# Android Summary

## XML Basics:

## LinearLayout:

```
<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"
    tools:context=".MainActivity"
    android:orientation="vertical">
</LinearLayout>
```

## Button:

```
<Button
        android:id="@+id/btn1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dip"
        android:text="Click Here" />
```

## TextView:

```
<TextView
        android:id="@+id/txt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:text="This is TextView" />
```

## EditText:

```
<EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="type here something"
        android:inputType="textPersonName" />
        <!-- inputType = [number,date,time,textEmailAddress,textPassword,...] -->
```

## CheckBox:

```
    <CheckBox
        android:id="@+id/checkBox"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="choice 1 " />

    <CheckBox
        android:id="@+id/checkBox2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="choice 2" />
```

## RadioGroup and RadioButton:

```
<RadioGroup
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <RadioButton
            android:id="@+id/radioButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="A" />

        <RadioButton
            android:id="@+id/radioButton2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="B" />

        <RadioButton
            android:id="@+id/radioButton3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="C" />
    </RadioGroup>
```

## ImageView:

```
<ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/ic_launcher_background" />
```

## Spinner:

```
<Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
```

## ListView:

```
    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
```

## Menu:

```
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:android="http://schemas.android.com/apk/res/android">

<item
    android:icon="@drawable/imageName"
    android:title="Item"
    app:showAsAction="always" />
<item
    android:title="Item"
    app:showAsAction="never" />
</menu>

ifRoom: The menu item is displayed in the action bar if there is room for it.

always: The menu item is always displayed in the action bar, regardless of whether there is room for it or not. If there is not enough room, the menu item is displayed in the overflow menu.

withText: The menu item is displayed in the action bar with its text. If there is not enough room, the menu item is displayed in the overflow menu.

never: The menu item is not displayed in the action bar and is only displayed in the overflow menu.
```

# java Basics:

## MainActivity:

```
public class MainActivity extends AppCompatActivity {
    Button button;
    TextView textView;
    EditText editText;
    ImageView imageView;
    CheckBox checkBox;
    RadioButton radioButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button = findViewById(R.id.button);
        imageView = findViewById(R.id.imageView);
        textView = findViewById(R.id.textView);
        editText = findViewById(R.id.editText);
        checkBox = findViewById(R.id.checkBox);
        radioButton = findViewById(R.id.radioButton1);
    }
}
```

## setOnClickListener:

```
public class MainActivity extends AppCompatActivity {
    Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button = findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                function();
            }
        });
    }
    private void function(){
        //...
    }
}
```

## TextView:

```
public class MainActivity extends AppCompatActivity {
    TextView textView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textView = findViewById(R.id.textView);
        textView.setText("Hello World!");
        textView.setBackgroundColor(Color.RED);
        textView.append("Hello World Again!");
        textView.setVisibility(View.GONE);
        textView.setVisibility(View.VISIBLE);

    }
}
```

## EditText:

```
public class MainActivity extends AppCompatActivity {
    EditText editText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText = findViewById(R.id.editText);
        String result = editText.getText().toString();
    }
}
```

## CheckBox and RadioButton:

```
public class MainActivity extends AppCompatActivity {
    CheckBox checkBox;
    RadioButton radioButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        checkBox = findViewById(R.id.checkBox); checkBox.setChecked(false);

        radioButton = findViewById(R.id.radioButton1);
        if(checkBox.isChecked()){ //....}
        if(radioButton.isChecked()){//....}
    }
}
```

# imageView:

```
public class MainActivity extends AppCompatActivity {
    ImageView imageView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        imageView = findViewById(R.id.imageView);
        imageView.setImageResource(R.drawable.imageName);
    }
}
```

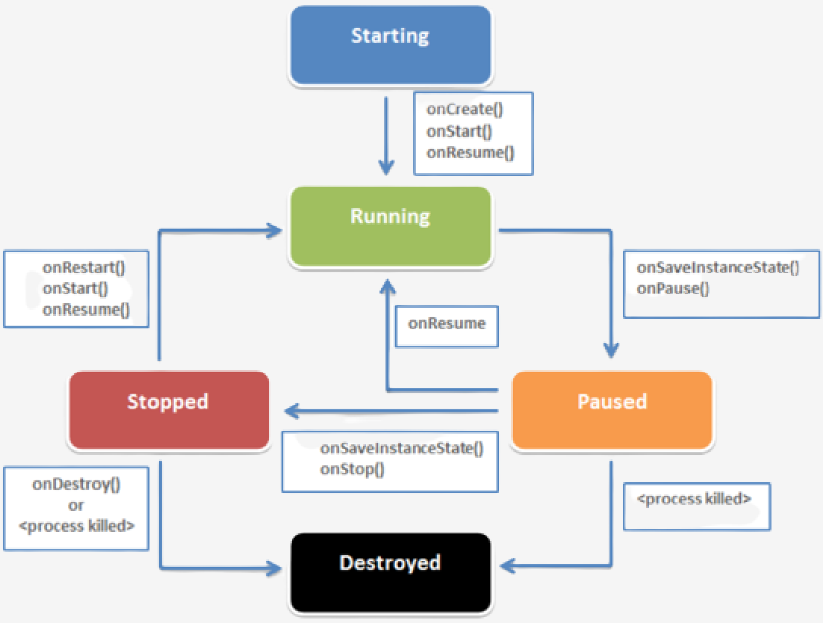
# Activity lifecycle:

```
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

   }

   @Override
   protected void onStart() {
      super.onStart();
   }

z   @Override
   protected void onResume() {
      super.onResume();
   }
   @Override
   protected void onPause() {
      super.onPause();
   }
   @Override
   protected void onStop() {
      super.onStop();
   }
   @Override
   protected void onDestroy() {
      super.onDestroy();
   }
   @Override
   protected void onRestart() {
      super.onRestart();
   }
}
```



# Spinner:

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Spinner spinner = findViewById(R.id.spinner2);
        TextView result = (TextView) findViewById(R.id.Text);

        ArrayAdapter<String> adapter =
        createFromResource(this,R.array.numbers, android.R.layout.simple_spinner_item);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner.setAdapter(adapter);

        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        public void onItemSelected(AdapterView<?> adapterView, View v, int i, long id) {
            String text = adapterView.getItemAtPosition(i).toString();
            result.setText("You chose " + text);
        }
        public void onNothingSelected(AdapterView<?> parent) {}
      });
    }
}
// in res/values/strings.xml
<resources>
    <string name="app_name">spinner</string>
    <string-array name="numbers">
        <item>One</item>
        <item>Two</item>
        <item>Three</item>
        <item>Four</item>
    </string-array>

</resources>
```

# Displaying Toasts:

```
    // Display a short Toast
    Toast.makeText(this, "This is a short Toast", Toast.LENGTH_SHORT).show();

    // Display a long Toast
    Toast.makeText(this, "This is a long Toast. It will take more time to
disappear than the short Toast.", Toast.LENGTH_LONG).show();
```

# ListView Adapter:

```
public class MainActivity extends AppCompatActivity {
  private ListView listView;
  private ArrayList<String> items;
  private ArrayAdapter<String> adapter;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    listView = findViewById(R.id.listView);
    items = new ArrayList<>();
    adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, items);
    listView.setAdapter(adapter);

    // Add some items to the list
    items.add("Item 1");
    items.add("Item 2");
    items.add("Item 3");

    // Notify the adapter that the data has changed
    adapter.notifyDataSetChanged();

    // Set the onItemClickListener
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
      @Override
      public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        String item = (String) parent.getItemAtPosition(position);
        Toast.makeText(MainActivity.this, "Item clicked: " + item, Toast.LENGTH_SHORT).show();
      }
    });

    // Set the onItemLongClickListener
    listView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
      @Override
      public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
        String item = (String) parent.getItemAtPosition(position);
        Toast.makeText(MainActivity.this, "Item long clicked: " + item,
Toast.LENGTH_SHORT).show();
        return true;
      }
    });

    // Set the onItemSelectedListener
    listView.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
      @Override
      public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        String item = (String) parent.getItemAtPosition(position);
        Toast.makeText(MainActivity.this, "Item selected: " + item, Toast.LENGTH_SHORT).show();
      }
      @Override
      public void onNothingSelected(AdapterView<?> parent) {
        Toast.makeText(MainActivity.this, "Nothing selected", Toast.LENGTH_SHORT).show();
      }
    });
  }
}
```

## Intent:

```
public class MainActivity extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Create the Intent
    Intent intent = new Intent(MainActivity.this, SecondActivity.class);

    // Put data in the Intent
    intent.putExtra("message", "Hello from MainActivity");

    // Start the SecondActivity
    startActivity(intent);
  }
}
public class SecondActivity extends AppCompatActivity {

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

    // Get the data from the Intent
    String message = getIntent().getStringExtra("message");

    // Display the message in a TextView
    TextView textView = findViewById(R.id.textView);
    textView.setText(message);
  }
}
```

## SharedPreferences:

```
// To retrieve an instance of SharedPreferences, use the following code:
SharedPreferences sharedPreferences =
    getSharedPreferences("PrefName", this.MODE_PRIVATE);

// To save data to SharedPreferences, use the following code:
SharedPreferences.Editor editor = sharedPreferences.edit();
editor.putInt("keyName", value);
editor.apply();

editor.clear(); // to clear the stored data
editor.apply();

// To retrieve data from SharedPreferences, use the following code:
int retrievedValue = sharedPreferences.getInt("keyName", defaultValue);
```

## Files:

```
- getFilesDir() - returns internal directory for your app
- getCacheDir() - returns a "temp" directory for scrap files
- getResources().openRawResource(R.raw.id) -read an input file from res/raw/
- openFileInput("name" ) - opens a file for reading
- openFileOutput("name", mode) - opens a file for writing

// read a file, and put its contents into a TextView
// (assumes hello.txt file exists in res/raw/ directory)
public class MainActivity extends AppCompatActivity {
    private TextView myTextView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myTextView = findViewById(R.id.textView);

        String allText = "";
        try {
            InputStream inputStream =
getResources().openRawResource(getResources().getIdentifier("FILENAME_WITHOUT_EXTENSION", "raw", getPackageName()));
            Scanner scanner = new Scanner(inputStream);
            while (scanner.hasNextLine()) {
                allText += scanner.nextLine() + "\n";
            }
            myTextView.setText(allText);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```java
public class MainActivity extends AppCompatActivity {

    private TextView myTextView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myTextView = findViewById(R.id.my_text_view);
        writeToFile();
        readFromFile();
    }
    private void writeToFile() {
        try {
            PrintStream output = new PrintStream(openFileOutput("out.txt",
MODE_PRIVATE));
            output.println("Hello, world!");
            output.println("How are you?");
            output.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
    private void readFromFile() {
        try {
            Scanner scan = new Scanner(openFileInput("out.txt"));
            String allText = ""; // read entire file
            while (scan.hasNextLine()) {
                String line = scan.nextLine();
                allText += line + "\n";
            }
            myTextView.setText(allText);
            scan.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

```java
// Get the directory where private files are stored
File filesDirectory = this.getFilesDir();
// Create a new File object with the desired file name
File file = new File(filesDirectory, "users.txt");
// Check if the file exists and create it if it doesn't
if (!file.exists()) {
    try {
        file.createNewFile();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

## Menu:

```
    //specify the menu
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;                   //name_of_the_menu

    }

    //Menu options
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.register:
                Register();
                break;
        }
        return true;
    }

    //Register takes the user to the register activity
    private void Register() {
        Intent intent = new Intent(this, Register.class);
        startActivity(intent);
    }
```

## Database:

```
public class dbHelper extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 1;
    // Database Name
    private static final String DATABASE_NAME = "usersManager";
    // Users table name
    private static final String TABLE_USERS = "users";
    // Users Table Columns names
    private static final String KEY_ID = "id";
    private static final String KEY_NAME = "name";
    private static final String KEY_PASS = "password";

    public dbHelper(Context C) {
        super(C, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_CONTACTS_TABLE = "CREATE TABLE " + TABLE_USERS+
                "(" + KEY_ID + " INTEGER PRIMARY KEY, " + KEY_NAME + " TEXT, "
                + KEY_PASS + " TEXT" + ")";
        db.execSQL(CREATE_CONTACTS_TABLE);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {
        db.execSQL("Drop TABLE IF EXISTS " +  TABLE_USERS);
        onCreate(db);
    }

    List<User> getAllUsers(){
        SQLiteDatabase db = this.getReadableDatabase();
        String query = "SELECT * FROM " + TABLE_USERS;
        Cursor cursor = db.rawQuery(query, null);
        List<User> users = new ArrayList<User>();
        if (cursor.moveToFirst()) {
            do {
                User c = new User (Integer.parseInt(cursor.getString( 0)),
                        cursor.getString( 1), cursor.getString(2));
                users.add(c);
            }
            while (cursor.moveToNext());
        }
        db.close();
        return users;
    }
    Long addUser (User c) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(KEY_NAME, c.getName());
        values.put(KEY_PASS, c.getPassword());
        // Inserting Row
        long id= db.insert(TABLE_USERS,  null, values);
        db.close(); // Closing database connection
        return id;
    }

    public void deleteUser(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_USERS, KEY_ID + " = ?", new String[] { String.valueOf(id)
});
        db.close();
    }
    public boolean userExists(String name, String password) {
        SQLiteDatabase db = this.getReadableDatabase();
        String query = "SELECT * FROM " + TABLE_USERS + " WHERE " + KEY_NAME + " =
? AND " + KEY_PASS + " = ?";
        Cursor cursor = db.rawQuery(query, new String[]{name, password});
        boolean exists = cursor.getCount() > 0;
        cursor.close();
        db.close();
        return exists;
    }
}
```

```
// Database Name for Contact Manager

public class dbHelper extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 1;
    // Database Name
    private static final String DATABASE_NAME = "contactsManager";
    // Contacts table name
    private static final String TABLE_CONTACTS = "contacts";
    // Contacts Table Columns names
    private static final String KEY_ID = "id";
    private static final String KEY_NAME = "name";
    private static final String KEY_PH_NO = "phone_number";

    public dbHelper(Context C) {
        super(C, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_CONTACTS_TABLE = "CREATE TABLE " + TABLE_CONTACTS+
                "(" + KEY_ID + " INTEGER PRIMARY KEY, " + KEY_NAME + " TEXT, "
                + KEY_PH_NO + " TEXT" + ")";
        db.execSQL(CREATE_CONTACTS_TABLE);
    }


    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {
        db.execSQL("Drop TABLE IF EXISTS " +  TABLE_CONTACTS);
        onCreate(db);
    }


    List<Contact> getAllContacts(){
        SQLiteDatabase db = this.getReadableDatabase();
        String query = "SELECT * FROM " + TABLE_CONTACTS;
        Cursor cursor = db.rawQuery(query, null);
        List<Contact> contacts = new ArrayList<Contact>();
        if (cursor.moveToFirst()) {
            do {
                Contact c = new Contact (Integer.parseInt(cursor.getString( 0)),
                        cursor.getString( 1), cursor.getString(2));
                contacts.add(c);
            }
            while (cursor.moveToNext());
        }
        db.close();
        return contacts;
    }
```

```
      Long addContact (Contact c) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(KEY_NAME, c.get_name());
        values.put(KEY_PH_NO, c.get_phone_number());
        // Inserting Row
        long id= db.insert(TABLE_CONTACTS,  null, values);
        db.close(); // Closing database connection
        return id;
    }


    public Contact getContactById(int id) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.query(TABLE_CONTACTS, new String[] { KEY_ID, KEY_NAME,
KEY_PH_NO
                KEY_ID + "=?", new String[] { String.valueOf(id) }, null, null,
null, null);
        if (cursor != null)
            cursor.moveToFirst();
        Contact contact = new Contact(Integer.parseInt(cursor.getString(0)),
                cursor.getString(1), cursor.getString(2));
        db.close();
        return contact;
    }


    public int updateContact(int id, String name, String phone) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(KEY_NAME, name);
        values.put(KEY_PH_NO, phone);
        int rowsAffected = db.update(TABLE_CONTACTS, values, KEY_ID + "=?",
                new String[] { String.valueOf(id) });
        db.close();
        return rowsAffected;
    }

    public void deleteContactById(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_CONTACTS, KEY_ID + " = ?",
                new String[] { String.valueOf(id) });
        db.close();
    }
}
```

# Fragments

Create a Fragment:

```
public class FragmentExample extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        return inflater.inflate(R.layout.fragment_example, false);
    }

    @Override
    public void onActivityCreated(Bundle savedState) {
    super.onActivityCreated(savedState);
    TextView example = (TextView) getActivity().findViewById(R.id.example);
    }
}
```

Add a Fragment to an Activity:

```
FragmentManager manager = getSupportFragmentManager();
FragmentTransaction transaction = manager.beginTransaction();
FragmentExample f = new FragmentExample();
transaction.add(R.id.fragment, f);
transaction.commit();
```

Replace a Fragment:

```
FragmentExample f = new FragmentExample();
transaction = manager.beginTransaction()
transaction.replace(R.id.fragment, f);
transaction.addToBackStack(null);
transaction.commit();
```

# Broadcast Receivers

Creating a Broadcast Receiver

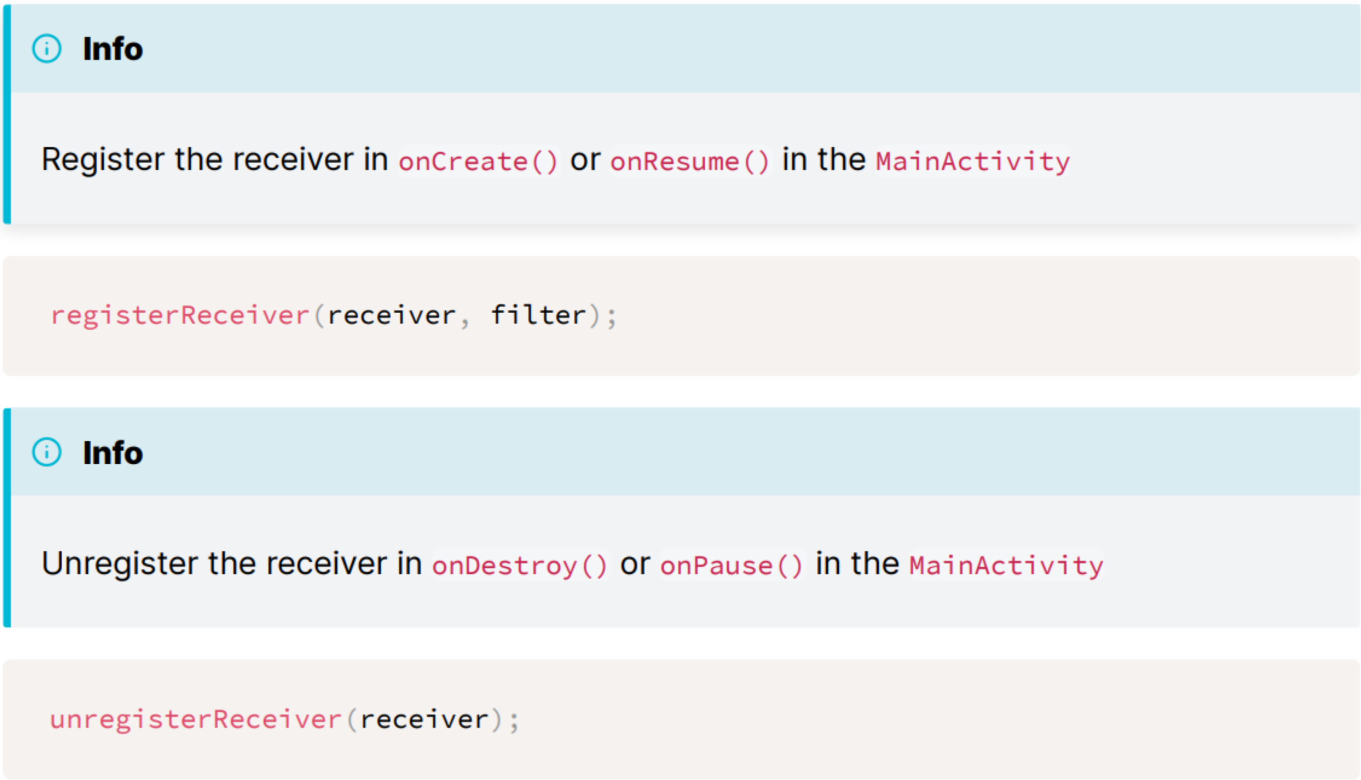
```
public CustomReciever extends BroadCastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
    // this method is called when the BroadCastReceiver is
    receiving an Intent broadcast
    String intentAction = intent.getAction();
    if (intentAction.equals(Intent.ACTION_POWER_CONNECTED))
    // do something
    }
}
```

Registering a Broadcast Receiver

Statically

```
AndroidManifest.xml
<receiver
    android:name=".CustomReciever"
    android:enabled="true"
    android:exported="trure" >
    <intent-filter>
    <action name="android.intent.action.BOOT_COMPLETED"
/>
</intent-filter>
</receiver>
```

Dynamically



**Info**

Register the receiver in `onCreate()` or `onResume()` in the `MainActivity`

```
registerReceiver(receiver, filter);
```

**Info**

Unregister the receiver in `onDestroy()` or `onPause()` in the `MainActivity`

```
unregisterReceiver(receiver);
```