a perfect single app in this world. Everything is a constant iteration
    - The first car was not a Tesla, it was first a car pulled by horses, then a car with fuel, then better cars with fuel, you get it.
    - Life is about iterations. Create something, publish it fast. I wasted many years of my life because I didn’t understand this.
    - Check about (MVP) minimum viable products for more information. But again, don't waste too much time on this. Just make your app.

- User feedback > What you think your app needs.
    - Start your app with 1 feature (The most important one), then publish, ask feedback from users, then add the most wanted second feature.
    - Don’t waste time trying to publish the perfect app, otherwise you will never publish. Trust me.

- Create your app visual before you even start coding!
    - Start with 1 feature, use figma (https://www.figma.com/) to make the design of your single feature app first.
    - Keep in mind, just like your code, your design (UI) will never be perfect. And it’s okay. Follow the 80/20 rule. When you are 80% satisfied with your design, then it's good, you can move to coding.
        - What is the 80/20 rule?
            - 80% of your accomplishment comes from 20% of work.
            - The next 20% of your accomplishment will come for the next 80% or work.
    - Once you design is 80%, move to coding the UI
    - Trust me, you will save so much time splitting the design from the code.

- 10X your productivity
    - Deep Work is all you need.
        - What is deep working?
            - Remove all distraction for 90 minutes and work for real
        - What is working for real?
            - Shallow work = Sending emails, making a logo, finding a name for your app. (This is now real work, this is wasting your time)
            - Real work = The things that are actually hard to do. (Use your common sense)

- Commands

    Ctrl+space (check the available arguments/parameters)

    Ctrl+b (Hide/show the explorer)
    Ctrl+j (Hide/show the terminal)

    Ctrl+Maj+f (Make a search in every files)
    Ctrl+f (Search in the current file)

    Shift+alt+f (format document)
    Ctrl+s (Save and format document)
    Ctrl+shift+r (Refactor, which is used to wrap with another Widget)

    - Note: You can right click to see the commands

- What else do you need?
    - Nothing
    - Just learn what I told you:
        - Level I: Dart basics
        - Level II: Flutter basics
        - Level III: Practice Dart and Flutter basics stuff
        - Level IV: Flutter advanced
        - Level V: Practice Flutter advanced stuff

I wish you the best in your Flutter journey,
Flutter Mapp