Sources:

MS Bing Chat Copilot

A book with a picture of a bus

AI-generated content may be incorrect.

Chapter 12-12\_4\_Displaying\_Images

Chapter 12-12\_6\_Button\_Controls\_and\_Events

A book with a picture of a van

AI-generated content may be incorrect.

Chapter 14-14\_7\_Handling\_Mouse\_Events

**+Contributors:**

Trieu Huu Phuong

**I.Simple buttons containing one image, icon, or symbol**

In order to create buttons including images or icons inside it, first make sure that you’ve already studied the resource materials above, then follow the steps below:

1)Using **ImageView container** to create an image, add image into it, adjust the size, then attach **.setOnMouseClicked event method** to activate any action whenever an user clicks on this **ImageView**

ImageView imageView = new ImageView(new Image("path/to/image.png"));

imageView.setFitWidth(100);

imageView.setFitHeight(100);

// Add click functionality

imageView.setOnMouseClicked(event -> {

System.out.println("Image clicked!");

// Add your desired action here

});

A screenshot of a computer

AI-generated content may be incorrect.

ImageView imageView = new ImageView(new Image("path/to/image.png"));

imageView.setFitWidth(100);

imageView.setFitHeight(100);

Button button = new Button();

button.setGraphic(imageView);

// Add action functionality

button.setOnAction(event -> {

System.out.println("Button with image clicked!");

// Add your desired action here

});

A screenshot of a computer

AI-generated content may be incorrect.

3)Final method is to add HTML symbols into JavaFX through Unicode string

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A black text on a white background

AI-generated content may be incorrect.

You can look up the Unicode equivalent for HTML symbols here or through other resources.

<https://unicode.org/charts/>

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

It’s not required that you only have to use the Miscellaneous Symbols section on this page from w3schools.com

<https://www.w3schools.com/charsets/ref_utf_symbols.asp>

The important things to look at is the Hex codes of each symbol

Ex:

A screenshot of a computer

AI-generated content may be incorrect.

You may only be able to use HTML Charsets and UTF-8 symbols, to use Emoji, follow these steps below:

A white background with black text

AI-generated content may be incorrect.

label.setFont(Font.font("Segoe UI Emoji", 16));

label.setText("\uD83D\uDE03"); // Unicode for 😃

Ex:

**package** application;

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Button;

**import** javafx.scene.text.Font;

**import** javafx.scene.layout.Pane;

**import** javafx.stage.Stage;

**public** **class** **Main** **extends** Application {

**@Override**

**public** **void** **start**(**Stage** primaryStage) {

**Pane** **pane** = **new** **Pane**(); // Simple Pane for absolute positioning

// Create a Button and set its position

**Button** **button** = **new** **Button**(**"\u20BF Click Me"**);

button.**setLayoutX**(**100**); // X-coordinate

button.**setLayoutY**(**50**); // Y-coordinate

// Create a Button and set its position

**Button** **button2** = **new** **Button**(**"\u20BF Click Me"**);

button2.**setFont**(**Font**.***font***(**"Segoe UI Emoji"**, **16**));

button2.**setText**(**"\uD83D\uDE03"**); // Unicode for 😃

pane.**getChildren**().**add**(button);

pane.**getChildren**().**add**(button2);

**Scene** **scene** = **new** **Scene**(pane, **400**, **300**);

primaryStage.**setTitle**(**"Absolute Position Example"**);

primaryStage.**setScene**(scene);

primaryStage.**show**();

}

**public** **static** **void** **main**(**String**[] args) {

***launch***(args);

}

}

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**II. Advanced Buttons that combine both images, symbols, or icons and text**

**+Example case:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**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.scene.text.Text;**

**import javafx.stage.Stage;**

**public class MenuButtonExample extends Application {**

**@Override**

**public void start(Stage stage) {**

**// Create the symbol for Menu Bar**

**ImageView menuIcon = new ImageView(new Image("path/to/menu\_icon.png")); // Replace with actual image path**

**menuIcon.setFitHeight(16); // Adjust size**

**menuIcon.setFitWidth(16);**

**// Create the text for "Menu"**

**Text menuText = new Text("Menu");**

**// Create the arrow-down symbol**

**Text arrowSymbol = new Text("\u25BC"); // Unicode for ▼ (arrow down)**

**// Add all elements to an HBox**

**HBox buttonContent = new HBox(5, menuIcon, menuText, arrowSymbol); // Spacing of 5 between elements**

**// Create the button**

**Button menuButton = new Button();**

**menuButton.setGraphic(buttonContent);**

**// Add to a scene**

**Scene scene = new Scene(menuButton, 200, 100);**

**stage.setScene(scene);**

**stage.setTitle("Menu Button Example");**

**stage.show();**

**}**

**public static void main(String[] args) {**

**launch(args);**

**}**

**}**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A close up of a text

AI-generated content may be incorrect.**

**.button {**

**-fx-font-size: 14px;**

**-fx-background-color: lightgray;**

**}**

This approach ensures your button contains all the elements—symbol, text, and arrow—while maintaining a clean and professional look.