

CUSTOM VIEW

What is a Custom View in Android?

First, the concept of Custom View in Android is nothing but having our own customized view model other than the prebuilt ones. And how do we achieve that? By simply bringing together different types of prebuilt views and then use the combined one as a single view.

Why do we even need them in our apps?

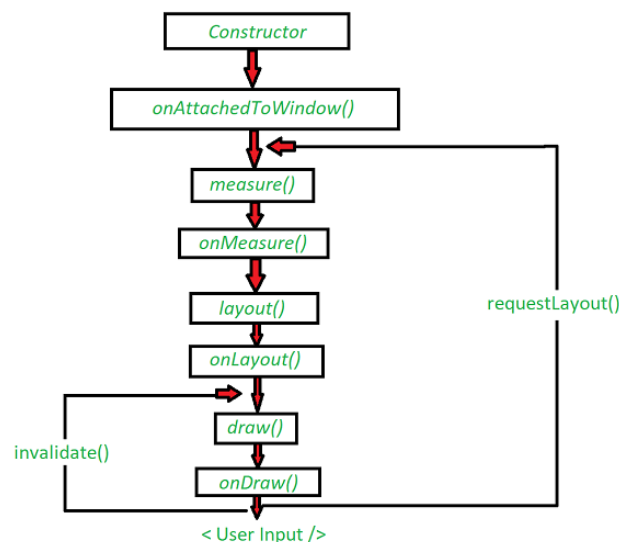
We all know that in the beginning android platform provides us some basic views for example

– [TextView](#), [ImageView](#), [EditText](#), [Button](#), [ImageButton](#), [RadioButton](#), etc. But, sometimes we need a more interactive and complex type of view where our prebuilt layouts and widgets fail to meet our needs. There the custom views come into the picture. Now the custom views in Android are fully customized by the developers in order to achieve the goal. Although custom views aren't needed everywhere. But, at a higher development level where complexity is more, we need them.

How are we going to create our own Custom View?

Whenever we're going to create a custom view for our apps, we need some items under consideration. Introduce yourself to the lifecycle of a view in Android.

GeeksforGeeks



Now, 3 basic methods are to be taken under importance.

1. **onMeasure()**
2. **onLayout()**
3. **onDraw()**

These 3 methods will be overridden in the corresponding Java/Kotlin class.

1. onMeasure() Method

As suggested by the method name it is used for measurement purposes. Basically, we can control the width and height of our custom view.

- **If not overridden:** The size of the view will be either 'match_parent' or 'wrap_content'.
- **If overridden:** On overriding this method we have more control of size over the custom view. Do not call the method '**super.onMeasure()**'. Instead, we'll be calling the method '**setMeasuredDimension(width, height)**'.

Overriding onMeasure():

When this method is called we

get '**widthMeasureSpec**' & '**heightMeasureSpec**' as parameters. A size mode is a constraint that is set by the parent for its child's views. Available 3 types of modes are listed below.

- **UNSPECIFIED:** No constraints are given, so it can be of whatever size it wants.
- **EXACTLY:** Exact size is determined for the child views.
- **AT_MOST:** Child can be as large as it wants up to the specified size.

2. onLayout() Method

This method is used by the parent view to notify our custom view about its position. One should use it to calculate their drawing width and height. Point to remember whatever happens in **onMeasure()** affects the positions, got from the parent. This is recommended one should always calculate the drawing size here before proceeding to draw the view.

3. onDraw() Method

All the drawing happens within this part. It's not hard at all as you get an instance of Canvas object and you are free to draw whatever you want.

Overriding onDraw() :

First for simplicity get the width and height of the canvas object with '**getWidth()**' & '**getHeight()**' methods.