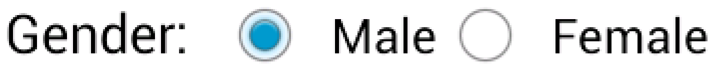
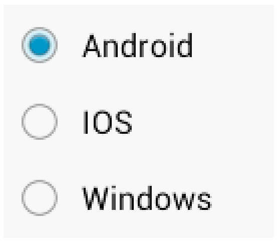
Android Radio Button Using Kotlin.

In this lesson, we will learn about android radio button using kotlin. We will see different attributes of radio button that can be used to customise this widget.

Example





Getting Started

Android Radio Button can be defined as below –

> An Android Radio Button is a two states button that can either be selected or unselected. There is no other option. Also, It can not be unselected, by clicking on it, once it is selected. That’s why we use this widget with radio group. When we use several radio buttons in a radio group, selecting one radio button make sure other radio buttons in the radio group are unselected.

Different Attributes of Android Radio Button widget

Attributes of Radio Button widget are inherited from TextView, Compound Button and View. Some of the popular attributes of Radio Button inherited from TextView are –

Sr.	XML Attributes	Description
1.	android:ems	Defines width of view in ems.
2.	android:gravity	It decides how text inside this view will be aligned. For example,

		center, left, right etc.
3.	android:height	Specifies height of the view.
4.	android:maxWidth	Specifies maximum allowed width of the view.
5.	android:minWidth	Specifies minimum allowed width of the view.
6.	android:width	Specifies width of the view.

Attributes of Radio Button inherited from Compound Button are –

Sr.	XML Attributes	Description
1.	android:button	Drawable for button graphic
2.	android:buttonTint	Specifies Tint to button graphic
3.	android:buttonTintMode	Blending mode used to apply the button graphic tint.

Attributes of Radio Button inherited from View are –

Sr.	XML Attributes	Description
1.	android:id	Specifies unique id of the view.
2.	android:padding	Specifies padding of view.
3.	android:onClick	Decides what to do when view is clicked.
4.	android:visibility	Decides whether to show or hide the view.
5.	android:tooltipText	Text to be shown when cursor id hovered on this view.
6.	android:background	Sets background to view.
7.	android:alpha	Sets alpha in view.

Example of Android Radio Button Using Kotlin

At first, we will create a new android application. Then, we will use use radio button using kotlin file in this application.

1. Creating New Project in Kotlin

Follow steps below to create new project. Please ignore the steps if you have already created the project.

Step	Description
1.	Open Android Studio.
2.	Go to **File** => **New** => **New Project**. Write application name as **RadioButton**. Then, check **Include Kotlin Support** and click **next** button.
3.	Select minimum SDK you need. However, Then, click **next** button
4.	Then, select **Empty Activity** => click **next** => click **finish**.
5.	You will get a newly created project successfully if you have followed steps properly.

Use Radio Button widget in xml file

Open `res/layout/activity_main.xml` file. Then, add code for radio button in it.

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="center"
    android:gravity="center">

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="20dp">

        <RadioButton
            android:id="@+id/radioMale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="10dp"
            android:text="Male"/>

        <RadioButton
            android:id="@+id/radioFemale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="10dp"
            android:text="Female"/>

    </RadioGroup>

</LinearLayout>
```

Here, we have implemented a scenario where you need to select gender(Male or Female) while filling job application. So, in activity_main.xml file, we have added two radio buttons inside a radio group. One radio button represents gender male and another represents female. Now, only one radio button can be selected at a time.

Now, we will access this widget in kotlin file and show a proper message whenever a radio button is selected.

Open **src/main/java/com.yourpackage/MainActivity.kt** file and add below code into it.

```
package com.yourpackage
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.RadioGroup
import android.widget.Toast
class Calc : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val radioGroup = findViewById<RadioGroup>(R.id.radioGroup)
        radioGroup?.setOnCheckedChangeListener { group, checkedId ->
            var text = "You selected: "
            text += if (R.id.radioMale == checkedId)"male "else "female"
            Toast.makeText(applicationContext, text,
Toast.LENGTH_SHORT).show()
        }
    }
}
```

In MainActivity.kt file, we have accessed radio group in which we have added two radio buttons. Then, we have set a listener to display toast message whenever radio button selection changes.

Since AndroidManifest.xml file is very important in any android application, we are also going to mention it here

AndroidManifest.xml file

Code inside **main/AndroidManifest.xml** file is as below.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest package="com.yourpackage"
          xmlns:android="http://schemas.android.com/apk/res/
android">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
```

```
              <action android:name="android.intent.action.MAIN"/>

                <category
android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```