Widgets:

- Widgets are classes used to create UIs
- They create UI elements and layouts

Types of Widgets:

Widgets can be widely classified into two:

1) Stateful Widget: A stateful Widget can change through interacting with the user. Examples, Checkbox, forms etc.
2) Stateless Widget: A stateless widget always remains constant. Examples: Text, Icon, etc.

Examples of Widgets and Layouts:

1) Container: A container is a widget that includes other widgets such as padding, background color, row, column, etc. under what is known as a child.
2) Column: A column widget arranges its attributes in a vertical way. It uses child as well to add features to the column.
3) Row: A row widget arranges its attributes in a horizontal way. It uses child as well to add features to the row.
4) Expanded: An expanded widget makes a child of a row or column expand horizontally or vertically respectively.
5) Padding: When a constraint is given regarding the size of its child, padding ensures to implement it. There are several options such as EdgeInsets.all(), EdgeInsets.symmetric, etc.
6) Listview: A listview helps scroll through its children one after the other.
7) Icon: An Icon is a pictorial representation of an app. Icons cannot be clicked on in flutter and are merely a representation.

8) Image: This displays an image. You can choose images from the internet, computer or flutter itself for the same.
9) Text: A text widget is used to type certain text that has to be present in the app. There are many attributes to it such as TextStyle, textColor, and textAlign.
10) Button: A button widget is used to click on something that results in performing an action it was designed to do. There are several types of buttons such as Flat button, Outline button. A button must include an onPressed function and a child text with several different fonts and colors to it as well.
11) SafeArea: A SafeArea widget makes sure that its child has decent padding so that it does not interrupt with an operating system.

setState(): It notifies the framework that an object's internal state has changed. It changes the state of the stateful widget. It notifies that it's a way that might impact the user interface, which causes the framework to schedule a build for this State object. It helps work with user input. If this is not done, any change made under stateful widget will not be reflected in the output.