

CSCI 2120:

Software Design & Development II

UNIT4: UI management

GUI framework

JavaFX Controls: Label

Overview

1. Introduction
2. Creating a Label
3. Adding a Label to the Scene Graph
4. Displaying Images in a Label
5. Changing the Text of a Label
6. Set Label Font

Introduction

- The JavaFX `Label` control can `display a text or image` label inside a JavaFX GUI.
- The `Label` control must be added to the `scene graph` to be visible.
- The JavaFX `Label` control is represented by the class `javafx.scene.control.Label` .

Creating a Label

You create a **label control** instance by creating an instance of the **Label** class.

Here is a JavaFX **Label** instantiation example:

```
Label label = new Label("My Label");
```

As you can see, the **text to display** in the label is passed as **parameter** to the **Label** constructor.

Adding a Label to the Scene Graph

To make the `Label` visible you must add it to the `scene graph`. This means adding it to a `Scene` object, or as `child of a layout` which is attached to a `Scene` object.

Here is an example that attaches a JavaFX Label to the scene graph:

Adding a Label to the Scene Graph

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.stage.Stage;

public class LabelExperiments extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        primaryStage.setTitle("HBox Experiment 1");

        Label label = new Label("My Label");

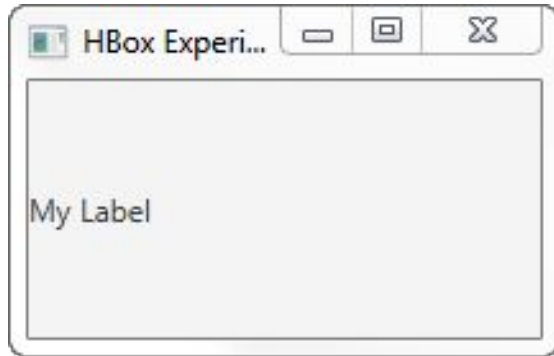
        Scene scene = new Scene(label, 200, 100);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

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

Adding a Label to the Scene Graph

Notice that the `Label` is added directly to the `Scene` object. Normally you would nest the `Label` inside a layout component of some kind. We have left that out here to keep the example simple.

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



Displaying Images in a Label

It is possible to display an **image** inside a **label** next to the label text. The JavaFX **Label** class contains a constructor that can take a **Node** as extra parameter.

Here is a JavaFX **label** example that adds an image to the label using an **JavaFX ImageView** component:

Displaying Images in a Label

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.stage.Stage;
import java.io.FileInputStream;

public class LabelExperiments extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        primaryStage.setTitle("HBox Experiment 1");

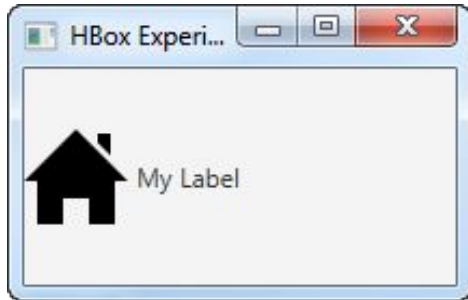
        FileInputStream input = new FileInputStream("resources/images/iconmonstr-home-6-48.png");
        Image image = new Image(input);
        ImageView imageView = new ImageView(image);
        Label label = new Label("My Label", imageView);

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

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

Displaying Images in a Label

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



Changing the Text of a Label

You can **change the text** of a **label** using its **setText()** method. This can be done while the application is running.

Here is an example of setting the text of a JavaFX **Label**:

```
label.setText("New label text");
```

Set Label Font

You can **change the font** used by a JavaFX **Label** by calling its **setFont()** method. This is useful if you need to change the size of the text, or want to use a different text style. You can learn more about how to create JavaFX fonts in the **JavaFX Fonts** lecture.

Here is an example of setting the font of a JavaFX **Label**:

```
Label label = new Label("A label with custom font set.");  
label.setFont(new Font("Arial", 24));
```

This example tells the **Label** to use the **Arial** font with a size of **24**.

END