JAVAFX UI CONTROLS

Every user interface considers the following three main aspects −

- **UI elements** − These are the core visual elements which the user eventually sees and interacts with. JavaFX provides a huge list of widely used and common elements varying from basic to complex, which we will cover in this tutorial.
- **Layouts** − They define how UI elements should be organized on the screen and provide a final look and feel to the GUI (Graphical User Interface). This part will be covered in the Layout chapter.
- **Behavior** − These are events which occur when the user interacts with UI elements. This part will be covered in the Event Handling chapter.

JavaFX provides several classes in the package **javafx.scene.control**. To create various GUI components (controls), JavaFX supports several controls such as date picker, button text field, etc.

Each control is represented by a class; you can create a control by instantiating its respective class.

Following is the list of commonly used controls while the GUI is designed using JavaFX.

S.No	Control & Description
1	**Label** A Label object is a component for placing text.
2	**Button** This class creates a labeled button.
3	**ColorPicker** A ColorPicker provides a pane of controls designed to allow a user to manipulate and select a color.

4	**CheckBox** A CheckBox is a graphical component that can be in either an on(true) or off (false) state.
5	**RadioButton** The RadioButton class is a graphical component, which can either be in a ON (true) or OFF (false) state in a group.
6	**ListView** A ListView component presents the user with a scrolling list of text items.
7	**TextField** A TextField object is a text component that allows for the editing of a single line of text.
8	**PasswordField** A PasswordField object is a text component specialized for password entry.
9	**Scrollbar** A Scrollbar control represents a scroll bar component in order to enable user to select from range of values.
10	**FileChooser** A FileChooser control represents a dialog window from which the user can select a file.
11	**ProgressBar** As the task progresses towards completion, the progress bar displays the task's percentage of completion.

12	**Slider** A Slider lets the user graphically select a value by sliding a knob within a bounded interval.

Example

The following program is an example which displays a login page in JavaFX. Here, we are using the controls **label, text field, password field** and **button**.

Save this code in a file with the name **LoginPage.java**.

```
import javafx.application.Application;
import static javafx.application.Application.launch;
import javafx.geometry.Insets;
import javafx.geometry.Pos;

import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.PasswordField;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

public class LoginPage extends Application {
  @Override
  public void start(Stage stage) {
    //creating label email
    Text text1 = new Text("Email");

    //creating label password
    Text text2 = new Text("Password");

    //Creating Text Filed for email
    TextField textField1 = new TextField();
```

```
//Creating Text Filed for password
PasswordField textField2 = new PasswordField();

//Creating Buttons
Button button1 = new Button("Submit");
Button button2 = new Button("Clear");

//Creating a Grid Pane
GridPane gridPane = new GridPane();

//Setting size for the pane
gridPane.setMinSize(400, 200);

//Setting the padding
gridPane.setPadding(new Insets(10, 10, 10, 10));

//Setting the vertical and horizontal gaps between the columns
gridPane.setVgap(5);
gridPane.setHgap(5);

//Setting the Grid alignment
gridPane.setAlignment(Pos.CENTER);

//Arranging all the nodes in the grid
gridPane.add(text1, 0, 0);
gridPane.add(textField1, 1, 0);
gridPane.add(text2, 0, 1);
gridPane.add(textField2, 1, 1);
gridPane.add(button1, 0, 2);
gridPane.add(button2, 1, 2);

//Styling nodes
button1.setStyle("-fx-background-color: darkslateblue; -fx-text-fill: white;");
```

```
    button2.setStyle("-fx-background-color: darkslateblue; -fx-text-fill: white;");

    text1.setStyle("-fx-font: normal bold 20px 'serif' ");
    text2.setStyle("-fx-font: normal bold 20px 'serif' ");
    gridPane.setStyle("-fx-background-color: BEIGE;");

    //Creating a scene object
    Scene scene = new Scene(gridPane);

    //Setting title to the Stage
    stage.setTitle("CSS Example");

    //Adding scene to the stage
    stage.setScene(scene);

    //Displaying the contents of the stage
    stage.show();
  }
  public static void main(String args[]){
    launch(args);
  }
}
```