DatePicker

#### **Methods of DatePicker**

Let’s discuss some common methods of a datepicker which are used to configure a DatePicker in our application.

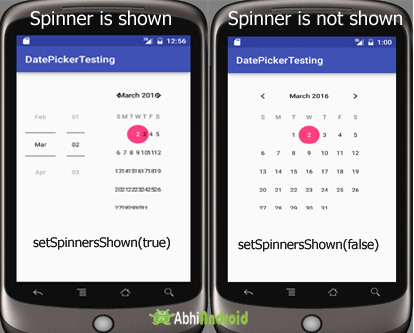
**1. setSpinnersShown(boolean shown):**

This method is used to set whether the [spinner](http://abhiandroid.com/ui/spinner) of the date picker in shown or not. In this method you have to set a Boolean value either true or false. True indicates spinner is shown, false value indicates spinner is not shown. Default value for this function is true.

Below we show the use of setSpinnerShown() function by setting false value.

DatePicker simpleDatePicker = (DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

simpleDatePicker.setSpinnersShown(false); // set false value for the spinner shown function



**2. getDayOfMonth():**

This method is used to get the selected day of the month from a date picker.  This method returns an integer value.

Below we get the selected day of the month from a date picker.

/\*Add in Oncreate() funtion after setContentView()\*/

DatePicker simpleDatePicker = (DatePicker) findViewById(R.id.simpleDatePicker); // initiate a date picker

int day = simpleDatePicker.getDayOfMonth(); // get the selected day of the month

**3. getMonth():**

This method is used to get the selected month from a date picker.  This method returns an integer value.

Below we get the selected month from a date picker.

DatePicker simpleDatePicker = (DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

int month = simpleDatePicker.getMonth(); // get the selected month

**4. getYear():**

This method is used to get the selected year from a date picker.  This method returns an integer value.

Below code is used to get the selected year from a date picker.

DatePicker simpleDatePicker = (DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

int year = simpleDatePicker.getYear(); // get the selected year

**5. getFirstDayOfWeek():**

This method is used to get the first day of the week. This method returns an integer value.

Below code is used to get the first day of the week.

DatePicker simpleDatePicker = (DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

int firstDay=simpleDatePicker.getFirstDayOfWeek(); // get the first day of the week

#### **Attributes of DatePicker**

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

**1. id:**id is an attribute used to uniquely identify a date picker.

<DatePicker

android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

/>

**2. datePickerMode:** This attribute is used to set the Date Picker in mode either spinner or calendar. Default mode is calendar but this mode is not used after api level 21, so from api level 21 you have to set the mode to spinner.

Below is an example code in which we set the mode to spinner for a date picker.

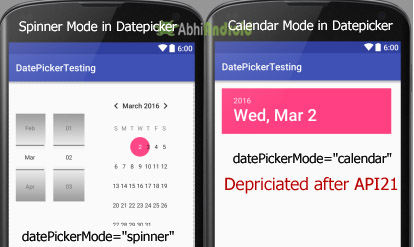
<DatePicker

android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:datePickerMode="spinner" /> <!-- spinner mode of a date picker -->



**3. background:** background attribute is used to set the background of a date picker. We can set a color or a drawable image in the background.

Below we set the red color for the background of a date picker.

<DatePicker

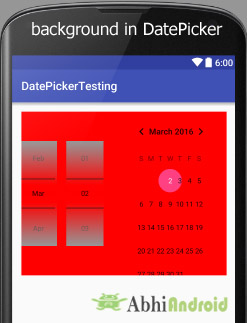
android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:datePickerMode="spinner"

android:background="#f00"/> <!-- red color for the background of the date picker -->



**Setting background of DatePicker In Java Class:**

DatePicker simpleDatePicker=(DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

simpleDatePicker.setBackgroundColor(Color.RED); //  red color for the background of a date picker

**4. padding:** padding attribute is used to set the padding from left, right, top or bottom for a date picker.

* **paddingRight:** set the padding from the right side of the date picker**.**
* **paddingLeft:** set the padding from the left side of the date picker**.**
* **paddingTop:** set the padding from the top side of the date picker**.**
* **paddingBottom:** set the padding from the bottom side of the date picker**.**
* **Padding:** set the padding from the all side’s of the date picker**.**

Below code of padding attribute set the 40dp padding from all the side’s of the date picker.

<DatePicker

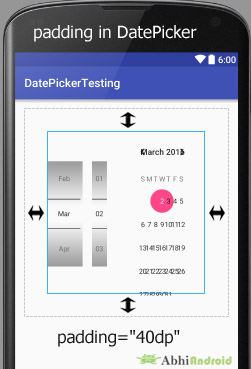
android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:datePickerMode="spinner"

android:padding="40dp"/> <!-- 40dp padding from all the sides of a date picker -->



#### **DatePicker Example in Android Studio:**

**Step 1:** Create a new project and name it **DatePickerExample**

**Step 2:** Open res -> layout -> **activity\_main.xml (or) main.xml** and add following code:

In this step we open an xml file and add the code for displaying a datepicker with spinner mode and a button for getting the date from the datepicker.

<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">

<DatePicker

android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="#150"

android:datePickerMode="spinner" />

<Button

android:id="@+id/submitButton"

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/simpleDatePicker"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="50dp"

android:background="#150"

android:text="SUBMIT"

android:textColor="#fff"

android:textSize="20sp"

android:textStyle="bold" />

</RelativeLayout>

**Step 3:** Open app -> package -> **MainActivity.java**

In this step we open MainActivity where we add the code to initiate the datepicker & a button and then perform onClickListener() event on button so whenever a user clicks on the button the day of the month, month and  year will be displayed by using a [Toast](http://abhiandroid.com/programming/custom-toast-tutorial-example.html).

package example.abhiandroid.datepickerexample;

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.DatePicker;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

DatePicker simpleDatePicker;

Button submit;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// initiate the date picker and a button

simpleDatePicker = (DatePicker) findViewById(R.id.simpleDatePicker);

submit = (Button) findViewById(R.id.submitButton);

// perform click event on submit button

submit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// get the values for day of month , month and year from a date picker

String day = "Day = " + simpleDatePicker.getDayOfMonth();

String month = "Month = " + (simpleDatePicker.getMonth() + 1);

String year = "Year = " + simpleDatePicker.getYear();

// display the values by using a toast

Toast.makeText(getApplicationContext(), day + "\n" + month + "\n" + year, Toast.LENGTH\_LONG).show();

}

});

}

}

**Output:**

Now run the App in AVD and you will see datepicker will appear on the screen. Choose the date, month & year and click submit. The date you selected will appear on Screen.