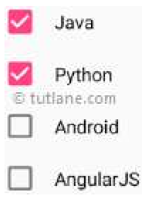
Android UI Control [CheckBox]

Android – UI Control

CheckBox

- **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.
- Following is the pictorial representation of using **CheckBox** control in android applications.



- By default, the android **CheckBox** will be in the **OFF** (**Unchecked**) state. We can change the default state of CheckBox by using **android:checked** attribute. Set **android:checked = “true”** in our XML layout file.
- In android, we can create **CheckBox** control in two ways either in the XML layout file or create it in the activity file programmatically.

Android – UI Control

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>
```

Android – UI Control

Create CheckBox Control in Activity File

- In Activity that hosts our XML layout file, we need to implement click event method like as shown below.

```
LinearLayout layout = (LinearLayout)findViewById(R.id.l_layout);
CheckBox cb = new CheckBox(this);
cb.setText("Tutlane");
cb.setChecked(true);
layout.addView(cb);
```

Android – UI Control

Handle Android CheckBox Click Events

- Whenever the user clicks on **CheckBox** to Select or Deselect the **CheckBox** object will receive an **on-click** event.

- In android, we can define the **CheckBox** click event in two ways either in the XML layout file or create it in the activity file programmatically.

- **Define CheckBox Click Event in XML Layout File**
  - We can define a click event handler for button by adding the **android:onClick** attribute to the <**CheckBox**> element in our XML layout file.

  - The value of **android:onClick** attribute must be the name of method which we need to call in response to a click event and the activity file which hosting XML layout must implement the corresponding method.

- Following is the example of defining a **CheckBox** click event using **android:onClick** attribute in XML layout file.

Android – UI Control

Handle Android CheckBox Click Events

```
<?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">
   <CheckBox
   android:id="@+id/chk1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:checked="true"
   android:text="Java"
   android:onClick="onCheckBoxClick"/>
</LinearLayout>
```

Android – UI Control

Handle Android CheckBox Click Events

- In Activity that hosts our XML layout file, we need to implement click event method like as shown below.

```
public void onCheckboxClicked(View view) {
   // Is the view now checked?
   boolean checked = ((CheckBox) view).isChecked();
   // Check which checkbox was clicked
   switch(view.getId()) {
      case R.id.chk1:
         if (checked)
         // Do your coding
      else
         // Do your coding
         break;
      // Perform your logic
   }
}
```

Android – UI Control

Define CheckBox Click Event in Activity File

- We can define **CheckBox** click event programmatically in activity file rather than XML layout file.

- To define checkbox click event programmatically, create **View.OnClickListener** object and assign it to the button by calling **setOnClickListener(View.OnClickListener)** like as shown below.

Android – UI Control

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
        }
    }
});
```

Android – UI Control

Android CheckBox Control Attributes

Attribute	Description
android:id	It is used to uniquely identify the control
android:checked	It is used to specify the current state of checkbox
android:gravity	It is used to specify how to align the text like left, right, center, top, etc.
android:text	It is used to set the text for a checkbox.
android:textColor	It is used to change the color of text.
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 – UI Control

Android CheckBox Control Attributes

Attribute	Description
android:background	It is used to set the background color for checkbox control.
android:padding	It is used to set the padding from left, right, top and bottom.
android:onClick	It’s the name of the method to invoke when the checkbox clicked.
android:visibility	It is used to control the visibility of control.

Android – UI Control

Android CheckBox Control Example

- In this example we define a multiple **CheckBox** controls and one **Button** control in **LinearLayout** to get the selected values of **CheckBox** controls when we click on **Button** in the android application.

- Create a new android application using android studio and give names as **CheckBoxExample**.

- Now open an **activity_main.xml** file from \res\layout path and write the code like as shown below

Android – UI Control

Android CheckBox Control Example

activity_main.xml

```
<?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">
   <CheckBox
      android:id="@+id/chkJava"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:padding="10dp"
      android:layout_marginTop="150dp"
      android:layout_marginLeft="100dp"
      android:text="Java"
      android:onClick="onCheckboxClicked"/>
<CheckBox
      android:id="@+id/chkPython"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:padding="10dp"
      android:layout_marginLeft="100dp"
      android:text="Python"

android:onClick="onCheckboxClicked"/>
   <CheckBox
      android:id="@+id/chkAndroid"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:padding="10dp"
      android:layout_marginLeft="100dp"
      android:text="Android"

android:onClick="onCheckboxClicked"/>
```

Android – UI Control

Android CheckBox Control Example

```
<CheckBox
      android:id="@+id/chkAngular"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:padding="10dp"
      android:layout_marginLeft="100dp"
      android:text="AngularJS"
      android:onClick="onCheckboxClicked"/>
   <Button
      android:id="@+id/getBtn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:text="Get Details" />
</LinearLayout>
```

Android – UI Control

Android CheckBox Control Example

```
MainActivity.java
package com.tutlane.checkboxexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
   CheckBox android, java, angular, python;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      android = (CheckBox)findViewById(R.id.chkAndroid);
      angular = (CheckBox)findViewById(R.id.chkAngular);
      java = (CheckBox)findViewById(R.id.chkJava);
      python = (CheckBox)findViewById(R.id.chkPython);
      Button btn = (Button)findViewById(R.id.getBtn);
```

Android – UI Control

Android CheckBox Control Example

```
btn.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            String result = "Selected Courses";
            if(android.isChecked()){
            result += "\nAndroid";
            }
            if(angular.isChecked()){
               result += "\nAngularJS";
            }
            if(java.isChecked()){
               result += "\nJava";
            }
            if(python.isChecked()){
               result += "\nPython";
            }
           Toast.makeText(getApplicationContext(), result,
           Toast.LENGTH_SHORT).show();
         }
 });
   }
```

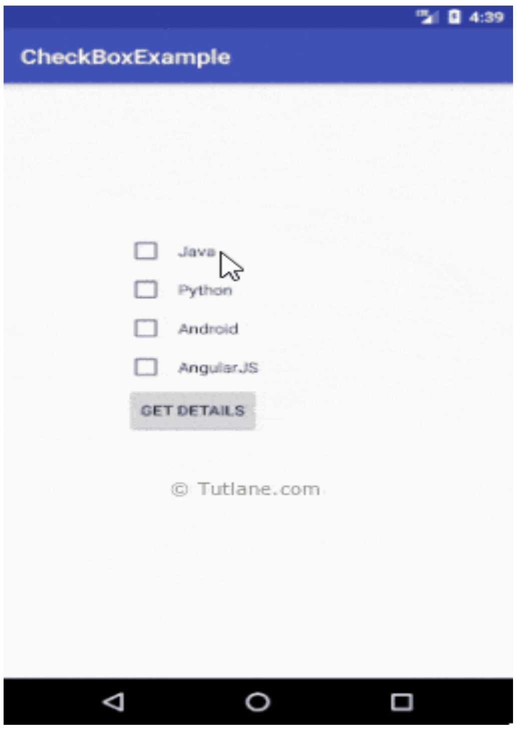
Android – UI Control

Android CheckBox Control Example

```
public void onCheckboxClicked(View view) {
        boolean checked = ((CheckBox) view).isChecked();
        String str="";
        // Check which checkbox was clicked
        switch(view.getId()) {
            case R.id.chkAndroid:
                str = checked?"Android Selected":"Android Deselected";
                break;
            case R.id.chkAngular:
                str = checked?"AngularJS Selected":"AngularJS Deselected";
                break;
            case R.id.chkJava:
                str = checked?"Java Selected":"Java Deselected";
                break;
            case R.id.chkPython:
                str = checked?"Python Selected":"Python Deselected";
                break;
        }
Toast.makeText(getApplicationContext(), str, Toast.LENGTH_SHORT).show();
    }
}
```

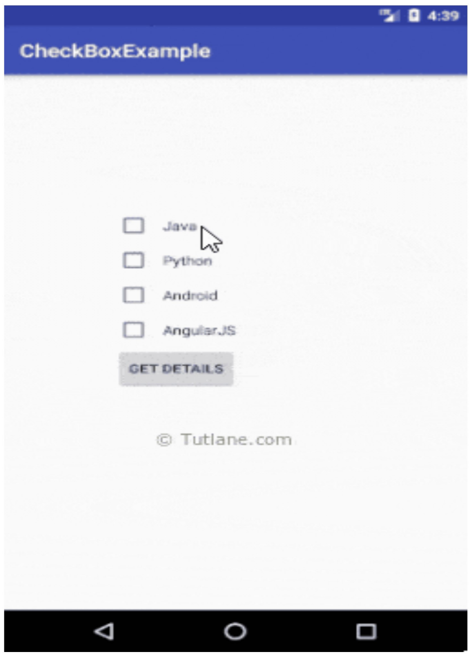
Android – UI Control

Output of Android CheckBox Example



Android – UI Control

Output of Android CheckBox Example
(Once we enter details in all fields and click on Button we will get a result like as shown below.**)**


4:39
CheckBoxExample
Java
Python
Android
AngularJS
GET DETAILS
© Tutlane.com