



<> Code - Revisions 2

<> MAD S6 UT2.md

1. Write the syntax for Intent-Filter tag.

Syntax:

```
<intent-filter  
  android:icon="drawable resource"  
  android:label="string resource"  
  android:priority="integer">  
</intent-filter>
```

2. Define Services in Android OS.

- A service is an application component which runs without direct interaction with the user in the background. Services are used for repetitive and potentially long running operations, i.e., Internet downloads, checking for new data, data processing, updating content providers and the like.
- Services are started with two methods namely, `Context.startService()`, `Context.bindService()`.
- Services are the faceless components of Android as they have their individual interfaces. They typically run in the background to perform long-running operations or work for remote processes.

1. Started

- A service is started when an application component, such as an activity, starts it by calling `startService()`.
- Now the service can run in the background indefinitely, even if the component that started it is destroyed.

2. Bound

- A service is bound when an application component binds to it by calling `bindService()`.
- A bound service offers a client-server interface that allows components to interact with the service, send requests, get results, and even do so across processes with `InterProcess`

3.Enlist the steps to publish the Android application>

Step 1: Sign up or Sign up for an account on the Android Developer Console. **Step 2:** Create a new application. **Step 3:** Prepare multimedia. **Step 4:** Prepare code for release. **Step 5:** Build a release-ready APK. **Step 6:** Upload AP K Complete the checklist on the left until all the items have a green checkmark.

4.Draw the activity life cycle.

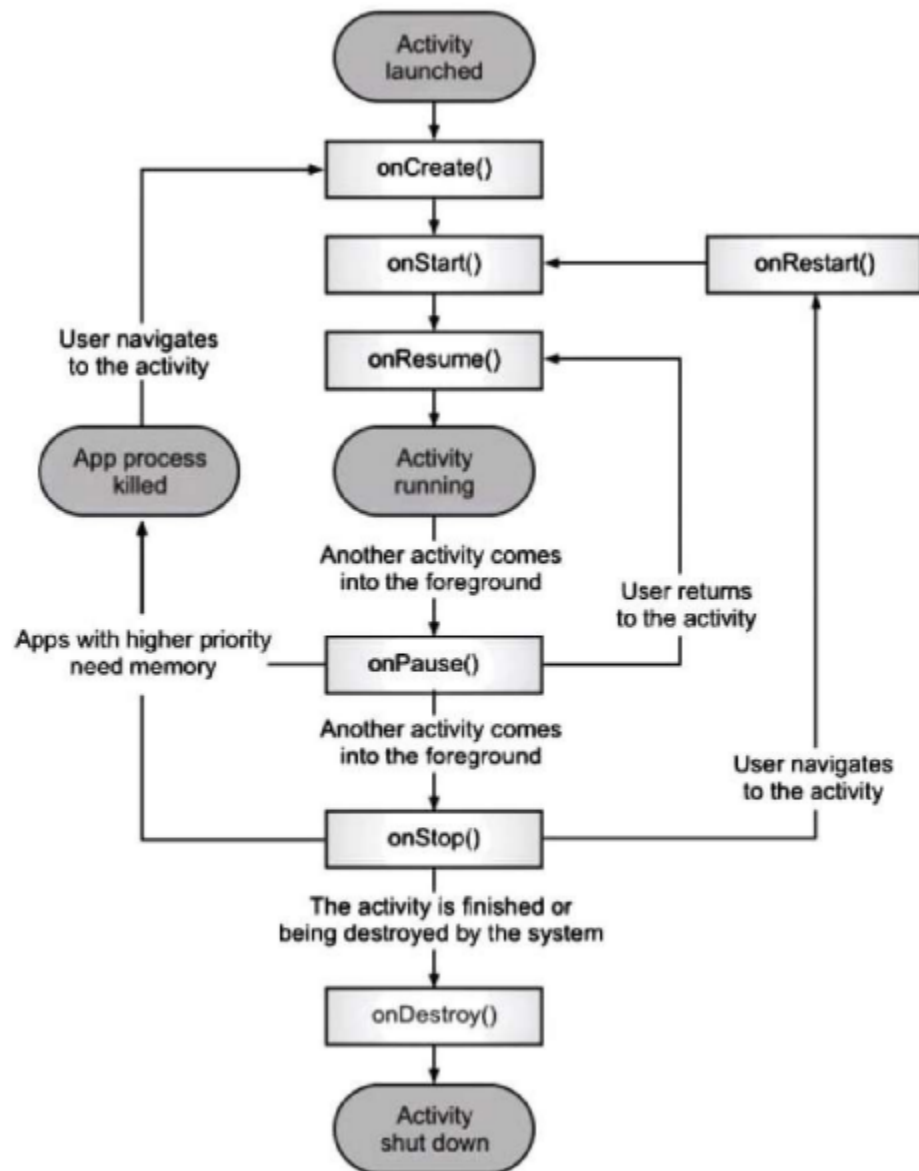


Fig. 5.2: Activity Life Cycle

5.What is Date and Time picker with its methods?

Date Picker

- Android Date Picker allows you to select the date consisting of day, month and year in your custom user interface. For this functionality android provides DatePicker and DatePickerDialog components.
- Apart from date attributes, DatePicker object is also passed into this function. You can use the following methods of the DatePicker to perform further operation.
- Methods of DatePicker:

1. **getDayOfMonth()** - This method gets the selected day of month
2. **getMonth()** - This method gets the selected month
3. **getYear()** - This method gets the selected year
4. **getFirstDayOfWeek()** - This Method returns first day of the week.
5. **getCalendarView()** - This method returns calendar view.

TimePicker

- Android Time Picker allows us to select the time of day in either 24 hour or AM/PM mode.
- The time consists of hours, minutes and clock format. Android provides this functionality through TimePicker class.
- Syntax:

```
<TimePicker
    android:id="@+id/timePicker1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

- Methods of TimePicker:
1. **setCurrentHour(Integer currentHour)** - This method sets the current hour
 2. **setCurrentMinute(Integer currentMinute)** - This method sets the current minute
 3. **is24HourView()** - This method returns true if this is in 24 hour view else false
 4. **setIs24HourView(Boolean is24HourView)** - This method set whether in 24 hour or AM/PM mode
 5. **isEnabled()** - This method returns the enabled status for this view

6. Write a program to display circular progress bar.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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">
```

```

<Button
    android:id="@+id/btnDownloadFile"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Download File"
    android:layout_centerInParent="true"/>
</RelativeLayout>

```

Java Code:

```

package com.example.progprogress;
import androidx.appcompat.app.AppCompatActivity;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.os.Bundle;
public class MainActivity extends
AppCompatActivity {
    Button btnDownloadFile;
    ProgressDialog progressDialog;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ma btnDownloadFile =
            while (progress <= 100) {
                try {
                    progressDialog.setPro
                    findViewById(R.id.btnDownloadFile);

btnDownloadFile.setOnClickListener(View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        progressDialog = new
                        ProgressDialog(MainActivity.this);
                        gress(progress);
                    }
                    progress++;
                    Thread.sleep(200);
                } catch (Exception ex) {}
                progressDialog.setProgressStyle(
                    ProgressDialog.STYLE_HORIZONTAL);
                progressDialog.setTitle("File }
            });
        progressDialog.dismiss();
        Downloading ");
        progressDialog.setProgress(100);
        Thread t = new Thread(new Runnable() {
            @Override
            public void run() {
                int progress = 0;
            }

```

```
    }  
    }  
};
```

7. List sensors in Android and explain anyone in detail.

The Android platform supports following three broad categories of sensors:

1. Motion Sensors
2. Environmental Sensors
3. Position sensors

Motion Sensors:

1. A motion sensor (or motion detector) is an electronic device that is designed to detect and measure movement.
2. Motion sensors are used primarily in home and business security systems.
3. These sensors measure acceleration forces and rotational forces along three axes.
4. This category includes accelerometers, gravity sensors, gyroscopes, and rotational vector sensors.

8. Develop the registration form using the following GUI.

```
<RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:paddingBottom="@dimen/activity_vertical_margin"  
    android:paddingLeft="@dimen/activity_horizontal_margin"  
    android:paddingRight="@dimen/activity_horizontal_margin"  
    android:paddingTop="@dimen/activity_vertical_margin"  
    tools:context="com.android_examples.com.edittextimageadd.MainActivity" >  
    <FrameLayout>  
        <ImageView  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:src="@drawable/banner_image"  
            android:contentDescription="@string/my_image_description" /> >  
    </FrameLayout>  
    <EditText  
        android:id="@+id/editText1"  
        android:hint="Enter your name"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_centerHorizontal="true"  
        android:layout_centerVertical="true"
```

```

        android:ems="10"
        android:drawableLeft="@drawable/name_image">
        <requestFocus />
</EditText>
<EditText
    android:id="@+id/editText1"
    android:hint="Enter your email"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:ems="10"
    android:drawableLeft="@drawable/email_image" >
    <requestFocus />
</EditText>
<EditText
    android:id="@+id/editText1"
    android:hint="Enter your password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:ems="10"
    android:drawableLeft="@drawable/pass_image" >
    <requestFocus />
</EditText>
<EditText
    android:id="@+id/editText1"
    android:hint="confirm password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:ems="10"
    android:drawableLeft="@drawable/conpass_image" >
    <requestFocus />
</EditText>
<EditText
    android:id="@+id/editText1"
    android:hint="Enter your mobile"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:ems="10"
    android:drawableLeft="@drawable/mobile_image" >
    <requestFocus />
</EditText>
<RadioButton
    android:id="@+id/radioMale"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="male"

```

```

        android:checked="true" />
<RadioButton
    android:id="@+id/radioFemale"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="female" />
<Button
    android:id="@+id/btnDisplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="register" />
</RelativeLayout>

```

9. Develop a simple calculator using TableLayout

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.calculatorexample.MainActivity">
    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TableRow
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
            <TextView
                android:id="@+id/txtResult"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_span="4"
                android:layout_gravity="right"/>
            </TableRow>
            <TableRow
                android:layout_width="wrap_content"
                android:layout_height="wrap_content">
                <EditText
                    android:id="@+id/edtInput"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_span="4"
                    android:layout_gravity="right"
                    android:inputType="number"/>
                </TableRow>
                <TableRow
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content">
                    <Button

```

```
        android:id="@+id/btnCE"
        android:text="CE"/>
    <Button
        android:id="@+id/btnC"
        android:text="C"/>
    <Button
        android:id="@+id/btnDelete"
        android:text="Delete"
        android:layout_span="2"/>
</TableRow>
<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <Button
        android:id="@+id/btnNumber7"
        android:text="7"/>
    <Button
        android:id="@+id/btnNumber8"
        android:text="8"/>
    <Button
        android:id="@+id/btnNumber9"
        android:text="9"/>
    <Button
        android:id="@+id/btnNumberAdd"
        android:text="+"/>
</TableRow>
<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <Button
        android:id="@+id/btnNumber4"
        android:text="4"/>
    <Button
        android:id="@+id/btnNumber5"
        android:text="5"/>
    <Button
        android:id="@+id/btnNumber6"
        android:text="6"/>
    <Button
        android:id="@+id/btnNumberSub"
        android:text="-"/>
</TableRow>
<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <Button
        android:id="@+id/btnNumber1"
        android:text="1"/>
    <Button
        android:id="@+id/btnNumber2"
        android:text="2"/>
    <Button
        android:id="@+id/btnNumber3"
```



```

        android:text="3"/>
    <Button
        android:id="@+id/btnNumberMul"
        android:text="*"/>
</TableRow>
<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <Button
        android:id="@+id/btnNumber0"
        android:text="0"/>
    <Button
        android:id="@+id/btnDot"
        android:text="."/>
    <Button
        android:id="@+id/btnResult"
        android:text="="/>
    <Button
        android:id="@+id/btnNumberDiv"
        android:text="/"/>
</TableRow>
</TableLayout>
</android.support.constraint.ConstraintLayout>

```

```

package com.jackrutorial.calculatorexample;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    Button btnNumber0;
    Button btnNumber1;
    Button btnNumber2;
    Button btnNumber3;
    Button btnNumber4;
    Button btnNumber5;
    Button btnNumber6;
    Button btnNumber7;
    Button btnNumber8;
    Button btnNumber9;

    TextView txtResult;

    EditText edtInput;

    Button btnCE;
    Button btnC;

```

```
Button btnDelete;
```

```
Button btnAdd;
```

```
Button btnSub;
```

```
Button btnMul;
```

```
Button btnDiv;
```

```
Button btnDot;
```

```
Button btnResult;
```

```
double val1=Double.NaN;
```

```
double val2;
```

```
String ACTION;
```

```
@Override
```

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

```
    btnNumber0 = (Button) findViewById(R.id.btnNumber0);
```

```
    btnNumber1 = (Button) findViewById(R.id.btnNumber1);
```

```
    btnNumber2 = (Button) findViewById(R.id.btnNumber2);
```

```
    btnNumber3 = (Button) findViewById(R.id.btnNumber3);
```

```
    btnNumber4 = (Button) findViewById(R.id.btnNumber4);
```

```
    btnNumber5 = (Button) findViewById(R.id.btnNumber5);
```

```
    btnNumber6 = (Button) findViewById(R.id.btnNumber6);
```

```
    btnNumber7 = (Button) findViewById(R.id.btnNumber7);
```

```
    btnNumber8 = (Button) findViewById(R.id.btnNumber8);
```

```
    btnNumber9 = (Button) findViewById(R.id.btnNumber9);
```

```
    txtResult = (TextView) findViewById(R.id.txtResult);
```

```
    edtInput = (EditText) findViewById(R.id.edtInput);
```

```
    btnCE = (Button) findViewById(R.id.btnCE);
```

```
    btnC = (Button) findViewById(R.id.btnC);
```

```
    btnDelete = (Button) findViewById(R.id.btnDelete);
```

```
    btnAdd = (Button) findViewById(R.id.btnNumberAdd);
```

```
    btnSub = (Button) findViewById(R.id.btnNumberSub);
```

```
    btnMul = (Button) findViewById(R.id.btnNumberMul);
```

```
    btnDiv = (Button) findViewById(R.id.btnNumberDiv);
```

```
    btnDot = (Button) findViewById(R.id.btnDot);
```

```
    btnResult = (Button) findViewById(R.id.btnResult);
```

```
    btnCE.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            edtInput.setText(null);  
        }  
    });
```

```

btnC.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        val1 = Double.NaN;
        txtResult.setText(null);
        edtInput.setText(null);
    }
});

btnDelete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String number = edtInput.getText().toString();
        if(number != null && number.length() > 0){
            number = number.substring(0, number.length() - 1);
        }
        edtInput.setText(number);
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnAdd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ACTION = "+";
        if(!Double.isNaN(val1)){
            val2 = Double.parseDouble(edtInput.getText().toString());
            val1 = val1 + val2;
        } else {
            val1 = Double.parseDouble(edtInput.getText().toString());
        }

        txtResult.setText(val1 + " + ");
        edtInput.setText(null);
    }
});

btnSub.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ACTION = "-";
        if(!Double.isNaN(val1)){
            val2 = Double.parseDouble(edtInput.getText().toString());
            val1 = val1 - val2;
        } else {
            val1 = Double.parseDouble(edtInput.getText().toString());
        }

        txtResult.setText(val1 + " - ");
        edtInput.setText(null);
    }
});

```

```

btnMul.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ACTION = "*";
        if(!Double.isNaN(val1)){
            val2 = Double.parseDouble(edtInput.getText().toString());
            val1 = val1 * val2;
        } else {
            val1 = Double.parseDouble(edtInput.getText().toString());
        }

        txtResult.setText(val1 + " * ");
        edtInput.setText(null);
    }
});

btnDiv.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ACTION = "/";
        if(!Double.isNaN(val1)){
            val2 = Double.parseDouble(edtInput.getText().toString());
            val1 = val1/val2;
        } else {
            val1 = Double.parseDouble(edtInput.getText().toString());
        }

        txtResult.setText(val1 + " / ");
        edtInput.setText(null);
    }
});

btnResult.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(ACTION != null && ACTION.equals("+")){
            double result = val1 +
Double.parseDouble(edtInput.getText().toString());
            txtResult.setText(null);
            edtInput.setText(String.valueOf(result));
        } else if(ACTION != null && ACTION.equals("-")){
            double result = val1 -
Double.parseDouble(edtInput.getText().toString());
            txtResult.setText(null);
            edtInput.setText(String.valueOf(result));
        } else if(ACTION != null && ACTION.equals("*")){
            double result = val1 *
Double.parseDouble(edtInput.getText().toString());
            txtResult.setText(null);
            edtInput.setText(String.valueOf(result));
        } else if(ACTION != null && ACTION.equals("/")){
            double result = val1 /
Double.parseDouble(edtInput.getText().toString());

```

```

        txtResult.setText(null);
        edtInput.setText(String.valueOf(result));
    }

    ACTION = null;
    val1 = Double.NaN;
    edtInput.setSelection(edtInput.getText().length());
}

});

btnNumber0.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + "0");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnNumber1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + "1");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnNumber2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + "2");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnNumber3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + "3");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnNumber4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + "4");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnNumber5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

```

```

        edtInput.setText(edtInput.getText() + "5");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnNumber6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + "6");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnNumber7.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + "7");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnNumber8.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + "8");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnNumber9.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + "9");
        edtInput.setSelection(edtInput.getText().length());
    }
});

btnDot.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtInput.setText(edtInput.getText() + ".");
        edtInput.setSelection(edtInput.getText().length());
    }
});
}
}

```