ASSIGNMENT-6

1. DP in Android

A virtual pixel unit that you should use when defining UI layout, to express layout dimensions or position in a density-independent way. The density- independent pixel is equivalent to one physical pixel on a 160 dpi screen, which is the baseline density assumed by the system for a “medium” density screen.

A dimension value defined in XML. A dimension is specified with a number followed by a unit of measure. For example: 10px, 2in, 5sp. The following units of measure are supported by Android:

dp

Density-independent Pixels - An abstract unit that is based on the physical density of the screen. These units are relative to a 160 dpi (dots per inch) screen, on which 1dp is roughly equal to 1px. When running on a higher density screen, the number of pixels used to draw 1dp is scaled up by a factor appropriate for the screen's dpi. Likewise, when on a lower density screen, the number of pixels used for 1dp is scaled down. The ratio of dp-to-pixel will change with the screen density, but not necessarily in direct proportion. Using dp units (instead of px units) is a simple solution to making the view dimensions in your layout resize properly for different screen densities. In other words, it provides consistency for the real-world sizes of your UI elements across different devices.

sp

Scale-independent Pixels - This is like the dp unit, but it is also scaled by the user's font size preference. It is recommend you use this unit when specifying font sizes, so they will be adjusted for both the screen density and the user's preference.

pt

Points - 1/72 of an inch based on the physical size of the screen, assuming a 72dpi density screen.

px

Pixels - Corresponds to actual pixels on the screen. This unit of measure is not recommended because the actual representation can vary across devices; each devices may have a different number of pixels per inch and may have more or fewer total pixels available on the screen.

mm

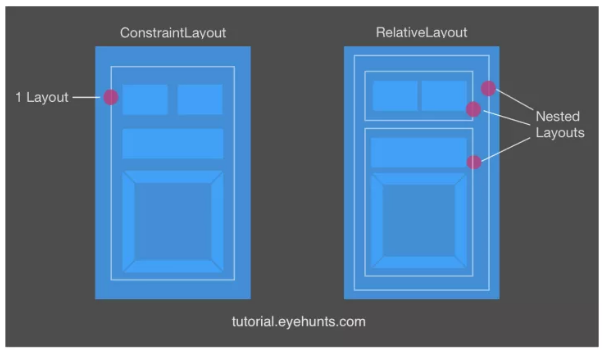
Millimeters - Based on the physical size of the screen.

in

Inches - Based on the physical size of the screen.

1. Differences between ConstraintLayout and RelativeLayout.

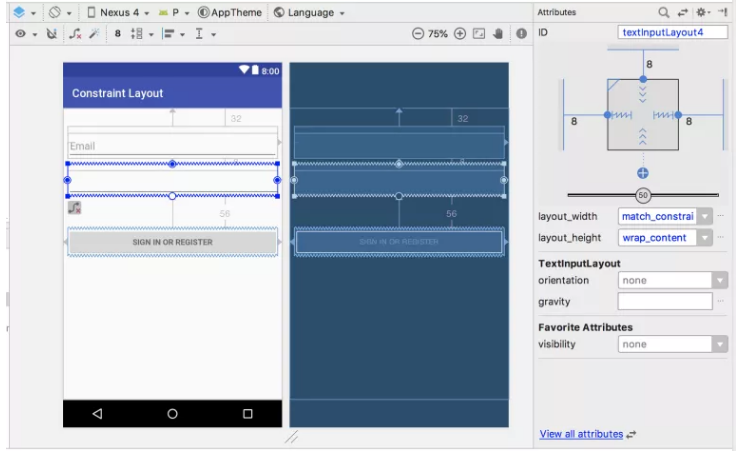
ConstraintLayout has **flat view hierarchy** unlike other layouts, so does **a better performance than relative layout**. Yes, this is the biggest advantage of Constraint Layout, the only single layout can handle your UI. Where in the Relative layout you needed multiple nested layouts (LinearLayout+ RelativeLayout).



In Android Studio you can drag and drop GUI component likeTextView , Button , TextInputLayout etc. So now its make life easier to developers and make they work faster UI development and more productive so they continue improving drag-drop GUI builder. However drag-and-drop gestures, the developer is only providing you with X/Y coordinates of a widget, based on where you release the mouse button and completes the drop.

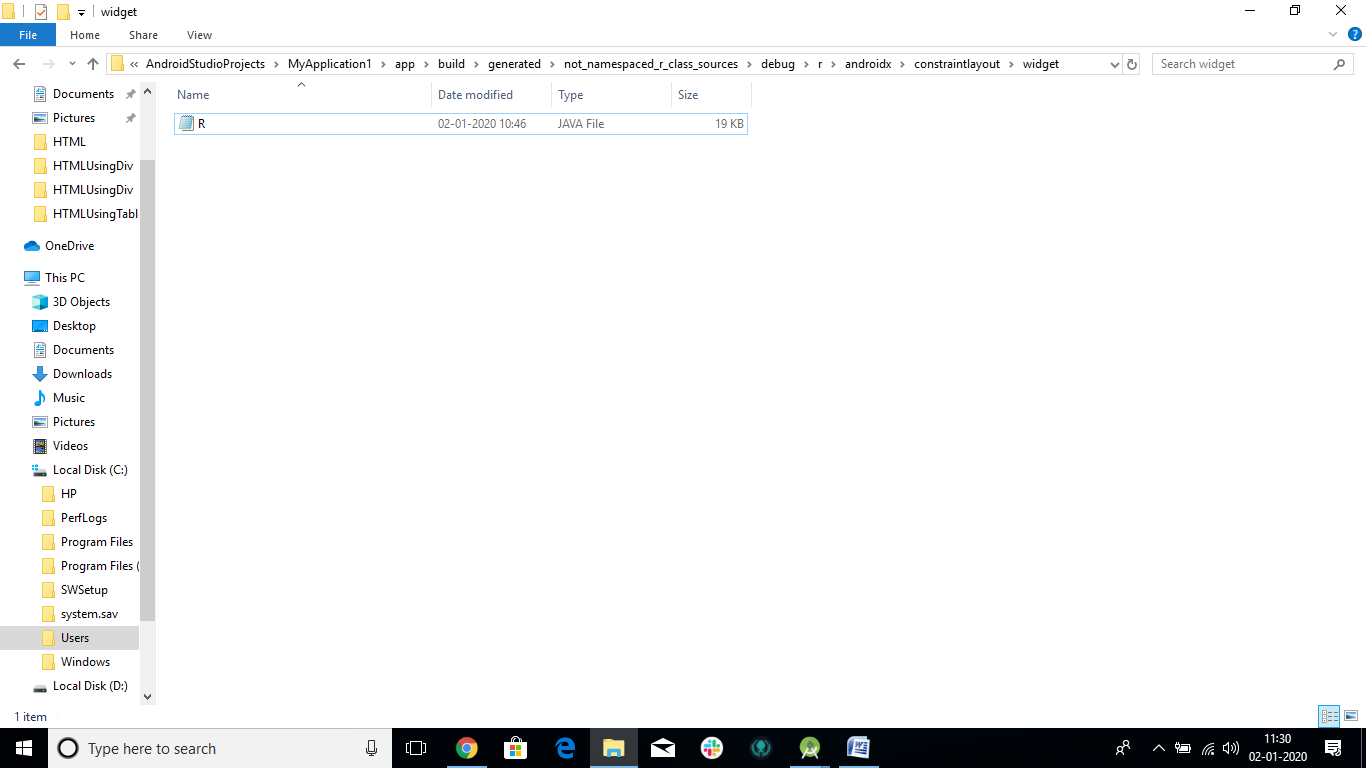
With Relative Layout  is difficult for GUI builder to handle drag-drop and probably you will have to dig inside the XML code to get things done.

But in ConstraintLayout have an option to constraint applying by the use of Blueprint and Visual Editor tool which makes it easy to design a page.



Each changes the detail of a widget often cause the sizes to have to be recomputed. Let’s take an example a change in EditText might starting point that a whole hierarchy to go through re-size and re-position work. If the application UI (user interface) has a container inside the container which is inside another container etc. Means that parents re-size or re-position their children and that can be very expensive (rework again on the user interface) for deep hierarchies.

1. Where is R.java file in Android?



C:\Users\HP\AndroidStudioProjects\MyApplication1\app\build\generated\not\_namespaced\_r\_class\_sources\debug\r\androidx\constraintlayout\widget

REFERENCES

1)<https://tutorial.eyehunts.com/android-interview-questions/differences-between-constraintlayout-relativelayout/>

2) <https://developer.android.com/reference/android/support/constraint/ConstraintLayout>

3) <https://developer.android.com/reference/android/support/constraint/ConstraintLayout>

4)<https://stackoverflow.com/questions/37321448/differences-between-constraintlayout-and-relativelayout>