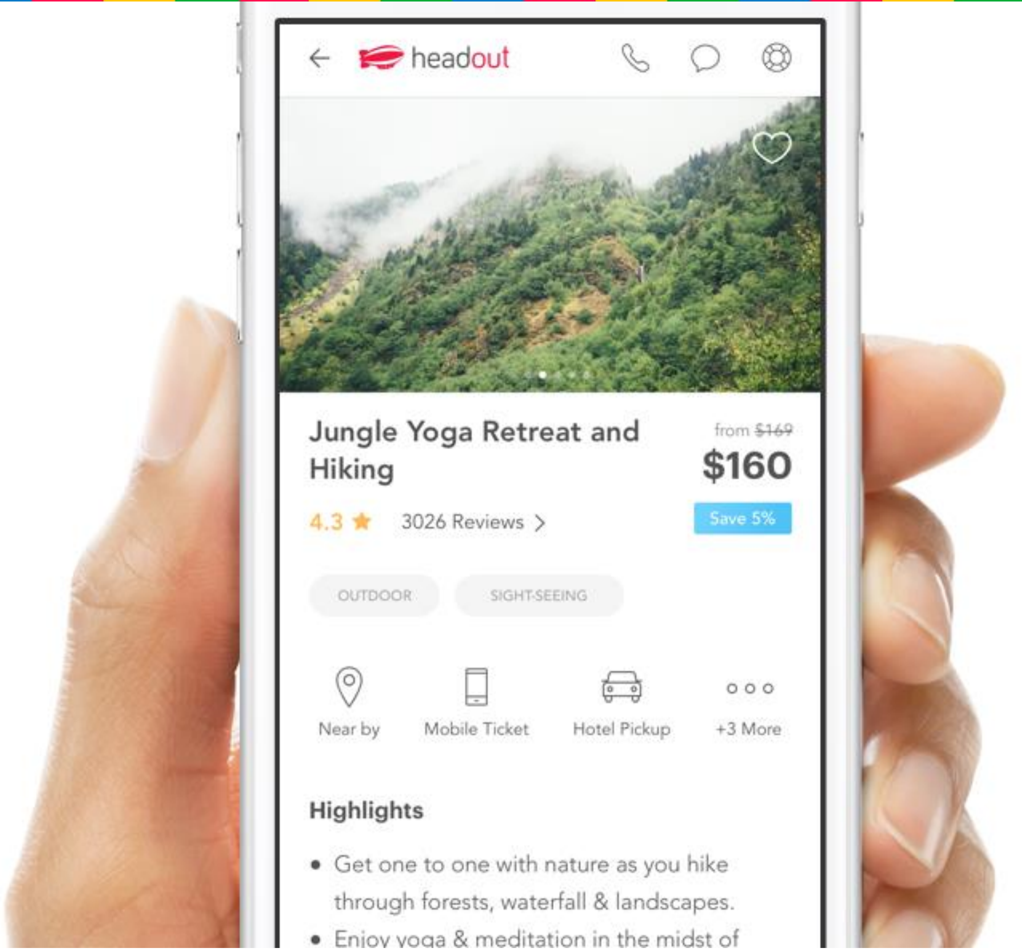
Modul 2: Application UI dan UX

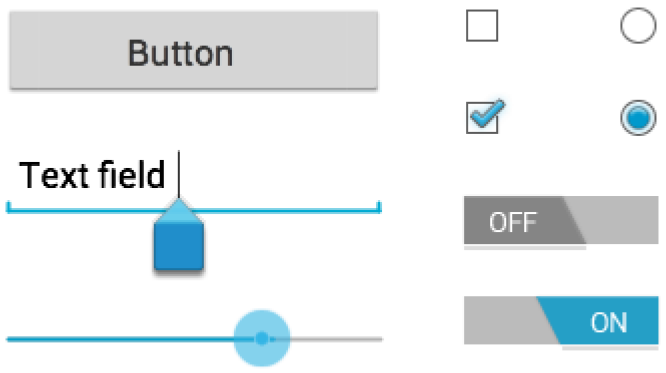
- Views and View Groups


headout
Jungle Yoga Retreat and Hiking
from $169
$160
4.3 ★ 3026 Reviews >
Save 5%
OUTDOOR
SIGHT-SEEING
Near by
Mobile Ticket
Hotel Pickup
+3 More
Highlights
Get one to one with nature as you hike through forests, waterfall & landscapes.
Enjoy yoga & meditation in the midst of

Jurusan Ilmu Komputer
Fakultas MIPA
Universitas Lampung

tutlane dicoding Google Developers Authorized Training Partner

Android
UI Control



Android UI Control

- In android **ui** or **input** controls are the interactive or View components which are used to design the user interface of an application.
- The View is a base class for all UI components in android and it is used to create an interactive UI components such as TextView, EditText, Checkbox, Radio Button, etc. and it responsible for event handling and drawing.

Android UI Control

In android, we can define a UI or input controls in two ways, those are:

- Declare UI elements in XML
- Create UI elements at runtime

Android TextView

- In android, **TextView** is a user interface control which is used to set and display the text to the user based on our requirements.
- The TextView control will act as like label control and it won’t allow users to edit the text.
- In android, we can create a TextView control in two ways either in XML layout file or create it in Activity file programmatically.

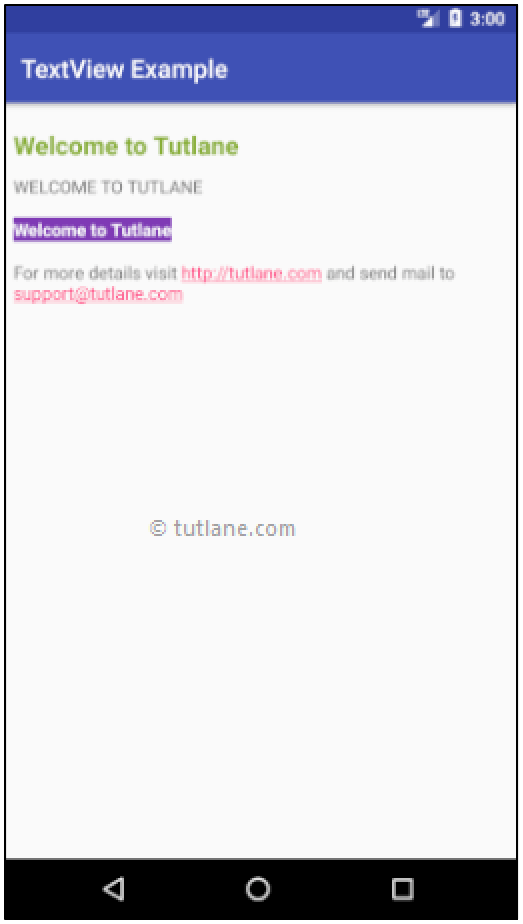
Attribute	Description
android: id	It is used to uniquely identify the control
android:autoLink	It will automatically found and convert urls and email addresses as a clickable links.
android: ems	It is used to make the textview be exactly this many ems wide.
android:hint	It is used to display the hint text when text is empty
android:width	It makes the TextView be exactly this many pixels wide.
android:height	It makes the TextView be exactly this many pixels tall.
android:text	It is used to display the text.
android:textColor	It is used to change the color of text.
android:gravity	It is used to specify how to align the text by the view's x and y axis.
android:maxWidth	It is used to make the TextView be at most this many pixels wide.
android:minWidth	It is used to make the TextView be at least this many pixels wide.
android:textSize	It is used to specify the size of text.
android:textStyle	It is used to change the style (bold, italic, bolditalic) of text.
android:textAllCaps	It is used to present the text in all CAPS
android:typeface	It is used to specify the Typeface (normal, sans, serif, monospace) for the text.
android:textColor	It is used to change the color of text.
android:textColorHighlight	It is used to change the color of text selection highlight.

Set the Text of Android TextView

```
<TextView
      android:id="@+id/textView1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Welcome to Tutlane" />
```

```
TextView tv = (TextView)findViewById(R.id.textView1);
tv.setText("Welcome to Tutlane");
```

Android TextView Example



3:00

TextView Example

Welcome to Tutlane

WELCOME TO TUTLANE

Welcome to Tutlane

For more details visit http://tutlane.com and send mail to support@tutlane.com

© tutlane.com

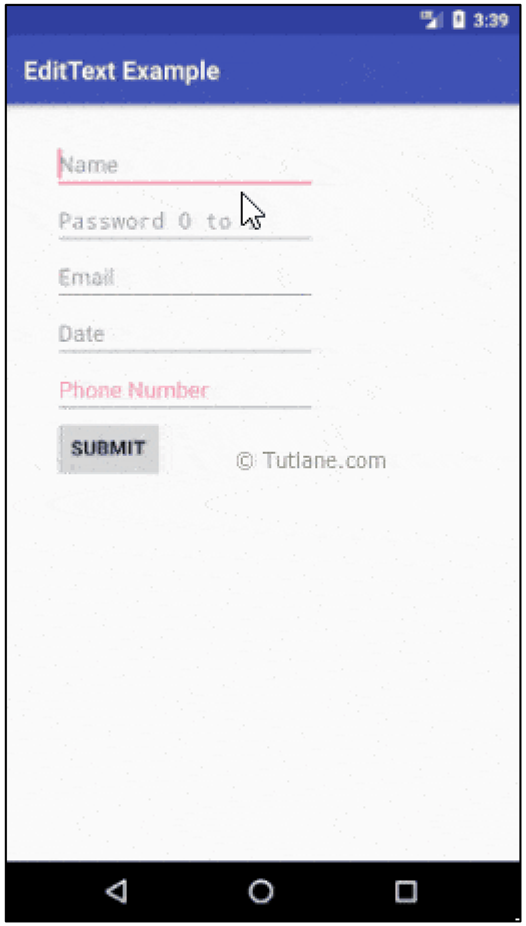
Android EditText

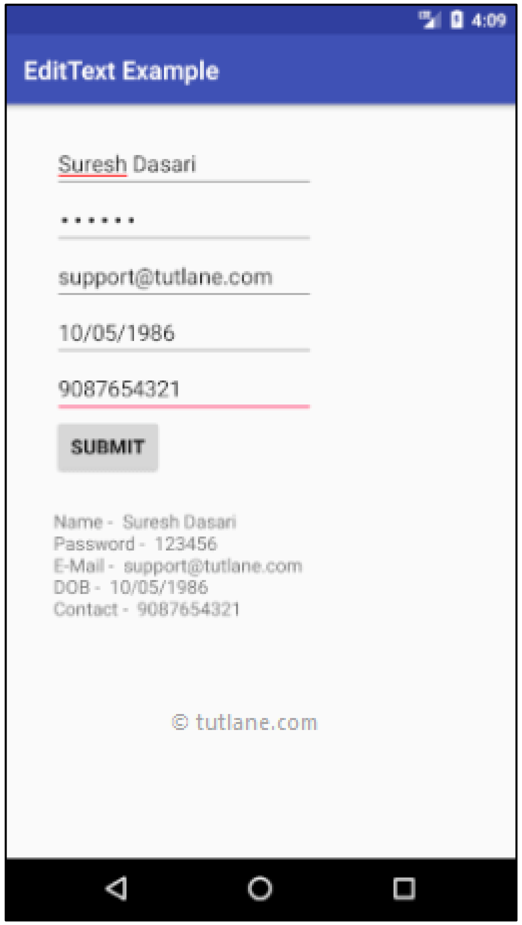
- In android, **EditText** is a user interface control which is used to allow the user to enter or modify the text.
- While using **EditText** control in our android applications, we need to specify the type of data the text field can accept using **inputType** attribute.
- For example, if it accept plain text, then we need to specify the inputType as “**text**”.

Android EditText

- In case if **EditText** field is for password, then we need to specify the inputType as “**textPassword**”.
- In android, **EditText** control is an extended version of **TextView** control with an additional features and it is used to allow users to enter input values.
- In android, we can create **EditText** control in two ways either in XML layout file or create it in Activity file programmatically.

Android EditText Example





Name - Suresh Dasari
Password - 123456
E-Mail - support@tutlane.com
DOB - 10/05/1986
Contact - 9087654321

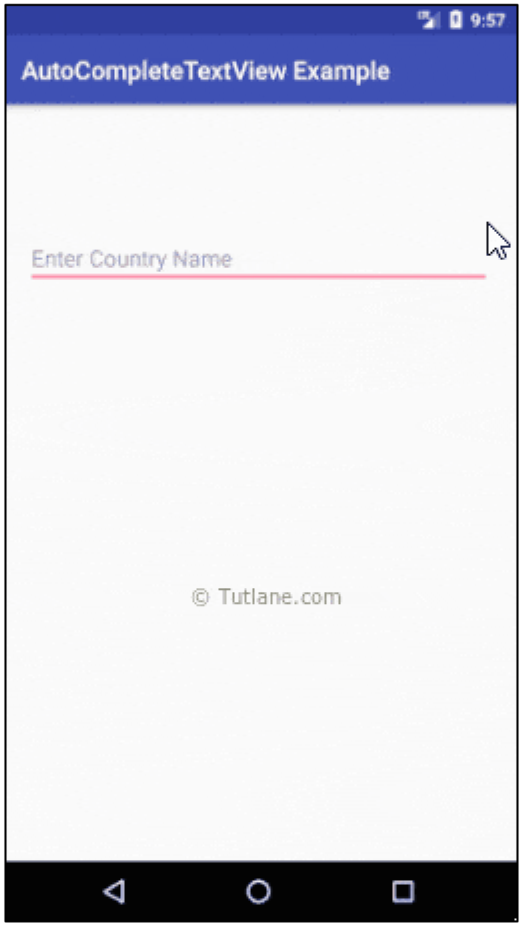
Android AutoCompleteTextView

- In android, **AutoCompleteTextView** is an editable text view which is used to show the list of suggestions based on the user typing text.
- The list of suggestions will be shown as a dropdown menu from which the user can choose an item to replace the content of textbox.
- The **AutoCompleteTextView** is a subclass of EditText class so we can inherit all the properties of EditText in AutoCompleteTextView based on our requirements.

Android AutoCompleteTextView

- Generally, the dropdown list of suggestions can be obtained from the data adaptor and those suggestions will be appeared only after giving the number characters defined in the Threshold limit.
- The Threshold property of AutoCompleteTextView is used to define the minimum number of characters the user must type to see the list of suggestions.
- The dropdown list of suggestions can be closed at any time in case if no item is selected from the list or by pressing the back or enter key.

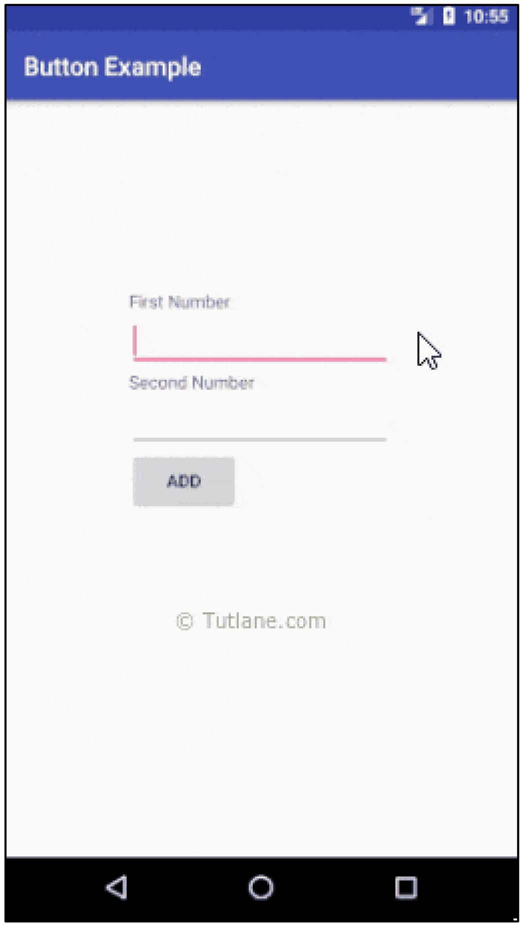
AutoCompleteTextView Example



Android Button

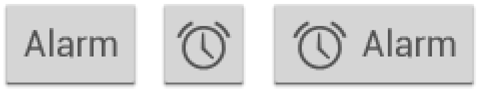
- In android, **Button** is a user interface control which is used to perform an action whenever the user click or tap on it.
- Generally, Buttons in android will contains a text or an icon or both and perform an action when user touches it.
- In android, we have a different type of buttons available to use based on our requirements, those are **ImageButton**, **ToggleButton**, **RadioButton**.

Android Button Example



Android ImageButton

- In android, **Image Button** is a user interface control which is used to display a button with image and to perform an action when user click or tap on it.
- By default, the ImageButton looks same as normal button and it perform an action when user click or touches it, but only difference is we will add a custom image to the button instead of text.



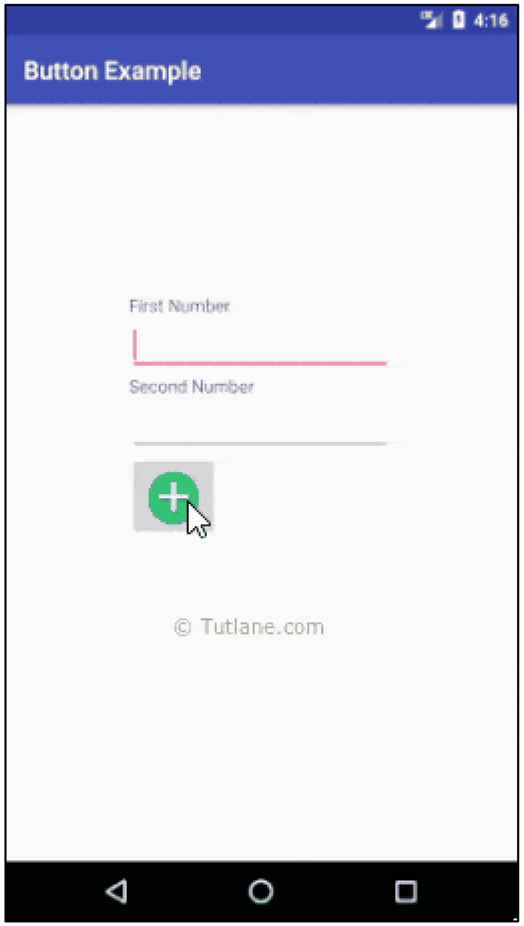
Android ImageButton

- In android, we have a different type of buttons available to use based on our requirements, those are:
  - Button,
  - ImageButton,
  - ToggleButton and
  - RadioButton.

Create ImageButton in XML Layout File

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical" android:layout_width="match_parent"
   android:layout_height="match_parent">
    <ImageButton
       android:id="@+id/addBtn"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:src="@drawable/add_icon" />
</LinearLayout>
```

Android ImageButton Example



Android Toggle Button

- In android, **Toggle Button** is a user interface control which is used to display **ON** (**Checked**) or **OFF** (**Unchecked**) states as a button with a light indicator.
- The **ToggleButton** is useful for the users to change the settings between two states either **ON** or **OFF**. We can add a **ToggleButton** to our application layout by using **ToggleButton** object.

Android Toggle Button

- By default, the android **ToggleButton** will be in **OFF** (**Unchecked**) state. We can change the default state of ToggleButton by using android:checked attribute.
- In case, if we want to change the state of ToggleButton to **ON** (**Checked**), then we need to set **android:checked = “true”**in our XML layout file.

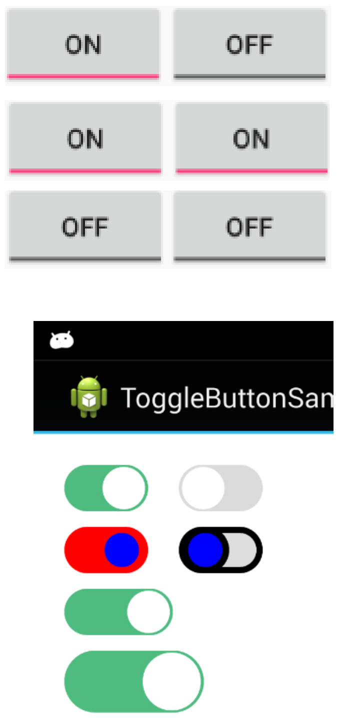
Create ToggleButton in XML Layout File

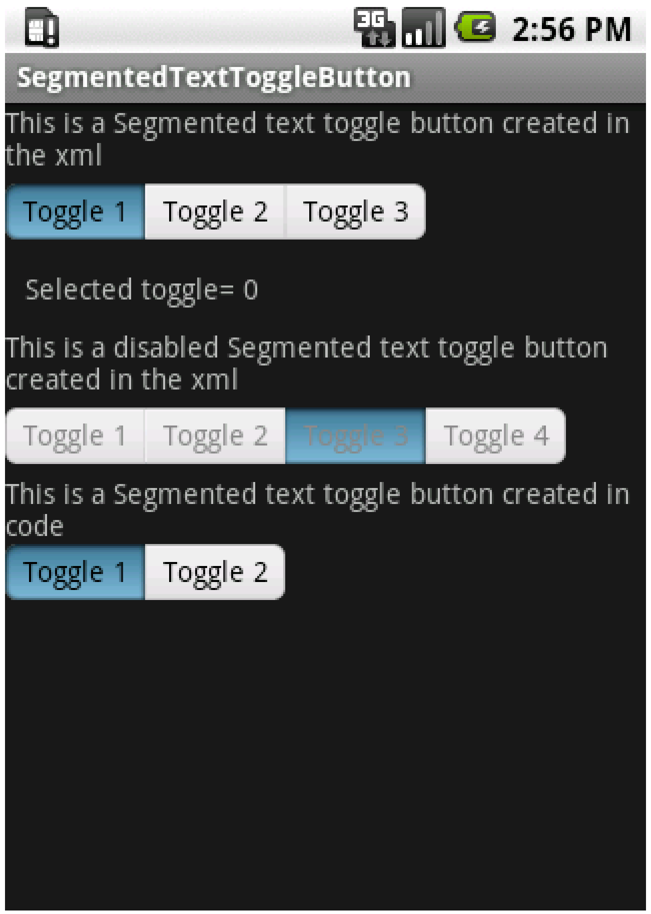
```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent">
    <ToggleButton
        android:id="@+id/toggle1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="120dp"
        android:checked="true"
        android:textOff="OFF"
        android:textOn="ON"/>
</RelativeLayout>
```

Handle Android ToggleButton Click Events

```
ToggleButton toggle = (ToggleButton) findViewById(R.id.togglebutton);
toggle.setOnCheckedChangeListener(new CompoundButton.OnCheckedChang
eListener() {
    public void onCheckedChanged(CompoundButton
buttonView, boolean isChecked) {
        if (isChecked) {
            // The toggle is enabled
        } else {
            // The toggle is disabled
        }
    }
});
```

Output of Android ToggleButton Example



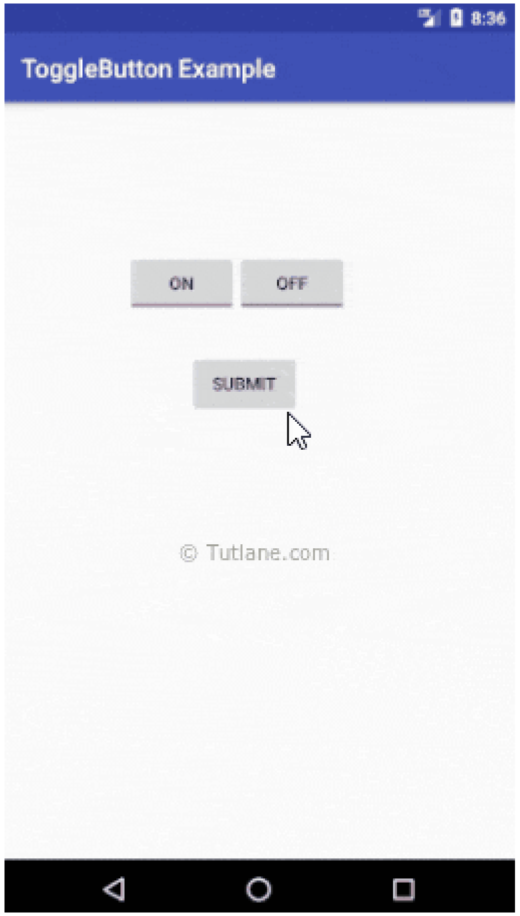


This is a Segmented text toggle button created in the xml

Selected toggle= 0

This is a disabled Segmented text toggle button created in the xml

This is a Segmented text toggle button created in code



Android CheckBox

- In android, **CheckBox** is a two states button that can be either checked (ON) or unchecked (OFF) and it will allow users to toggle between the two states (ON / OFF) based on the requirements.
- Generally, we can use multiple **CheckBox** controls in android application to allow users to select one or more options from the set of values.

Android CheckBox

- By default, the android **CheckBox** will be in **OFF** (**Unchecked**) state. We can change the default state of CheckBox by using **android:checked** attribute.
- In case, if we want to change the state of **CheckBox** to **ON** (**Checked**), then we need to set **android:checked = “true”** in our XML layout file.

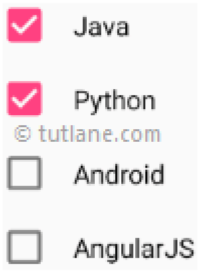
Create CheckBox in XML Layout File

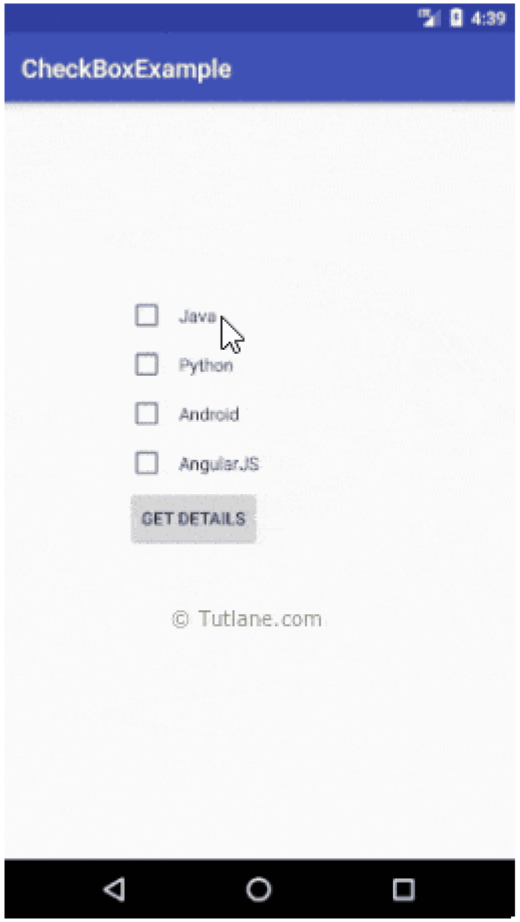
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent">
<CheckBox
   android:id="@+id/chk1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:checked="true"
   android:text="Java" />
</RelativeLayout>
```

Define CheckBox Click Event in Activity File

```
CheckBox chk = (CheckBox) findViewById(R.id.chk1);
chk.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        boolean checked = ((CheckBox) v).isChecked();
        // Check which checkbox was clicked
        if (checked){
            // Do your coding
        }
        else{
            // Do your coding
        }
    }
});
```

Output of Android CheckBox Example





Android RadioButton

- In android, **Radio Button** is a two states button that can be either checked or unchecked and it’s a same as CheckBox control, except that it will allow only one option to select from the group of options.
- The user can press or click on radio button to make it select. In android, CheckBox control allow users to change the state of control either Checked or Unchecked but the radiobutton cannot be unchecked once it is checked.

Android RadioButton

- Generally, we can use **RadioButton** controls in android application to allow users to select only one option from the set of values.
- In android, we use radio buttons with in a **RadioGroup** to combine multiple radio buttons into one group and it will make sure that user can select only one option from the group of multiple options.

Android RadioButton

- By default, the android **RadioButton** will be in **OFF** (**Unchecked**) state. We can change the default state of **RadioButton**by using **android:checked** attribute.
- In case, if we want to change the state of **RadioButton** to **ON** (**Checked**), then we need to set **android:checked = “true”**in our XML layout file.

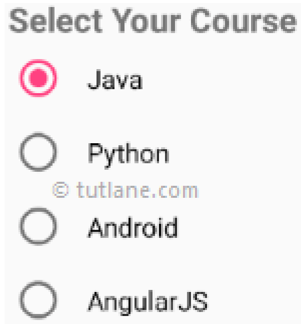
Create RadioButton in XML Layout File

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent">
<RadioGroup
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="vertical">
   <RadioButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Java"
      android:checked="true"/>
</RelativeLayout>
```

RadioButton Click Event in Activity File

```
RadioButton rdb = (RadioButton) findViewById(R.id.radiobutton1);
rdb.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        boolean checked = ((RadioButton) v).isChecked();
        // Check which radiobutton was pressed
        if (checked){
            // Do your coding
        }
        else{
            // Do your coding
        }
    }
});
```

Output of Android RadioButton Example

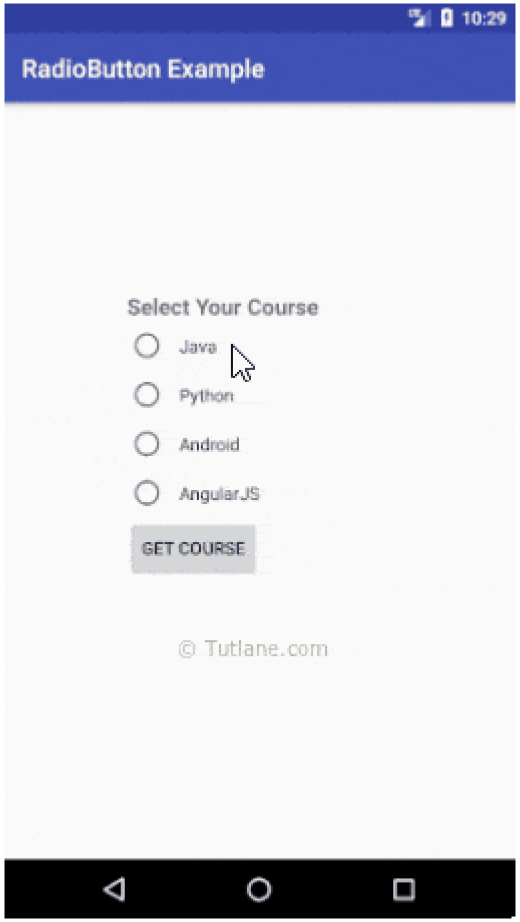

Select Your Course

● Java

○ Python

○ Android

○ AngularJS



Select Your Course

○ Java

○ Python

○ Android

○ AngularJS

Android ProgressBar

- In android, **ProgressBar** is a user interface control which is used to indicate the progress of an operation. For example, downloading a file, uploading a file.

CodeLab