

JavaFX: Getting Started with JavaFX

3 Hello World, JavaFX Style

The best way to teach you what it is like to create and build a JavaFX application is with a "Hello World" application. An added benefit of this tutorial is that it enables you to test that your JavaFX technology is properly installed.

The tool used in this tutorial is NetBeans IDE 7.4. Before you begin, ensure that the version of NetBeans IDE that you are using supports JavaFX 8. See the [Certified System Configurations](#) section of the Java SE 8 downloads page for details.

Construct the Application

1. From the **File** menu, choose **New Project**.
2. In the **JavaFX** application category, choose **JavaFX Application**. Click **Next**.
3. Name the project **HelloWorld** and click **Finish**.

NetBeans opens the HelloWorld.java file and populates it with the code for a basic Hello World application, as shown in [Example 3-1](#).

Example 3-1 Hello World

```
package helloworld;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

public class HelloWorld extends Application {

    @Override
    public void start(Stage primaryStage) {
        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });

        StackPane root = new StackPane();
        root.getChildren().add(btn);

        Scene scene = new Scene(root, 300, 250);

        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

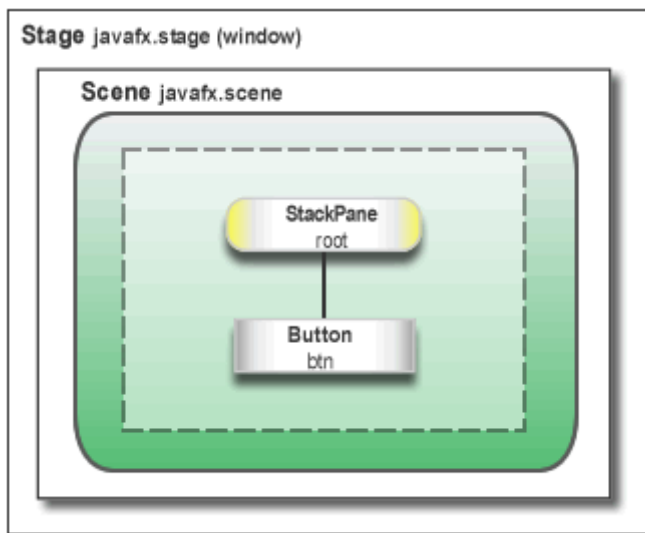
    public static void main(String[] args) {
        launch(args);
    }
}
```

Here are the important things to know about the basic structure of a JavaFX application:

- The main class for a JavaFX application extends the `javafx.application.Application` class. The `start()` method is the main entry point for all JavaFX applications.
- A JavaFX application defines the user interface container by means of a stage and a scene. The JavaFX **Stage** class is the top-level JavaFX container. The JavaFX **Scene** class is the container for all content. [Example 3-1](#) creates the stage and scene and makes the scene visible in a given pixel size.
- In JavaFX, the content of the scene is represented as a hierarchical scene graph of nodes. In this example, the root node is a **StackPane** object, which is a resizable layout node. This means that the root node's size tracks the scene's size and changes when the stage is resized by a user.
- The root node contains one child node, a button control with text, plus an event handler to print a message when the button is pressed.
- The `main()` method is not required for JavaFX applications when the JAR file for the application is created with the JavaFX Packager tool, which embeds the JavaFX Launcher in the JAR file. However, it is useful to include the `main()` method so you can run JAR files that were created without the JavaFX Launcher, such as when using an IDE in which the JavaFX tools are not fully integrated. Also, Swing applications that embed JavaFX code require the `main()` method.

[Figure 3-1](#) shows the scene graph for the Hello World application. For more information on scene graphs see [Working with the JavaFX Scene Graph](#).

Figure 3-1 Hello World Scene Graph



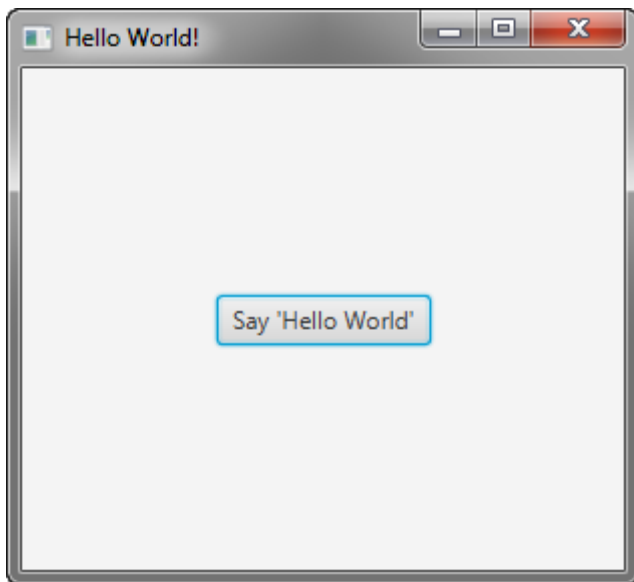
[Description of "Figure 3-1 Hello World Scene Graph"](#)

Run the Application

1. In the Projects window, right-click the **HelloWorld** project node and choose **Run**.
2. Click the Say Hello World button.
3. Verify that the text "Hello World!" is printed to the NetBeans output window.

[Figure 3-2](#) shows the Hello World application, JavaFX style.

Figure 3-2 Hello World, JavaFX style



[Description of "Figure 3-2 Hello World, JavaFX style"](#)

Where to Go Next

This concludes the basic Hello World tutorial, but continue reading for more lessons on developing JavaFX applications:

- [Creating a Form in JavaFX](#) teaches the basics of screen layout, how to add controls to a layout, and how to create input events.
- [Fancy Forms with JavaFX CSS](#) provides simple style tricks for enhancing your application, including adding a background image and styling buttons and text.
- [Using FXML to Create a User Interface](#) shows an alternate method for creating the login user interface. FXML is an XML-based language that provides the structure for building a user interface separate from the application logic of your code.
- [Animation and Visual Effects in JavaFX](#) shows how to bring an application to life by adding timeline animation and blend effects.