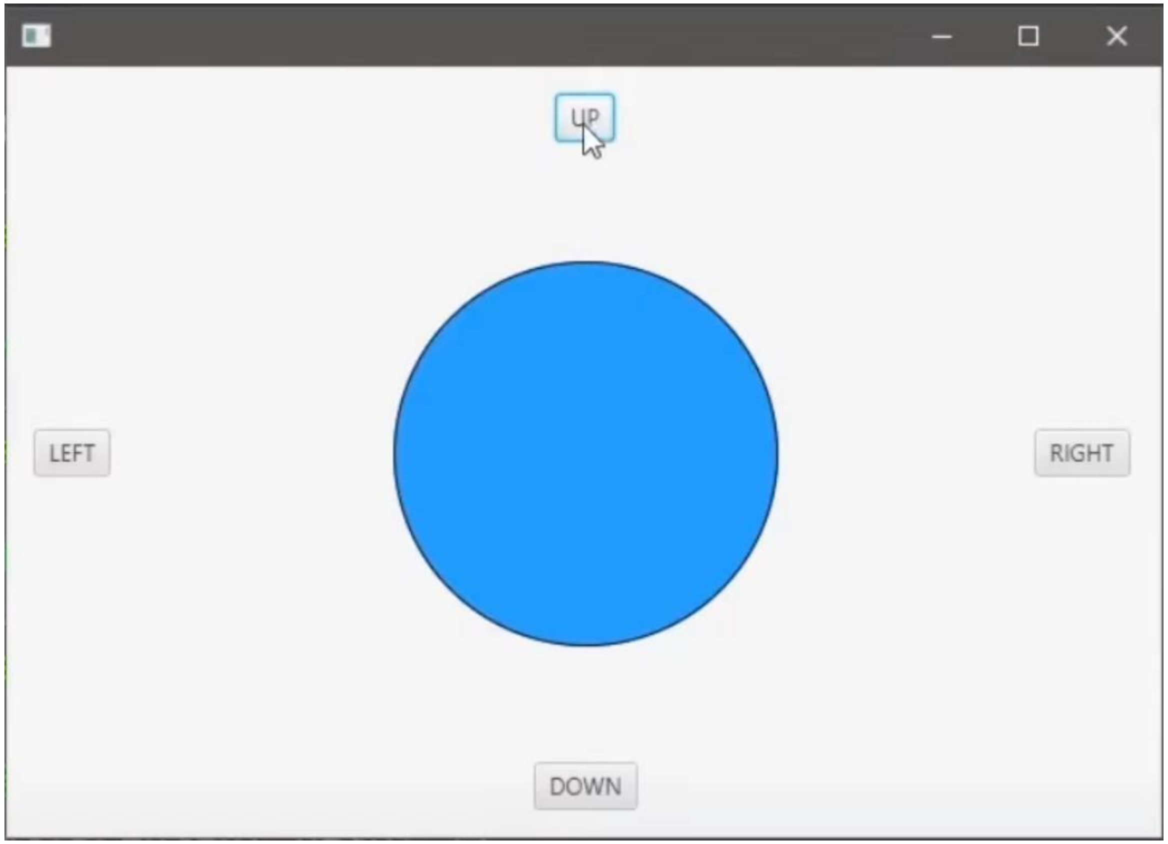
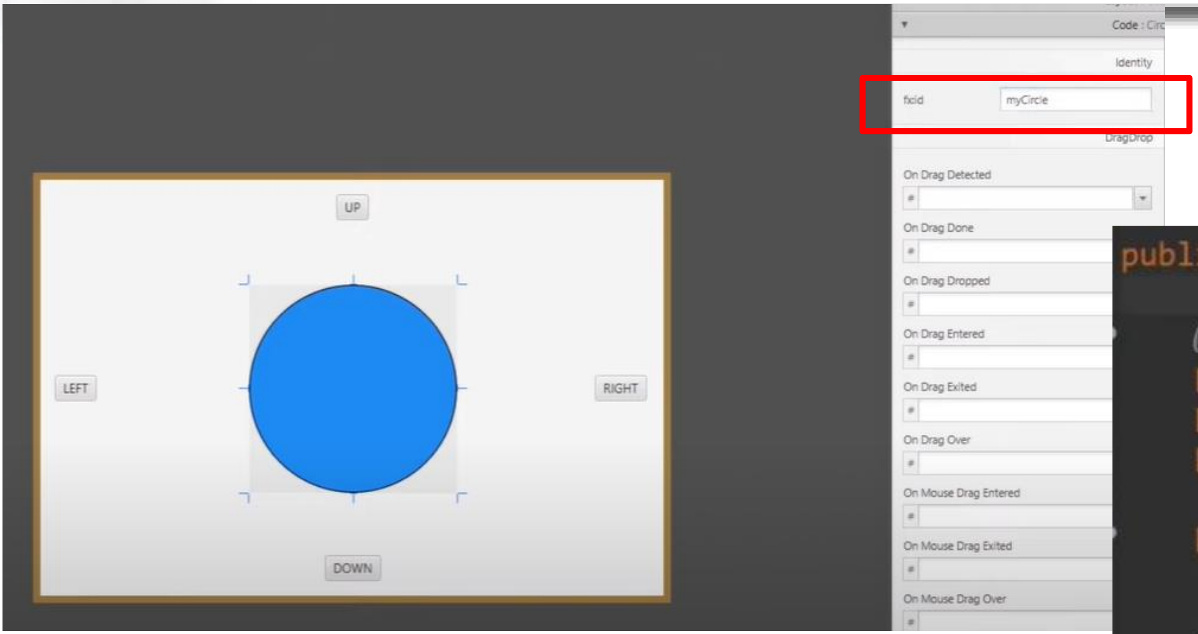
JavaFX II

Shin-Jie Lee (李信杰)
Associate Professor
Computer and Network Center
Department of Computer Science and Information Engineering
National Cheng Kung University

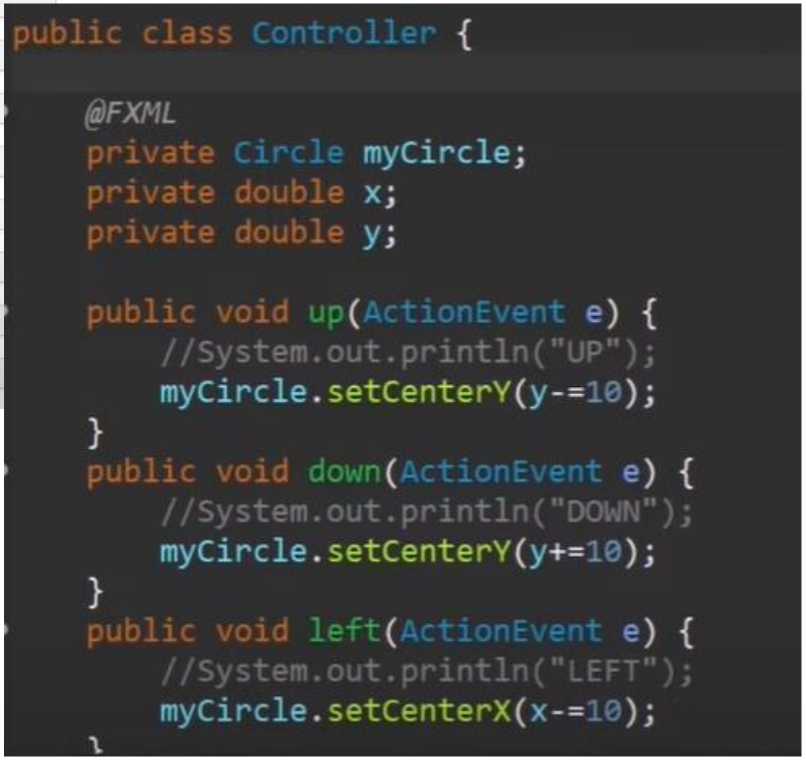
Did you do the lab?



Make a circle in SceneBuilder and name its fxid to “myCircle”



fxid myCircle



```java
public class Controller {

    @FXML
    private Circle myCircle;
    private double x;
    private double y;

    public void up(ActionEvent e) {
        //System.out.println("UP");
        myCircle.setCenterY(y-=10);
    }
    public void down(ActionEvent e) {
        //System.out.println("DOWN");
        myCircle.setCenterY(y+=10);
    }
    public void left(ActionEvent e) {
        //System.out.println("LEFT");
        myCircle.setCenterX(x-=10);
    }
```

5.
CSS Styling

What’s CSS?

CSS (Cascading Style Sheet)

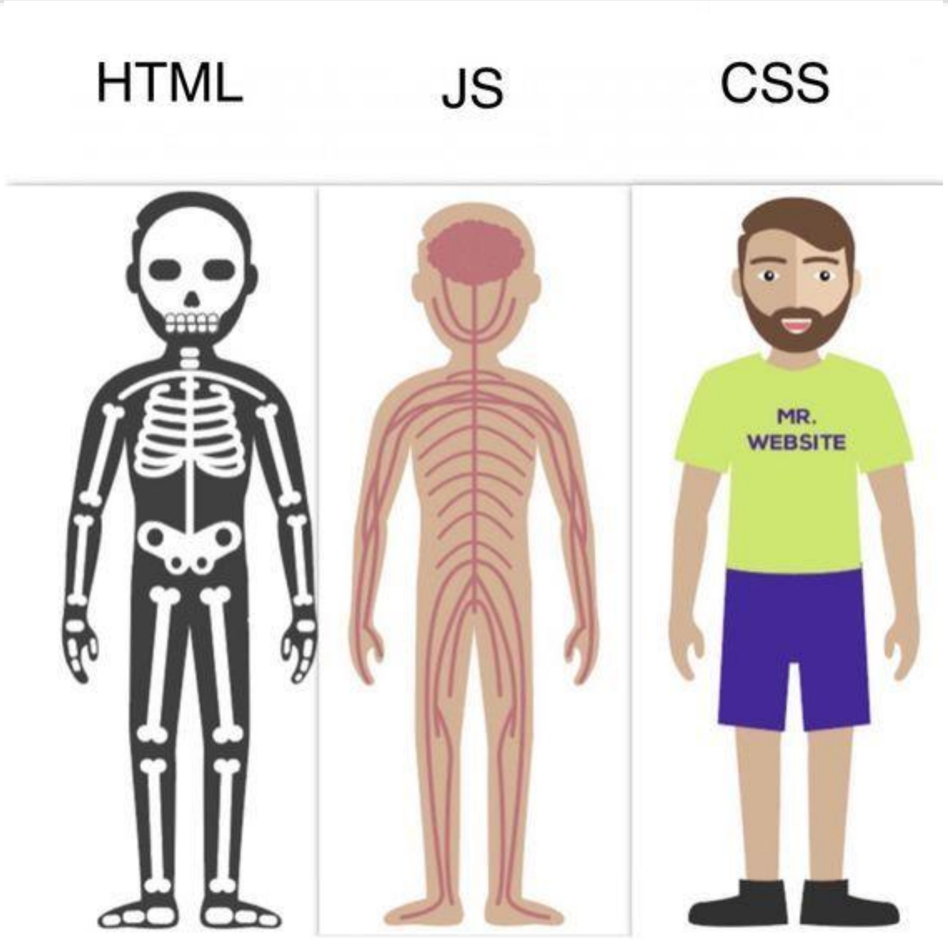
Style sheet that describes the presentation of a document written in markup language (HTML, fxml)

講人話！

HTML

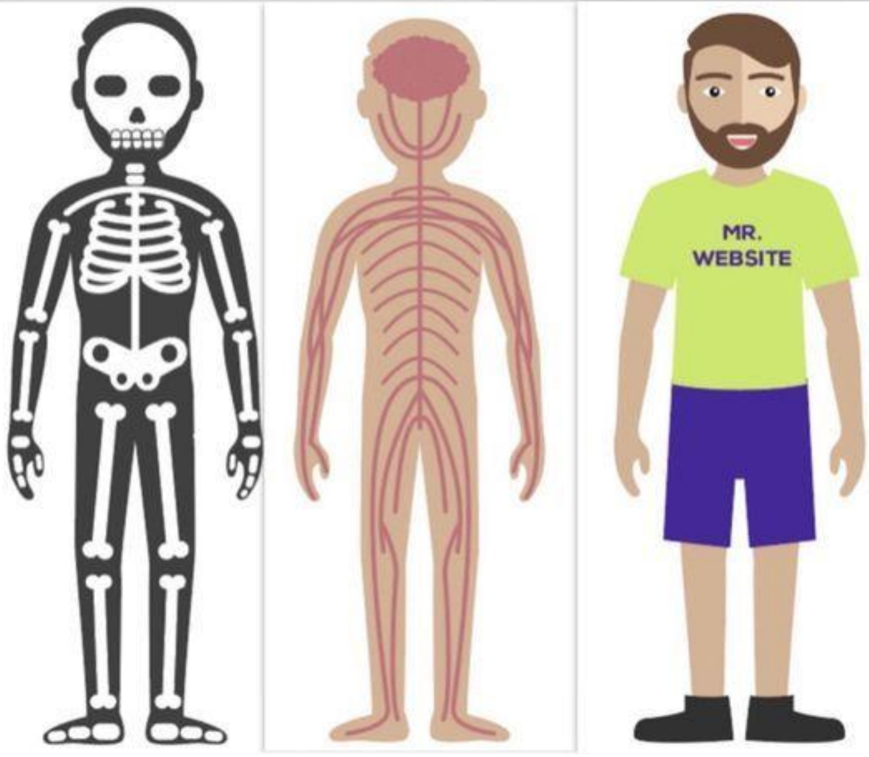
JS

CSS


HTML
JS
CSS
MR.
WEBSITE

~~HTML~~ ~~JS~~ ~~CSS~~

FXML Java Code CSS



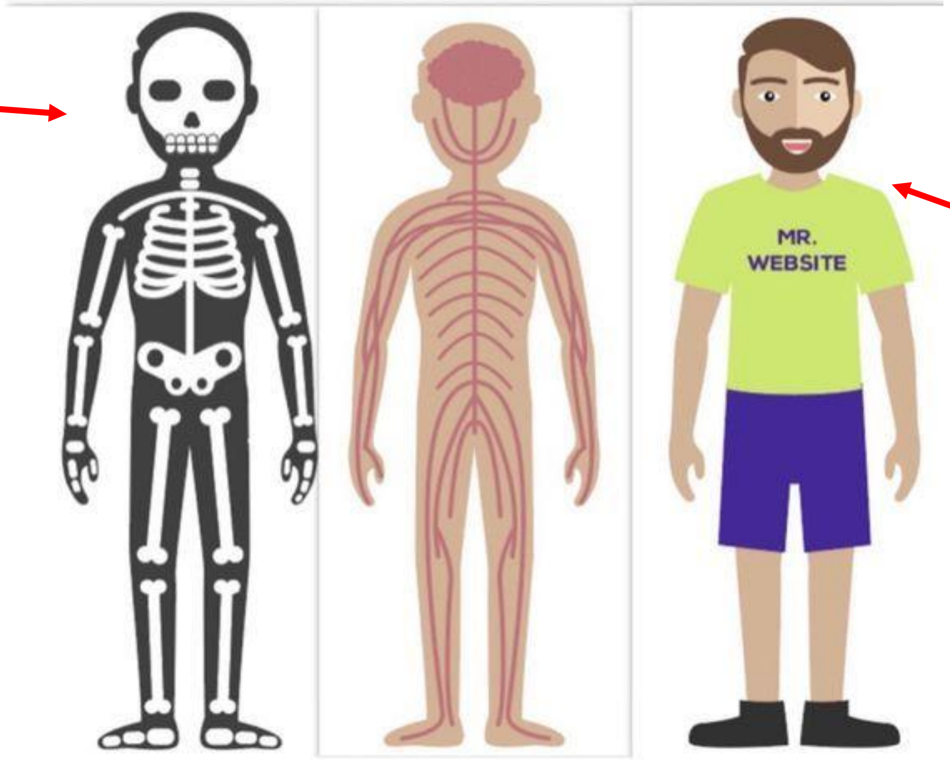
Functions

~~HTML~~ FXML

~~JS~~ Java Code

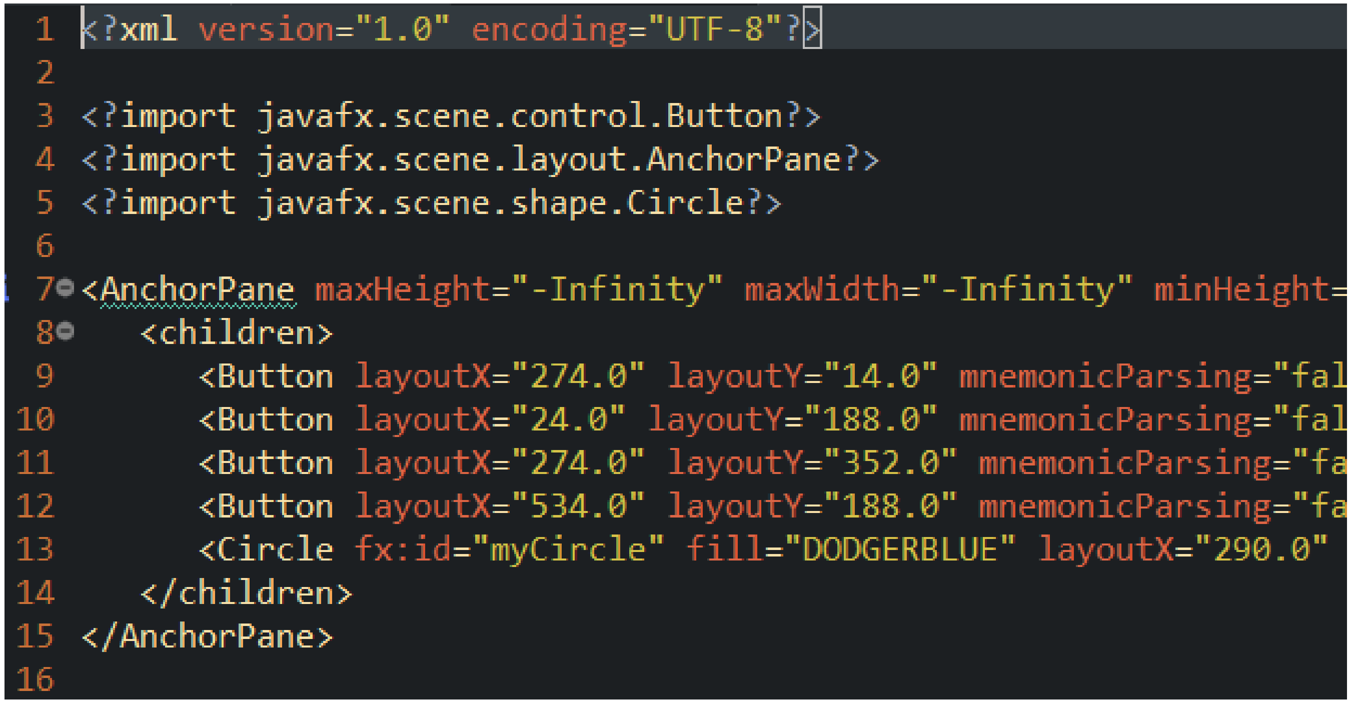
~~CSS~~ CSS

UI Structure



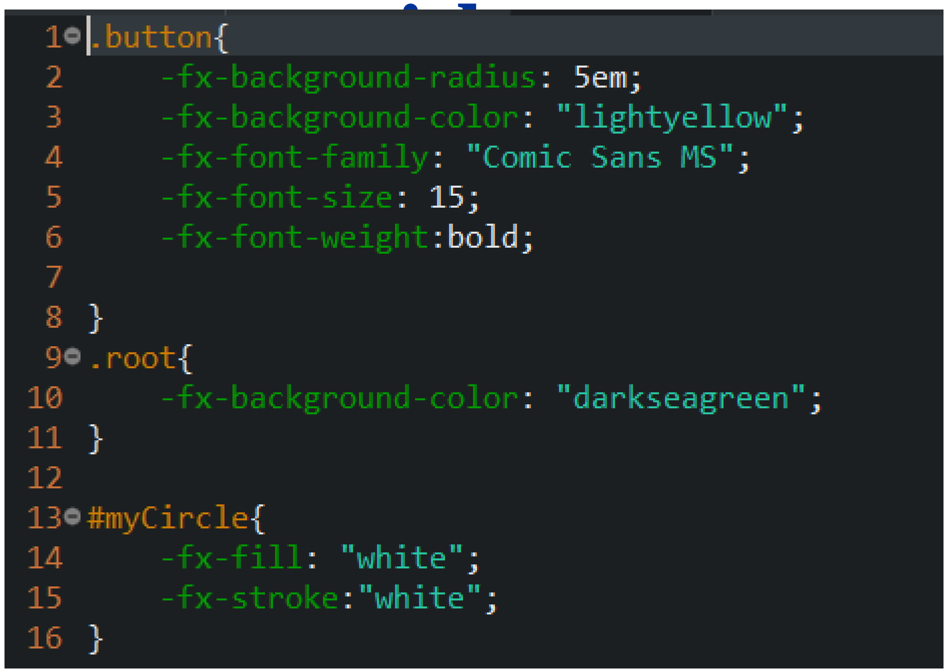
UI Style

Markup Language doesn’t have programming logic



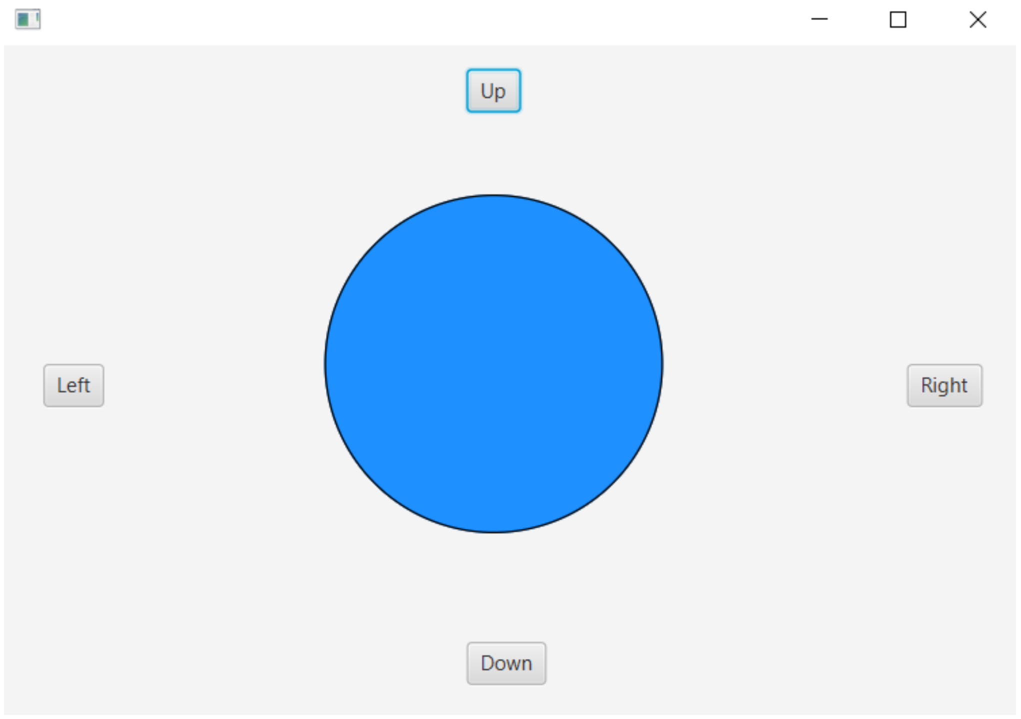
```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.shape.Circle?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight=
   <children>
      <Button layoutX="274.0" layoutY="14.0" mnemonicParsing="fal
      <Button layoutX="24.0" layoutY="188.0" mnemonicParsing="fal
      <Button layoutX="274.0" layoutY="352.0" mnemonicParsing="fa
      <Button layoutX="534.0" layoutY="188.0" mnemonicParsing="fa
      <Circle fx:id="myCircle" fill="DODGERBLUE" layoutX="290.0"
   </children>
</AnchorPane>

```

CSS doesn’t have programming logic

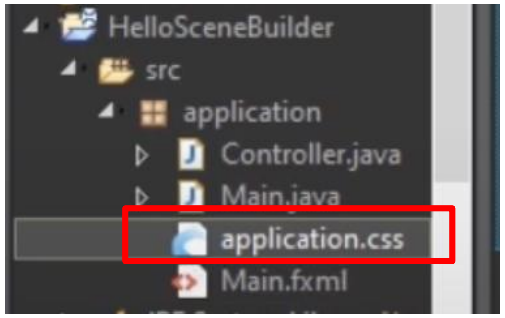


```css
.button{
    -fx-background-radius: 5em;
    -fx-background-color: "lightyellow";
    -fx-font-family: "Comic Sans MS";
    -fx-font-size: 15;
    -fx-font-weight:bold;

}
.root{
    -fx-background-color: "darkseagreen";
}

#myCircle{
    -fx-fill: "white";
    -fx-stroke:"white";
}
```

Let’s give this some styles!

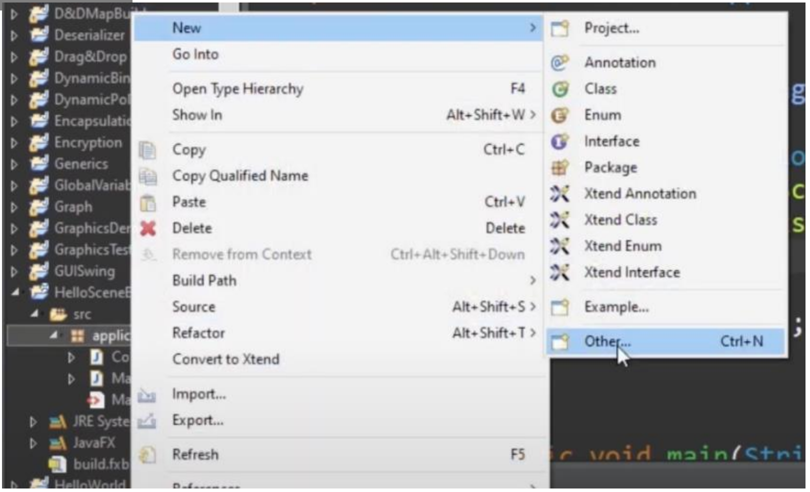


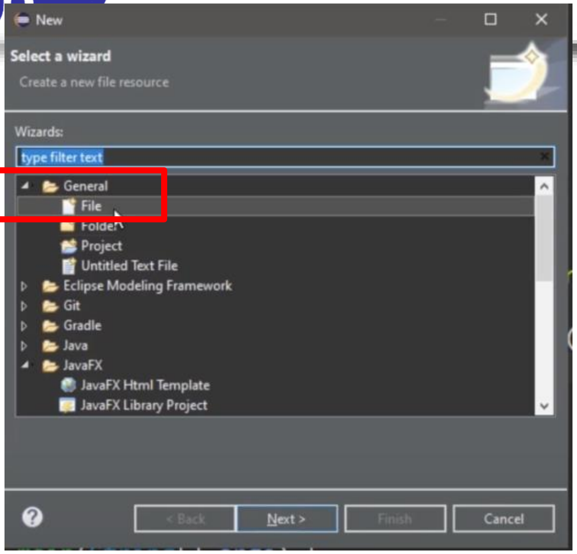
Create a CSS file



There should be a css file in your package

Or you can create your own by 1. Right click on your package > **New** > **Other**





New

Select a wizard

Create a new file resource

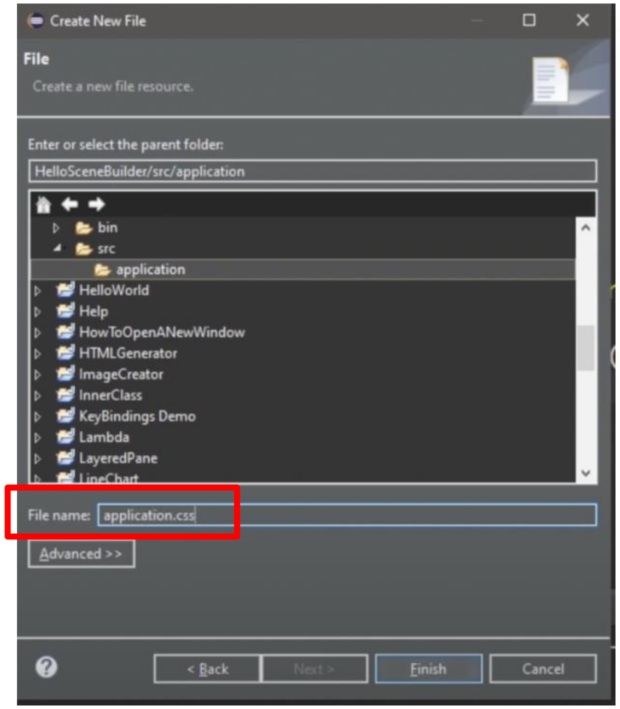
Wizards:

type filter text

- General
  - File
  - Folder
  - Project
  - Untitled Text File
- Eclipse Modeling Framework
- Git
- Gradle
- Java
- JavaFX
  - JavaFX Html Template
  - JavaFX Library Project

< Back | Next > | Finish | Cancel

2. Click on **General > File** and click Next



Create New File

File

Create a new file resource.

Enter or select the parent folder:

HelloSceneBuilder/src/application

- bin
- src
  - application
- HelloWorld
- Help
- HowToOpenANewWindow
- HTMLGenerator
- ImageCreator
- InnerClass
- KeyBindings Demo
- Lambda
- LayeredPane
- LineChart

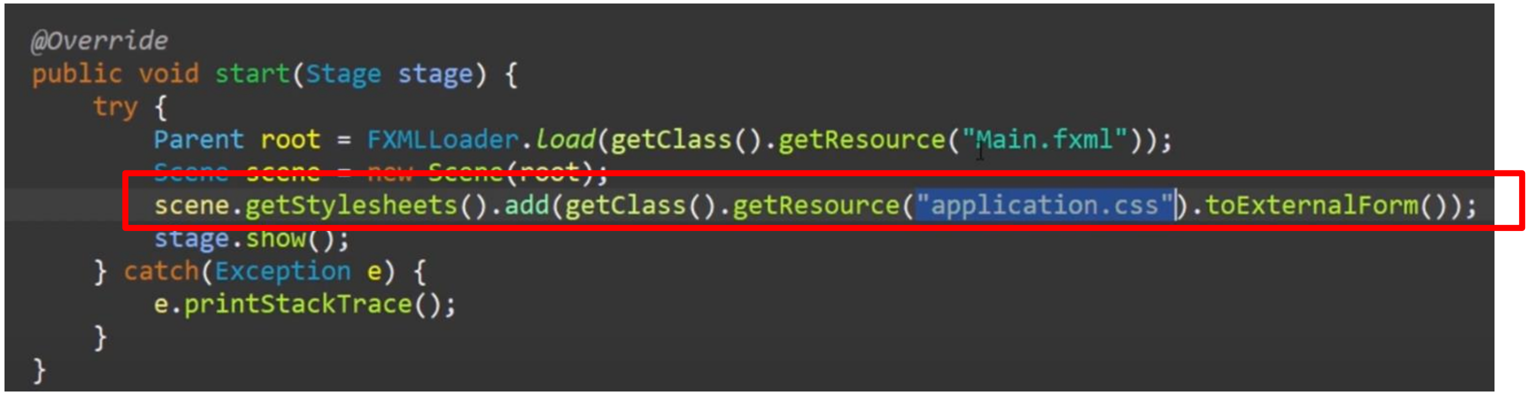
File name: application.css

Advanced >>

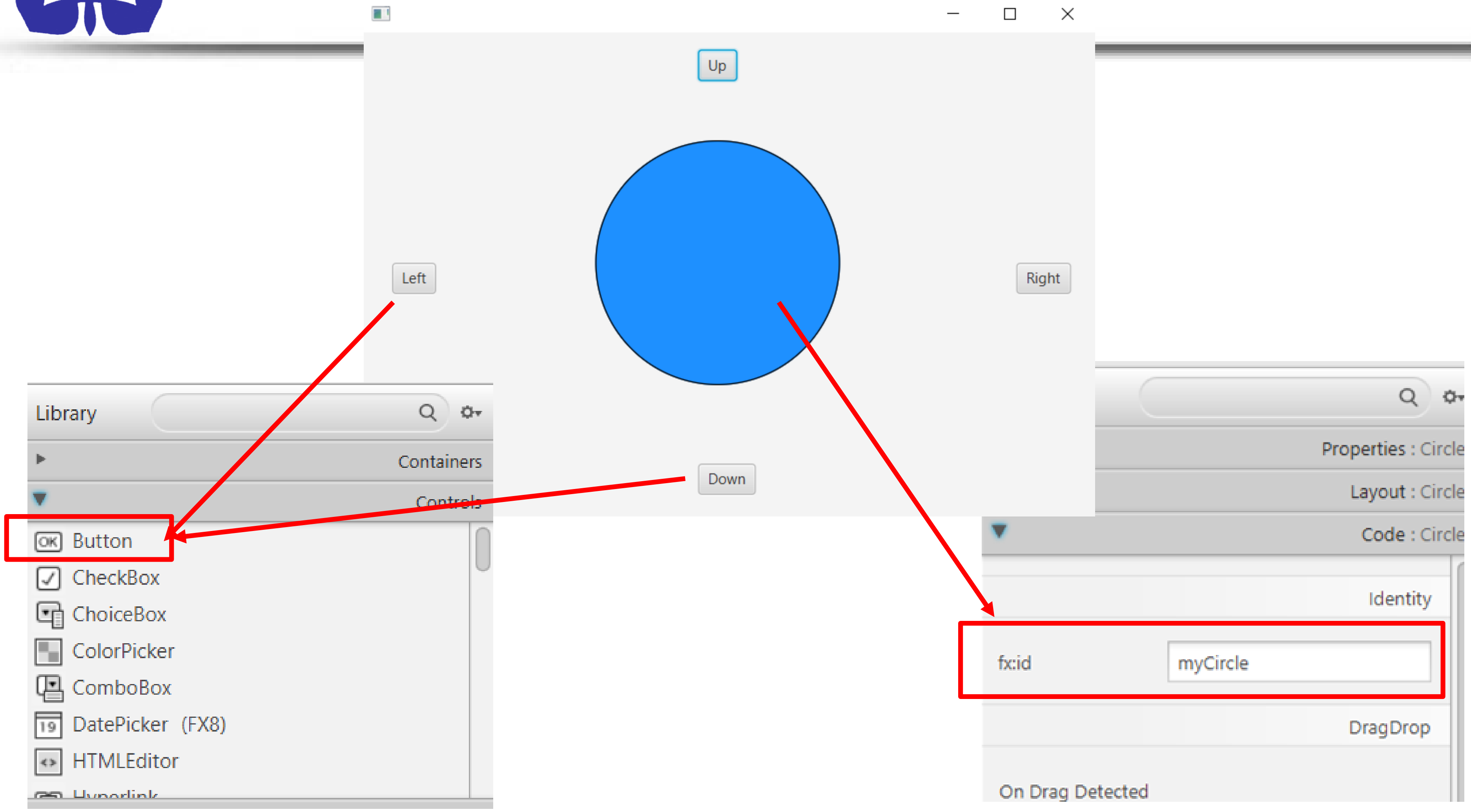
< Back | Next > | Finish | Cancel

3. Type in your file name with the file extension name as .css, and click Finish

Link your css to your Scene



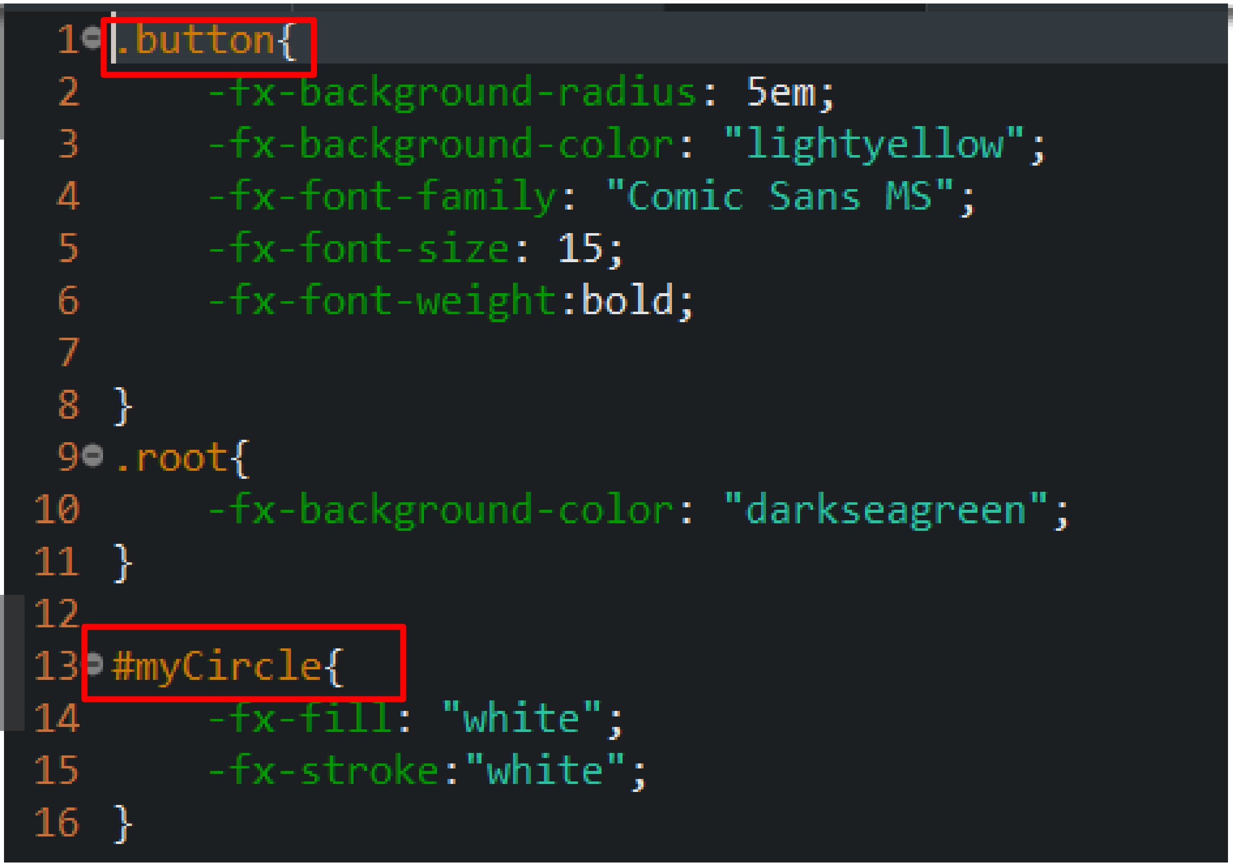
```
@Override
public void start(Stage stage) {
    try {
        Parent root = FXMLLoader.load(getClass().getResource("Main.fxml"));
        Scene scene = new Scene(root);
        scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
        stage.show();
    } catch(Exception e) {
        e.printStackTrace();
    }
}
```



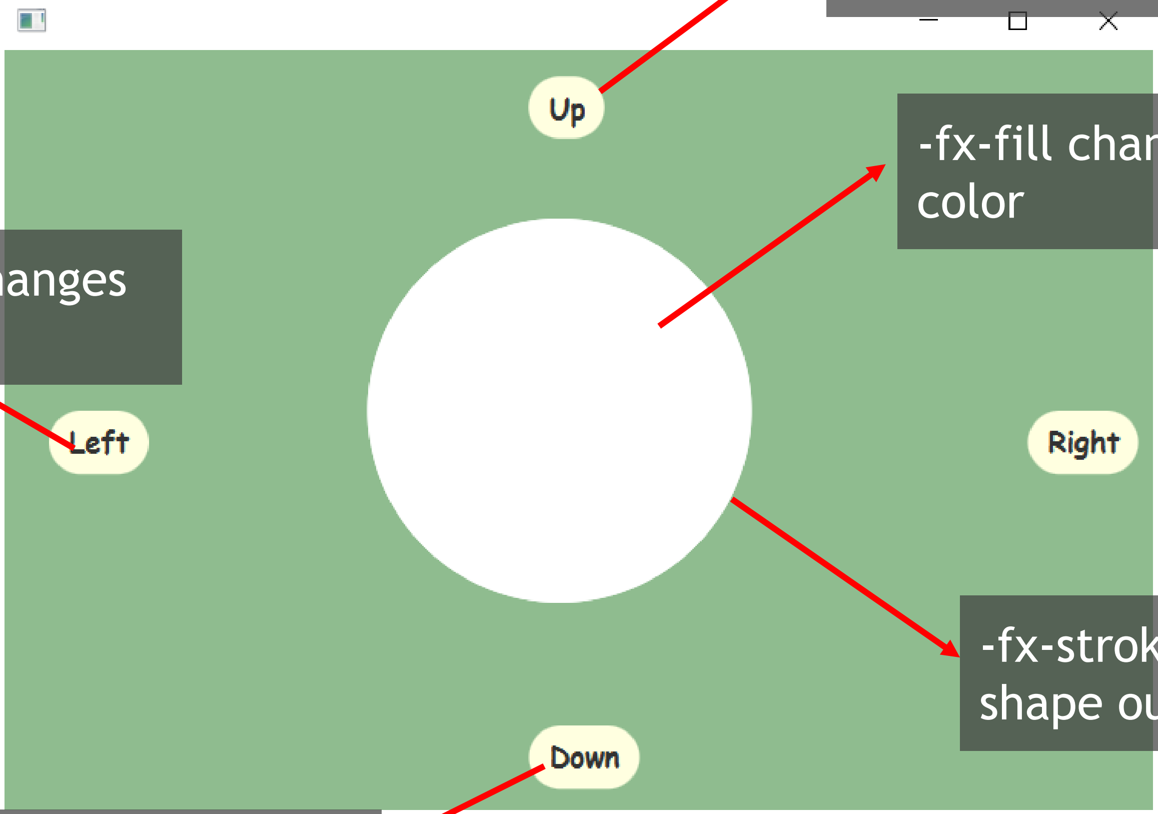
fx:id myCircle

For all the nodes that are “button”

For the node with the id “myCircle”



```
.button{
        -fx-background-radius: 5em;
        -fx-background-color: "lightyellow";
        -fx-font-family: "Comic Sans MS";
        -fx-font-size: 15;
        -fx-font-weight:bold;

}
.root{
        -fx-background-color: "darkseagreen";
}

#myCircle{
        -fx-fill: "white";
        -fx-stroke:"white";
}
```



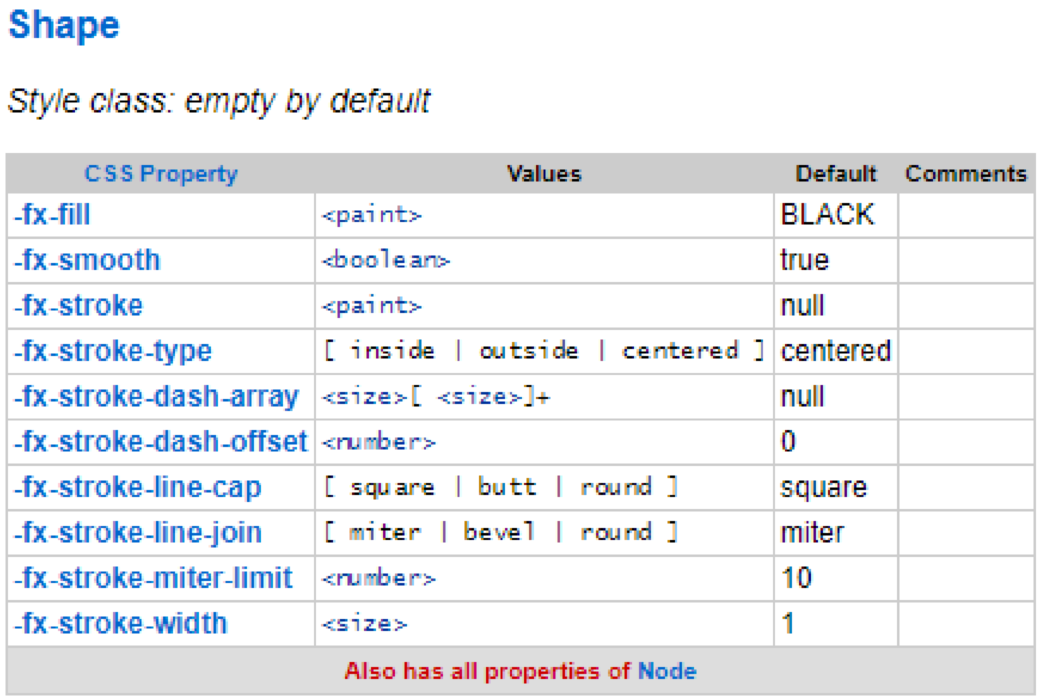
-fx-background-radius makes the button round

-fx-fill changes the shape color

-fx-font-family changes the font

-fx-stroke changes the shape outline color

-fx-background-color changes the background color of the button https://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html#button

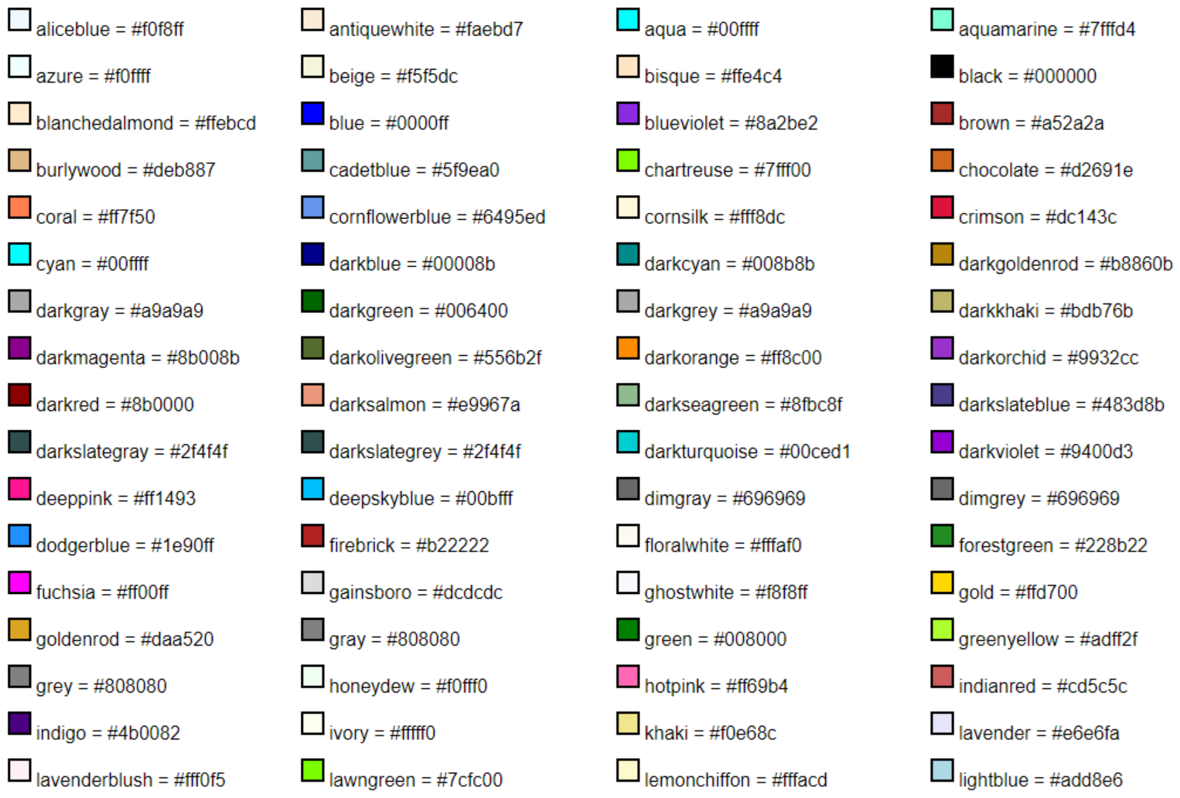


Shape

Style class: empty by default

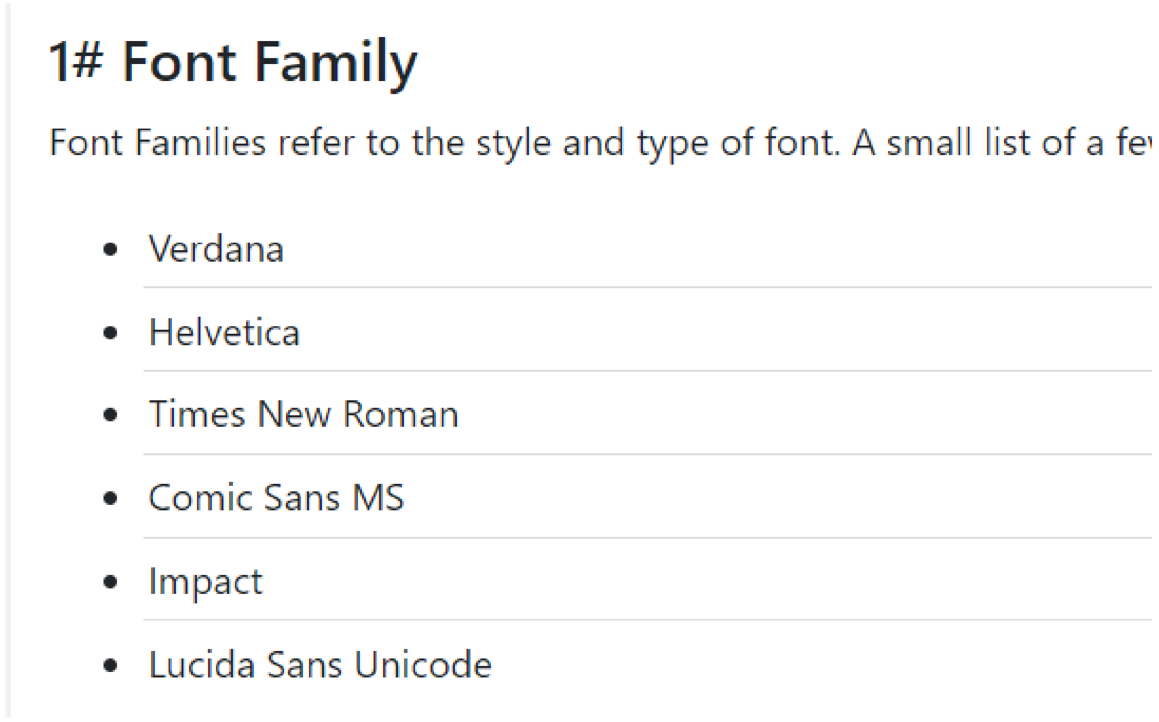
CSS Property	Values	Default	Comments
-fx-fill	<paint>	BLACK	
-fx-smooth	<boolean>	true	
-fx-stroke	<paint>	null	
-fx-stroke-type	[inside | outside | centered]	centered	
-fx-stroke-dash-array	<size>[<size>]+	null	
-fx-stroke-dash-offset	<number>	0	
-fx-stroke-line-cap	[square | butt | round]	square	
-fx-stroke-line-join	[miter | bevel | round]	miter	
-fx-stroke-miter-limit	<number>	10	
-fx-stroke-width	<size>	1	
Also has all properties of Node			

It also contains a list of standard colors you can use



aliceblue = #f0f8ff	antiquewhite = #faebd7	aqua = #00ffff	aquamarine = #7fffd4
azure = #f0ffff	beige = #f5f5dc	bisque = #ffe4c4	black = #000000
blanchedalmond = #ffebcd	blue = #0000ff	blueviolet = #8a2be2	brown = #a52a2a
burlywood = #deb887	cadetblue = #5f9ea0	chartreuse = #7fff00	chocolate = #d2691e
coral = #ff7f50	cornflowerblue = #6495ed	cornsilk = #fff8dc	crimson = #dc143c
cyan = #00ffff	darkblue = #00008b	darkcyan = #008b8b	darkgoldenrod = #b8860b
darkgray = #a9a9a9	darkgreen = #006400	darkgrey = #a9a9a9	darkkhaki = #bdb76b
darkmagenta = #8b008b	darkolivegreen = #556b2f	darkorange = #ff8c00	darkorchid = #9932cc
darkred = #8b0000	darksalmon = #e9967a	darkseagreen = #8fbc8f	darkslateblue = #483d8b
darkslategray = #2f4f4f	darkslategrey = #2f4f4f	darkturquoise = #00ced1	darkviolet = #9400d3
deeppink = #ff1493	deepskyblue = #00bfff	dimgray = #696969	dimgrey = #696969
dodgerblue = #1e90ff	firebrick = #b22222	floralwhite = #fffaf0	forestgreen = #228b22
fuchsia = #ff00ff	gainsboro = #dcdcdc	ghostwhite = #f8f8ff	gold = #ffd700
goldenrod = #daa520	gray = #808080	green = #008000	greenyellow = #adff2f
grey = #808080	honeydew = #f0fff0	hotpink = #ff69b4	indianred = #cd5c5c
indigo = #4b0082	ivory = #fffff0	khaki = #f0e68c	lavender = #e6e6fa
lavenderblush = #fff0f5	lawngreen = #7cfc00	lemonchiffon = #fffacd	lightblue = #add8e6

https://coderslegacy.com/java/javafx-font/

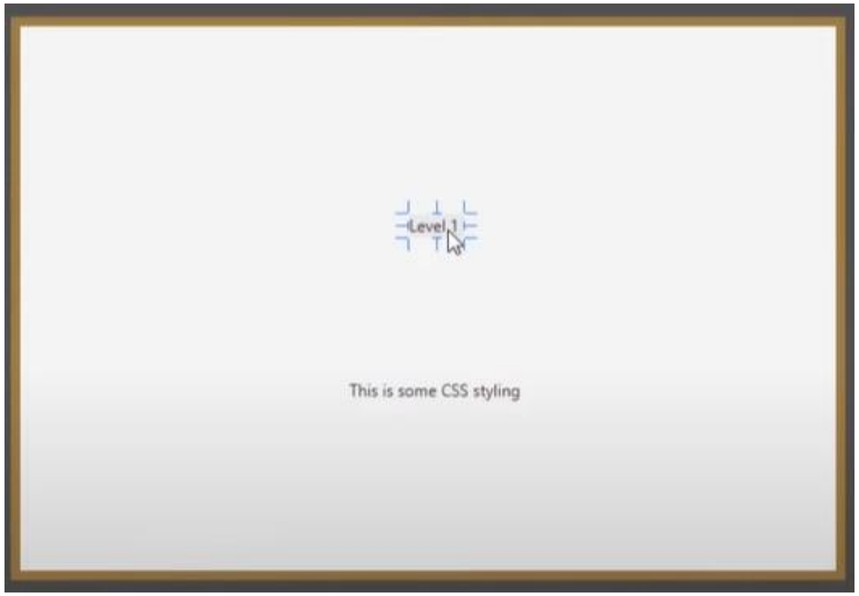


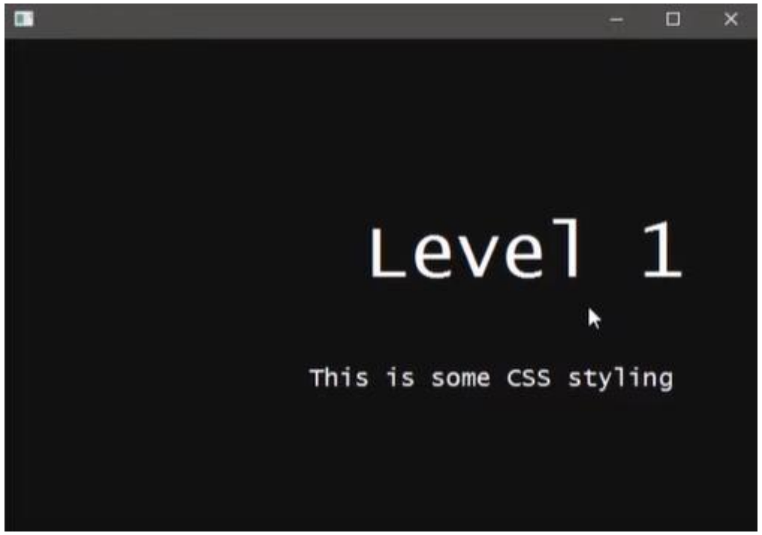
1# Font Family

Font Families refer to the style and type of font. A small list of a fe

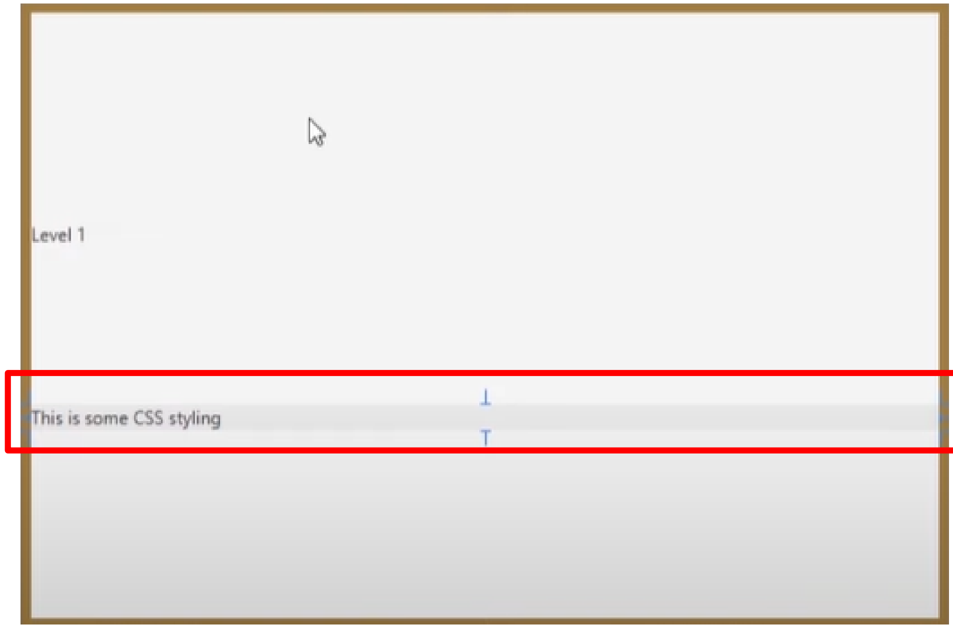
- Verdana
- Helvetica
- Times New Roman
- Comic Sans MS
- Impact
- Lucida Sans Unicode

A Common Mistake



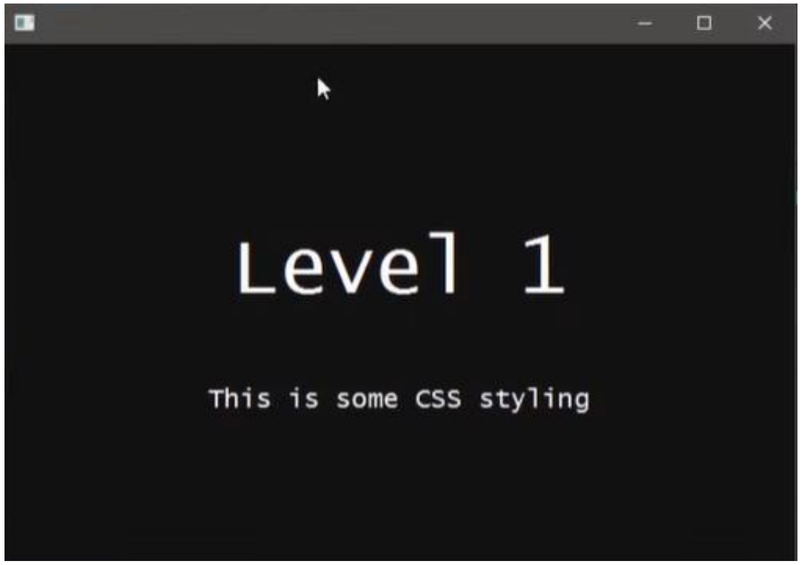


It won’t center unless the container is big enough



Level 1

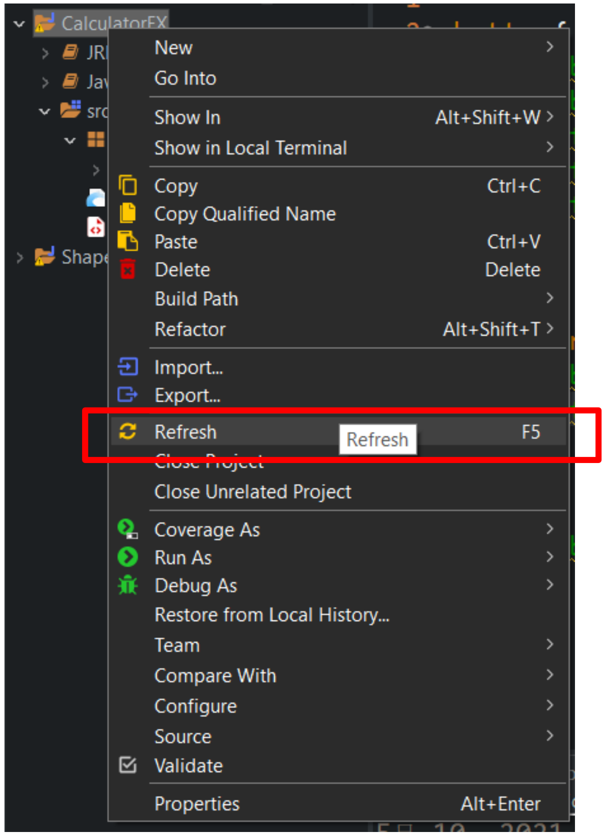
This is some CSS styling



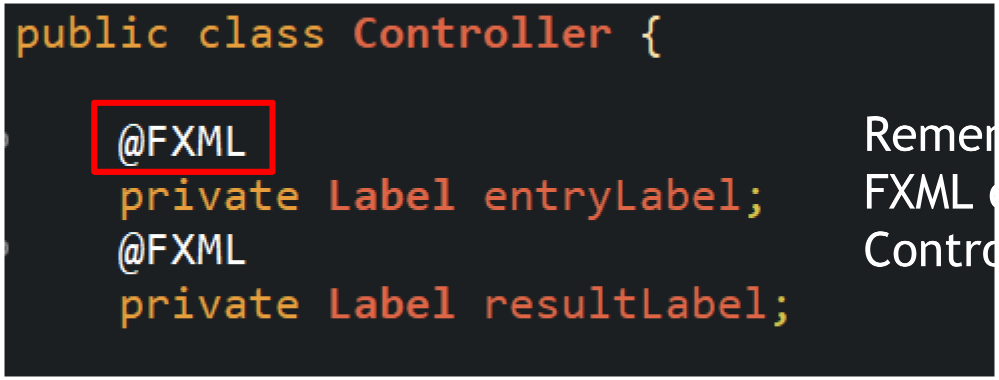
Level 1

This is some CSS styling

Mistake 2


CalculatorFX
JRE
Java
src
Shape
New
Go Into
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Refresh
Close Project
Close Unrelated Project
Coverage As
Run As
Debug As
Restore from Local History...
Team
Compare With
Configure
Source
Validate
Properties Alt+Enter

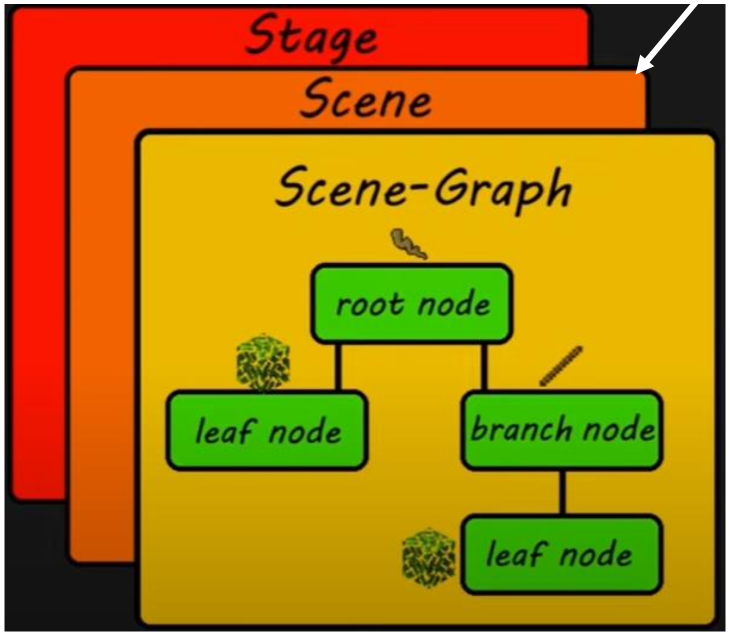
Mistake 3



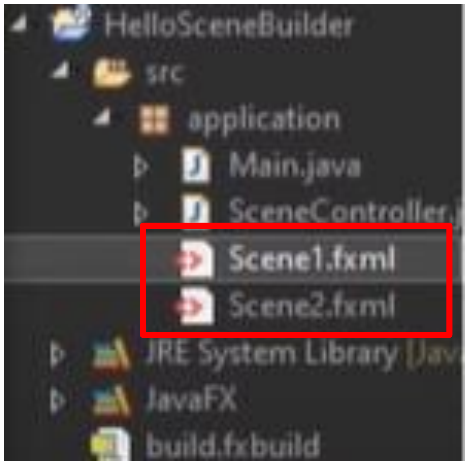
```
public class Controller {

    @FXML
    private Label entryLabel;
    @FXML
    private Label resultLabel;
```

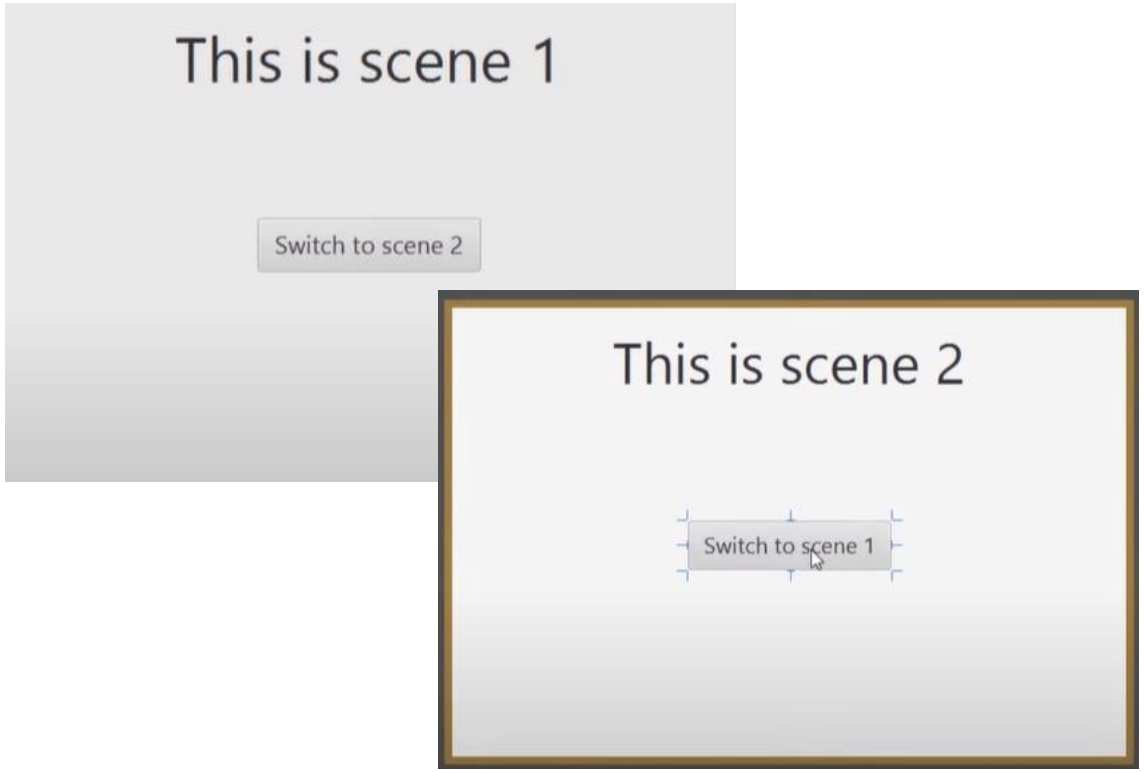
Remer
FXML
Contro

6.
Switch Scenes

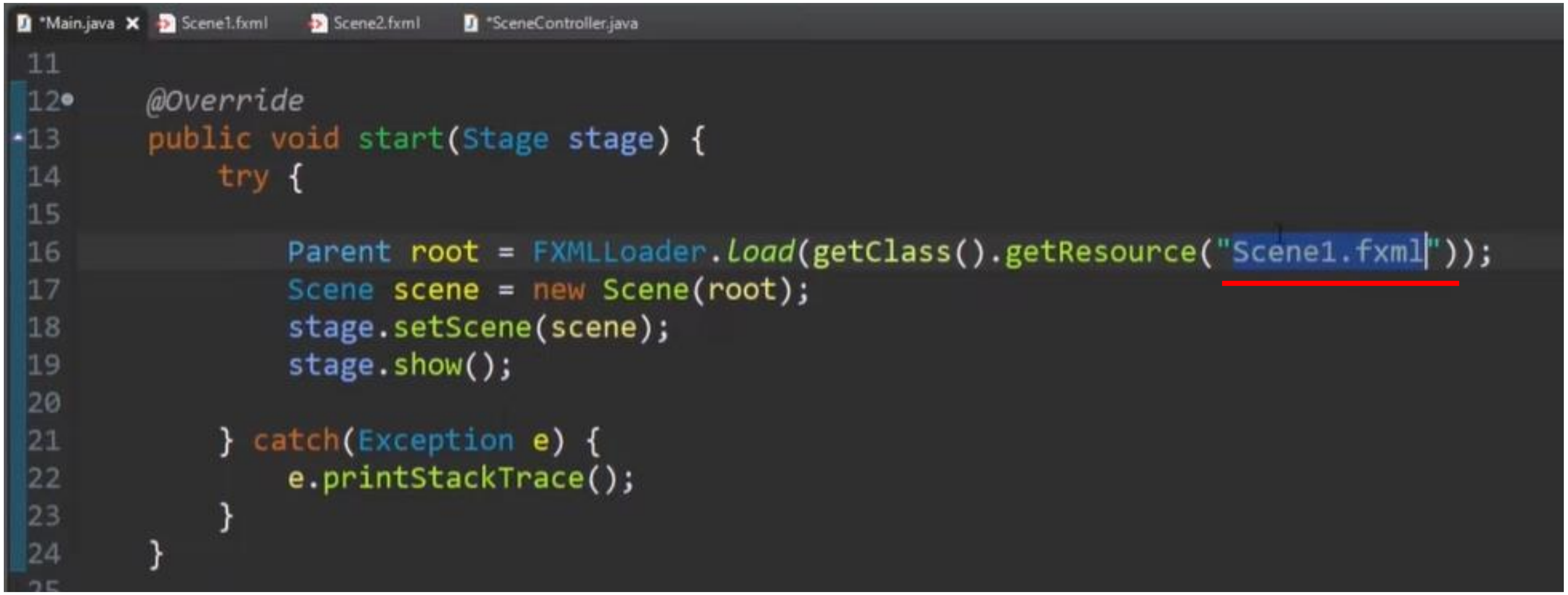

Stage
Scene
Scene-Graph
root node
leaf node
branch node
leaf node

Step 1: Create 2 FXML files



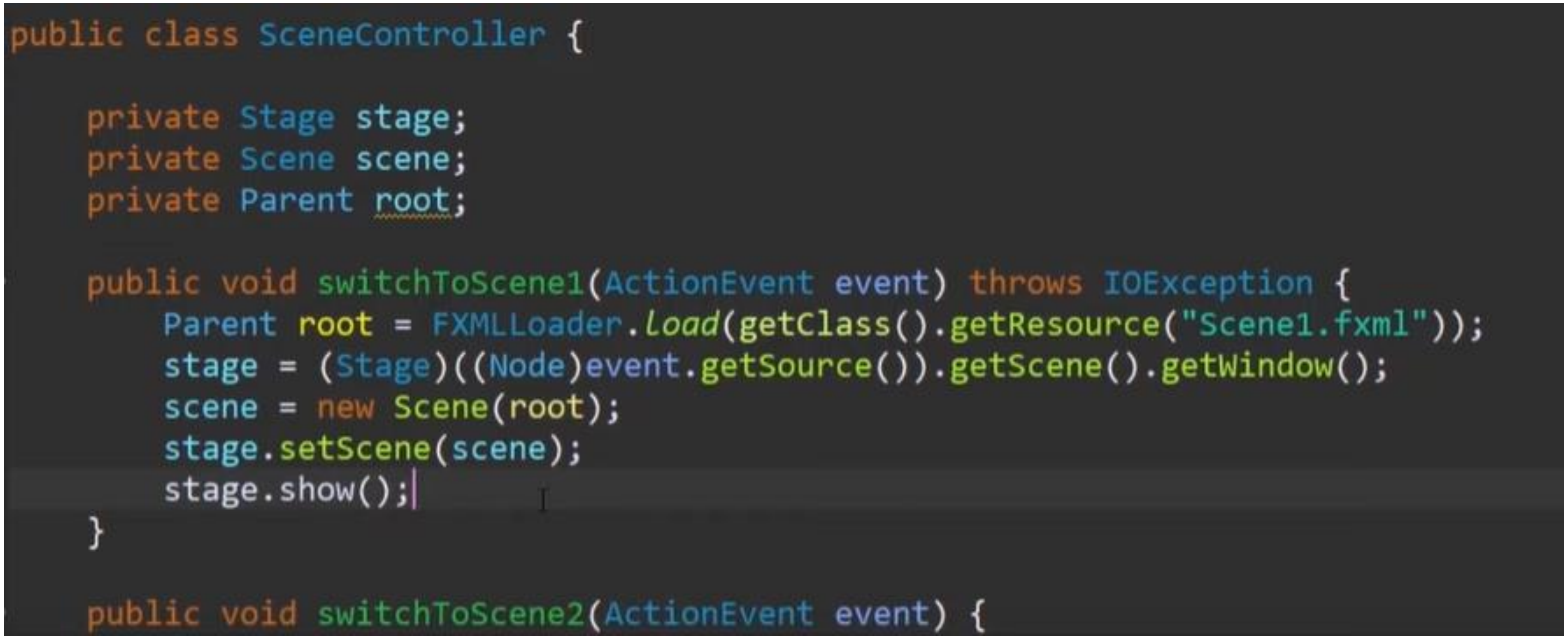


Step 2: Place Scene1 at the root node



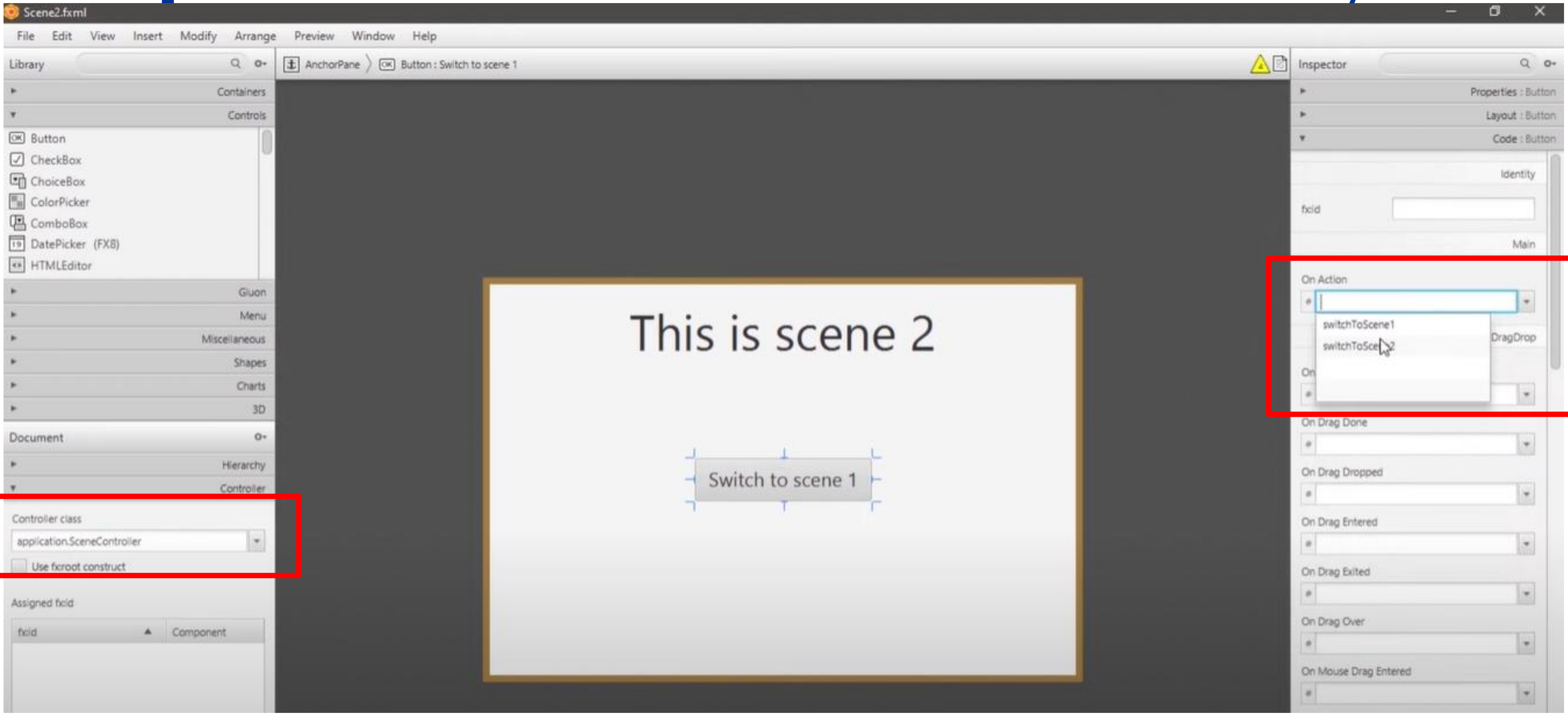
```
    @Override
    public void start(Stage stage) {
        try {

            Parent root = FXMLLoader.load(getClass().getResource("Scene1.fxml"));
            Scene scene = new Scene(root);
            stage.setScene(scene);
            stage.show();

        } catch(Exception e) {
            e.printStackTrace();
        }
    }
```

Step 3: Create a Controller class



```
public class SceneController {

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToScene1(ActionEvent event) throws IOException {
        Parent root = FXMLLoader.load(getClass().getResource("Scene1.fxml"));
        stage = (Stage)((Node)event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }

    public void switchToScene2(ActionEvent event) {
```

Step 4: Bind the Controller class to your



Controller class: application.SceneController

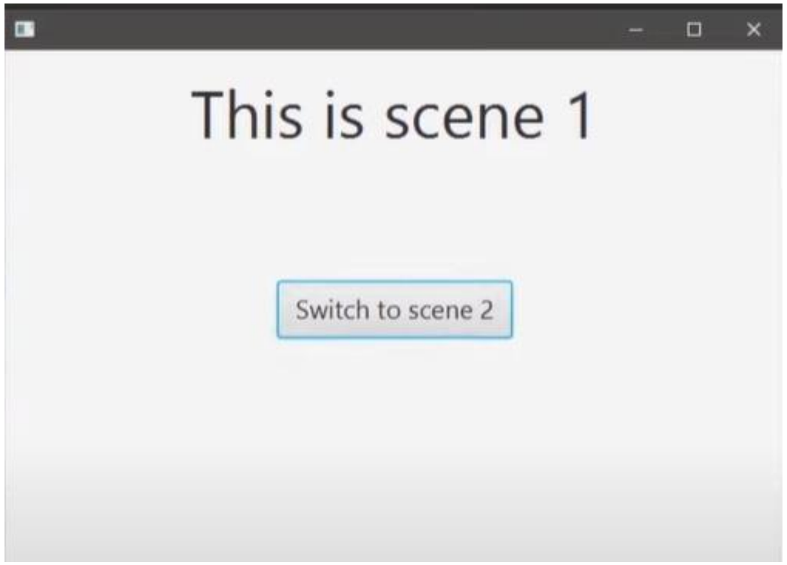
Use fx:root construct

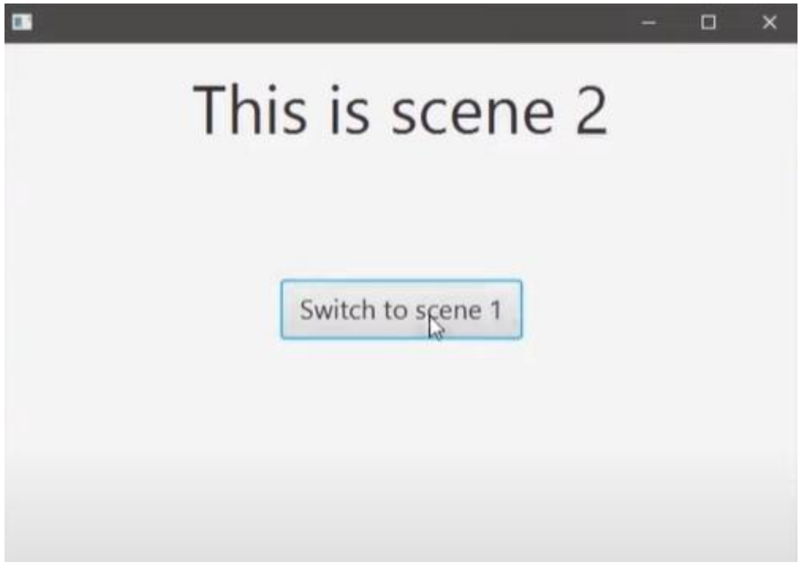
Assigned fx:id

fx:id	Component

On Action:
- switchToScene1
- switchToScene2

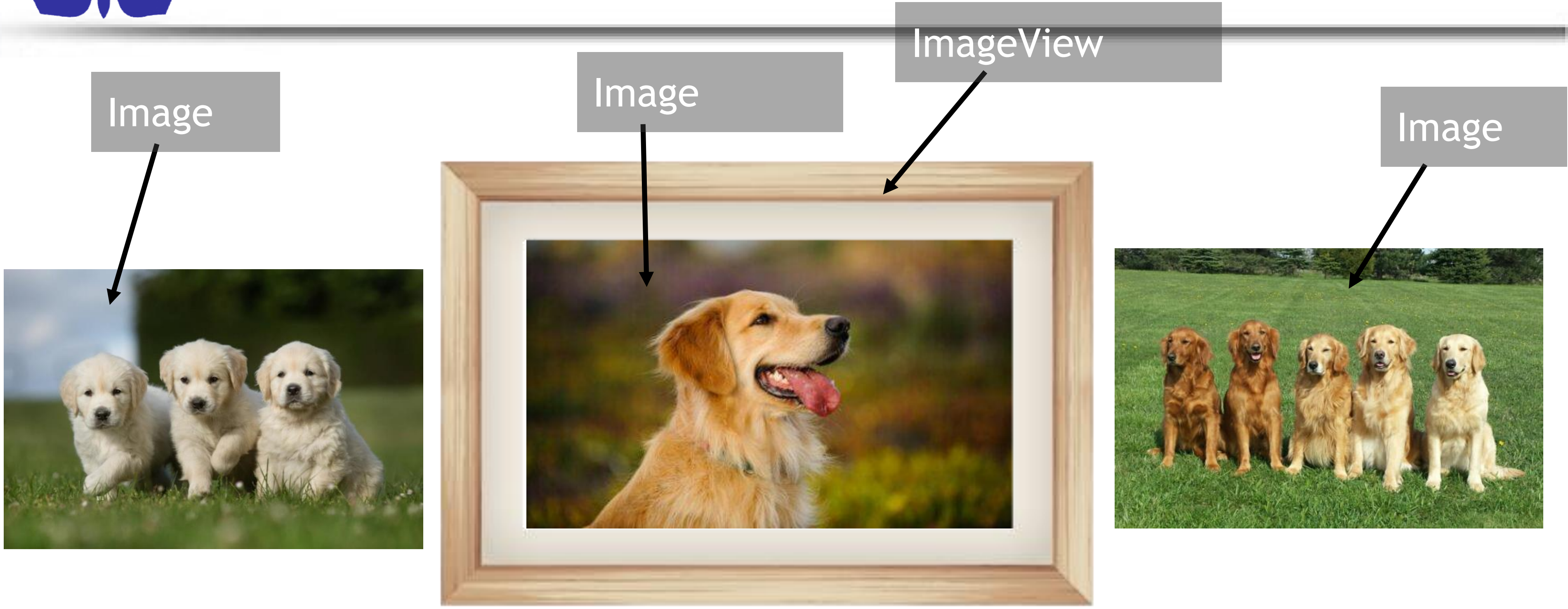
Try it yourself!





7.
More FX Components

7.1 ImageView


ImageView
Image
Image
Image

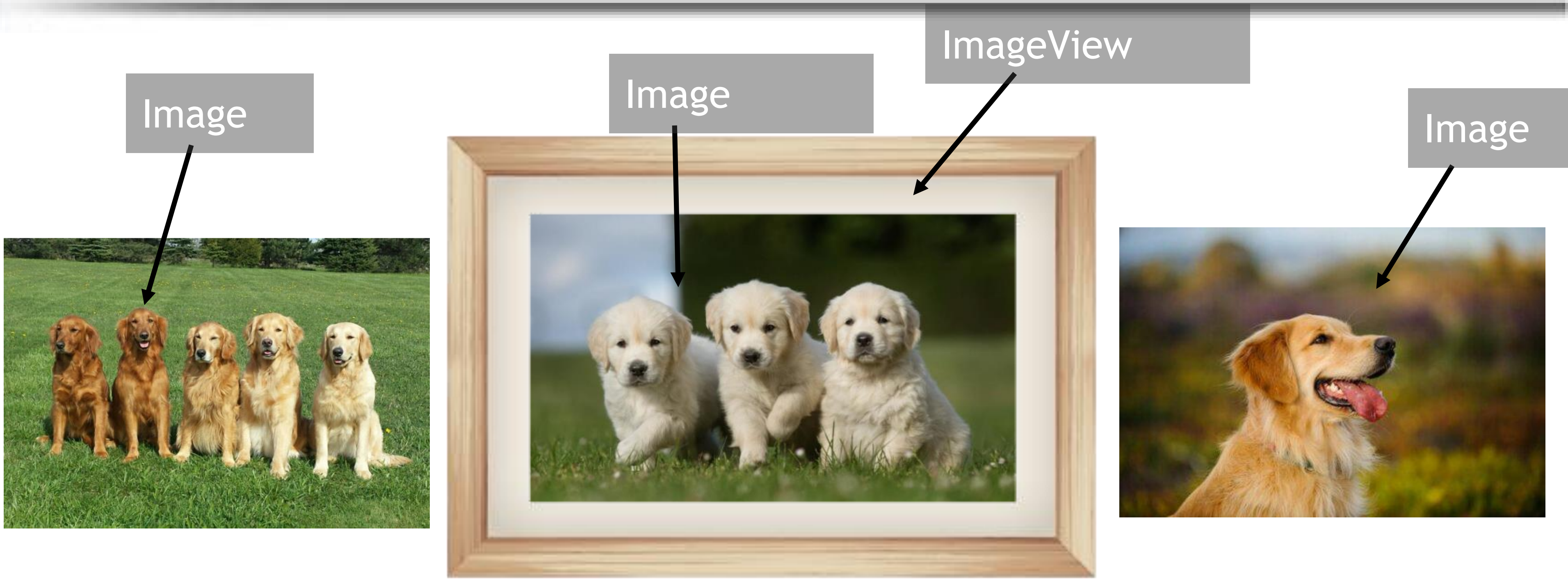


Image
Image
ImageView
Image

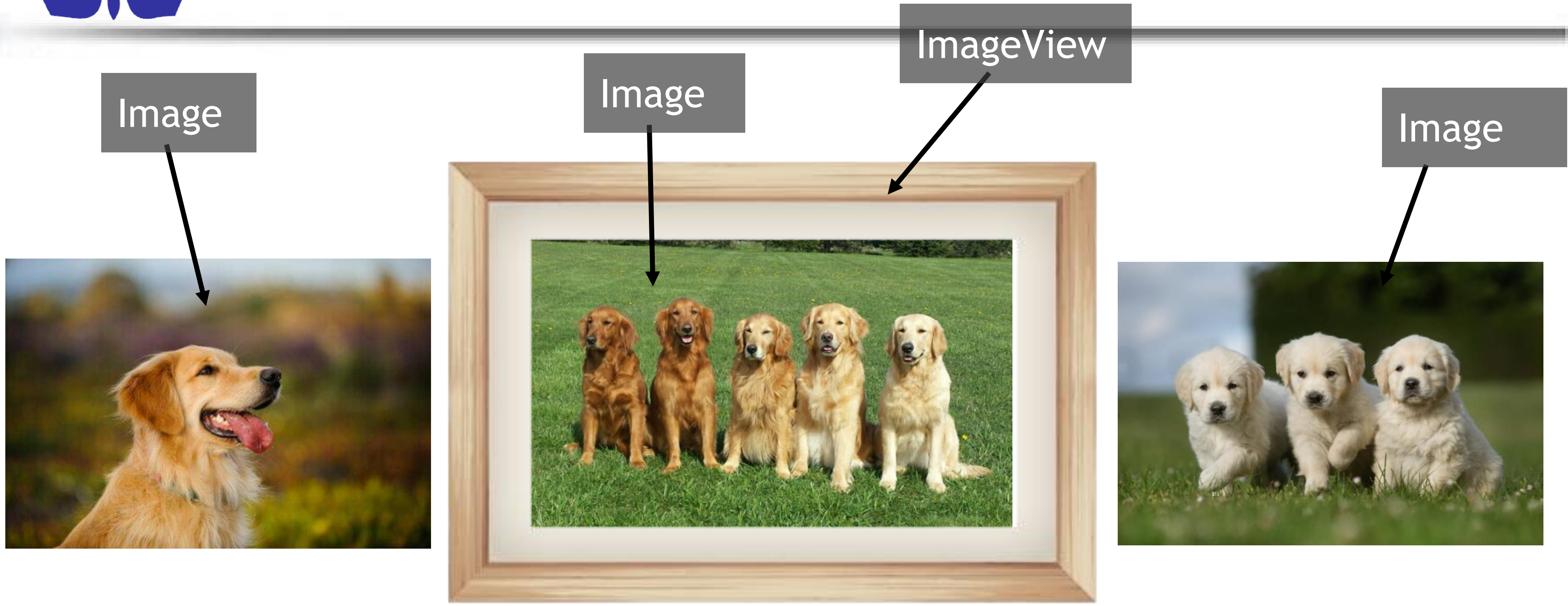
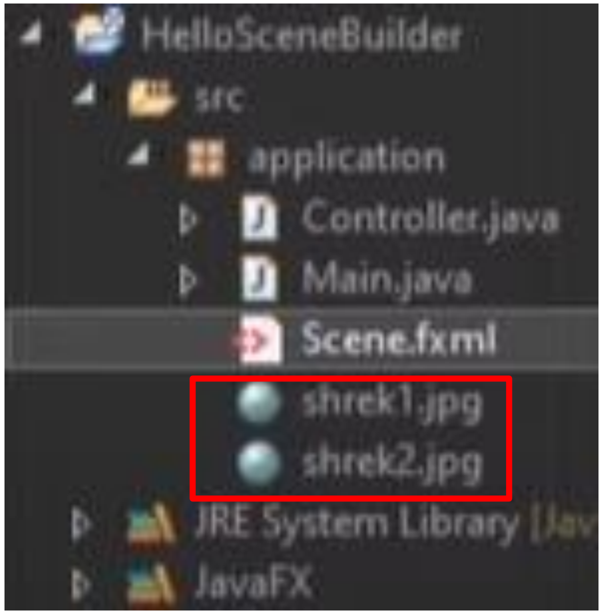
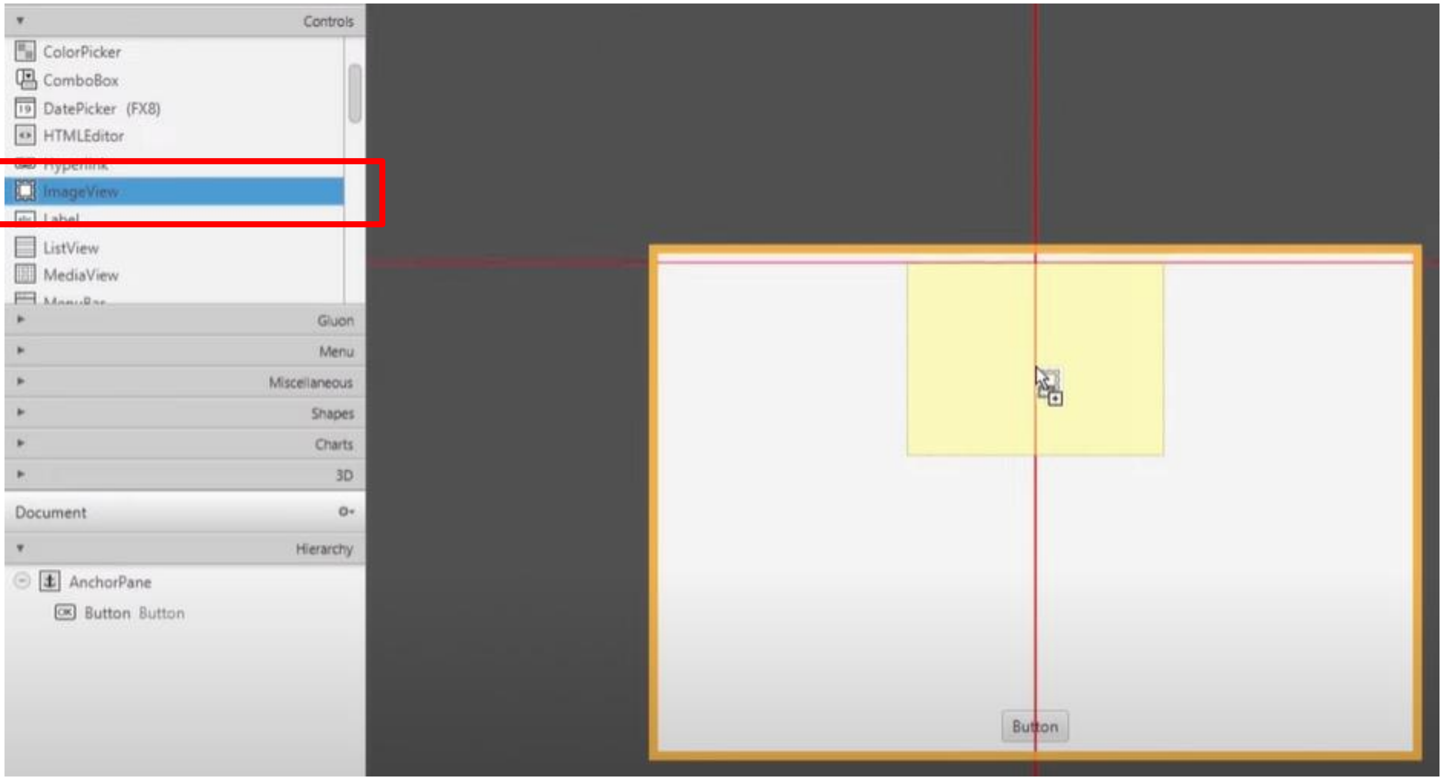


Image
Image
ImageView
Image

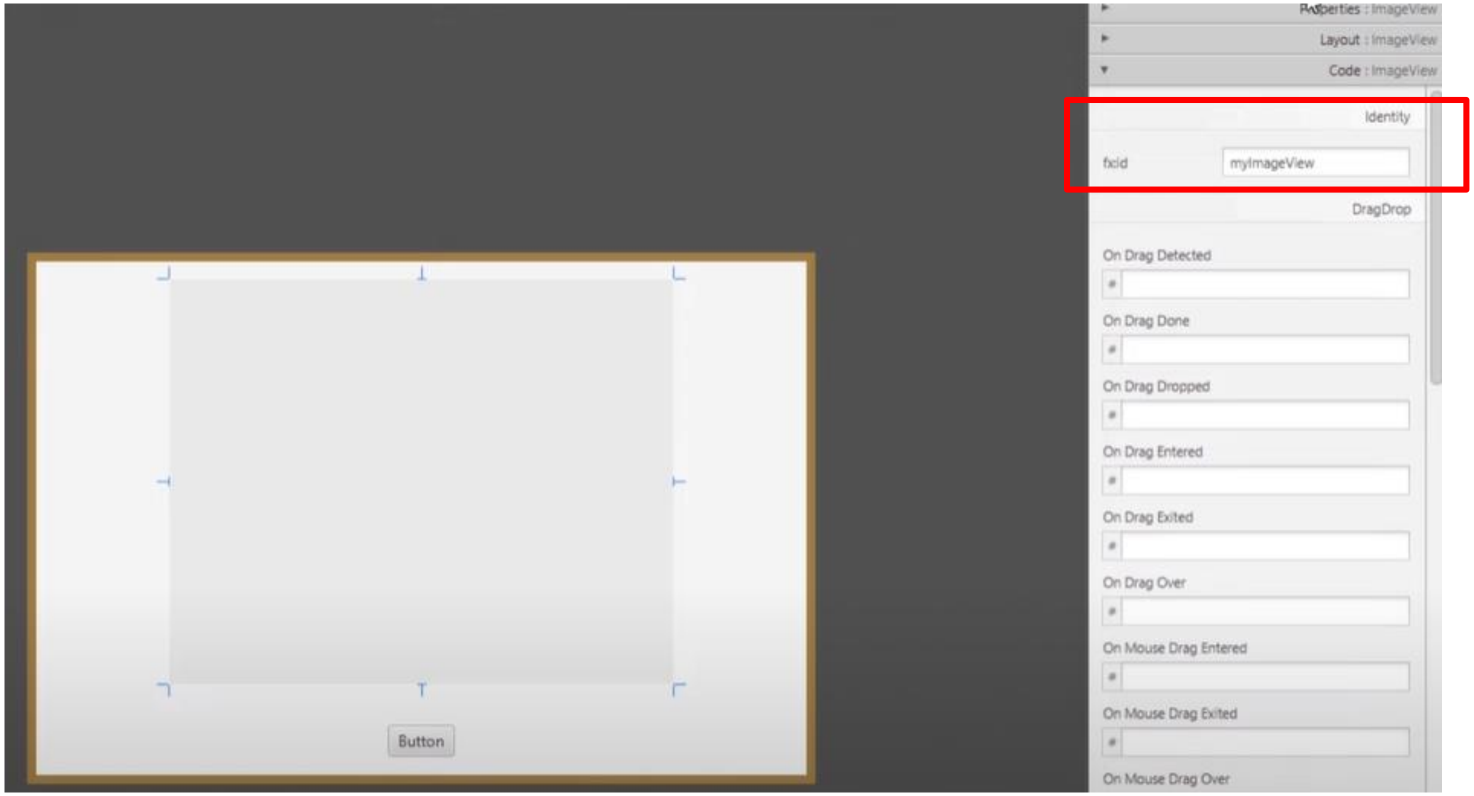
Step 1: Copy your images under your package



Step 1: Create a FXML file with ImageView

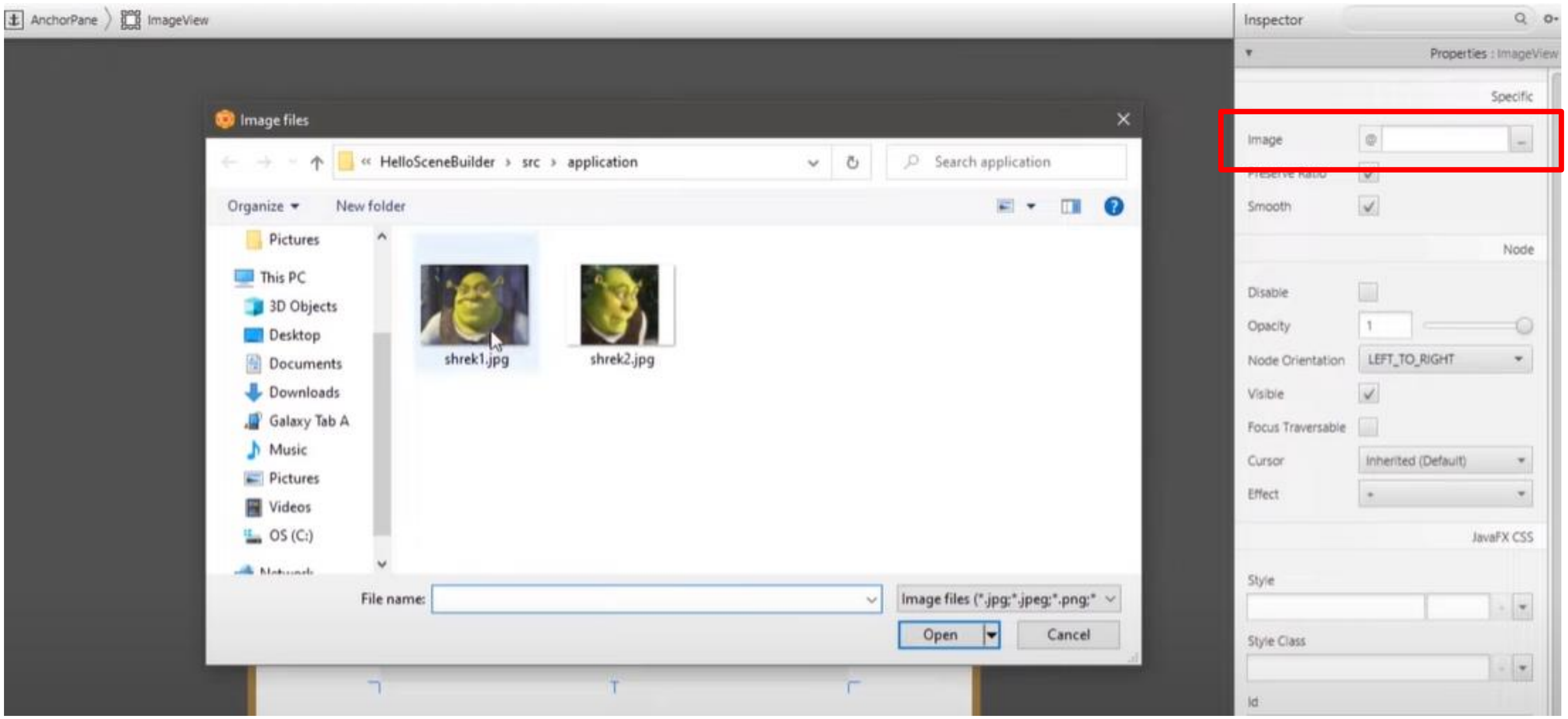


Remember to give it an fxid

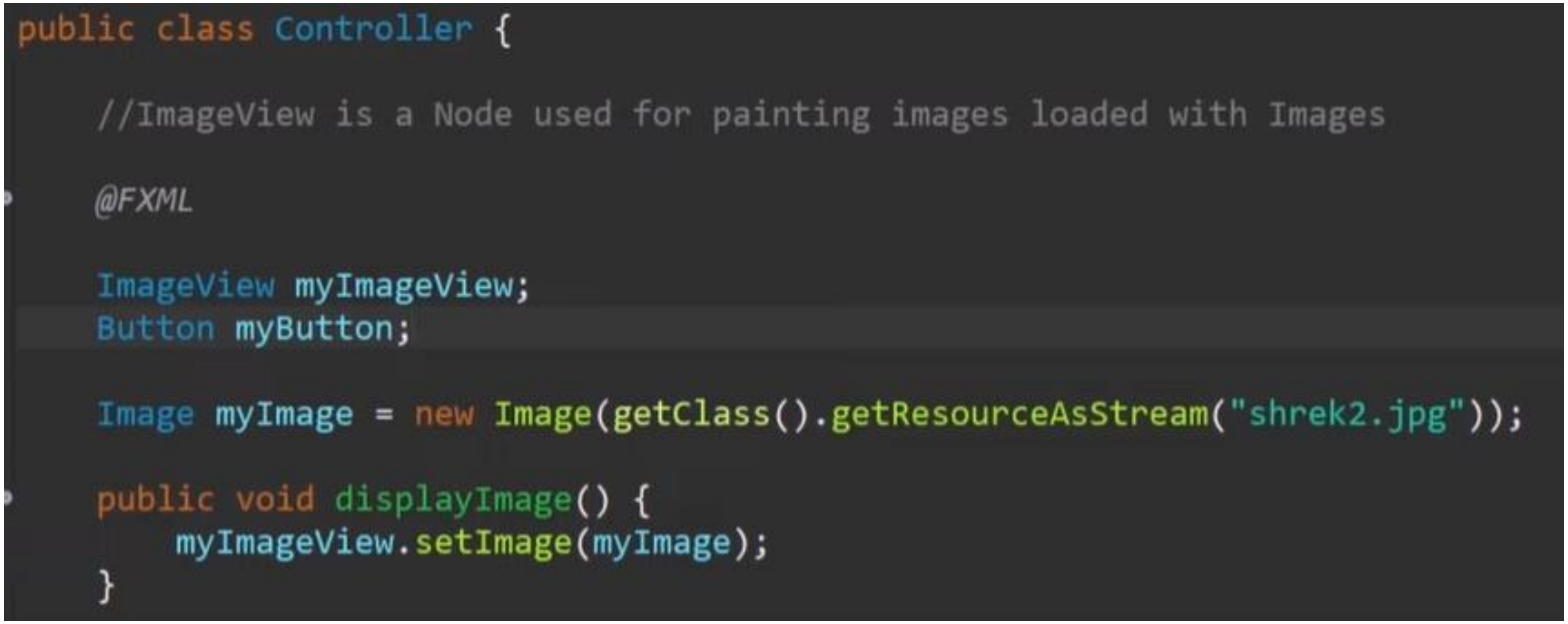


fxid: myImageView

Choose an initial Image for your

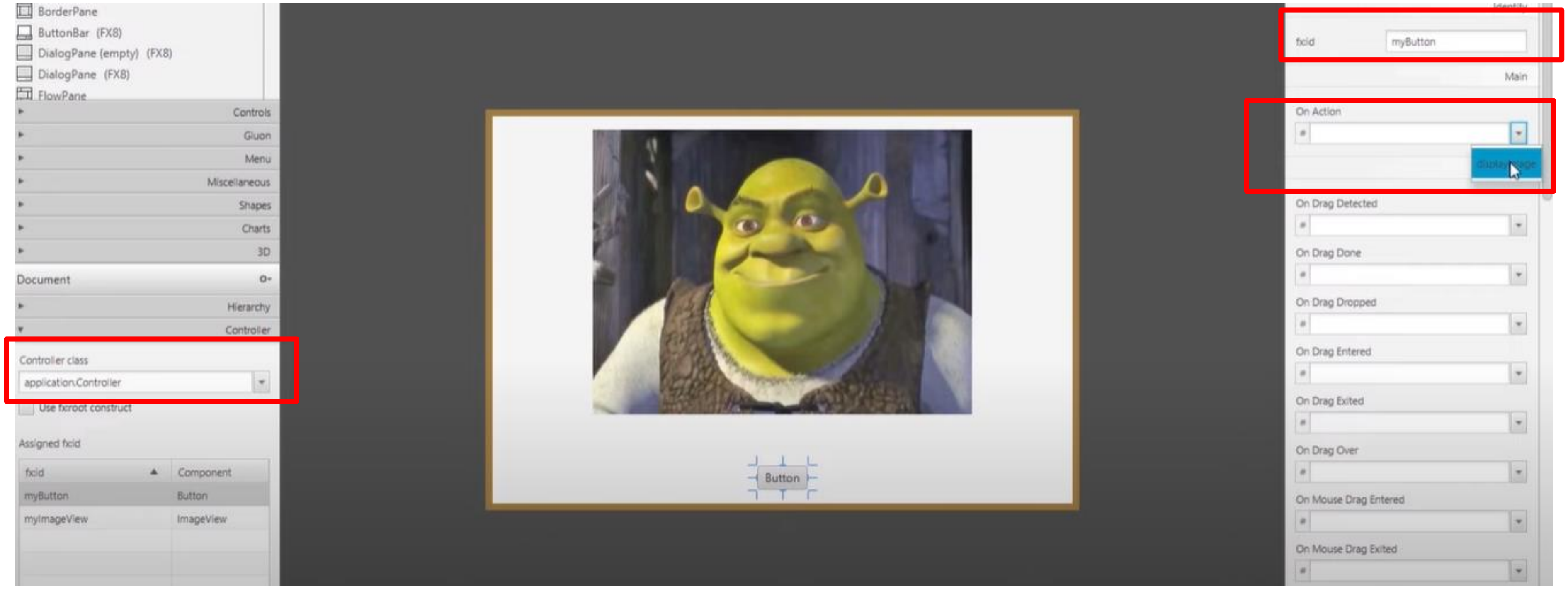


Step 2: Add your Controller class



```
public class Controller {

    //ImageView is a Node used for painting images loaded with Images

    @FXML

    ImageView myImageView;
    Button myButton;

    Image myImage = new Image(getClass().getResourceAsStream("shrek2.jpg"));

    public void displayImage() {
        myImageView.setImage(myImage);
    }
```

Step 3: Bind your function to the button



Controller class: application.Controller

Use fx:root construct

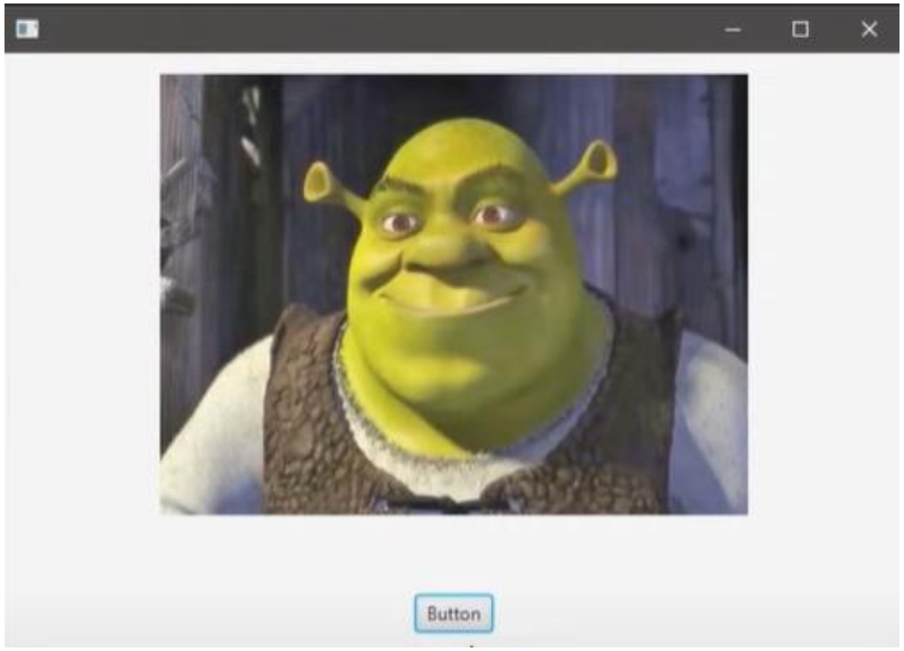
Assigned fx:id

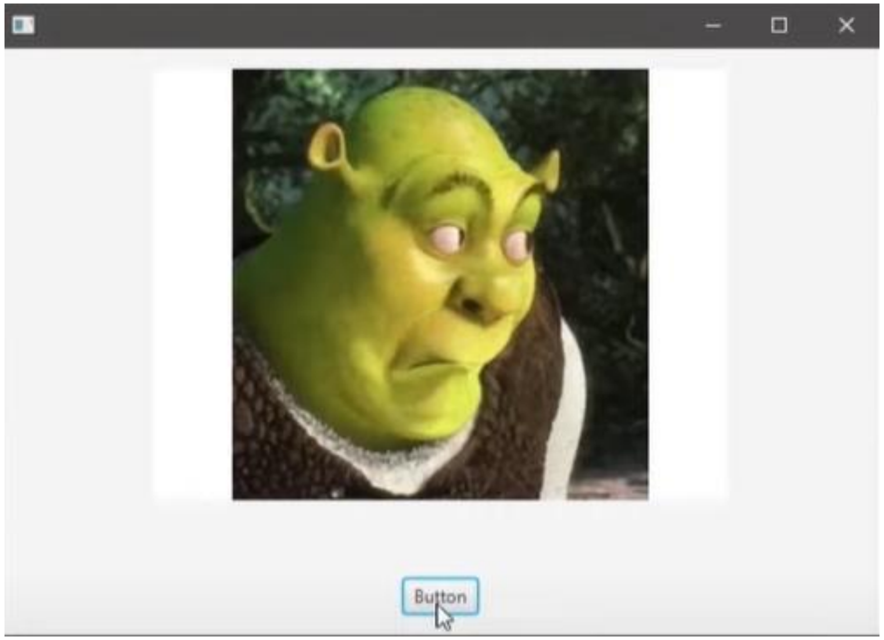
fx:id	Component
myButton	Button
myImageView	ImageView

fx:id: myButton

On Action: # displayImage

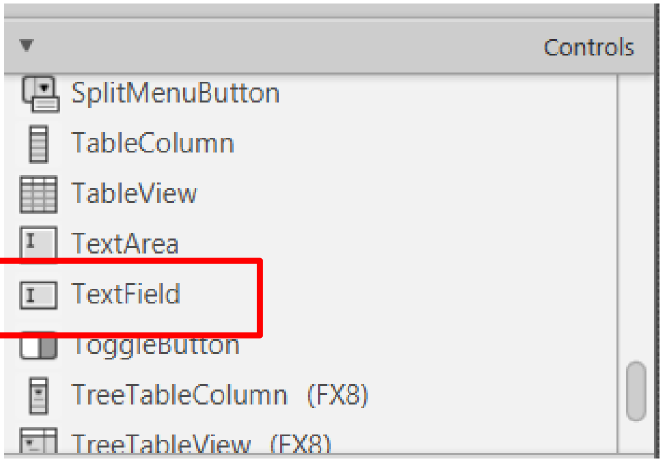
Try it!

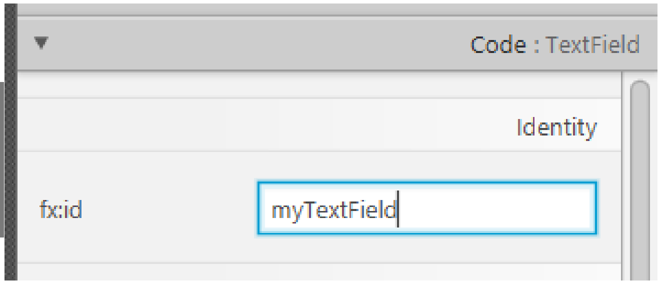




7.2 TextField

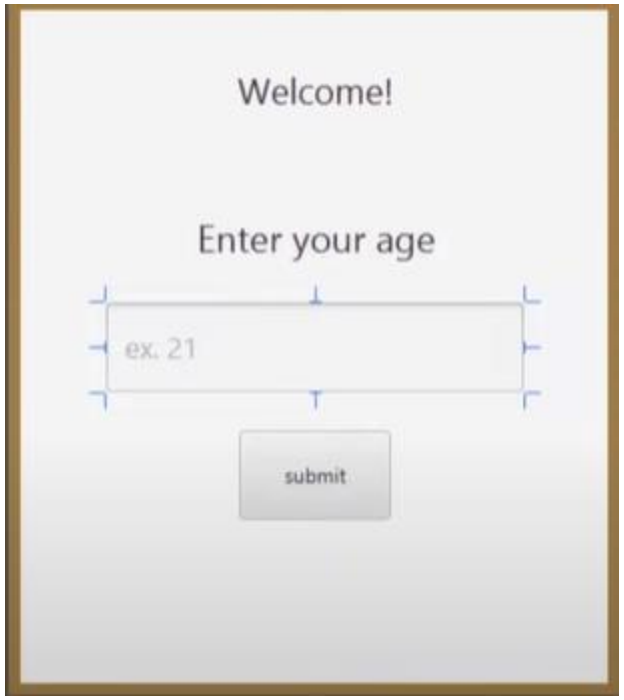
Add a fxml file with a TextField




Code : TextField

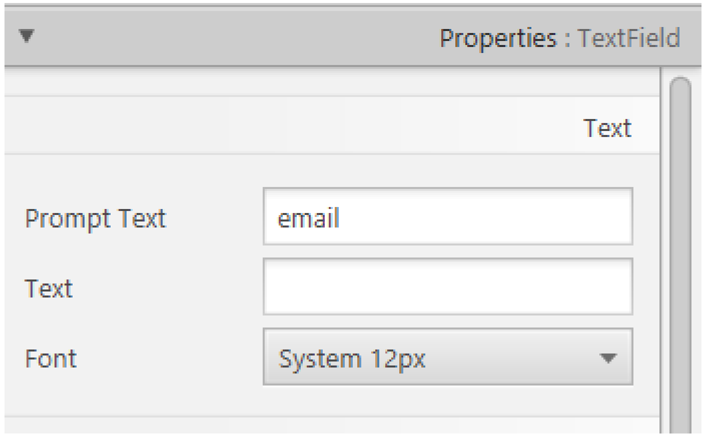
Identity

fx:id myTextField



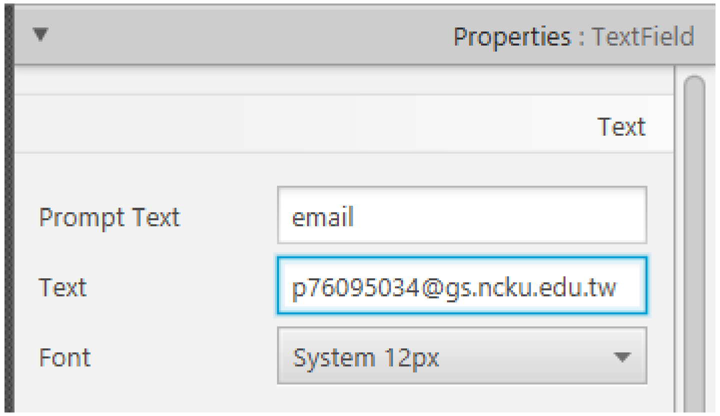
Don’t forget the fxid

Prompt Text vs. Text


Properties : TextField

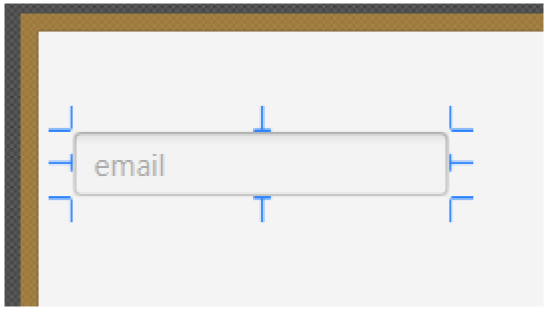
Text

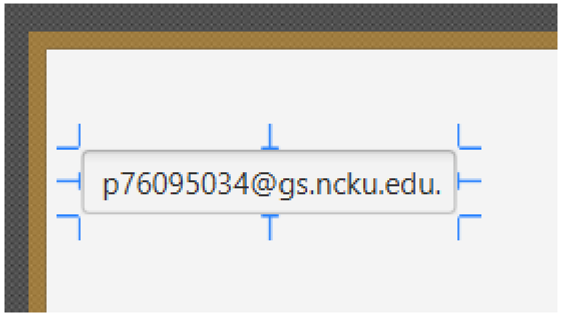
Prompt Text	email
Text	
Font	System 12px


Properties : TextField

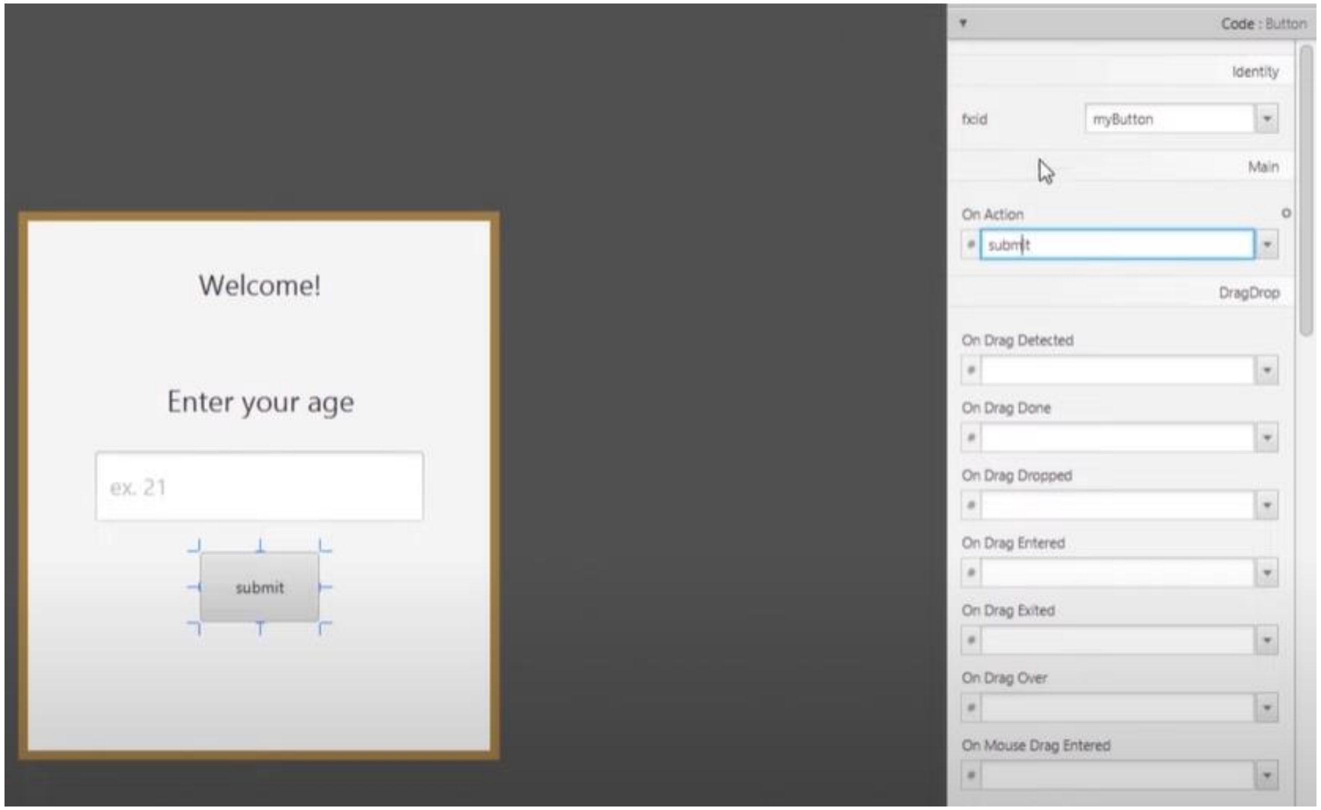
Text

Prompt Text	email
Text	p76095034@gs.ncku.edu.tw
Font	System 12px



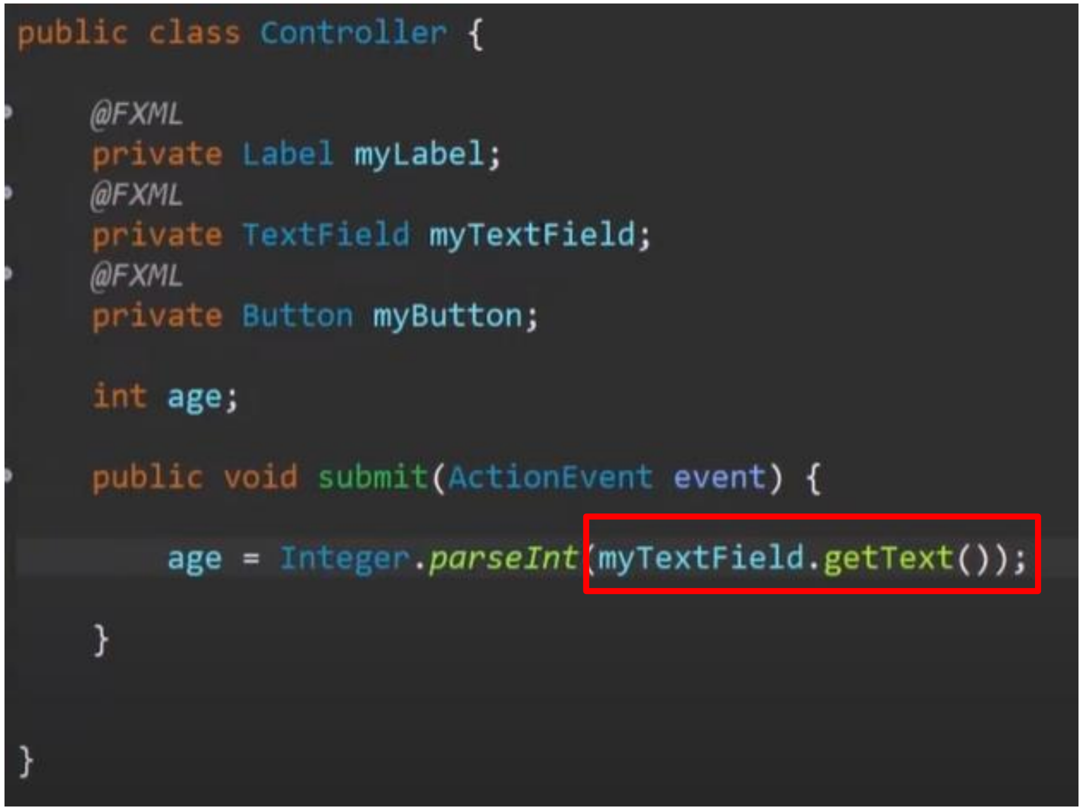


Link a “submit” function to the button



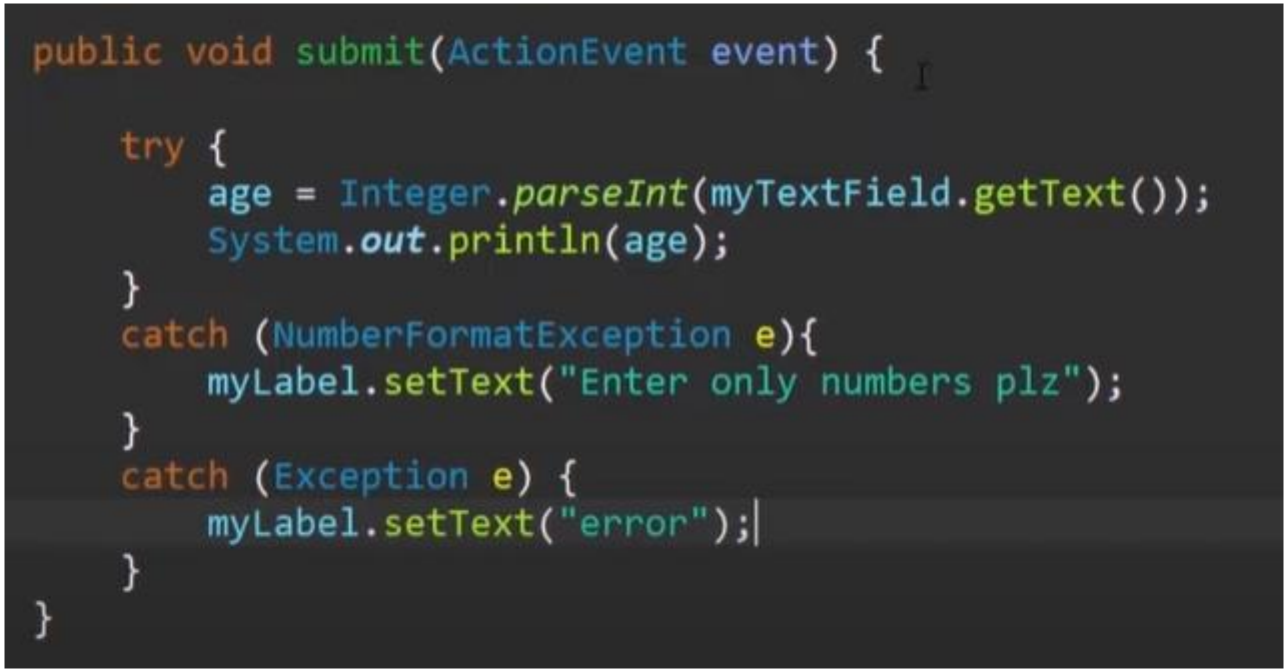
fx:id: myButton
On Action: # submit

Get the text within the TextField by using



```
public class Controller {

    @FXML
    private Label myLabel;
    @FXML
    private TextField myTextField;
    @FXML
    private Button myButton;

    int age;

    public void submit(ActionEvent event) {

        age = Integer.parseInt(myTextField.getText());

    }


}
```

To prevent users entering the wrong input

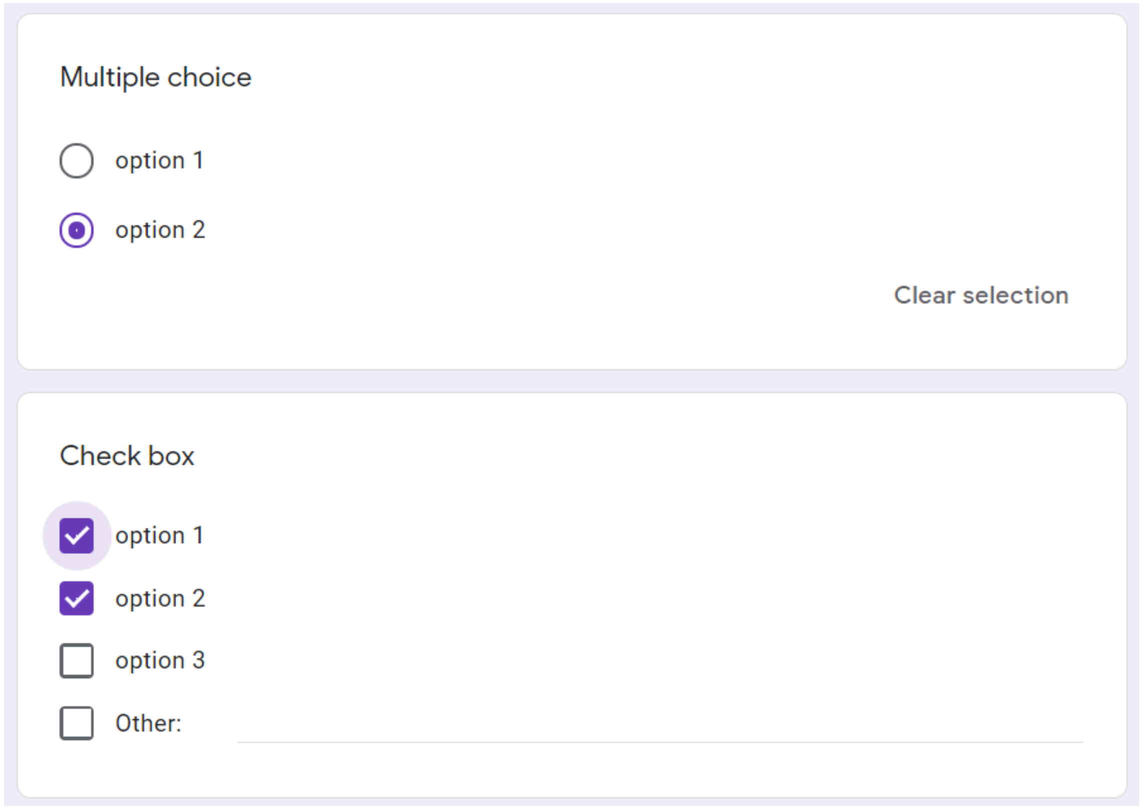


```
public void submit(ActionEvent event) {

    try {
        age = Integer.parseInt(myTextField.getText());
        System.out.println(age);
    }
    catch (NumberFormatException e){
        myLabel.setText("Enter only numbers plz");
    }
    catch (Exception e) {
        myLabel.setText("error");
    }
}
```

7.3 CheckBox and RadioButton

RadioButtons

CheckBoxes



Multiple choice

( ) option 1

[✓ option 2]

Check box

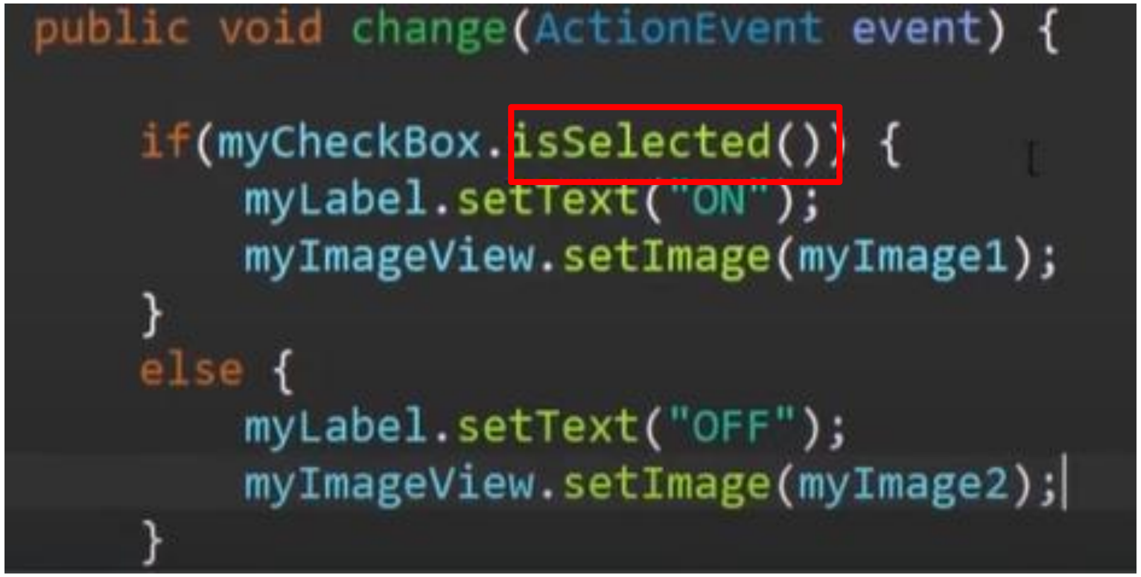
[✓ option 1]

[✓ option 2]

☐ option 3

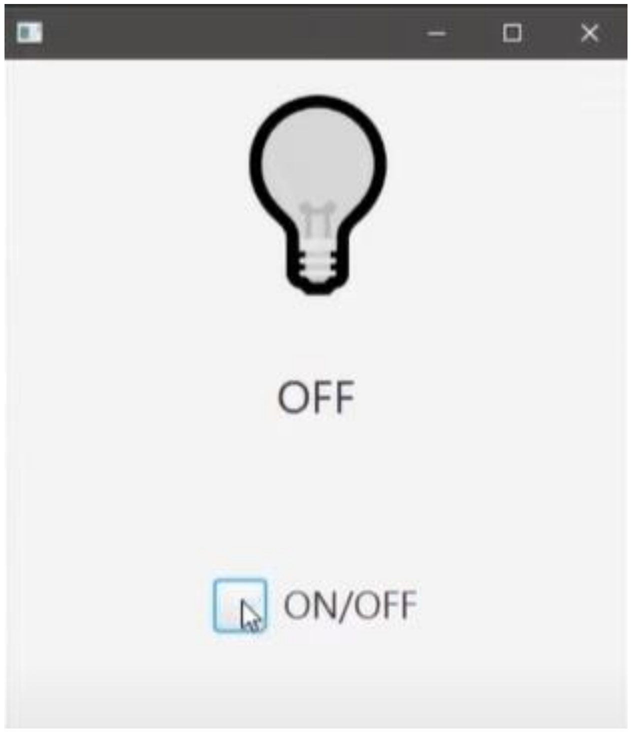
☐ Other:

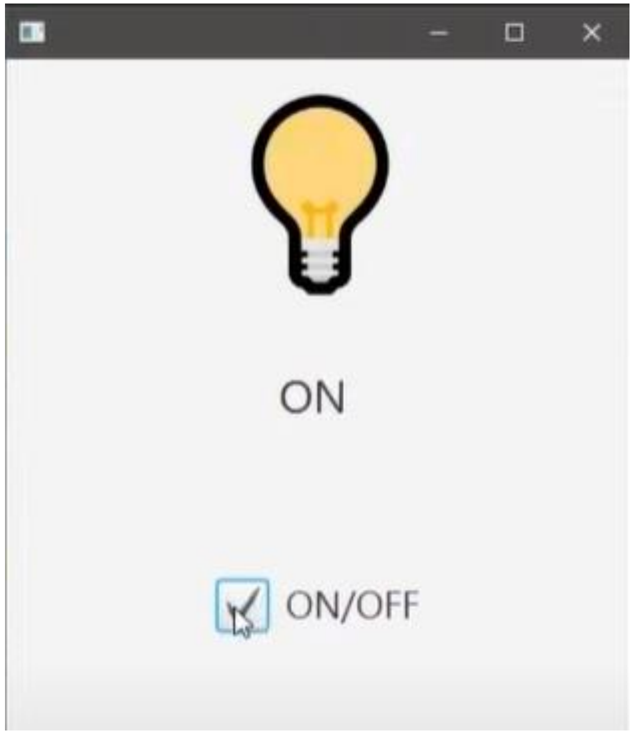
See if the CheckBox or RadioButton is selected by using isSelected()



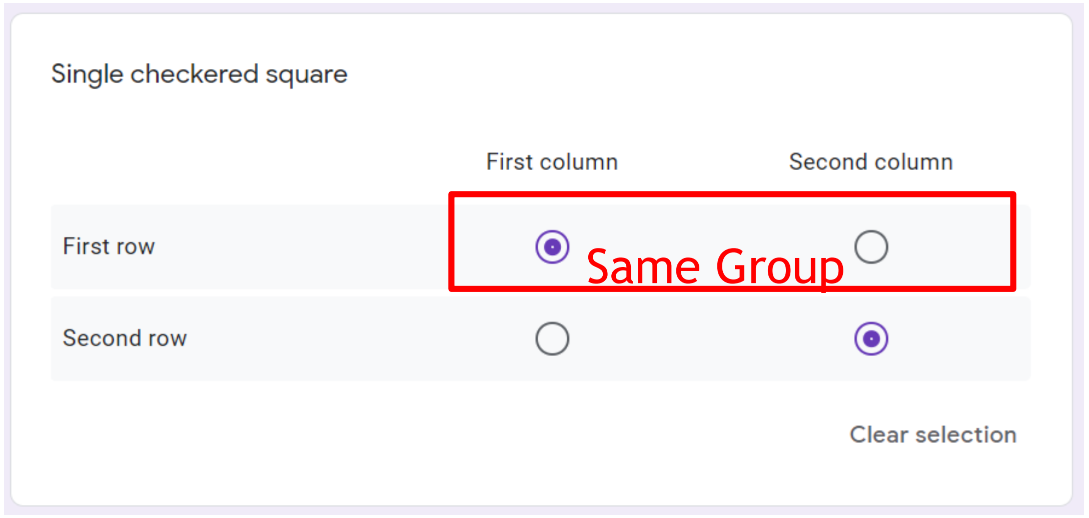
```
public void change(ActionEvent event) {

    if(myCheckBox.isSelected()) {
        myLabel.setText("ON");
        myImageView.setImage(myImage1);
    }
    else {
        myLabel.setText("OFF");
        myImageView.setImage(myImage2);
    }
```

Try it!





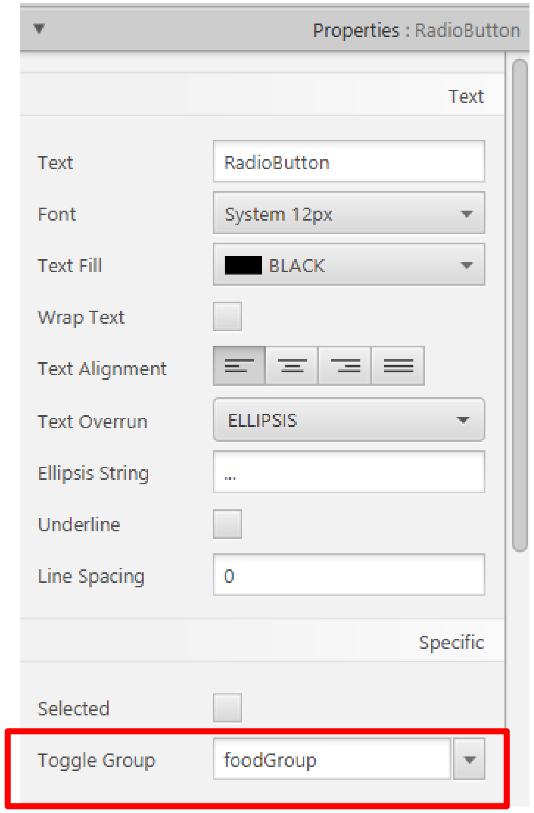
RadioButtons should belong to a group



Single checkered square

	First column	Second column
First row	●	○
Second row	○	●

Clear selection



Properties : RadioButton

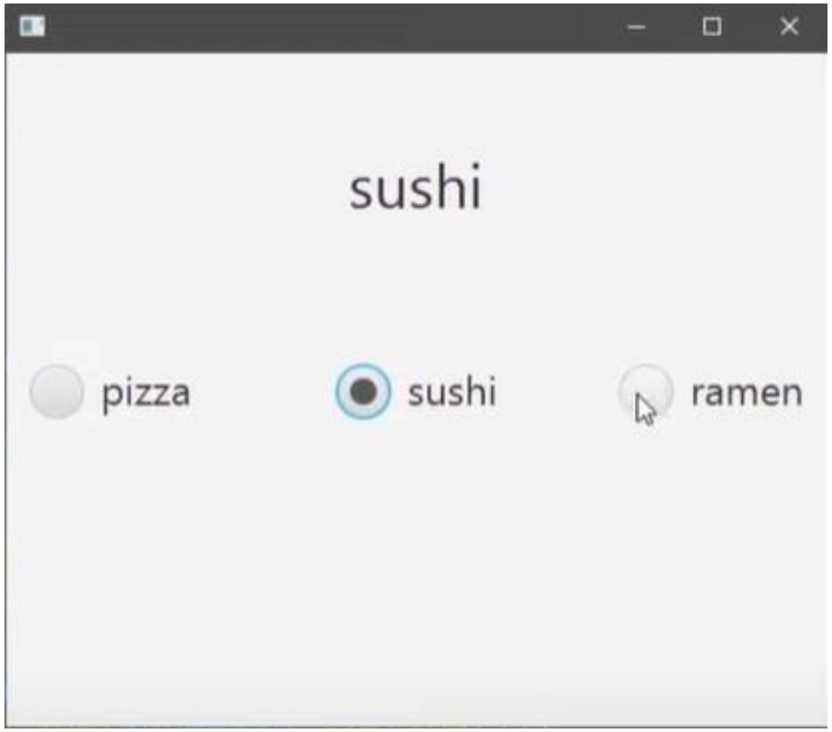
Text

Text	RadioButton
Font	System 12px
Text Fill	■ BLACK
Wrap Text	☐
Text Alignment	
Text Overrun	ELLIPSIS
Ellipsis String	...
Underline	☐
Line Spacing	0

Specific

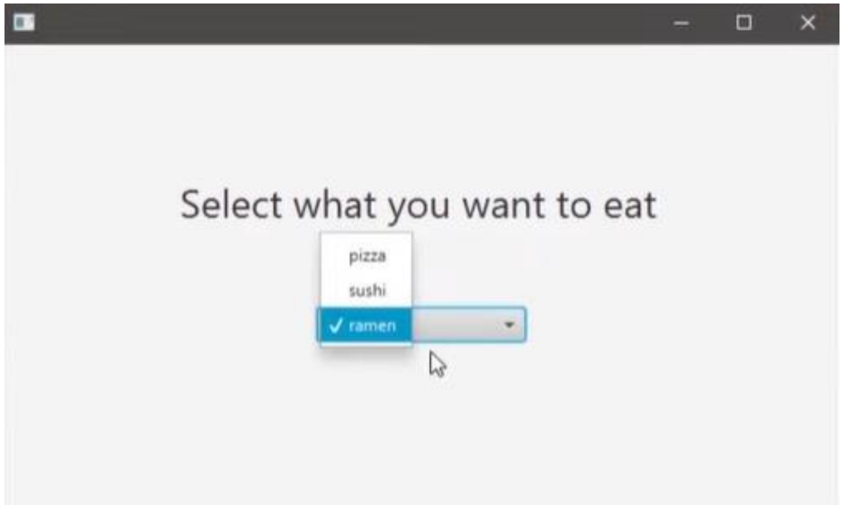
Selected	☐
Toggle Group	foodGroup

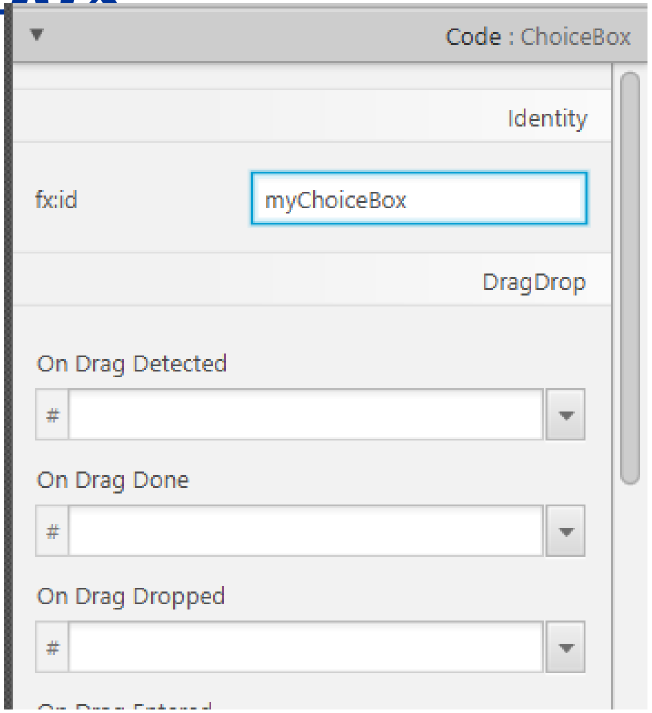
See that only one option can be selected at



7.4 ChoiceBox

Notice there is not onAction method for ChoiceBox




Code : ChoiceBox

Identity

fx:id myChoiceBox

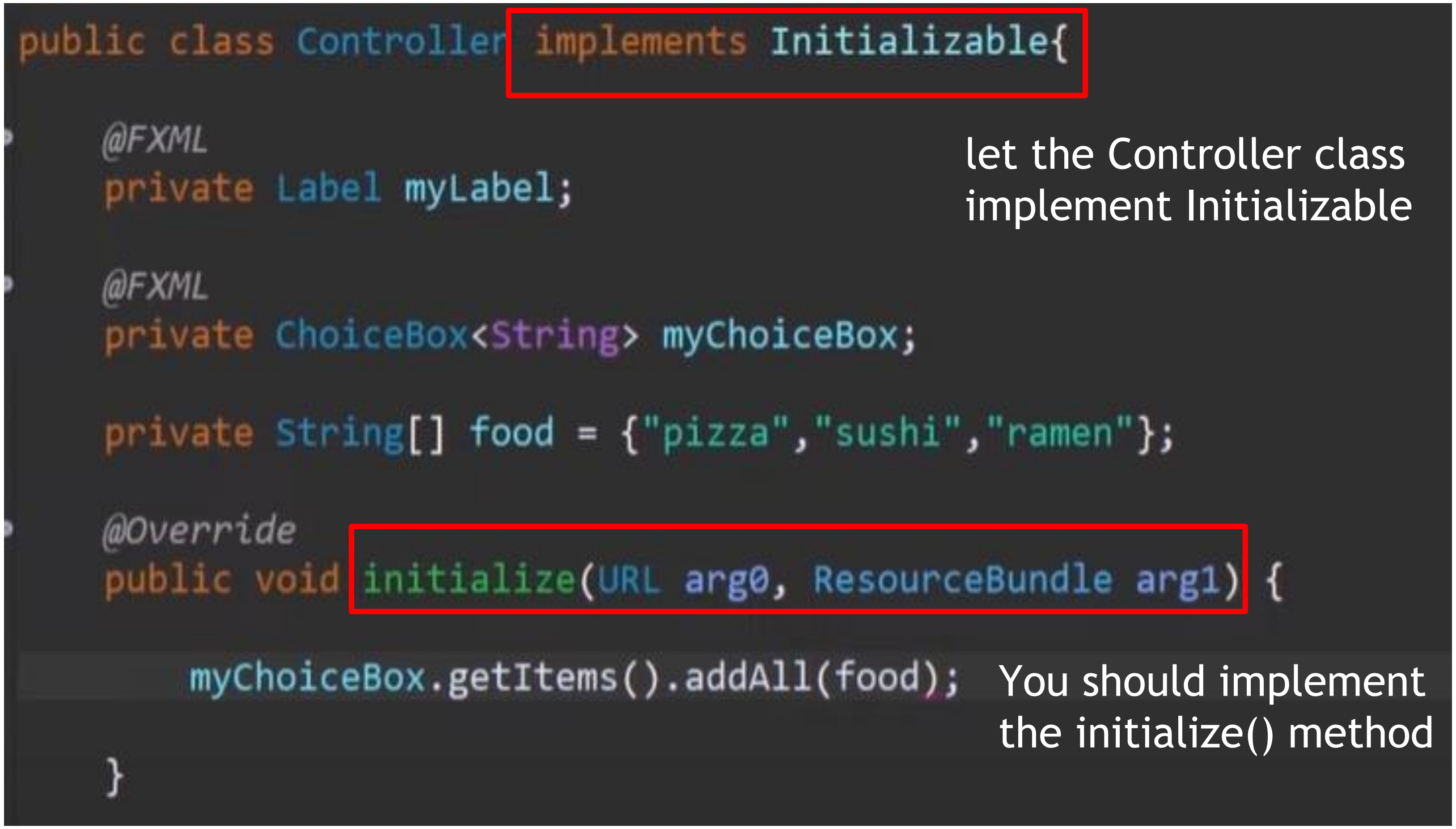
DragDrop

On Drag Detected

On Drag Done

On Drag Dropped

We have to initialize the ChoiceBox after the root has been built

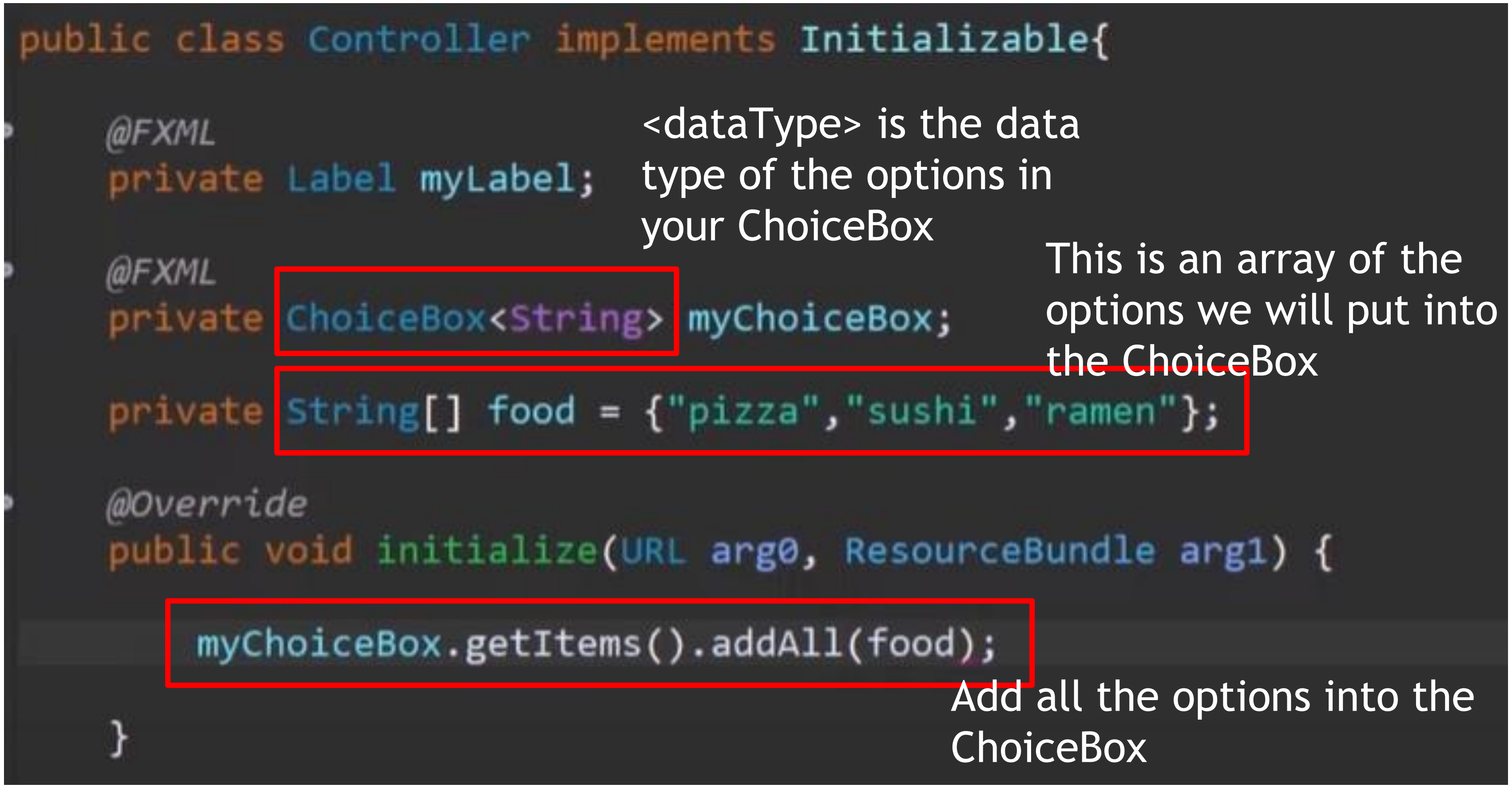


```
public class Controller implements Initializable{

    @FXML
    private Label myLabel;

    @FXML
    private ChoiceBox<String> myChoiceBox;

    private String[] food = {"pizza","sushi","ramen"};

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {

        myChoiceBox.getItems().addAll(food);

    }
```

let the Controller class implement Initializable

You should implement the initialize() method

Initialize our ChoiceBox



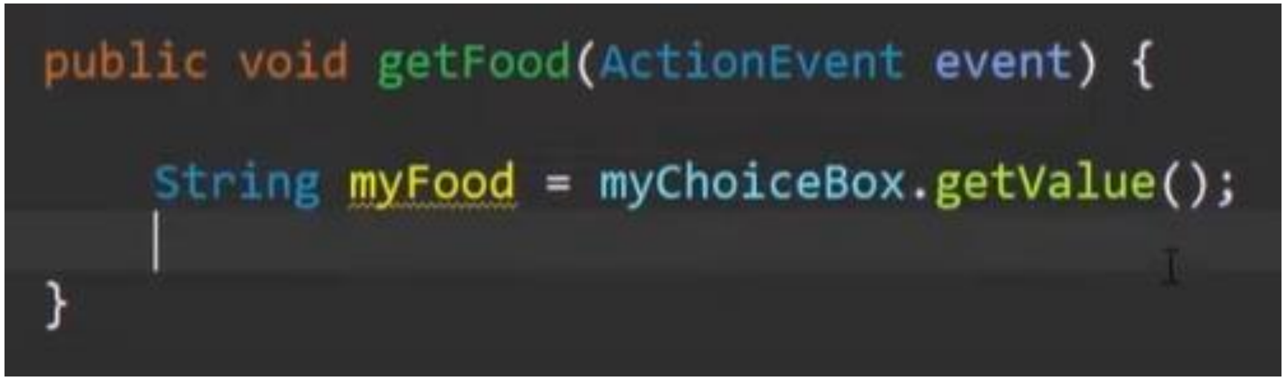
```
public class Controller implements Initializable{

    @FXML
    private Label myLabel;

    @FXML
    private ChoiceBox<String> myChoiceBox;

    private String[] food = {"pizza","sushi","ramen"};

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {

        myChoiceBox.getItems().addAll(food);

    }
```

<dataType> is the data type of the options in your ChoiceBox

This is an array of the options we will put into the ChoiceBox

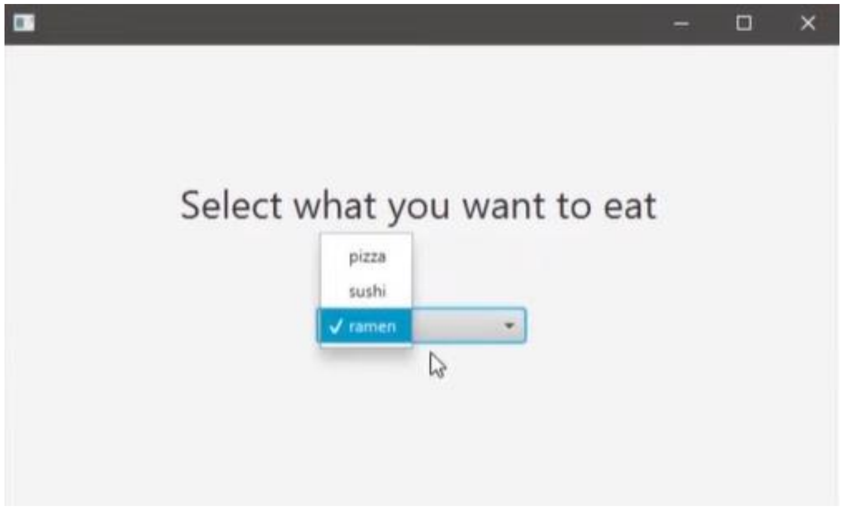
Add all the options into the ChoiceBox

To get the value of the selected options, use .getValue()



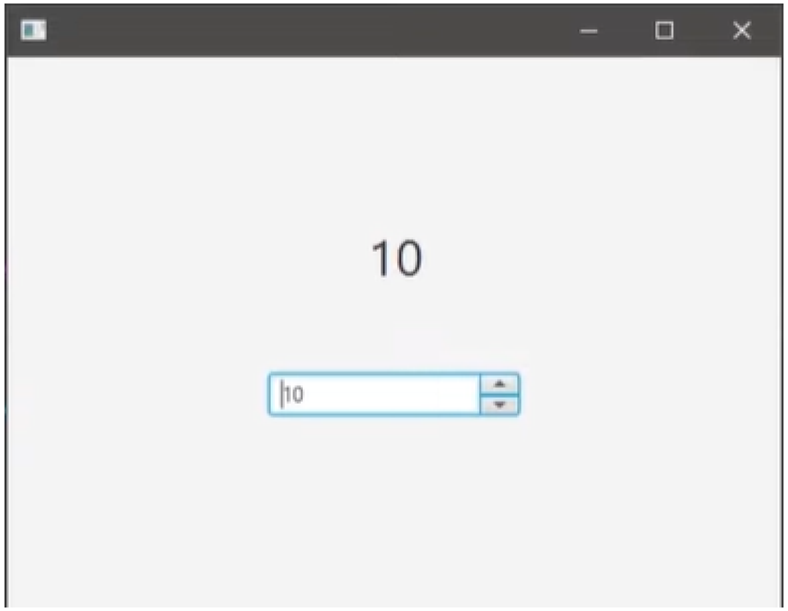
```
public void getFood(ActionEvent event) {

    String myFood = myChoiceBox.getValue();

}
```

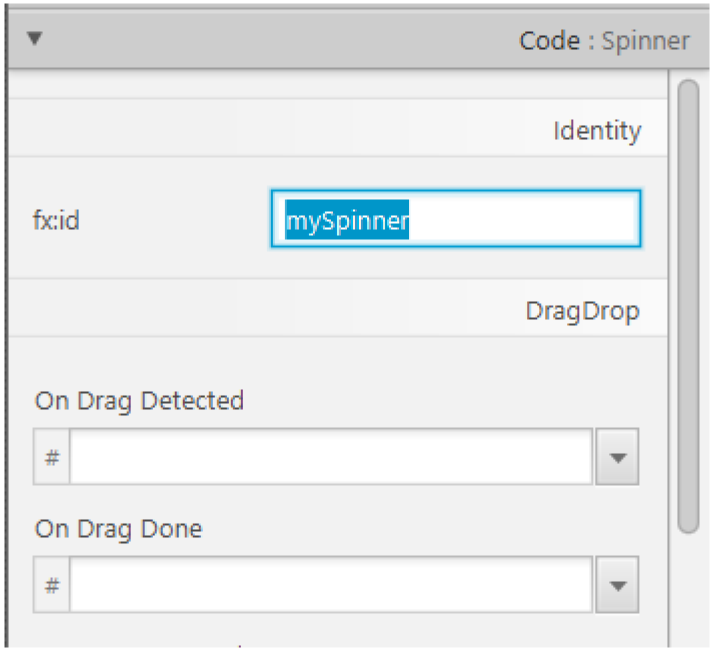
Try it!


Select what you want to eat

7.5 Spinners

Spinners don’t have onAction() either




Code : Spinner

Identity

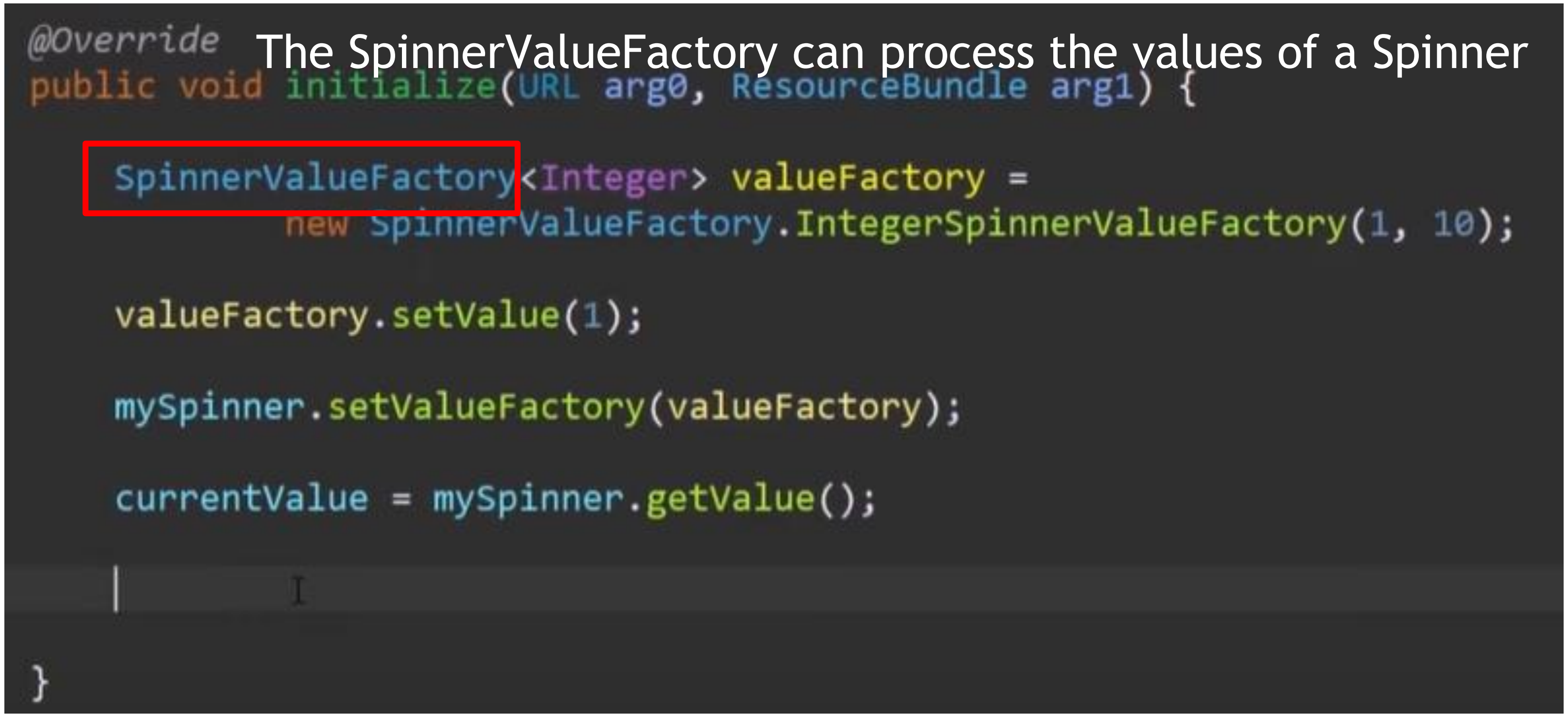
fx:id: mySpinner

DragDrop

On Drag Detected: #

On Drag Done: #

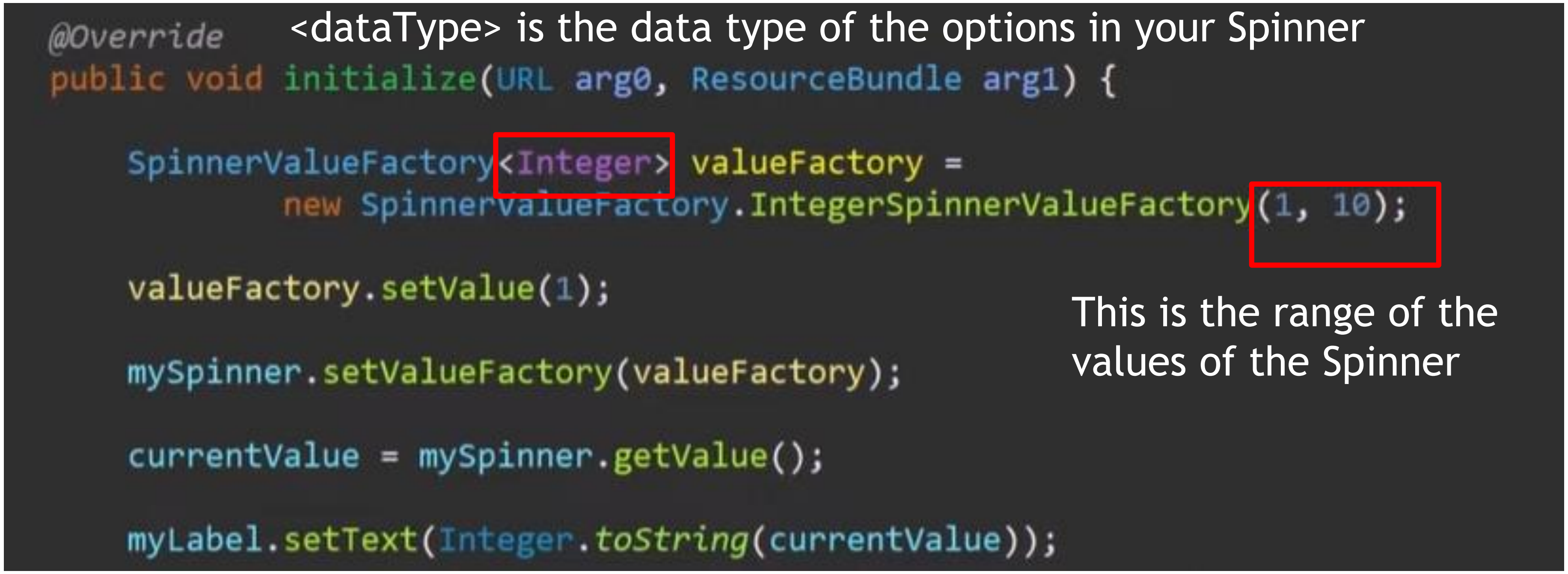
Initialize our Spinner



The SpinnerValueFactory can process the values of a Spinner

```
@Override
public void initialize(URL arg0, ResourceBundle arg1) {

    SpinnerValueFactory<Integer> valueFactory =
            new SpinnerValueFactory.IntegerSpinnerValueFactory(1, 10);

    valueFactory.setValue(1);

    mySpinner.setValueFactory(valueFactory);

    currentValue = mySpinner.getValue();

}
```

Initialize our Spinner

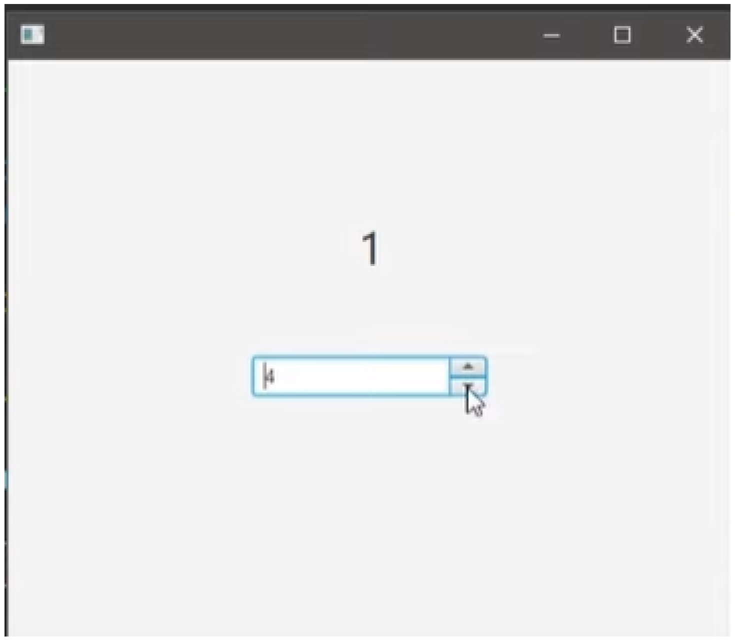


<dataType> is the data type of the options in your Spinner

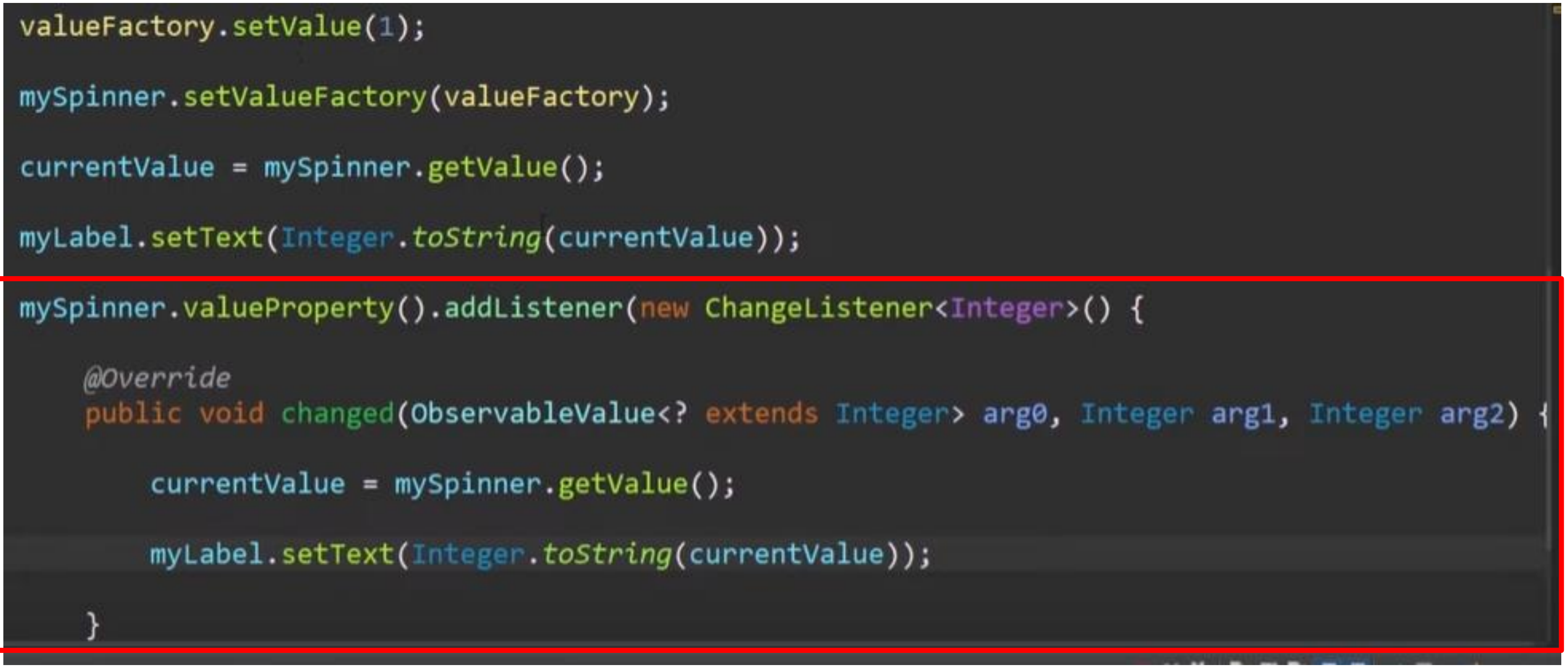
```
@Override
public void initialize(URL arg0, ResourceBundle arg1) {

    SpinnerValueFactory<Integer> valueFactory =
            new SpinnerValueFactory.IntegerSpinnerValueFactory(1, 10);

    valueFactory.setValue(1);

    mySpinner.setValueFactory(valueFactory);

    currentValue = mySpinner.getValue();

    myLabel.setText(Integer.toString(currentValue));
```

This is the range of the values of the Spinner

The Label doesn’t change with the Spinner’s value

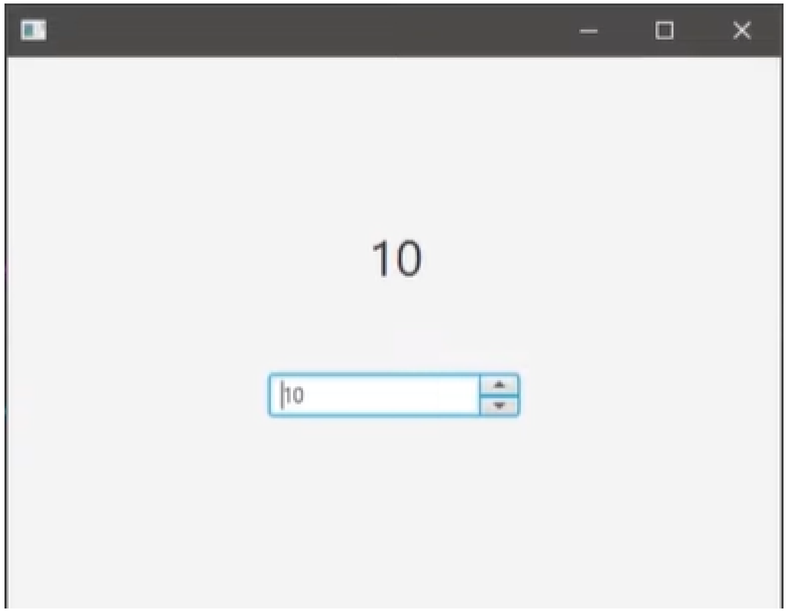


Add a listener to listen for the Spinner’s



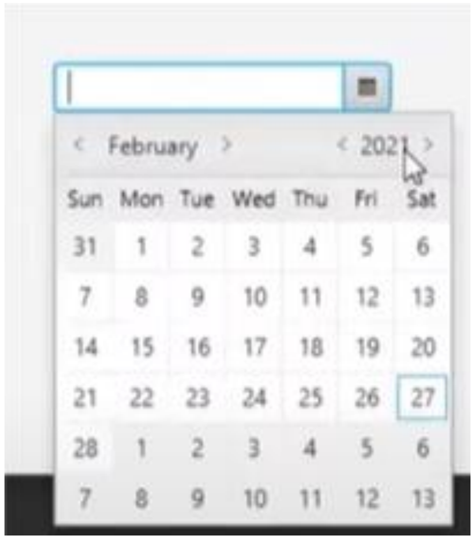
```
valueFactory.setValue(1);

mySpinner.setValueFactory(valueFactory);

currentValue = mySpinner.getValue();

myLabel.setText(Integer.toString(currentValue));

mySpinner.valueProperty().addListener(new ChangeListener<Integer>() {

    @Override
    public void changed(ObservableValue<? extends Integer> arg0, Integer arg1, Integer arg2) {

        currentValue = mySpinner.getValue();

        myLabel.setText(Integer.toString(currentValue));

    }
```

Try It

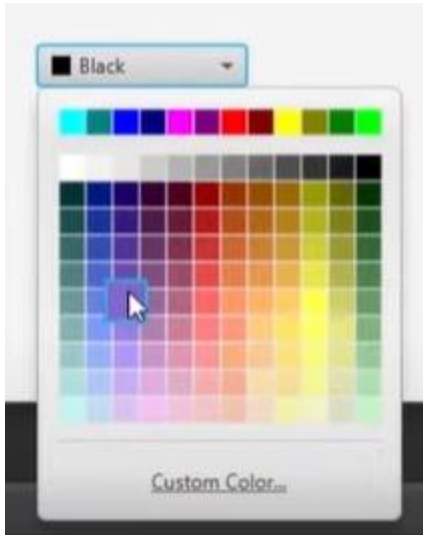


Other Components

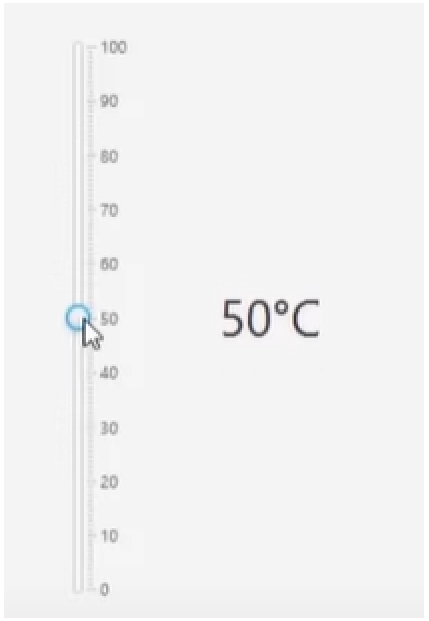
DatePick



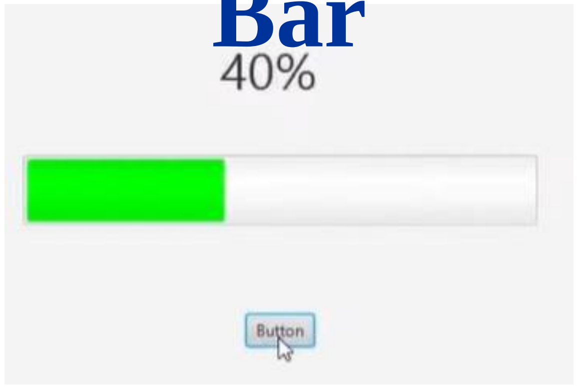
ColorPicker



Slider

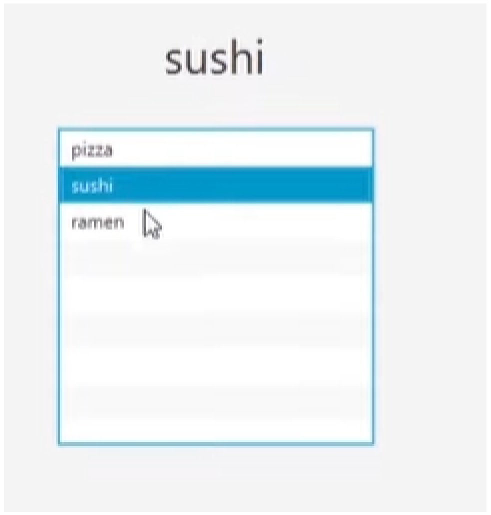


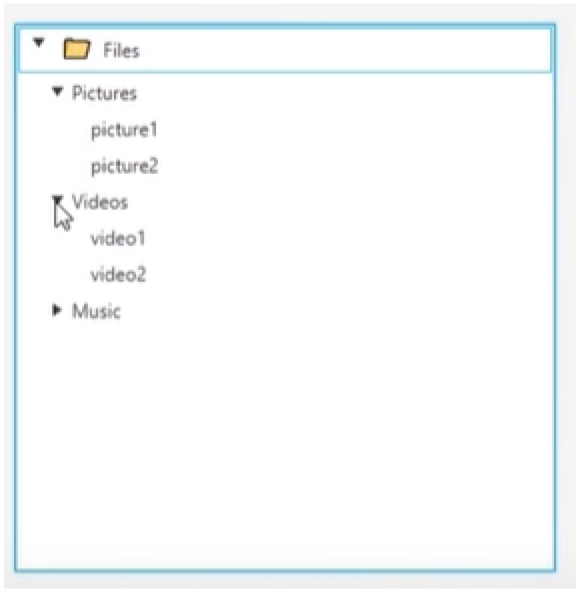
Progress Bar



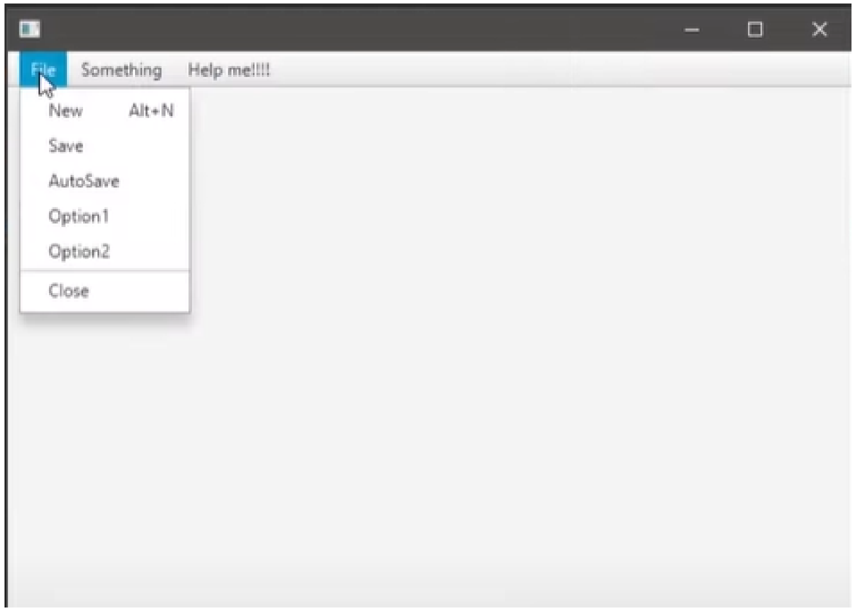
ListView

TreeView



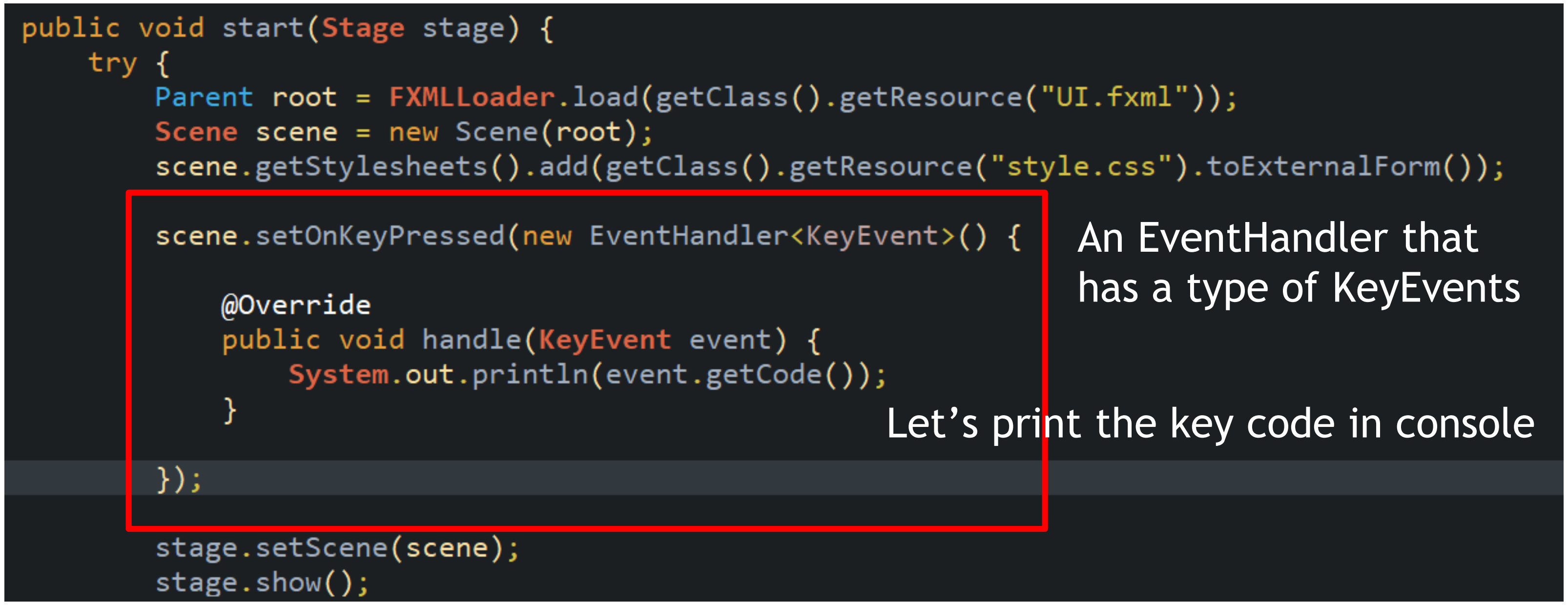


MenuBar


File
Something
Help me!!!!
New Alt+N
Save
AutoSave
Option1
Option2
Close

8.
Keyboard Events

Add event handler to the Scene

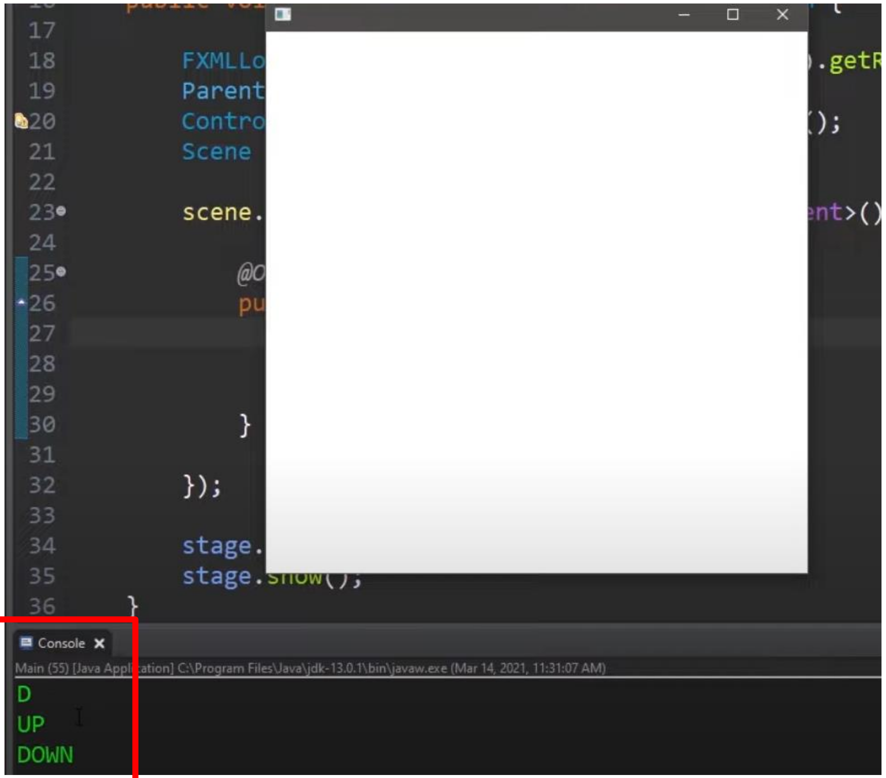


```java
public void start(Stage stage) {
    try {
        Parent root = FXMLLoader.load(getClass().getResource("UI.fxml"));
        Scene scene = new Scene(root);
        scene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

        scene.setOnKeyPressed(new EventHandler<KeyEvent>() {

            @Override
            public void handle(KeyEvent event) {
                System.out.println(event.getCode());
            }

        });

        stage.setScene(scene);
        stage.show();
```

An EventHandler that has a type of KeyEvents

Let’s print the key code in console

Try it



```
        FXMLLo                                              ).getR
        Parent
        Contro                                              ();
        Scene

        scene.                                              ent>()

            @O
            pu



            }

        });

        stage.
        stage.show();
    }
```

Console

```
Main (55) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Mar 14, 2021, 11:31:07 AM)
D
UP
DOWN
```