This tutorial guides you through creating a simple preference activity. And setting a preference listener

Remember the following shortcut keys to help speed things up

- ctrl-j live templates
- Alt-6 restore LogCat
- ctrl shift backspace- goto place last edited
- Alt+Enter quick fix/Extract string
- alt+Insert constructors, getter, setter
- ctrl-b - go to definition
- ctrl+Space code completion
- ctrl-o override members
- ctrl-H inheritance type heiarchy
- ctrl-alt-L Reformat code

1. Create an 'Empty Views Activity' project.
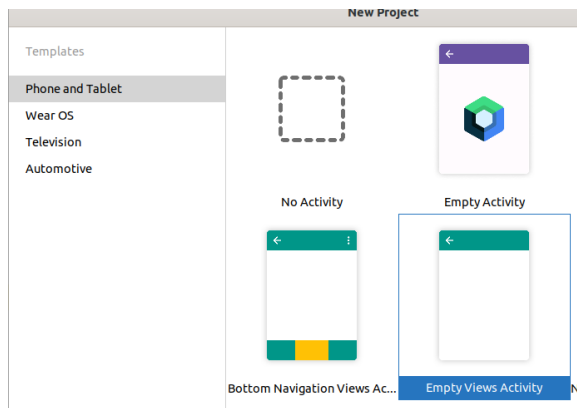
2. Add a 'Settings Views Activity' to this project.
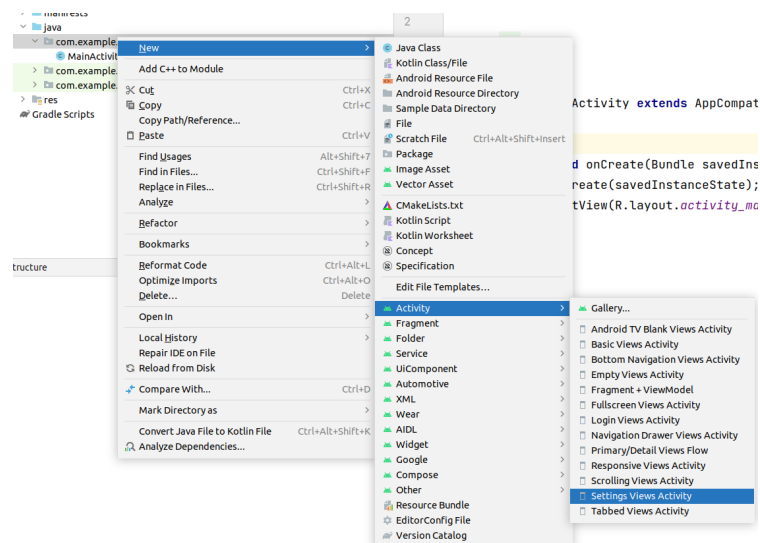
3. Modify activity_main.xml to have a switch and a FAB (floating action button)

By the end of this tutorial, when you set the switch, your app will start listening for changes in the settings activity and will react accordingly. The FAB will invoke a Settings Activity that you create.

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/cl"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@color/white"
    android:padding="10dp">

    <androidx.appcompat.widget.SwitchCompat
        android:id="@+id/switch1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="toggle_preference_change_listener"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintBottom_toBottomOf="parent" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/floatingActionButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:clickable="true"
        android:focusable="true"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:srcCompat="@android:drawable/ic_dialog_info"
        android:contentDescription="Show Settings Activity" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

4. Modify root_preferences.xml (in res/xml). Add the following to the bottom to include a seekbar (Notice that seekbar is not one of the Palette choices).

```xml
    <PreferenceCategory app:title="Seekbar demonstration">
      <SeekBarPreference xmlns:android="http://schemas.android.com/apk/res/android"
        android:max="50"
        app:defaultValue="50"
        app:enabled="true"
        app:key="seek_bar_key"
        app:min="0"
        app:showSeekBarValue="true"
        app:title="Main Activity background shade"
        app:summary="lower is lighter, higher is darker" />
    </PreferenceCategory>
```

```
//manage the FAB
FloatingActionButton fab = findViewById(R.id.floatingActionButton);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent myIntent = new Intent(MainActivity.this,SettingsActivity.class);
        startActivity(myIntent);
    }
});
```

```
//get a reference to the switch
SwitchCompat s=findViewById(R.id.switch1);

//you want to know if it's state has changed, it derives from CompoundButton
//use it's OnCheckedChangeListener to get notified when it changes and its state
s.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
        togglePreferenceChangeListener(b);
    }
});
```

```
//need these to track changes
private SharedPreferences myPreference;
```

```
// lets get a handle to default shared prefs
myPreference = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
```

```
private SharedPreferences.OnSharedPreferenceChangeListener listener = null;

private void togglePreferenceChangeListener(boolean enablePreferenceListener) {
        // this is the bit that listens for any preference changes to defaultsharedpreferences
    // (the prefs that the pref activity accesses)
    //you can also implements OnSharedPreferenceChangeListener for the mainactivity and then
    //register to have have the mainactivity listen for changes like this
    //myPreference.registerOnSharedPreferenceChangeListener(this);
    //and forgo whats below
    if (listener == null) {
       listener = new SharedPreferences.OnSharedPreferenceChangeListener() {
          @Override
          public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
             Toast.makeText(MainActivity.this, "Handle change of Key=" + key, Toast.LENGTH_SHORT).show();
          }
       };
    }

    if(enablePreferenceListener)
       // register the listener (turn it on)
       myPreference.registerOnSharedPreferenceChangeListener(listener);
    else
       //or unregister it (turn it off)
       myPreference.unregisterOnSharedPreferenceChangeListener(listener);
}
```
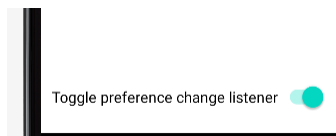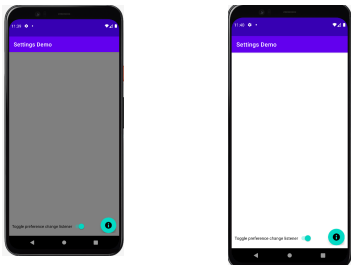
```java
package com.example.myapplication;
import android.graphics.Color;
/**
 * All this does is change the luminosity of the color gray
 * from black to white and all shades in between
 * Ripped off directly from
 * https://gist.github.com/martintreurnicht/f6bbb20a43211bc2060e
 */
public class colorutil {
    public static int mult(int color, double fraction) {
        int red = Color.red(color);
        int green = Color.green(color);
        int blue = Color.blue(color);
        red = multColor(red, fraction);
        green = multColor(green, fraction);
        blue = multColor(blue, fraction);
        int alpha = Color.alpha(color);
        return Color.argb(alpha, red, green, blue);
    }
    private static int multColor(int color, double fraction) {
        return (int)Math.max((color * fraction), 0);
    }
}
```

```java
private void setBackgroundColor(String key) {
    //get the new value of the slider and convert it to a fraction between .5 and 1.0
    double fract= (MainActivity.this.myPreference.getInt(key,50)/100.0) + .5;

    //get the original white
    int color = ContextCompat.getColor(MainActivity.this, R.color.white);

    //scale the white
    color=colorutil.mult(color,fract);

    //set the background color for this viewgroup
    ConstraintLayout cl=findViewById(R.id.cl);
    cl.setBackgroundColor(color);
}
```

```java
if(key.equals(SEEK_BAR_KEY))
    setBackgroundColor(key);
```

```java
private static final String SEEK_BAR_KEY = "seek_bar_key";
```

```java
//set the background color according to defaultsharedprefs saved value
setBackgroundColor(SEEK_BAR_KEY);
```