

# CSCI 2120:

## Software Design & Development II

*UNIT4: UI management*

*GUI framework*

JavaFX Core: **CSS Styling**

# Overview

1. Introduction
2. CSS Styling Methods
  - a. Default CSS Stylesheet
  - b. Scene Specific CSS Stylesheet
3. Parent Specific CSS Stylesheets
4. Component Style Property
5. JavaFX CSS Properties

# Introduction

JavaFX enables you to style JavaFX components using **CSS**, just like you can style HTML and SVG element in web pages with CSS. JavaFX uses the same CSS syntax as CSS for the web, but the CSS properties are specific to JavaFX and therefore have slightly different names than their web counterparts.

Styling your JavaFX applications using CSS helps you separate styling (looks) from the application code. This results in cleaner application code and makes it easier to change the styling of the application. You do not have to look inside the Java code to change the styling. You can also change the styling for many components at once, by using shared CSS stylesheets.

In this JavaFX CSS lecture we will take a deeper look at the many different ways you can apply CSS styles to your JavaFX applications, as well as look at what can and cannot be styled. The JavaFX CSS features are quite advanced, so there is a lot you can do with just CSS.

## Note:

---

I assume that you are already somewhat familiar with the core concepts of CSS, like CSS syntax, CSS rules, CSS properties etc. If not, it might be a good idea to read the basics about CSS

# CSS Styling Methods

There are several different methods to apply a CSS style to a JavaFX component. These methods are:

- JavaFX default CSS stylesheet
- Scene specific CSS stylesheet
- Parent specific CSS stylesheet
- Component **style** property

I will briefly explain how each of these CSS styling mechanisms work in the following sections.

# Default CSS Stylesheet

JavaFX applications have a default CSS stylesheet which is applied to all JavaFX components. If you provide no styling of the components, the default CSS stylesheet will style the JavaFX components so they look pretty.

The default stylesheet for JavaFX is called Modena, and is located in the JavaFX JAR file.

# Scene Specific CSS Stylesheet

You can set a CSS stylesheet for a JavaFX **Scene** object. This CSS stylesheet is then applied to all JavaFX components added to the scene graph for that **Scene** object. The scene specific CSS stylesheet will override the styles specified in the default stylesheet, in case both stylesheets sets the same CSS properties.

Here is an example of setting a CSS stylesheet on a **Scene** object:

```
scene.getStylesheets().add("style1/button-styles.css");
```

This example tells JavaFX to look for a stylesheet file called **button-styles.css** which is located in a directory called **style1**. JavaFX looks for this file on the classpath, so the directory **style1** should be located in a directory which is at root of one of the directories (or JAR files) which are included in the classpath for the application.

The string pointing to the CSS stylesheet file is interpreted as a URL. That means that you can also specify full paths to a file in the file system. However, it is good practice to think of CSS files as resources and bundle them with the code for your application. The users running the application will typically not be changing the styles, so there is no need to distribute the file outside the code (like you would with config file that users were intended to change).

# Parent Specific CSS Stylesheets

It is also possible to set a CSS stylesheet on all subclasses of the JavaFX **Parent** class. The **Parent** class is a base class for all components that can have children, meaning they can contain other components inside them. CSS properties specified in a stylesheet set on a **Parent** subclass will normally take precedence over CSS rules specified in a scene stylesheet, and the default stylesheet.

The JavaFX layout components are typical examples of **Parent** subclasses. If you set a CSS stylesheet on a layout component, the stylesheet will be applied to all components inside that layout component.

Setting a CSS stylesheet on a **Parent** subclass looks similar to setting it on a **Scene** object. Here is an example of setting a CSS stylesheet on a **VBox** which is a **Parent** subclass:

```
Button button1 = new Button("Button 1");
Button button2 = new Button("Button 2");

VBox vbox = new VBox(button1, button2);
vbox.getStylesheets().add("style1/button-styles.css");
```

## Code Explanation:

---

This code will set the **style1/button-styles.css** stylesheet on the **VBox**. The stylesheet will thus be applied to the two buttons inside the **VBox**.

# Component Style Property

You can set CSS styles for a specific component by setting the CSS properties directly on the component. This is done by setting a String containing the CSS properties in the component's **style** property.

CSS properties set via the **style** property take precedence over CSS properties specified in any **Parent** subclasses the component is nested inside, the scene stylesheet and the default stylesheet.

Here is an example that sets the **style** property for a JavaFX **Button** :

```
Button button = new Button("Button 2");  
button.setStyle("-fx-background-color: #0000ff");
```

This example sets the background color CSS property in the **style** property to a blue color.

You can set multiple CSS properties inside the same style string. Here is an example of how that looks:

```
String styles =  
    "-fx-background-color: #0000ff;" +  
    "-fx-border-color: #ff0000;" ;  
  
Button button = new Button("Button 2");  
button.setStyle(styles);
```



# JavaFX CSS Properties

As mentioned earlier JavaFX contains its own set of CSS properties. The JavaFX CSS properties are named somewhat differently from the CSS properties used with HTML. However, the JavaFX team have kept the names of the JavaFX CSS properties very close to the CSS properties used in HTML. If you are familiar with CSS for the web, you will often be able to guess what the corresponding JavaFX CSS property is called.

Here is a list of the most commonly used JavaFX CSS properties. Not all CSS properties can be applied to all JavaFX components, but many CSS properties can be applied to several JavaFX components.

JavaFX CSS Property	Description
-fx-background-color	Sets the background color of a Java

JAVAFX CSS PROPERTIES & SELECTORS LIST:

- <https://wheelercode.wordpress.com/javafx-css-properties-selectors-list/>
- <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/doc-files/cssref.html>

END