SharedPreferences

Shared Preferences

- Storing local data
- Best suited for storing settings
- User Key-Value pairs

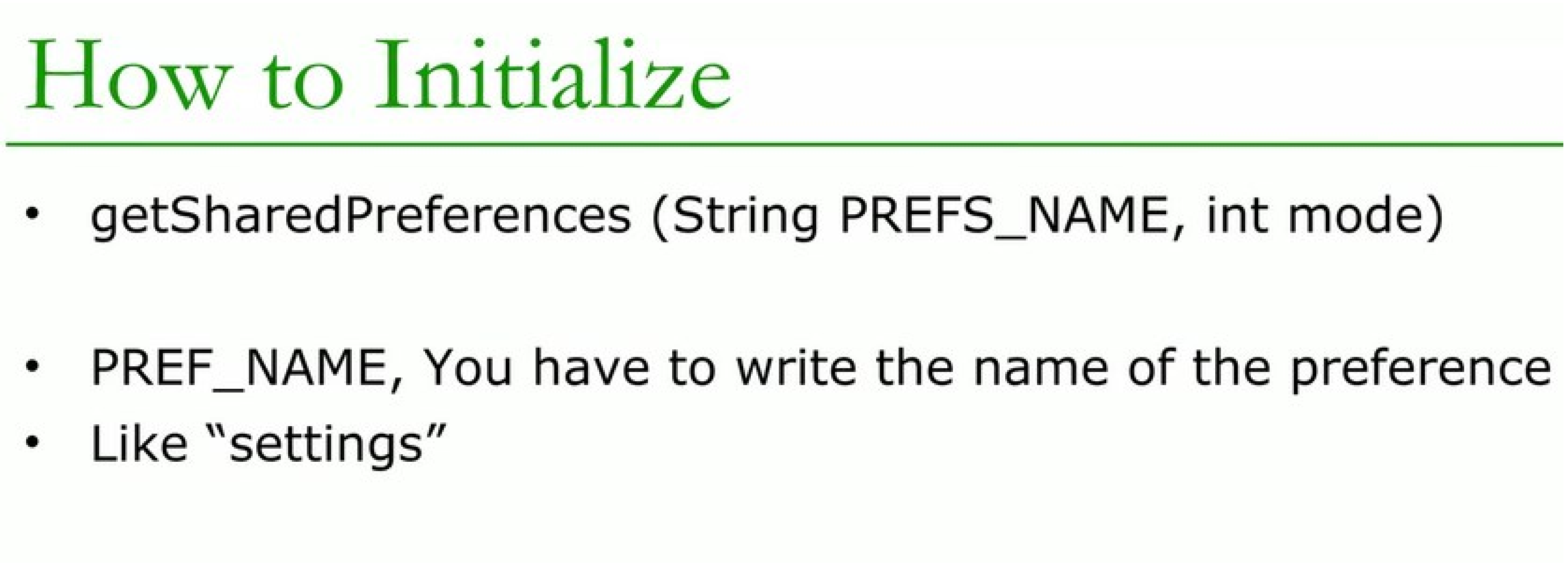
How to use it?

- **getSharedPreferences()**
- Used from within your activity (or other application context), to access application-level preferences.
- This method is used if you need multiple SharedPreference files.

- **getPreferences()**
- Used from within your activity, to access activity-specific preferences.
- This method is used only if you need one preference file for the activity. In this method, we don’t require a name as it will be the only preference file for this activity.

How to use it?

- **getDefaultSharedPreferences()**
- Used on the PreferenceManager, to get the shared preferences that work in concert with Android’s overall preference framework.



How to Initialize

- getSharedPreferences (String PREFS_NAME, int mode)

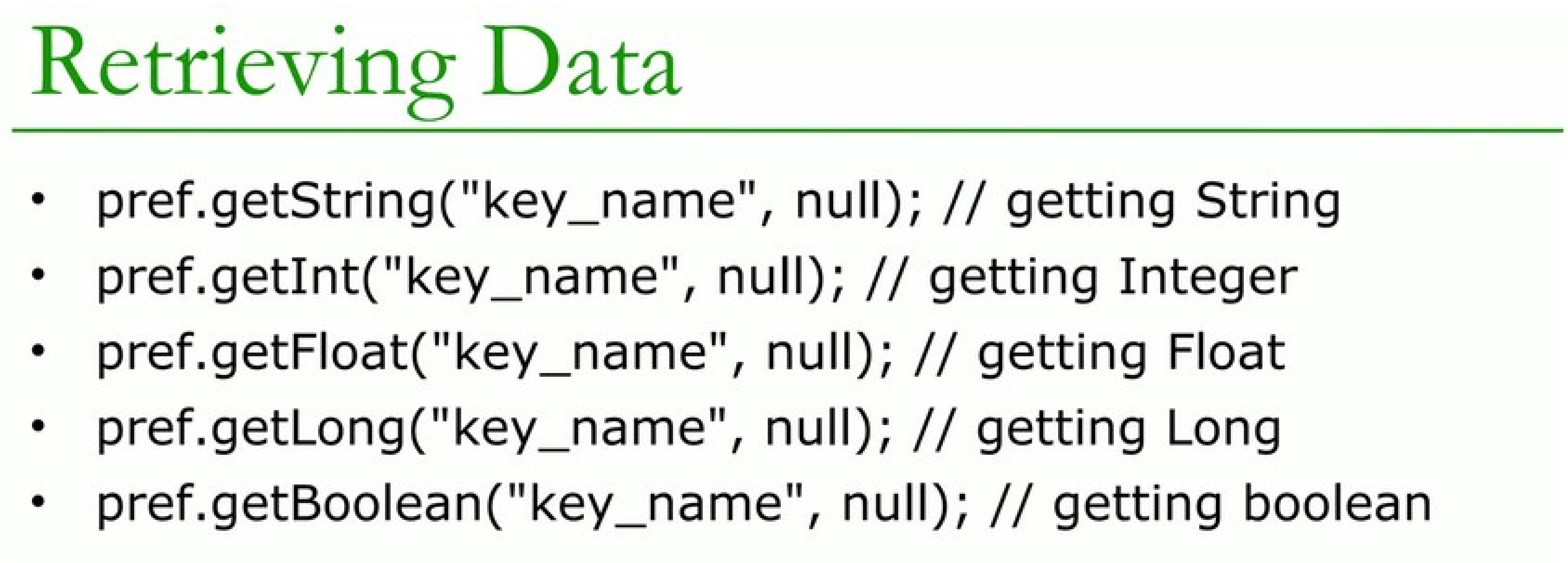
- PREF_NAME, You have to write the name of the preference
- Like “settings”

Modes

- **MODE_PRIVATE**: the default mode, where the created file can only be accessed by the calling application
- **MODE_WORLD_READABLE**: Creating world-readable files is very dangerous, and likely to cause security holes in applications
- **MODE_WORLD_WRITEABLE**: Creating world-writable files is very dangerous, and likely to cause security holes in applications
- **MODE_MULTI_PROCESS**: This method will check for modification of preferences even if the Shared Preference instance has already been loaded
- **MODE_APPEND**: This will append the new preferences with the already existing preferences
- **MODE_ENABLE_WRITE_AHEAD_LOGGING**: Database open flag. When it is set, it would enable write ahead logging by default

Storing Data

- SharedPreferences pref = getApplicationContext().getSharedPreferences(“settings", MODE_PRIVATE);
- // 0 - for private mode
- Editor editor = pref.edit();
- editor.putBoolean("key_name", true); // Storing boolean - true/false
- editor.putString("key_name", "string value"); // Storing string
- editor.putInt("key_name", "int value"); // Storing integer
- editor.putFloat("key_name", "float value"); // Storing float
- editor.putLong("key_name", "long value"); // Storing long
- editor.commit(); // commit changes

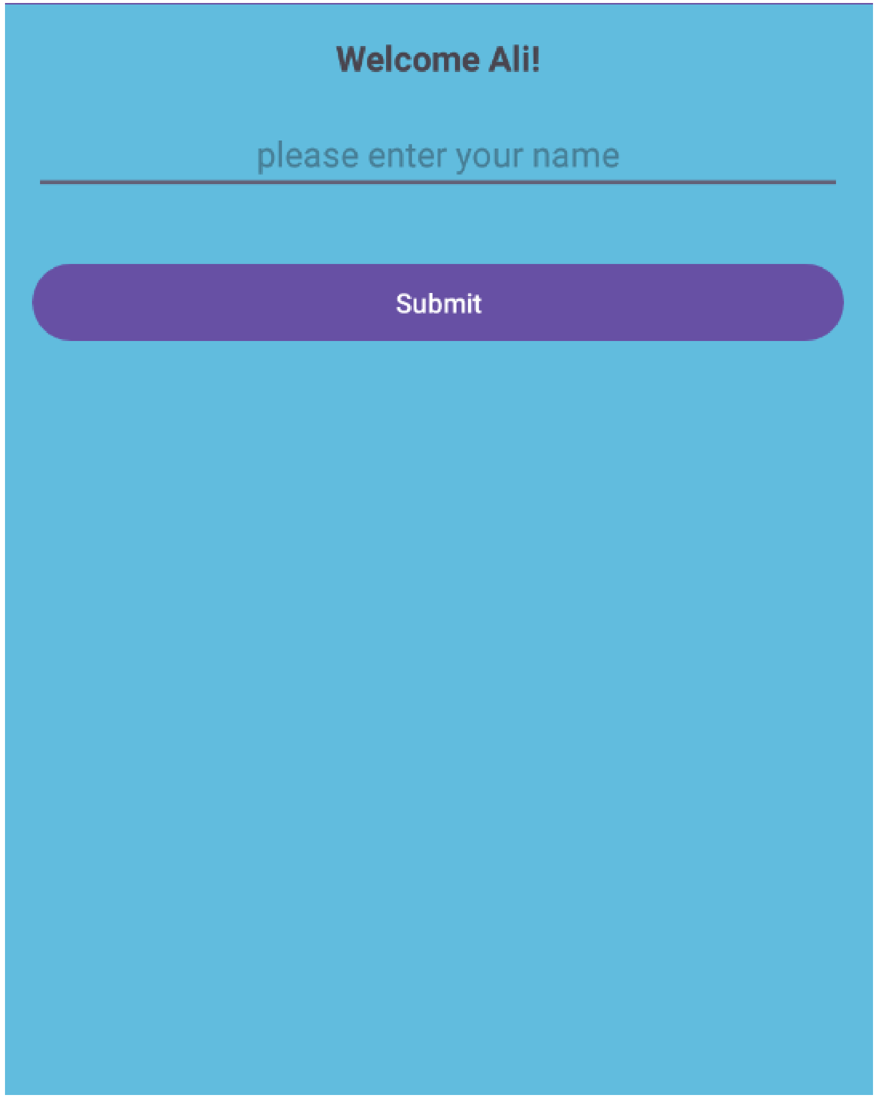


Retrieving Data

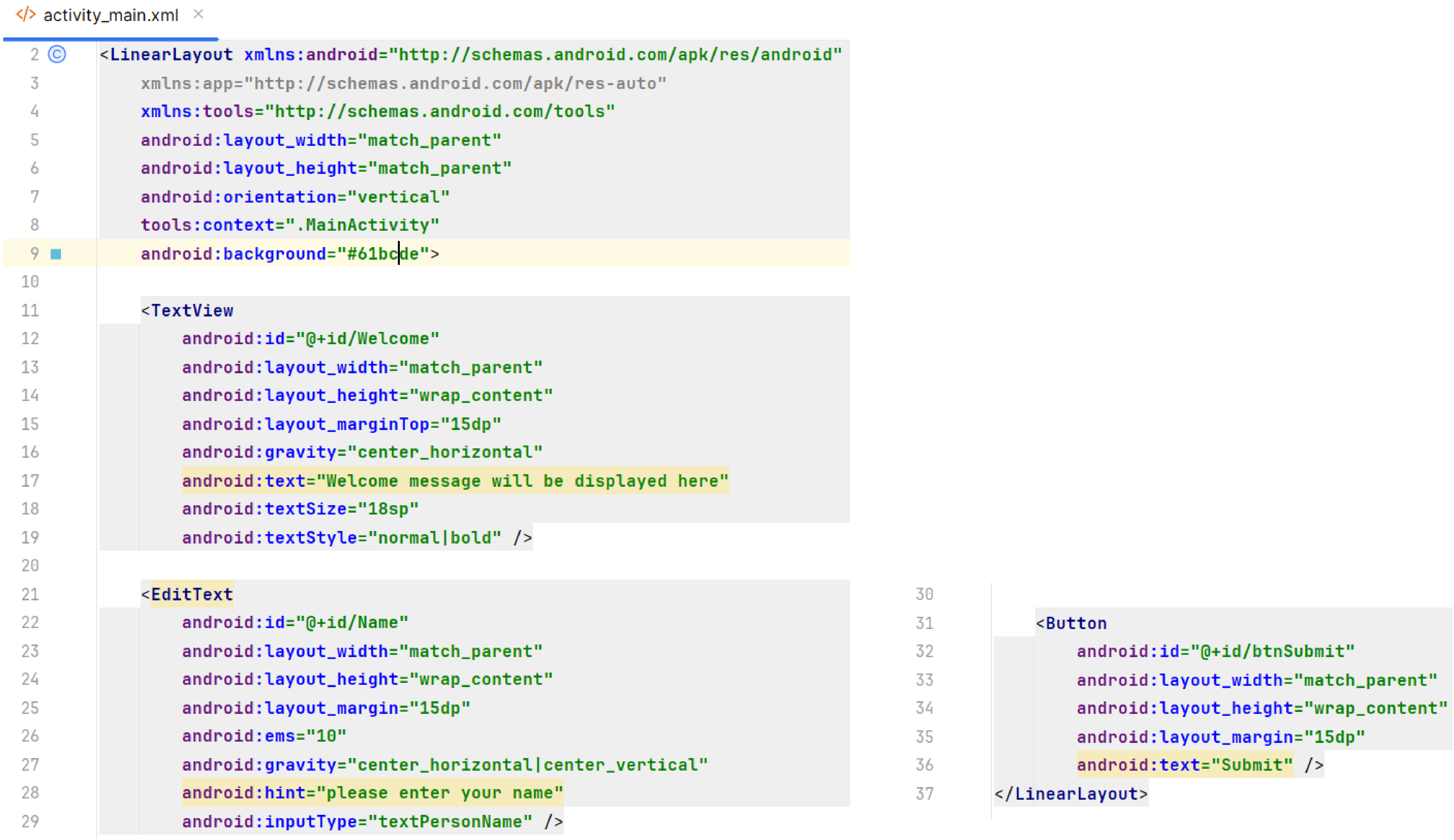
- pref.getString("key_name", null); // getting String
- pref.getInt("key_name", null); // getting Integer
- pref.getFloat("key_name", null); // getting Float
- pref.getLong("key_name", null); // getting Long
- pref.getBoolean("key_name", null); // getting boolean

How to save the changes?

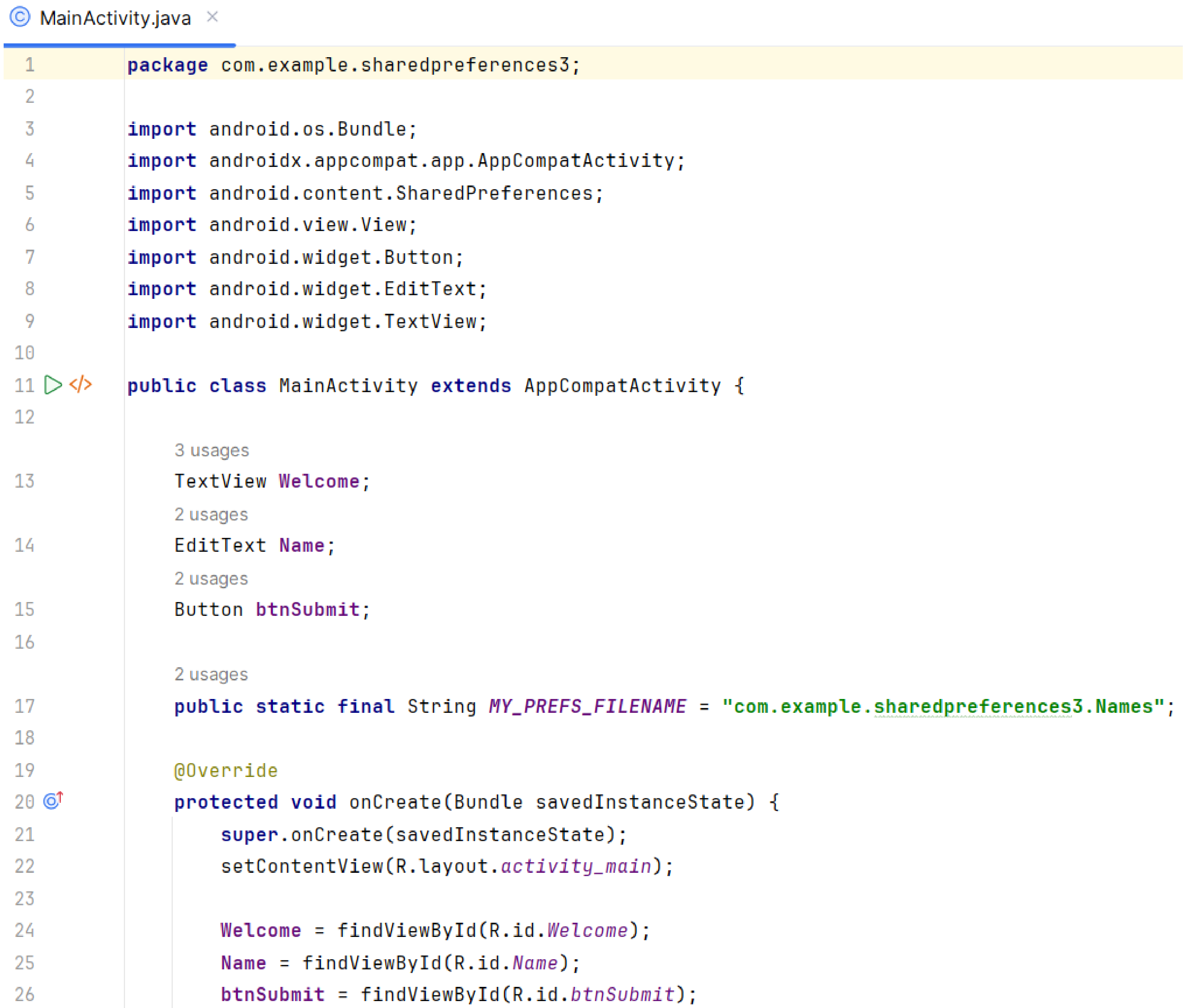
- **apply()** and **commit()** - These methods save the changes. Use **apply()** method to write the updates to disk asynchronously and use **commit()** method to write the data to disk synchronously.



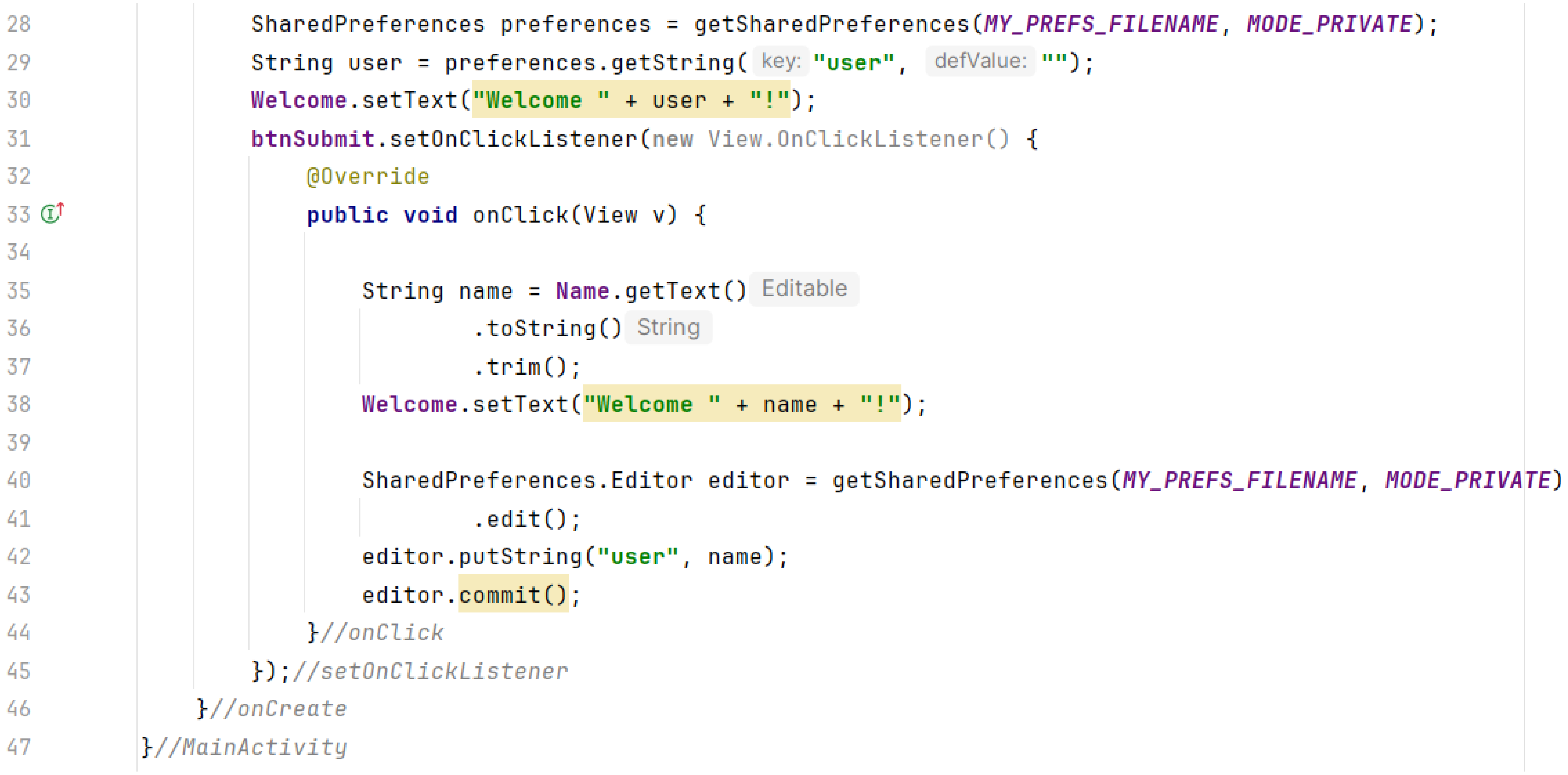
Welcome Ali!



```xml
<LinearLayout 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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity"
    android:background="#61bcde">

    <TextView
        android:id="@+id/Welcome"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:gravity="center_horizontal"
        android:text="Welcome message will be displayed here"
        android:textSize="18sp"
        android:textStyle="normal|bold" />

    <EditText
        android:id="@+id/Name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:ems="10"
        android:gravity="center_horizontal|center_vertical"
        android:hint="please enter your name"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/btnSubmit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:text="Submit" />
</LinearLayout>
```



```java
package com.example.sharedpreferences3;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.content.SharedPreferences;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    TextView Welcome;
    EditText Name;
    Button btnSubmit;

    public static final String MY_PREFS_FILENAME = "com.example.sharedpreferences3.Names";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Welcome = findViewById(R.id.Welcome);
        Name = findViewById(R.id.Name);
        btnSubmit = findViewById(R.id.btnSubmit);
```



```
        SharedPreferences preferences = getSharedPreferences(MY_PREFS_FILENAME, MODE_PRIVATE);
        String user = preferences.getString( key: "user",  defValue: "");
        Welcome.setText("Welcome " + user + "!");
        btnSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String name = Name.getText() Editable
                        .toString() String
                        .trim();
                Welcome.setText("Welcome " + name + "!");

                SharedPreferences.Editor editor = getSharedPreferences(MY_PREFS_FILENAME, MODE_PRIVATE)
                        .edit();
                editor.putString("user", name);
                editor.commit();
            }//onClick
        });//setOnClickListener
    }//onCreate
}//MainActivity
```