❖ Set up an editor:

- You can build apps with Flutter using any text editor combined with our command-line tools. However, we recommend using one of our editor plugins for an even better experience. These plugins provide you with code completion, syntax highlighting, widget editing assists, run & debug support, and more.

- We can use any one of them:

  - Android Studio or IntelliJ IDE.
  - Visual Studio Code.
  - Emacs

  Note: Here we are using Visual Studio Code.

❖ Install VS Code

VS Code is a lightweight editor with Flutter app execution and debug support

- VS Code , latest stable version

❖ **Install the Flutter and Dart plugins**:

- Start VS Code.
- Invoke **View > Command Palette ….**
- Type “install” and select **Extensions: Install Extensions.**
- Type “flutter” in the extensions search fields, select Flutter in the list, and click Install. This is also installs the required Dart plugin.

❖ **Validate your setup with the Flutter Doctor:**

- Invoke **View > Command Palette ….**
- Type “doctor” and select the **Flutter: Run Flutter Doctor.**
- Review the output in the **OUTPUT** pane for any issues. Make sure to select Flutter from the dropdown in the different Output Options.

Furthermore, information visits the site in order to install the flutter in other platforms as well.

Click for references - Flutter- installation – windows/mac/Linux

❖ Test drive:

We will learn to create a new Flutter app from templates, run it, and experience “hot reload” after you make changes to the app.

Select your development tool of choice for writing, building and running Flutter apps.

As I have mentioned earlier, we are going to use **Visual Studio Code.**

❖ **Create the app:**

- **Steps:**
  - Invoke **View > Command Palette.**
  - Type “***flutter***” and select the **Flutter: New Project**.
  - Select Application.

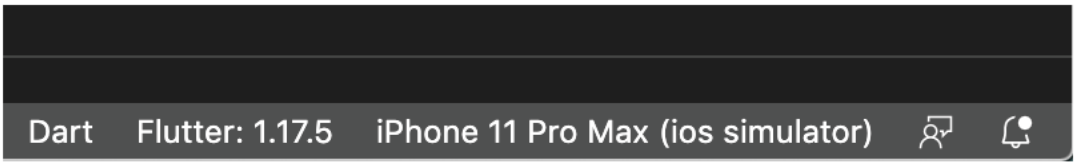
- Create or select the parent directory for the new project folder.
- Enter a project name, such as "***myapp"***, and press **Enter**.
- Wait for project creation to complete and the "***main.dart***" file to appear.

The above commands create a Flutter project directory called "***myapp***" that contains a simple demo app that uses **Material Components**.

Note: if VS Code was running during your initial Flutter setup, you may need to restart if for VS Code's Flutter plugin to detect the Flutter SDK.

❖ **Run the app:**

➢ Locate the VS Code status bar (the blue bar at the bottom of window):



➢ Select a device from the Device Selector area.

- If no device is available and you want to use a device simulator, click No Devices and launch a simulator.

  Note: you may not see Start IOS Simulator option when you click No Devices in VS Code. If you are on Mac, then you may have to run following command in terminal to launch a simulator.

  Open -a simulator

  In Android it is not possible to launch IOS simulator

- To setup a real device, follow the device-specific instruction on the install page for your OS.

- Invoke Run > Start Debugging or press F5.
- Wait for the app to launch – progress is printed in the Debug Console view.

After the app build completes, you'll see the starter app on your device.