Button

#### **Attributes of Button in Android:**

Now let’s  we discuss some important attributes that helps us to configure a Button in your xml file (layout).

**1. id:** id is an attribute used to uniquely identify a text Button. Below is the example code in which we set the id of a Button.

<Button

android:id="@+id/simpleButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Abhi Android"/>

**2. gravity:**The gravity attribute is an optional attribute which is used to control the alignment of the text like left, right, center, top, bottom, center\_vertical, center\_horizontal etc.

Below is the example code with explanation included in which we set the right and center vertical gravity for text of a Button.

<Button

android:id="@+id/simpleButton"

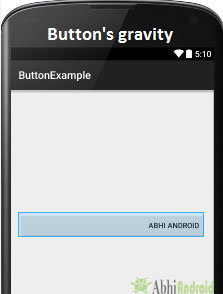
android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Abhi Android"

android:layout\_centerInParent="true"

android:gravity="right|center\_vertical"/><!--set the gravity of button-->



**3. text**: text attribute is used to set the text in a Button. We can set the text in xml as well as in the java class.

Below is the example code with explanation included in which we set the text “Learning Android @ AbhiAndroid” in a Button.

<Button

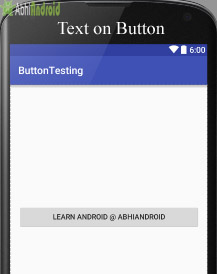
android:id="@+id/simpleButton"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:text="Learn Android @ AbhiAndroid"/><!--display text on button-->



**Setting Text Using Java class:**

Below is the example code in which we set the text on Button programmatically means in java class. The output will be same as the above.

Button button = (Button) findViewById(R.id.simpleButton);

button.setText("Learn Android @ AbhiAndroid");//set the text on button

**4.textColor:** textColor attribute is used to set the text color of a Button. Color value is in the form of “#argb”, “#rgb”, “#rrggbb”, or “#aarrggbb”.

Below is the example code with explanation included in which we set the red color for the displayed text of a Button.

<Button

android:id="@+id/simpleButton"

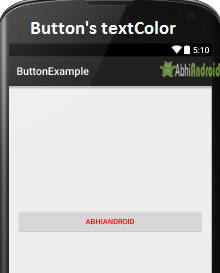
android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:text="AbhiAndroid"

android:textColor="#f00"/><!--red color for the text-->



**Setting Text Color On Button Inside Java class:**

Below is the example code in which we set the text color of a Button programmatically means in java class.

Button simpleButton=(Button) findViewById(R.id.simpleButton);

simpleButton.setTextColor(Color.RED);//set the red color for the text

**5. textSize:** textSize attribute is used to set the size of the text on Button. We can set the text size in sp(scale independent pixel) or dp(density pixel).

Below is the example code in which we set the 25sp size for the text of a Button.

<Button

android:id="@+id/simpleButton"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:text="AbhiAndroid"

android:textSize="25sp" /><!--25sp text size-->

**Setting textSize In Java class:**

Below is the example code in which we set the text size of a Button programmatically means in java class.

Button simpleButton=(Button)findViewById(R.id.*simpleButton*);

simpleButton.setTextSize(25);//set the text size of button

**6. textStyle:** textStyle attribute is used to set the text style of a Button. The possible text styles are bold, italic and normal. If we need to use two or more styles for a Button then “|” operator is used for that.

Below is the example code with explanation included, in which we set the bold and italic text styles for text of a button.

<Button

android:id="@+id/simpleButton"

android:layout\_width="fill\_parent"

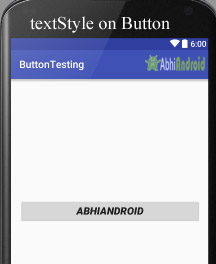
android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:text="AbhiAndroid"

android:textSize="20sp"

android:textStyle="bold|italic"/><!--bold and italic text style-->



**7. background:** background attribute is used to set the background of a Button. We can set a color or a drawable in the background of a Button.

Below is the example code in which we set the gren color for the background, Black color for the displayed text and set 15dp padding from all the side’s for Button.

<Button

android:id="@+id/simpleButton"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

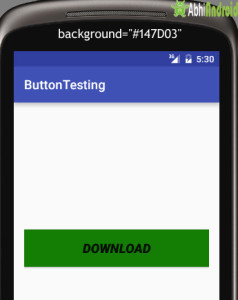
android:text="Download"

android:textSize="20sp"

android:padding="15dp"

android:textStyle="bold|italic"

android:background="#147D03" /><!--Background green color-->



**Setting background in Button In Java class:**

Below is the example code in which we set the background color of a Button programmatically means in java class.

Button simpleButton=(Button)findViewById(R.id.*simpleButton*);

simpleButton.setBackgroundColor(Color.*BLACK*);//set the black color of button background

**8. padding:** padding attribute is used to set the padding from left, right, top or bottom. In above example code of background we also set the 10dp padding from all the side’s of button.

**9. drawableBottom:**drawableBottom is the drawable to be drawn to the below of the text.

**Below is the example code in which we set the icon to the below of the text.**

Make sure you have image saved in your drawable folder name ic\_launcher.

<Button

android:id="@+id/simpleButton"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:background="#147D03"

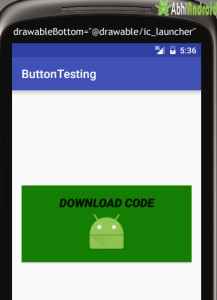
android:text="Download Code"

android:textSize="20sp"

android:padding="15dp"

android:textStyle="bold|italic"

android:drawableBottom="@drawable/ic\_launcher"/><!--image drawable on button-->



**10. drawableTop, drawableRight And drawableLeft:** Just like the above attribute we can draw drawable to the left, right or top of text.

**In the Below example we set the icon to the right of the text. In the same way you can do for other two attribute by your own:**

<Button

android:id="@+id/simpleButton"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:background="#147D03"

android:text="Download Code"

android:textSize="20sp"

android:padding="15dp"

android:textStyle="bold|italic"

android:drawableRight="@drawable/ic\_launcher"/><!--image drawable on Right side of Text on button-->

**Step 1:** Create a new project in Android Studio and name it ButtonExample.

Select File -> New -> New Project and Fill the forms and click "Finish" button.

**Step 2:** Now open res -> layout -> xml (or) activity\_main.xml and add following code. Here we are designing the UI of two button in [Relative Layout](http://abhiandroid.com/ui/relative-layout).

<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=".MainActivity">

<Button

android:id="@+id/simpleButton1"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="100dp"

android:background="#00f"

android:drawableRight="@drawable/ic\_launcher"

android:hint="AbhiAndroid Button1"

android:padding="5dp"

android:textColorHint="#fff"

android:textSize="20sp"

android:textStyle="bold|italic" />

<Button

android:id="@+id/simpleButton2"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:background="#f00"

android:drawableLeft="@drawable/ic\_launcher"

android:hint="AbhiAndroid Button2"

android:padding="5dp"

android:textColorHint="#fff"

android:textSize="20sp"

android:textStyle="bold|italic" />

</RelativeLayout>

**Step 3:** Now Open  app -> package -> MainActivity.java and the following code. Here using setOnClickListener() [method](http://abhiandroid.com/java/method) on button and using [Toast](http://abhiandroid.com/programming/custom-toast-tutorial-example.html) we will display which button is clicked by user.

package example.abhiandriod.buttonexample;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

Button simpleButton1, simpleButton2;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

simpleButton1 = (Button) findViewById(R.id.simpleButton1);//get id of button 1

simpleButton2 = (Button) findViewById(R.id.simpleButton2);//get id of button 2

simpleButton1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Toast.makeText(getApplicationContext(), "Simple Button 1", Toast.LENGTH\_LONG).show();//display the text of button1

}

});

simpleButton2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Toast.makeText(getApplicationContext(), "Simple Button 2", Toast.LENGTH\_LONG).show();//display the text of button2

}

});

}

}

**Output:**

Now start the AVD in Emulator and run the App. You will see two button. Click on any button and you will see the message on screen which button is clicked.