

CS8662 Mobile Application Development Lab

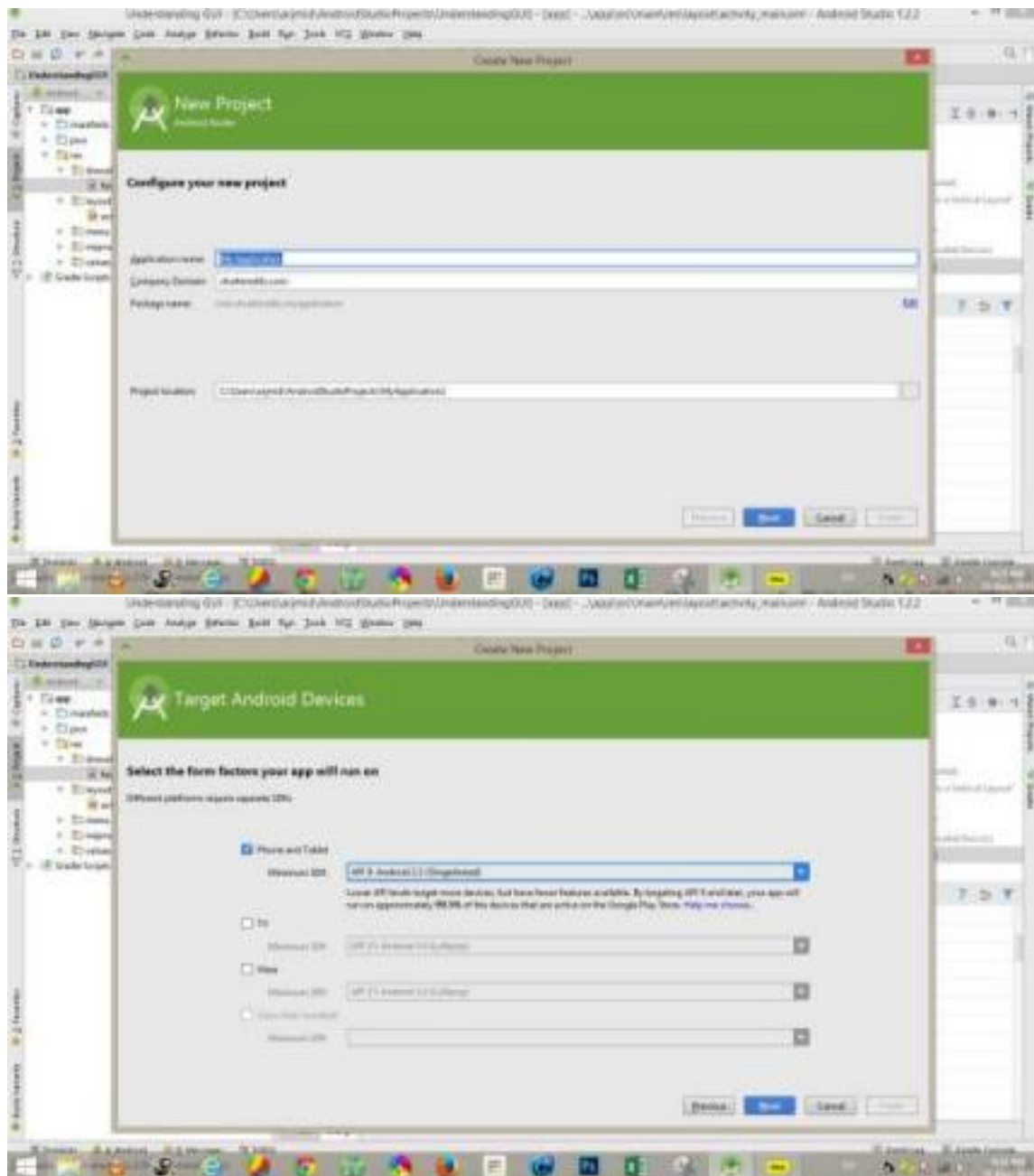
Develop an application that uses GUI components, Font and Colours

Understanding GUI

In this chapter we'll learn about different GUI in Android Studio and develop an app that use GUI, Font and Colors.

Part 1:

Start a project “Understanding GUI”



*You can learn more about the Android Studio Interface from
<http://developer.android.com/tools/studio/index.html>.*

Part 2:

UI design for android application is done using a layout xml file.

Android studio offers a layout editor which allows you to drag and drop UI elements into the interface.

You can learn more about layout editor from

<http://developer.android.com/sdk/installing/studio-layout.html>

Part 3:

Learn more on Material Design at

<https://www.google.com/design/spec/material-design/introduction.html#introduction-goals>

UI Elements:

Layouts:

- Linear Layout
 - Horizontal
 - Vertical
- Relative Layout
- Table Layout
- Grid Layout (Requires API 14 or more)
- Frame Layout

Widgets:

- TextView
- Button
- RadioButton
- ToggleButton
- ImageView
- ImageButton
- CheckBox
- ProgressBar
- SeekBar
- RatingBar
- WebView
- Spinner

Text Fields

- EditText
 - Plain text, Name, Number, Email etc.

Containers

- ScrollView
 - Vertical
 - Horizontal
- ListView
- GridView
- SearchView
- TabHost

Common Attributes

- Layout Width
- Layout Height
- Gravity
 - centre
 - left
 - right etc.

- match_parent
- fill_parent
- wrap_content
- <<pixel value>> in dp

- style
 - uses drawable.
- background
 - color or drawable.
- id
- padding
- margin
- textColor
 - color value (Hex,Rgba etc.)
 - drawable
 - color resource
- textStyle
- typeface

Changing attributes (text color, background, typeface etc.) in java

```
TextView <<variable>> = (TextView) findViewById(R.id.<<id>>);  
<<variable>>.setTextColor(getResources().getColor(R.color.<<name>>));
```

The color resource is defined in a "color.xml" resource file inside "values" resource directory.

Eg:



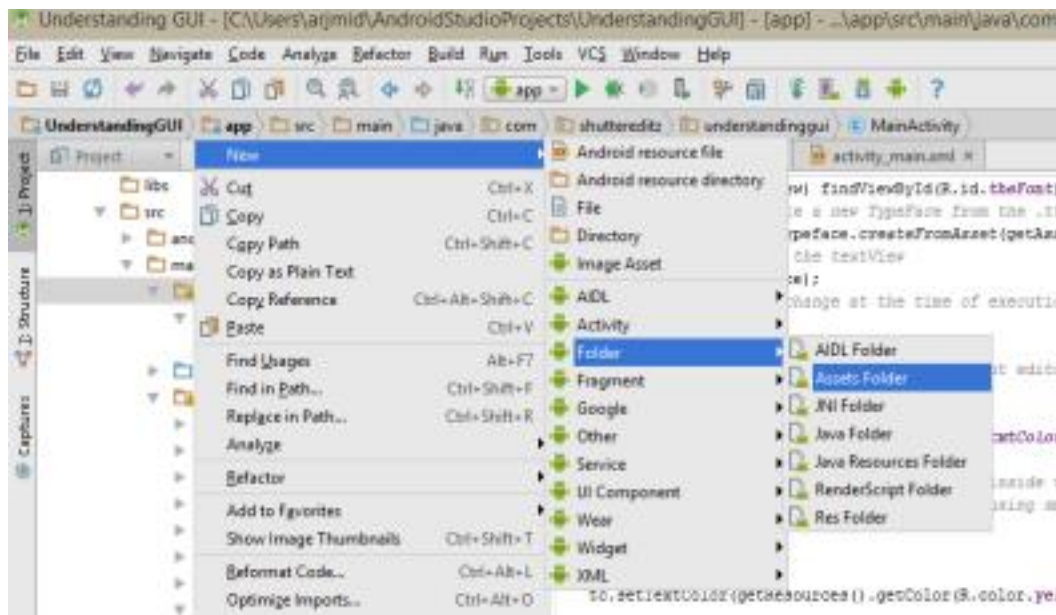
```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <color name="yellow">#FFFF00</color>  
</resources>
```

```
<<variable>>.setBackgroundColor(getResources().getColor(R.color.<<name>>));
```

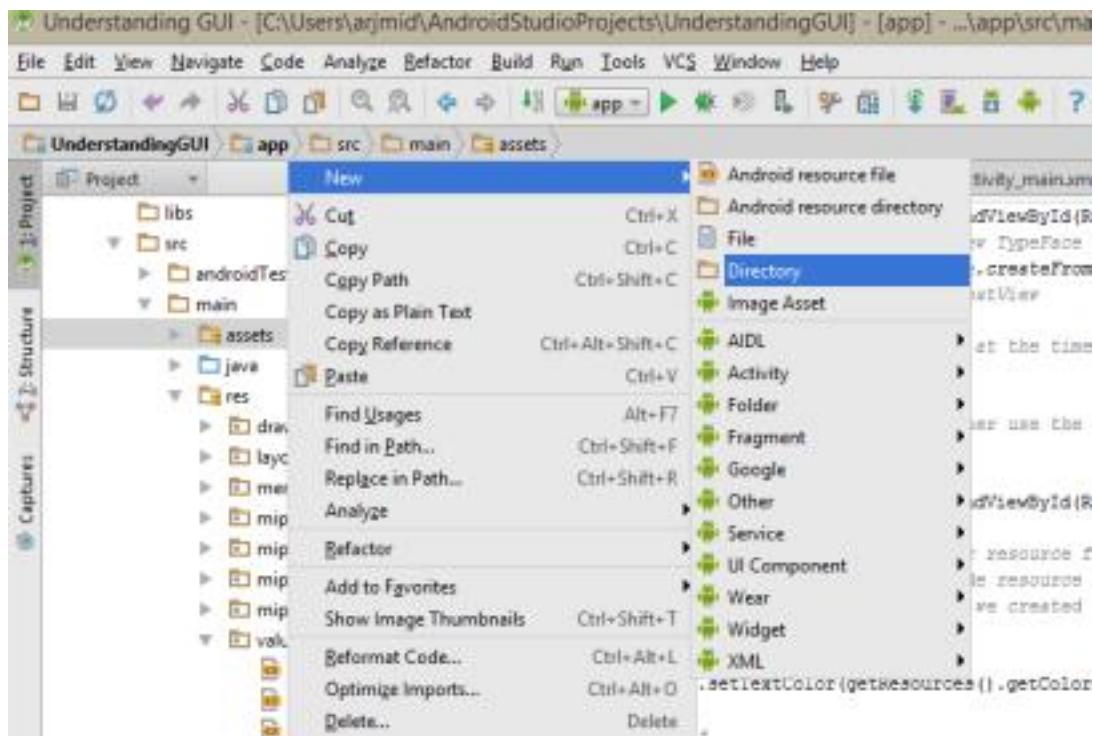
Changing Type Face

First we need to import a font-face to assets directory.

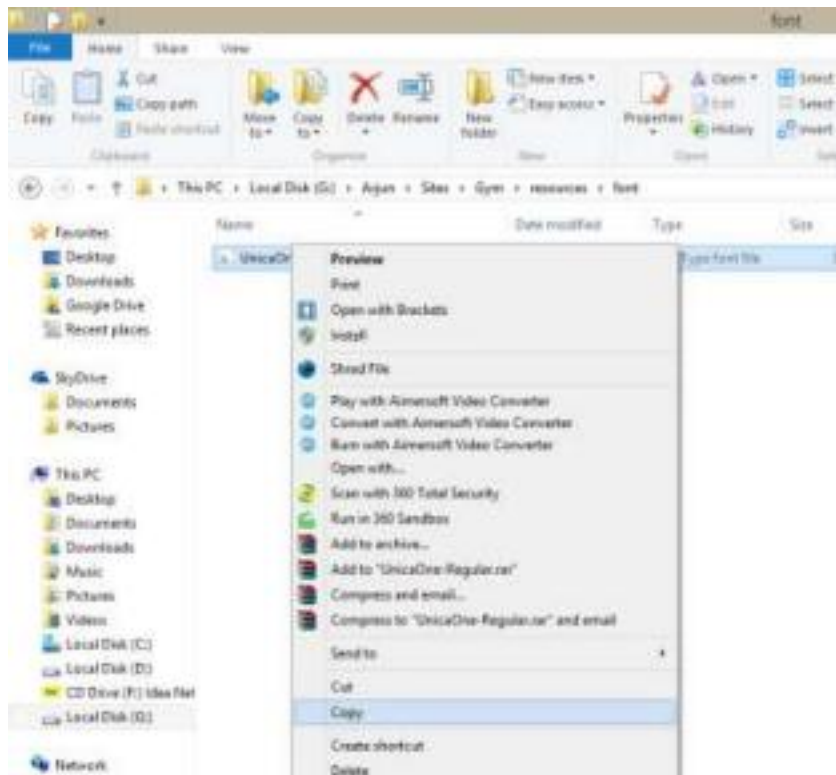
Create an assets directory.



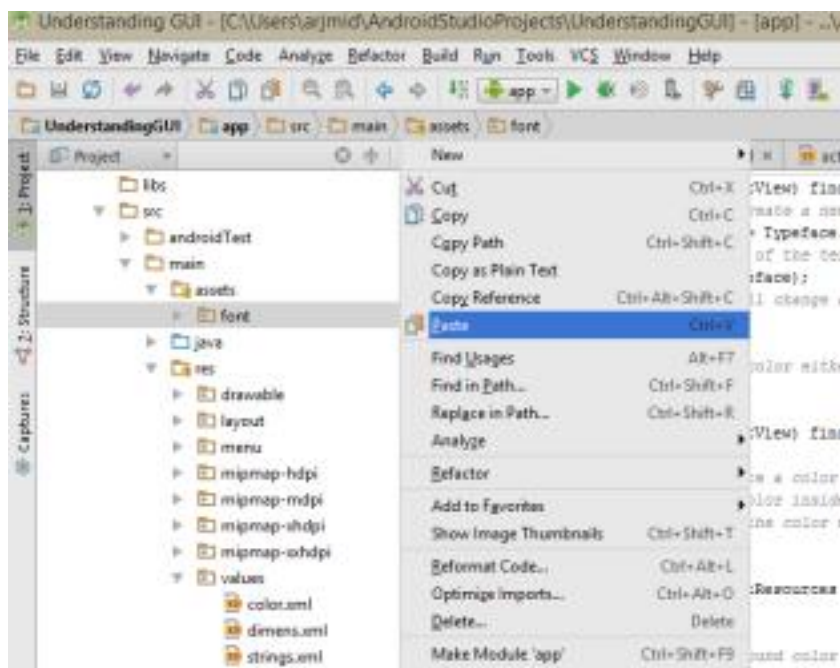
Create a new directory “font” – *not necessary* –



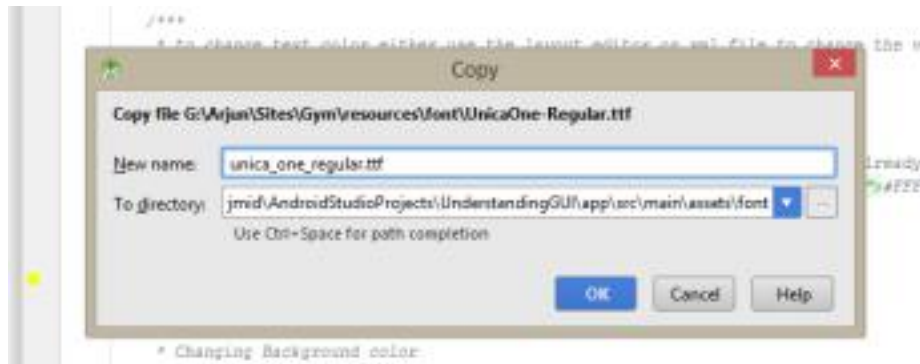
Copy the font



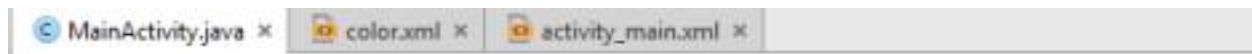
Paste it in the directory.



Change all caps to small and special characters (space, dash etc.) to underscore. Else it'll give an error on gradle build.



Now use the java script:



```
TextView tF = (TextView) findViewById(R.id.theFont); //find the
textView with id theFont
//Now we need to create a new Typeface from the .ttf file we imported
Typeface typeface = Typeface.createFromAsset(getAssets(),
"font/unica_one_regular.ttf");
//Set the typeface of the textView
tF.setTypeface(typeface);
//the type-face will change at the time of execution
```