

This site uses cookies to improve the user experience. [OK](#)



JavaFX

1. [JavaFX Tutorial](#)
2. [JavaFX Overview](#)
3. [Your First JavaFX Application](#)
4. [JavaFX FXML](#)
5. [JavaFX CSS Styling](#)
6. [JavaFX ImageView](#)
7. [JavaFX Label](#)
8. [JavaFX Button](#)
9. [JavaFX MenuButton](#)
10. [JavaFX ToggleButton](#)
11. [JavaFX RadioButton](#)
12. [JavaFX CheckBox](#)
13. [JavaFX ChoiceBox](#)
14. [JavaFX ComboBox](#)
15. [JavaFX ListView](#)
16. [JavaFX DatePicker](#)
17. [JavaFX TextField](#)
18. [JavaFX PasswordField](#)
19. [JavaFX TextArea](#)
20. [JavaFX ToolBar](#)
21. [JavaFX Tooltip](#)
22. [JavaFX FileChooser](#)
23. [JavaFX Group](#)
24. [JavaFX HBox](#)
25. [JavaFX VBox](#)
26. [JavaFX FlowPane](#)
27. [JavaFX TilePane](#)
28. [JavaFX GridPane](#)
29. [JavaFX MenuBar](#)
30. [JavaFX PieChart](#)
31. [JavaFX BarChart](#)
32. [JavaFX StackedBarChart](#)
33. [JavaFX ScatterChart](#)
34. [JavaFX LineChart](#)
35. [JavaFX AreaChart](#)
36. [JavaFX StackedAreaChart](#)



JavaFX ImageView

- [Creating an ImageView](#)
- [Adding an ImageView to the Scene Graph](#)
- [ImageView in Labels and Buttons](#)

Jakob Jenkov
Last update: 2016-05-13



The JavaFX `ImageView` control can display an image inside a JavaFX GUI. The `ImageView` control must be added to the scene graph to be visible. The JavaFX `ImageView` control is represented by the class `javafx.scene.image.ImageView`.

Creating an ImageView

You create an `ImageView` control instance by creating an instance of the `ImageView` class. The constructor of the `ImageView` class needs an instance of a `javafx.scene.image.Image` as parameter. The `Image` object represents the image to be displayed by the `ImageView` control.

Here is a JavaFX `ImageView` instantiation example:

```
FileInputStream input = new FileInputStream("resources/images/iconmonstr-home-6-48.png");
Image image = new Image(input);
ImageView imageView = new ImageView(image);
```

First a `FileInputStream` is created which points to the image file of the image to display.

Second an `Image` instance is created, passing the `FileInputStream` as parameter to the `Image` constructor. This way the `Image` class knows where to load the image file from.

Third an `ImageView` instance is created, passing the `Image` instance as parameter to the `ImageView` constructor.

Adding an ImageView to the Scene Graph

To make the `ImageView` visible you must add it to the scene graph. This means adding it to a `Scene` object. Since `ImageView` is not a subclass of `javafx.scene.Parent` it cannot be added directly to the scene graph. It must be nested inside another component, for instance a layout component.

Here is an example that attaches a JavaFX `ImageView` to the scene graph by nesting it inside a `HBox` layout component:

```
package com.jenkov.javaafx.controls;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;

import java.io.FileInputStream;

public class ImageViewExperiments extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception {
        primaryStage.setTitle("ImageView Experiment 1");

        FileInputStream input = new FileInputStream("resources/images/iconmonstr-home-6-48.png");
        Image image = new Image(input);
        ImageView imageView = new ImageView(image);

        HBox hbox = new HBox(imageView);

        Scene scene = new Scene(hbox, 200, 100);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

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

The result of running the above JavaFX `ImageView` example is an application that looks like this:



ImageView in Labels and Buttons

It is possible to use an `ImageView` in both a JavaFX `Label` and `Button`. This will cause the `Label` and `Button` to display the `ImageView` to the left of the text in the `Label` or `Button`. See the texts about [JavaFX Label](#) and [JavaFX Button](#) for information about how to do that.

Next: [JavaFX Label](#)

[G+](#) [Teilen](#) [Tweet](#)

Jakob Jenkov





Copyright Jenkov Aps