Toolbar

- It is similar to an ActionBar(now called as App Bars).

- Toolbar is a Viewgroup that can be placed at anywhere in the Layout. We can easily replace an ActionBar with Toolbar.

- It provides more feature than ActionBar. A Toolbar may contain a combination of element from start to end.

Toolbar View on Activity

Toolbar

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

APP   PIXEL 3A API 29

tomToolBarExample > app > src > main > res > values > styles.xml >

activity_main.xml ×   MainActivity.java ×   styles.xml ×

```xml
 3      <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
 4          <!-- Customize your theme here. -->
 5          <item name="colorPrimary">@color/colorPrimary</item>
 6          <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
 7          <item name="colorAccent">@color/colorAccent</item>
 8      </style>
 9
10      <style name="MyThemeNoActionBar" parent="Theme.AppCompat.Light.NoActionBar">
11          <!-- Customize your theme here. -->
12          <item name="colorPrimary">@color/colorPrimary</item>
13          <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
14          <item name="colorAccent">@color/colorAccent</item>
15      </style>
16
```

resources style item

Terminal   Build   6: Logcat   TODO

```java
Toolbar toolbar;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    toolbar = findViewById(R.id.toolbaar);

    setSupportActionBar(toolbar);

    if(getSupportActionBar()!=null)
    {
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setTitle(" New ");
    }

    toolbar.setTitle("NEW");
}


@Override
public boolean onCreateOptionsMenu(Menu menu) {

    new MenuInflater( context: this).inflate(R.menu.customtool,menu);
    return super.onCreateOptionsMenu(menu);
}


@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {

    int itemId = item.getItemId();
```

--> ya par by default app name save hota hai toolbar and uski priority high hoti hai. Ise wja sai toolbar.setTitle() mth. work nahi karta hai.

--> iska iya hma getSupportActionbar().setTitle() ka use karna hota hai mostly.

activity_main.xml ✕   customtool.xml   MainActivity.java

```xml
1   <?xml version="1.0" encoding="utf-8"?>
2   <LinearLayout
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       xmlns:app="http://schemas.android.com/apk/res-auto"
5       xmlns:tools="http://schemas.android.com/tools"
6       android:layout_width="match_parent"
7       android:layout_height="match_parent">
8
9
10      <com.google.android.material.appbar.AppBarLayout
11          android:layout_width="match_parent"
12          android:layout_height="90dp">
13
14          <androidx.appcompat.widget.Toolbar
15              android:background="#E8DC74"
16              android:id="@+id/toolbaar"
17              android:layout_width="match_parent"
18              android:layout_height="100dp" />
19
20      </com.google.android.material.appbar.AppBarLayout>
```

activity_main.xml ∨

--> In general appbar layout and toolbar same hota hai.

--> Agar XML mai, ham sirf appcompact ka use karenga to jab ham scroll ka use karenga page par to content jab tool bar ka pass aayaga to wo aapas mai aisa lgaga jaisa meet ho raha hai.

--> Lekin, appbar layout ka use karna sai wo content tolbar kai nicha jata lgaga jo lookwise acha lgta hai.

--> Appbar layout ek container hai to usee mai ek child hona compulsory hai.