Creating a Hello World App

1

Creating a New App

- 1. Start Android Studio.
- 2. Select File ⇒ New ⇒ New Flutter Project. (Start A New Flutter Project).
- 3. Select Flutter Application.



- 1. Enter the project name ch2_my_counter and click Next.
- 2. For the Platform Channel Language, select both the options for Kotlin and Swift.
- 3. Click the Finish button.

📥 New Project		×
Project name:	ch2_my_counter	
Project location:	C:\Users\alkoum\AndroidStudioProjects\ch2_my_counter]
Description:	Chapter 2 - My Counter	
Organization:	com.example	
Android language: (🔾 Java 💽 Kotlin	
iOS language: (Objective-C OSwift	
Platforms:	🗹 Android 🗹 iOS 🗌 Linux 🗌 MacOS 🔲 Web 🗌 Windows	
	Platform availability might depend on your Flutter SDK channel, and which desktop platforms have been enabled.	
	Additional desktop platforms can be enabled by, for example, running "flutter configenable-linux-desktop" on the command line.	
	When created, the new project will run on the selected platforms (others can be added later).	
	Create project	offline
▼ More Settings -		
Module name:	ch2_my_counter	
Content root:	C:\Users\alkoum\AndroidStudioProjects\ch2_my_counter	
Module file locat	tion: C:\Users\alkoum\AndroidStudioProjects\ch2_my_counter	
Project format:	.idea (directory based)	-
	Previous Finish Cancel He	elp

Running the App

1. click the Flutter **device selection button** on the right of the toolbar.



- 2. Select the Android emulator.
- 3. Click the **Run** icon in the toolbar.



4. Click the + floating action button on the bottom right, and you will see the counter increase each time you click it.

- 5. In the main.dart file, change MyHomePage (title: 'Flutter Demo Home Page') to MyHomePage(title: 'Hello World') and save.
- 6. Click **hot reload** button you will see that the **app bar title changes** and the state of the counter remains the same.



USING THEMES TO STYLE YOUR APP

There are two ways to use theme widgets:

- To style the look and feel globally.
- To style just a **portion** of the app.

Using a Global App Theme

- let's change it to light green.
- Add a new line below the primarySwatch and add code to change the background color (canvasColor) to lightGreen.

primarySwatch: Colors.blue, // Change it to primarySwatch: Colors.lightGreen, canvasColor: Colors.lightGreen.shade100,

- Save by pressing (in Windows). Hot reload is invoked.

- To show a little Flutter awesomeness, add a platform property of TargetPlatform.iOS after the canvasColor property, and run the app from the Android emulator.
- Suddenly, the iOS traits are running on Android. The app bar's title is not left aligned but changed to the center, which is the customary iOS style.

primarySwatch: Colors.blue, // Change it to primarySwatch: Colors.lightGreen, canvasColor: Colors.lightGreen.shade100, platform: TargetPlatform.iOS

This can be done in reverse by using TargetPlatform. to show Android traits on iOS, change the platform property to TargetPlatform.android.

primarySwatch: Colors.blue, // Change it to primarySwatch: Colors.lightGreen, canvasColor: Colors.lightGreen.shade100, platform: TargetPlatform.android



Searching for Packages

In the app, if you need to store the **user preferences** on both **iOS** and **Android** and want to find a **package** to do that for you.

- 1. Start your web browser and navigate to <u>https://pub.dartlang.org/flutter</u> Packages are published often by other **developers** and **Google**.
- 2. Click the search bar. Enter shared preferences.
- 3. Click the link for the **shared_preferences** package. (The direct link is https://pub.dev/flutter/packages?q=shared+preferences)
- 4. Details on how to install and use the **shared_preferences** package are available at this location.

Installing Packages

how to implement the **shared_preferences** external package in your app?

- 1. Open the **ch2_my_counter** app with Android Studio.
- 2. Open the **pubspec.yaml** file.
- 3. In the **dependencies** section: add **shared_preferences: ^2.0.5**. (Your version might be **higher**.)
- 4. Save the file, and the package will install. the message will say Process finished with exit code 0.

Image: Second	Pub get Pub upgrade Pu	b outdated Flutter docto
idStuc Flutter commands 18 19 dependencies: 20 flutter: 21 sdk: flutter	Pub get Pub upgrade Pu	b outdated Flutter docto
18 19 ⊖dependencies: 28 ⊖ flutter: 21 ⊖ sdk: flutter		☆3 ^
19 ⊖dependencies: 28 ⊖ 21 ⊖ sdk: flutter		Nov 3
28 \ominus flutter: 21 \ominus sdk: flutter		
21 o sdk: flutter		
ZI - Suk: FLUCCEP		
22		
23 # The following adas the cupertino icons font to	o your application.	
24 # Use with the Cupertinolcons class for 10S styl	le icons.	
25 cupertino_icons: ^1.0.3		
26		
27 of shared_preferences: ^2.0.5		
28		
29 odev_dependencies:		
30 🖯 flutter_test:		
31 😑 sdk: flutter		
32		
33		
34 # For information on the generic Dart part of this	s file, see the	
35 # following page: <u>https://www.dartlang.org/tools/p</u>	pub/pubspec	
36		
37 # The following section is specific to Flutter.		
Document 1/1 >>> dependencies:>>> shared_preferences:>>> ^2.0.5		
ttar.⊻ [ch2 mu counter]Dub ∐narada.⊻		
	<pre>23 # The following adds the Cupertino Icons font t 24 # Use with the CupertinoIcons class for iOS sty 25 cupertino_icons: ^1.0.3 26 27</pre>	<pre>23 # The following adds the Cupertino Icons font to your application. 24 # Use with the CupertinoIcons class for iOS style icons. 25 cupertino_icons: ^1.0.3 26 27 charac_preferences: ^2.0.5 28 29 cdev_dependencies: 30 chutter_test: 31 charac_statestatestatestatestatestatestatestat</pre>

5. Import the package in the **main.dart** file after the **material.dart** import line. **Save the changes**.

import 'package:flutter/material.dart';

import 'package:shared_preferences/shared_preferences.dart';

Using Packages

- Each package has its unique way of being implemented.
- It's always good to read the documentation.
- For the shared_preferences package, you need to add a few lines to implement it.
- The main point here is not how to use this package but how to add external packages to your app in general.

- Implementing and Initializing a Package

In the <u>MyHomePageState</u> class, add a function called <u>updateSharedPreferences()</u>.

```
class _MyHomePageState extends State<MyHomePage> {
 // ...
 void _updateSharedPreferences() async {
   SharedPreferences prefs = await SharedPreferences.getInstance();
   int counter = (prefs.getInt('counter')?? 0) + 1;
     print('Pressed $counter times.');
    await prefs.setInt('counter', counter);
```

- How It Works

This package saves users' preferences in both **iOS** and **Android** with a few lines of **Dart** code, There's no need to write native code for iOS or Android. This is the power of using packages, but be careful of overdoing it because you are relying on the authors of the packages to keep them up-to-date.