|  |
| --- |
| Android Tutorial – Part 2 |

|  |
| --- |
| 6-2-2018 |



Table of Contents

[Introduction 2](#_Toc515745059)

[Android Studio’s UI Design View - Layout Editor 2](#_Toc515745060)

[The Design View of Layout Editor 3](#_Toc515745061)

[Toolbar items to Change the preview appearance 4](#_Toc515745062)

[Edit View attributes 5](#_Toc515745063)

[The Text View of Layout Editor 6](#_Toc515745064)

[Android Views and View Groups 6](#_Toc515745065)

[Different Types of layouts 6](#_Toc515745066)

[Design a User interface 6](#_Toc515745067)

[Attributes of UI components 6](#_Toc515745068)

[Usage of string.xml file 6](#_Toc515745069)

[Usage of colours.xml file 6](#_Toc515745070)

[Usage of styles.xml file 7](#_Toc515745071)

[Access and Modify UI from Activity class 7](#_Toc515745072)

[Toasts 7](#_Toc515745073)

[Intents 7](#_Toc515745074)

[References 7](#_Toc515745075)

# Introduction

This is the part two of the android tutorial series. It is a continuation from last week. In order to follow this successfully, it is required to have,

* A basic understanding given about android in last session.
* The environment set up.
* The project created during last, opened in Android Studio.
* AVD or an Actual device ready for app deployment.

To catch up, in the last session,

* An introduction to android was given
* Android studio was introduced
* It was shown how to create a new android project using android studio.
* The structure of an android project was explained.
* A brief introduction about each component of the project was given.
* Shown how to run an android app in an Android virtual device (AVD) and a Real device.

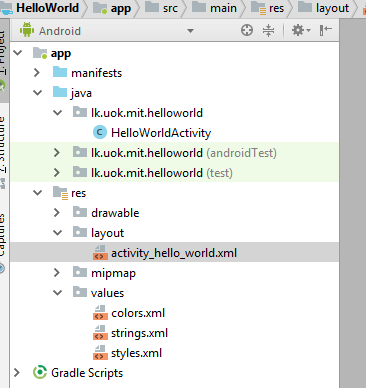
With that knowledge in hand, in this session below areas will be covered,

* Android Studio’s UI Design View (Layout Editor)
* Android Views and View Groups
* Different Types of layouts
* Design a User interface
  + Attributes of UI components
  + Usage of string.xml file
  + Usage of colours.xml file
  + Usage of styles.xml file
* Access and Modify UI from Activity class
* Toasts
* Intents

# Android Studio’s UI Design View - Layout Editor

Since the main focus of this tutorial is to show how to properly design user interfaces of android applications, as the first step, it’s required to get familiar with the Layout Editor of Android Studio first.

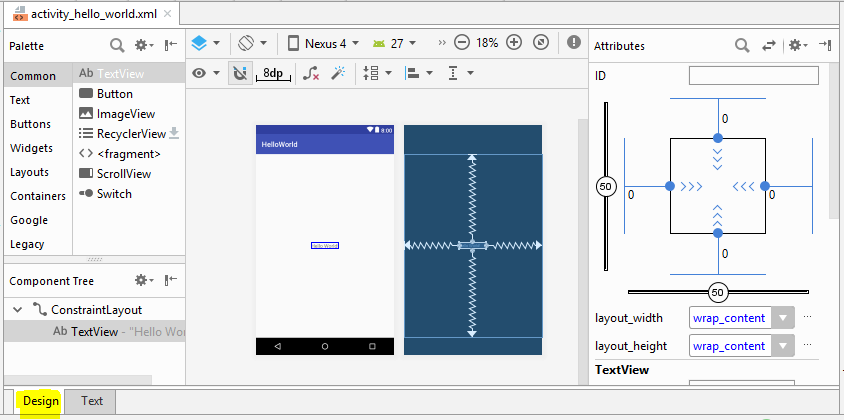
To open the layout editor, a layout xml file should be opened. If you could recall, we added an “Empty Activity” during creation of a new project, and named it as “HelloWorldActivity”. Along with that, an xml file was created as “**activity\_hello\_world.xml**”. This layout xml file could be seen at res->layout folder.



* Click on “activity\_hello\_world.xml” to open it in the Layout Editor of Android Studio.
* The layout editor view appears as below, there are two main views as;
  + **Design** – Facilitate UI design by providing Drag and Drop from palette
  + **Text** – Facilitate UI design by allowing to edit XML

## The Design View of Layout Editor

* The **Design View** of Layout Editor and its crucial items are marked and listed below,



* **Palette**: List of **views** and **view groups** that you can **drag** into your **layout**, in a categorized view
* **Component Tree**: View hierarchy for your layout.
  + Note there are only two components in the layout which was auto generated for our “HelloWorld” activity
    - ConstraintLayout – the parent layout or “View Group”
    - TextView – a text view with text “HelloWorld”, which is indicated as a child view under parent ConstraintLayout in Component Tree
* **Toolbar**: Buttons to configure your layout appearance in the editor and to change some layout attributes.
* **Design editor**: Layout in Design or Blueprint view, or both, allows to view either real world design of UI, or the blueprint, or both as shown above.
* **Attributes**: Controls for the selected view's attributes, this lets edit attributes of the selected view of the layout file

### Toolbar items to Change the preview appearance

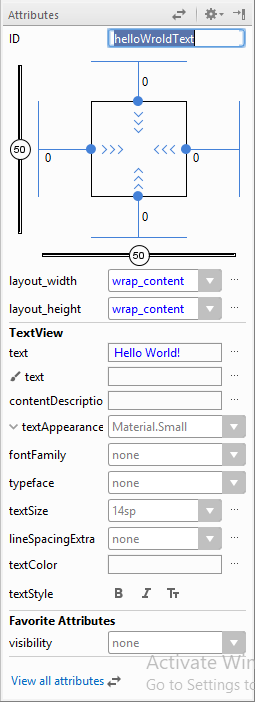
The buttons in the top row of the design editor allow you to configure the appearance of your layout in the editor. This toolbar is also available in the text editor's Preview window.



Corresponding to the numbers in above figure, the buttons available are as follows:

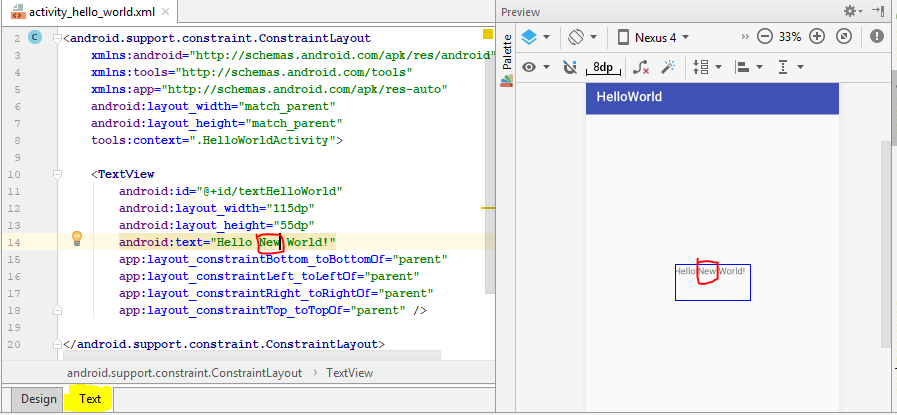
* **Design and blueprint**: Select how you'd like to view your layout in the editor; select either the
  + **Design view** (a real-world preview of the layout),
  + **The Blueprint view** (only outlines for each view),
  + OR Design + Blueprint for both side by side.
* **Screen orientation and layout variants**: Select between
  + - Landscape orientation
    - portrait screen orientation,
  + Also allows to choose other screen modes for which your app provides alternative layouts, such as night mode.
  + This menu also contains commands for creating a new layout variant.
* **Device type and size**: Select the
  + - device type (phone/tablet, Android TV, or Wear OS) and
    - Screen configuration (size and density).
  + Can select from several pre-configured device types and own AVD definitions
  + Also allows to start a new AVD by selecting Add Device Definition from the list.
* **API version**: Select the version of Android on which to preview your layout.
* **App theme**: Select which UI theme to apply to the preview. (This works only for supported layout styles; thus many themes in this list result in an error.)
* **Language**: Select the language to show for your UI strings.
  + This list displays only the languages available in your string resources.
  + If you'd like to edit your translations, click Edit Translations from the drop-down menu.
* **Zoom Out** – allows to zoom out the design view
* **Zoom level** – displays the current zoom level
* **Zoom In** – allows to zoom in the design view
* **Zoom to Fit** – Automatically select the appropriate zoom level based on window size to fit the screen
* **Issues Indication** – notifies (turns red) if there are any issues in the current UI design.

### Edit View attributes

* Each UI view of a layout has a set of attributes.
  + E.g.:- id, with, height, color, text, etc.
* These can be edited either in “Attributes” window or in “Text” view by editing xml
* The “Attribute” Window is available only when the design editor is open, so be sure you've selected the Design tab at the bottom of the window.
* When a view is selected, either by,
  + Clicking it in the Component Tree or
  + Clicking it in the design editor
* The Attributes window will look like below when the TextView “Hello Wrold” is seleted,  
    
  
* **ID** – the id attribute is used to **uniquely** **identify** a **view**.
  + ID attribute is optional.
  + An **android:id** attribute is required to be added to any View component in the layout file only if you want a **reference** to it
    - Access the view in Java code.
    - Access the view when defining layout in XML.
* **View inspector** - with controls for width/height style, margins, and bias (**available only for views in a ConstraintLayout**) in a Responsive UI.
* **Common attributes** - A list of common attributes for the selected view.
  + To see all available attributes, click View all attributes at the top of the window.
* **Favorite attributes** - Favorite attributes you've selected, most commonly used attributes
  + To add attributes to this, click View all attribute Favorite attributes and then click the star that appears when you hover your mouse over the left side of an attribute name.
* **View all attributes** - To search for a specific view attribute, click View all attributes and then click Search at the top of the window.

## The Text View of Layout Editor

* The **Text View** of Layout Editor and its crucial items are marked and listed below,



* This view mainly consists of two parts as shown above,
  + XML editor – allows to edit the xml code of layout
  + Preview – same as in the design view, show the real design of UI
* Note the change in text of “Hello World” text box, the **android:text** attribute has been **modified**  using text view
* Some may find its comfortable to design UI by editing the xml file in text view and some may find its comfortable to use the design view, choose according to your preference and always can be used interchangeably

# Android Views and View Groups

Activities make up the bulk of the graphical user interface associated with Android applications. In Android, there are two ways to construct Activities.

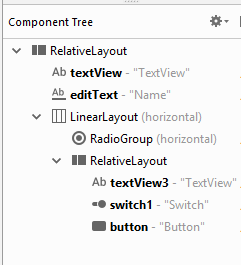
1. Programmatically add View and ViewGroup components (i.e. “widgets”) to the Activity in Java program code.
   * The programmatic approach is typically reserved for more dynamic and/or complicated UI scenarios.
2. Use XML layout files.
   * The typical and most common way to construct an Activity.

Composition of an android UI is a combination of **View** s and **ViewGroup** s

* The user interface for an Android app is built using a hierarchy of
  + **Layouts** (**ViewGroup**) objects
    - invisible containers that control how its child views are positioned on the screen
  + **Widgets** (**View**) objects.
    - UI components such as buttons and text boxes

An Illustration of how ViewGroup objects form branches in the layout and contain View objects is given below,

In the “Component Tree” view of the Android Studio, above arrangement will look like below,



http://www.vogella.com/tutorials/Android/article.html#using-views-and-view-groups-to-design-the-user-interface

https://developer.android.com/training/basics/firstapp/building-ui

http://abhiandroid.com/ui/xml

http://abhiandroid.com/ui/

# Design a User interface

## Attributes of UI components

## Usage of string.xml file

## Usage of colours.xml file

## Usage of styles.xml file

# Access and Modify UI from Activity class

# Toasts

# Intents

# References

*Application Fundamentals*. (2018, May 2018). Retrieved from Android Developers: https://developer.android.com

MULLIS, A. (2017, November 11). *Android Studio tutorial for beginners*. Retrieved from Android Authority: https://www.androidauthority.com/android-studio-tutorial-beginners-637572/

SINICKI, A. (2017, December 16). *Android SDK tutorial for beginners*. Retrieved from Android Authority: https://www.androidauthority.com/android-sdk-tutorial-beginners-634376/

Vogel, L. (2016, June 20). *Getting started with Android development - Tutorial*. Retrieved from vogella: http://www.vogella.com/tutorials/Android/article.html#high-level-overview-of-android-development