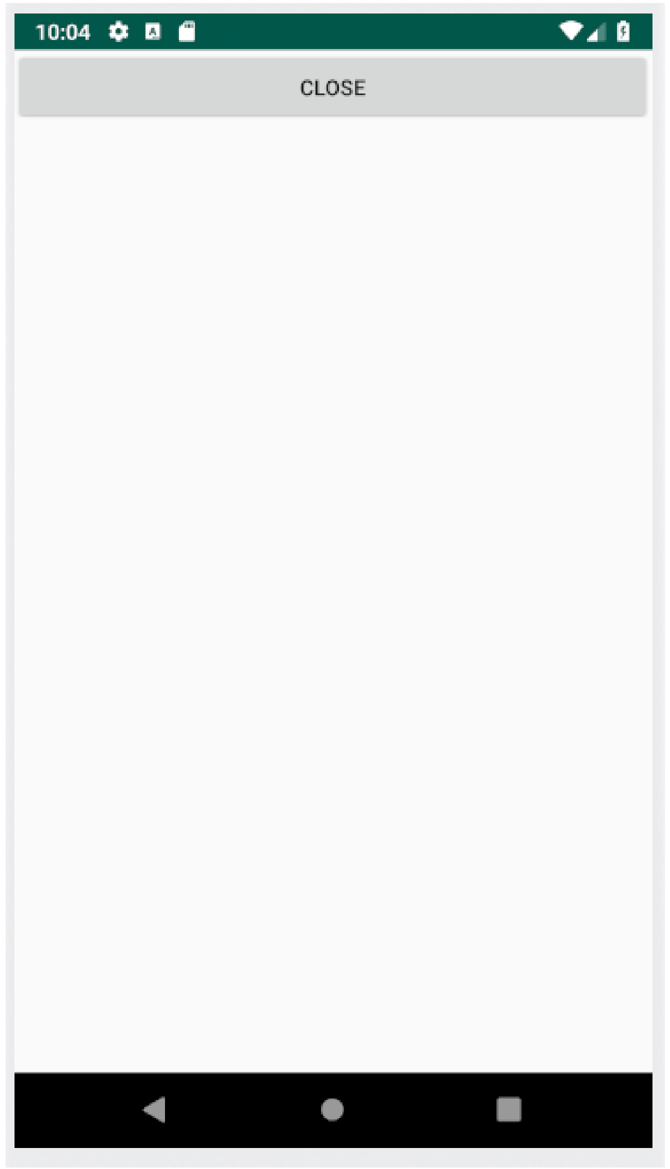
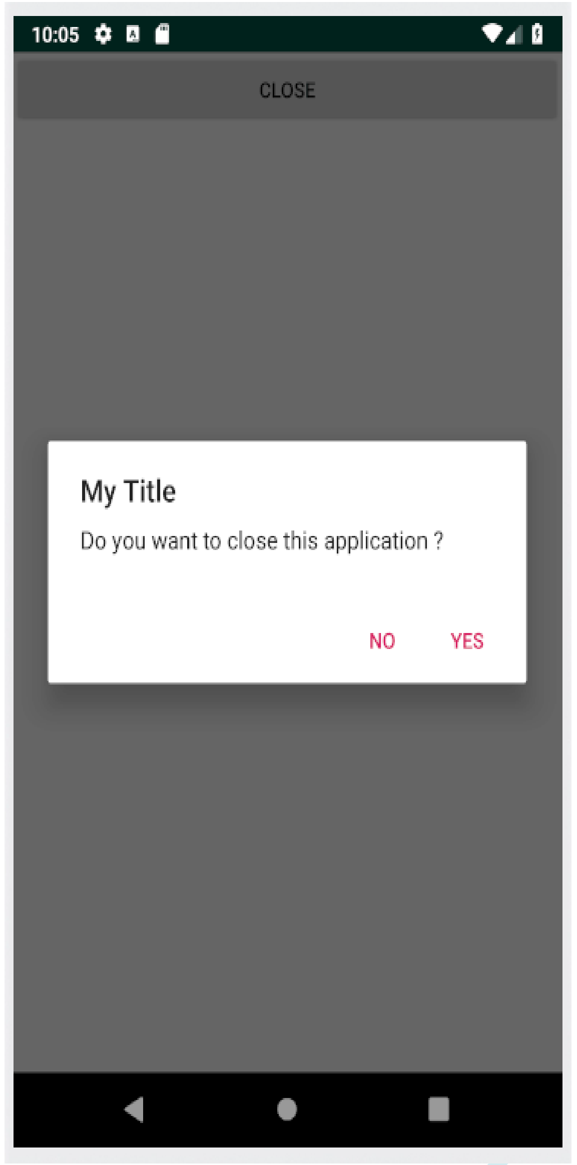
# Widgets and Layouts - 2

# AlertDialog

AlertDialog can be used to display the dialog message with **OK** and **Cancel** buttons.

It can be used to interrupt and ask the user about his/her choice to continue or discontinue.

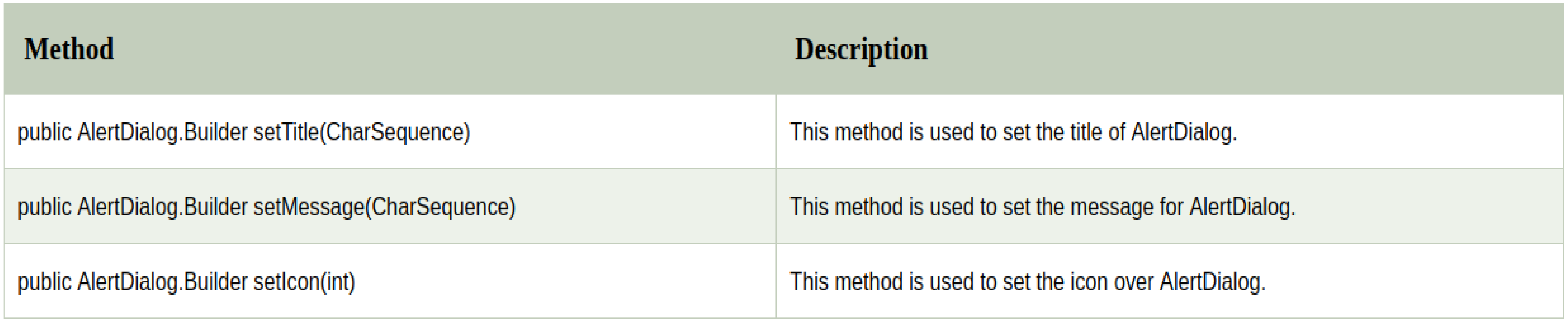




My Title

Do you want to close this application ?

# Methods of AlertDialog class



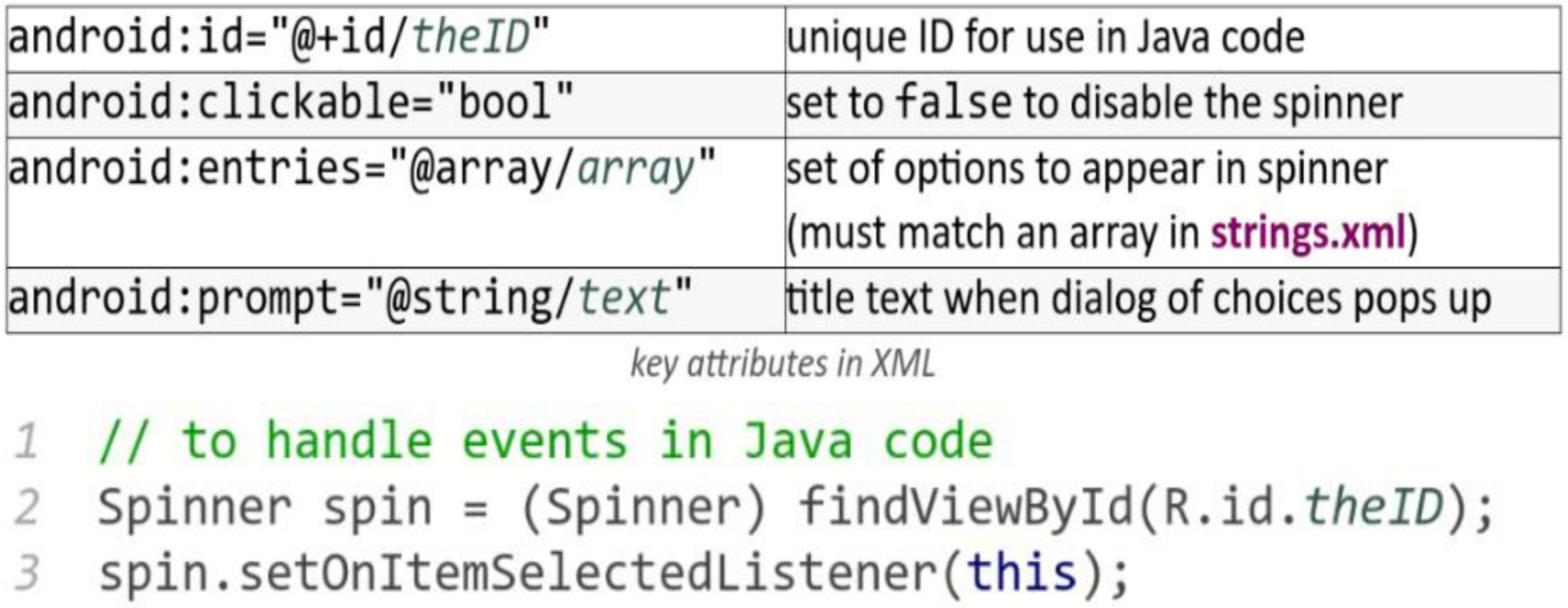
| Method | Description |
|---|---|
| public AlertDialog.Builder setTitle(CharSequence) | This method is used to set the title of AlertDialog. |
| public AlertDialog.Builder setMessage(CharSequence) | This method is used to set the message for AlertDialog. |
| public AlertDialog.Builder setIcon(int) | This method is used to set the icon over AlertDialog. |

```java
Button closeButton;
AlertDialog.Builder builder;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_views);

    closeButton = (Button) findViewById(R.id.btnClose);
    builder = new AlertDialog.Builder( context: this);

    closeButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            builder.setMessage("Do you want to close this application ?")
                    .setCancelable(false)
                    .setPositiveButton( text: "Yes", new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
                            finish();
                            Toast.makeText(getApplicationContext(), text: "You clicked Yes",
                                    Toast.LENGTH_SHORT).show();
                        }
                    })
                    .setNegativeButton( text: "No", new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
                            //  Action for 'NO' Button
                            dialog.cancel();
                            Toast.makeText(getApplicationContext(), text: "You clicked No",
                                    Toast.LENGTH_SHORT).show();
                        }
                    });
            //Creating dialog box
            AlertDialog alert = builder.create();
            alert.setTitle("My Title");
            alert.show();
        }
    });
```

Fr

# Spinner

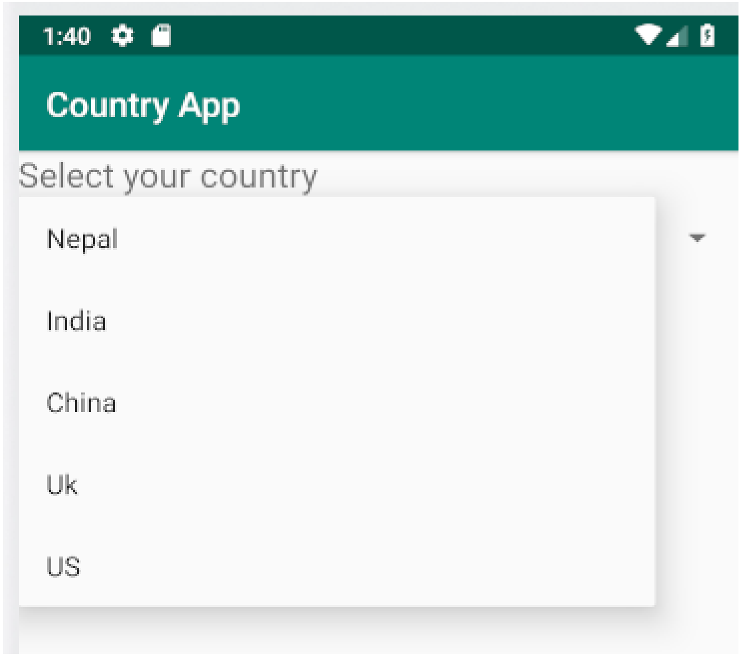
*A drop-down menu of selectable items.*



| | |
|---|---|
| `android:id="@+id/theID"` | unique ID for use in Java code |
| `android:clickable="bool"` | set to `false` to disable the spinner |
| `android:entries="@array/array"` | set of options to appear in spinner (must match an array in **strings.xml**) |
| `android:prompt="@string/text"` | title text when dialog of choices pops up |

*key attributes in XML*

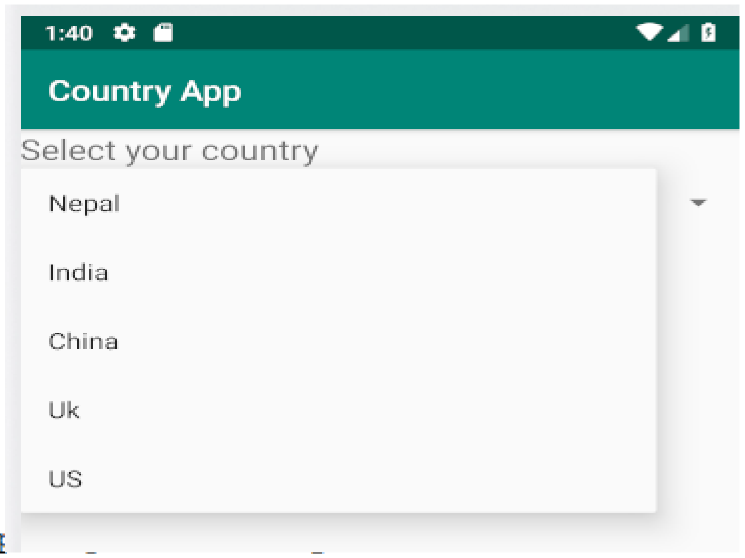
```java
// to handle events in Java code
Spinner spin = (Spinner) findViewById(R.id.theID);
spin.setOnItemSelectedListener(this);
```



# Spinner

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".SpinnerActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="Select your country"/>

    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/spinCountry"/>

</LinearLayout>
```

```java
public class SpinnerActivity extends App

    private Spinner spinCountry;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_spinner);

        spinCountry = findViewById(R.id.spinCountry);

        String countries[] ={"Nepal","India","China","Uk","US"};
        ArrayAdapter adapter = new ArrayAdapter<>(
                context: this,
                android.R.layout.simple_list_item_1 ,
                 countries
        );
        spinCountry.setAdapter(adapter);
    }
}
```


1:40

Country App

Select your country

Nepal

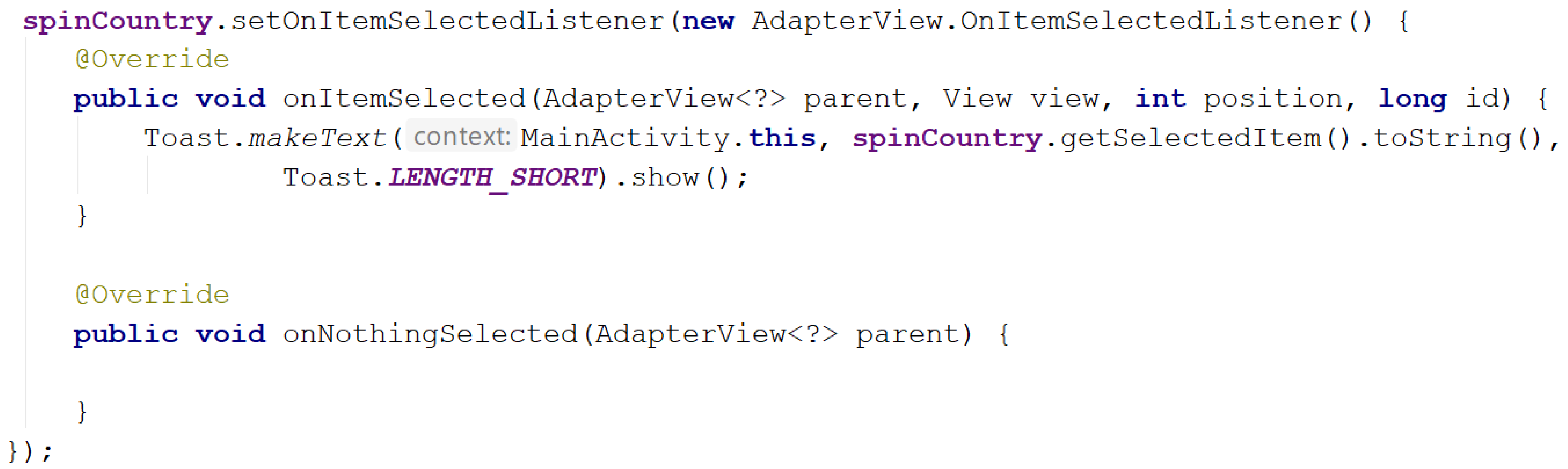
India

China

Uk

US

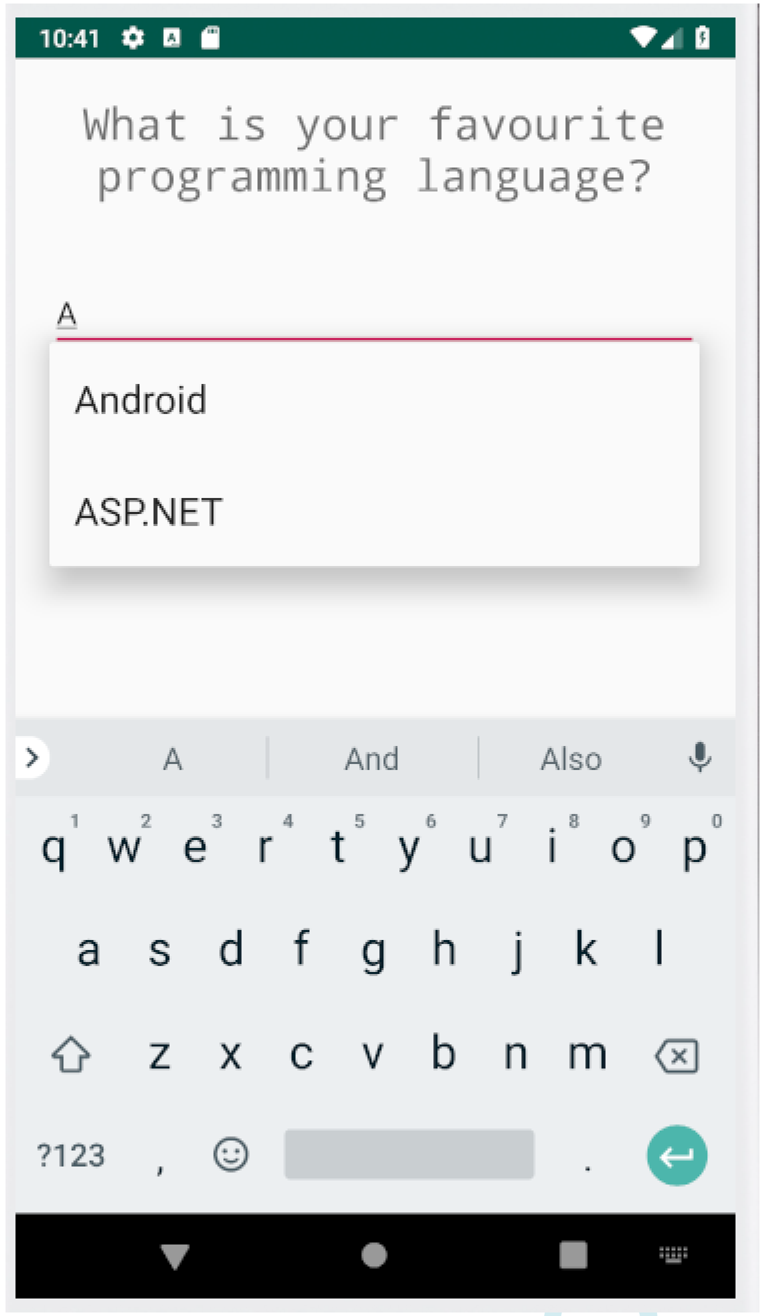
## Spinner setOnItemSelectedListener



```
spinCountry.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
   @Override
   public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
       Toast.makeText( context: MainActivity.this, spinCountry.getSelectedItem().toString(),
               Toast.LENGTH_SHORT).show();
   }

   @Override
   public void onNothingSelected(AdapterView<?> parent) {

   }
});
```

# AutoCompleteTextView

- Android AutoCompleteTextView **completes** the word based on the reserved words, so no need to write all the characters of the word.
- Android AutoCompleteTextView is a editable text field, it displays a list of suggestions in a drop down menu from which user can select only one suggestion or value.
- Android AutoCompleteTextView is the subclass of EditText class. The MultiAutoCompleteTextView is the subclass of AutoCompleteTextView class.



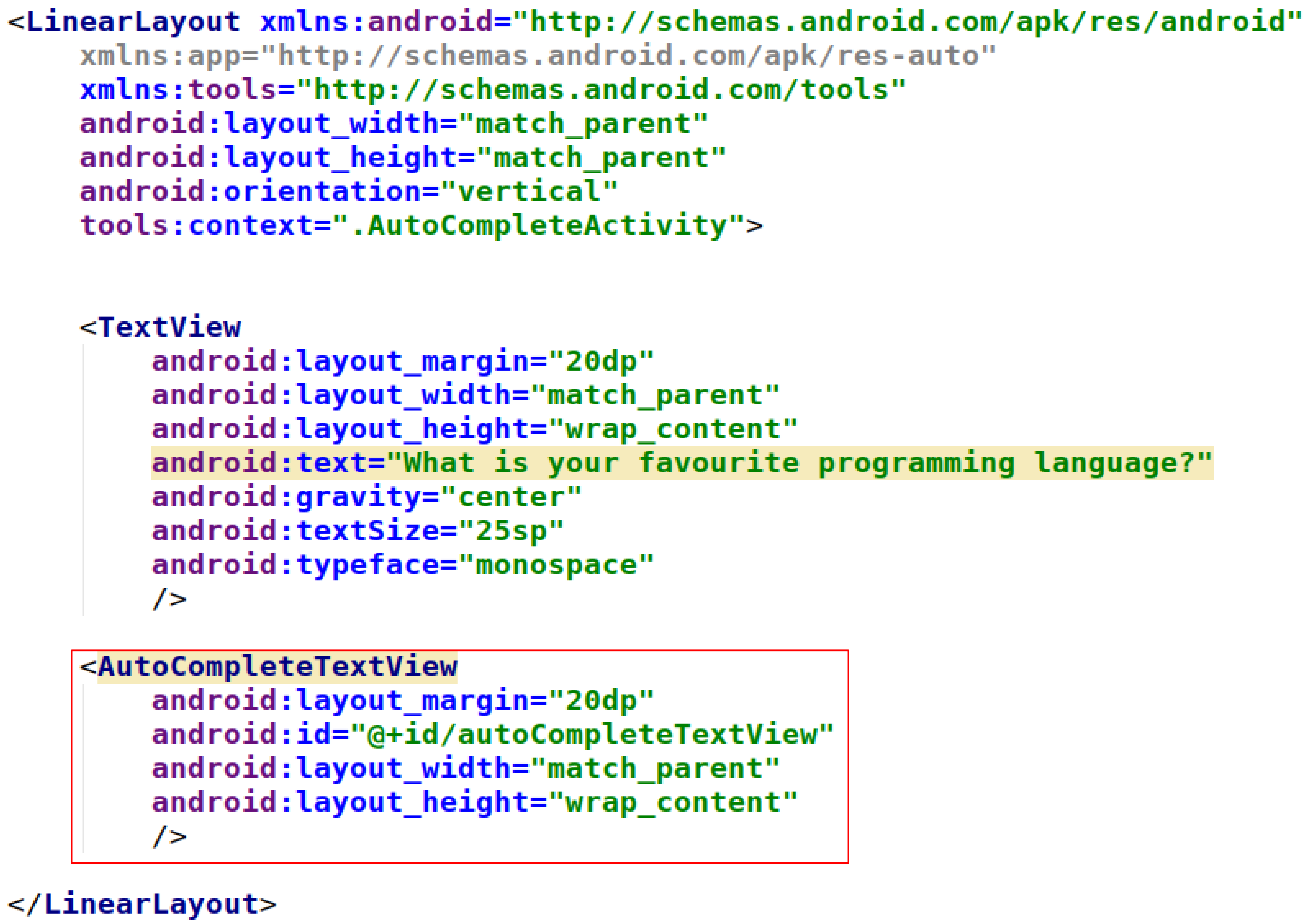
What is your favourite programming language?

A

Android

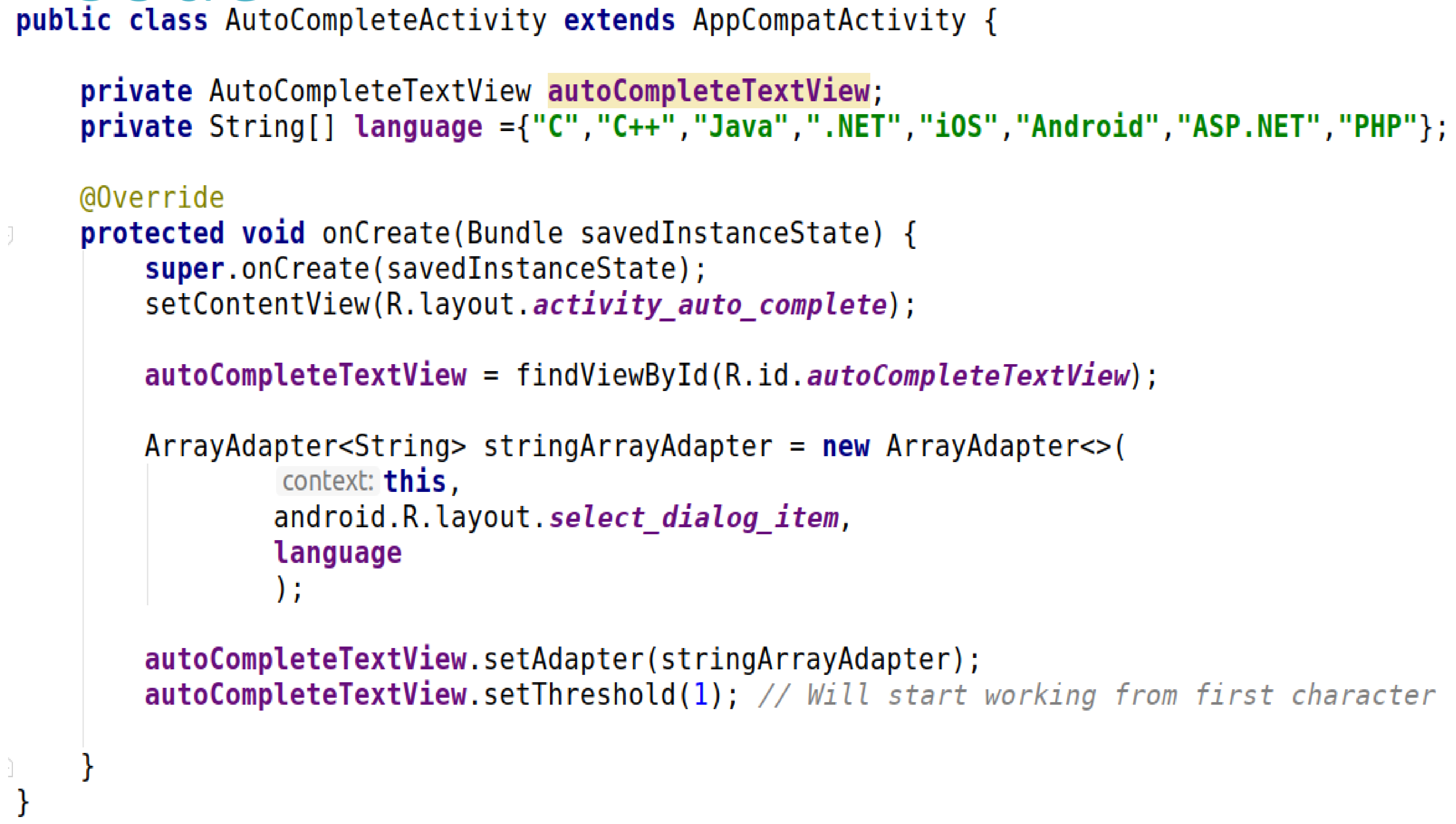
ASP.NET

## Design



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".AutoCompleteActivity">


    <TextView
        android:layout_margin="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="What is your favourite programming language?"
        android:gravity="center"
        android:textSize="25sp"
        android:typeface="monospace"
        />

    <AutoCompleteTextView
        android:layout_margin="20dp"
        android:id="@+id/autoCompleteTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

</LinearLayout>
```

## Code

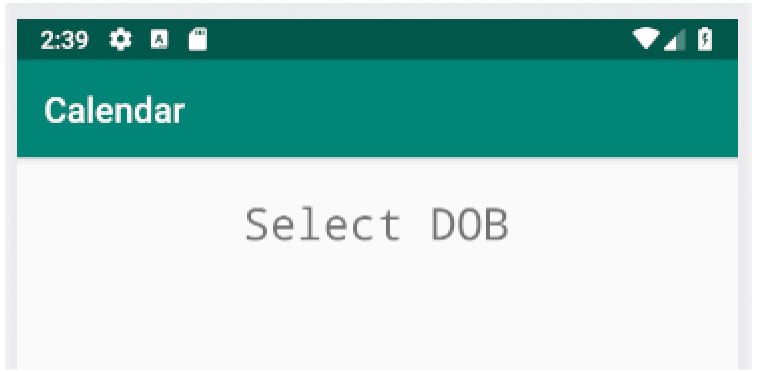


```java
public class AutoCompleteActivity extends AppCompatActivity {

    private AutoCompleteTextView autoCompleteTextView;
    private String[] language ={"C","C++","Java",".NET","iOS","Android","ASP.NET","PHP"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_auto_complete);

        autoCompleteTextView = findViewById(R.id.autoCompleteTextView);

        ArrayAdapter<String> stringArrayAdapter = new ArrayAdapter<>(
                context: this,
                android.R.layout.select_dialog_item,
                language
                );

        autoCompleteTextView.setAdapter(stringArrayAdapter);
        autoCompleteTextView.setThreshold(1); // Will start working from first character

    }
}
```

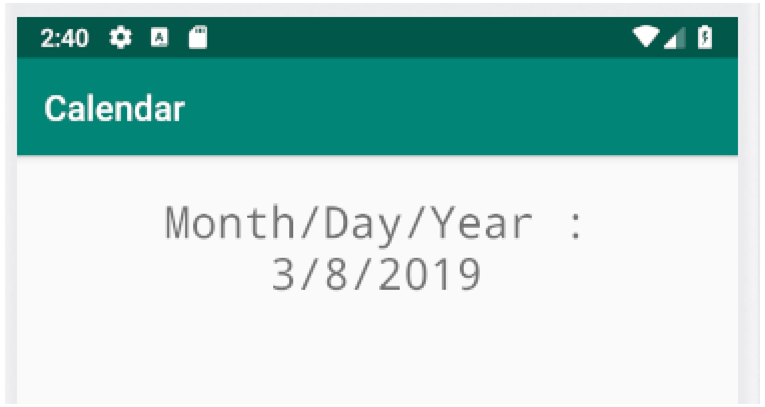
# Android DatePicker

Android **Date Picker** allows you to select the date consisting of day, month and year in your custom user interface.

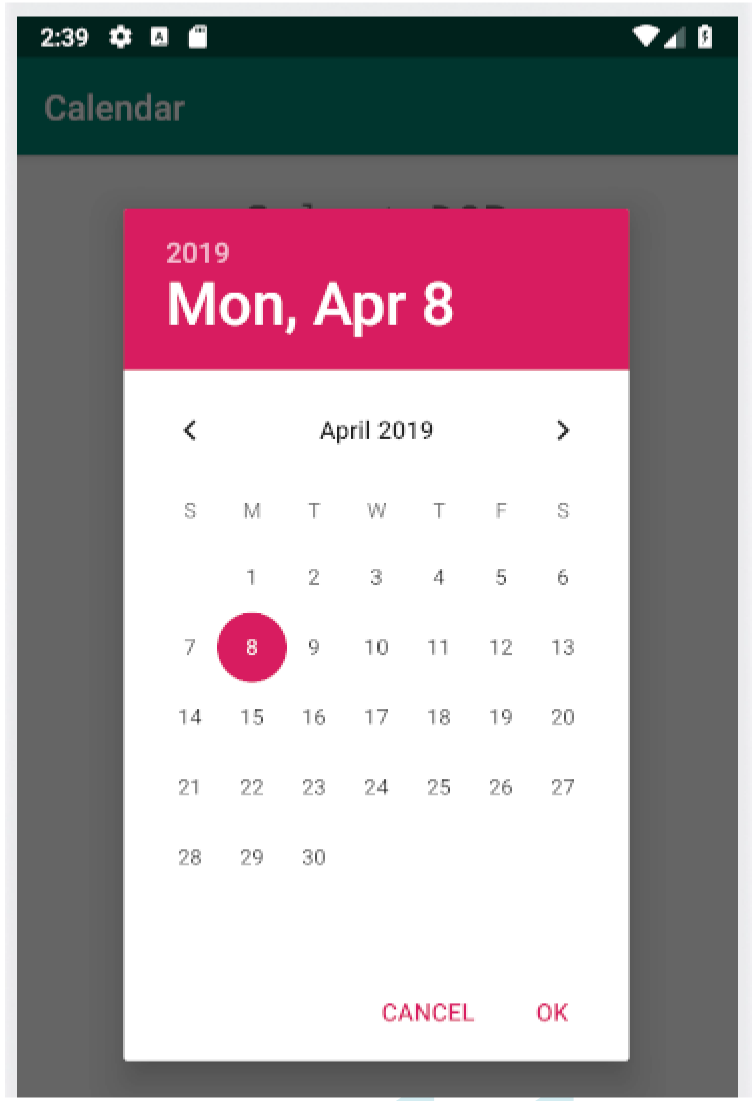
For this functionality android provides **DatePicker** and **DatePickerDialog** components.



Select DOB



Month/Day/Year : 3/8/2019

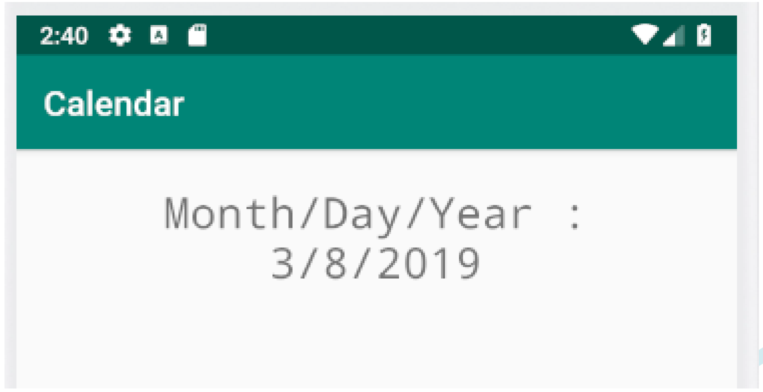


# Android DatePicker

```java
public class DatePickerActivity extends AppCompatActivity implements DatePickerDialog.OnDateSetListener {
private TextView tvDOB;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_date_picker);
    tvDOB = findViewById(R.id.tvDOB);
    tvDOB.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            loadDatePicker();
        }
    });
}

private void loadDatePicker() {
    // Use the current date as the default date in the picker
    final Calendar c = Calendar.getInstance();
    int year = c.get(Calendar.YEAR);
    int month = c.get(Calendar.MONTH);
    int day = c.get(Calendar.DAY_OF_MONTH);

    DatePickerDialog datePickerDialog = new DatePickerDialog(
            context: this, listener: this, year, month, day);
    datePickerDialog.show();
}
```

After selecting a date from date picker

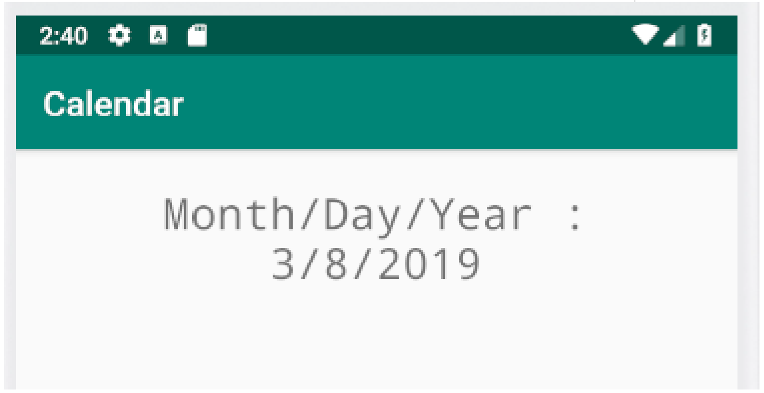
```java
@Override
public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
    String date = "Month/Day/Year : " + month + "/" + dayOfMonth + "/" + year;
    tvDOB.setText(date);
}
```



Month/Day/Year :
3/8/2019

# Another way

```
private void loadCalendar() {
    // Use the current date as the default date in the picker
    final Calendar c = Calendar.getInstance();
    int year = c.get(Calendar.YEAR);
    int month = c.get(Calendar.MONTH);
    int day = c.get(Calendar.DAY_OF_MONTH);
    DatePickerDialog datePickerDialog = new DatePickerDialog( context: this, new DatePickerDialog.OnDateSetListener()
        @Override
        public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
            String date = "Month/Day/Year : " + month + "/" + dayOfMonth + "/" + year;
            tvDOB.setText(date);
        }
    },year,month,day);
    datePickerDialog.show();
}
```

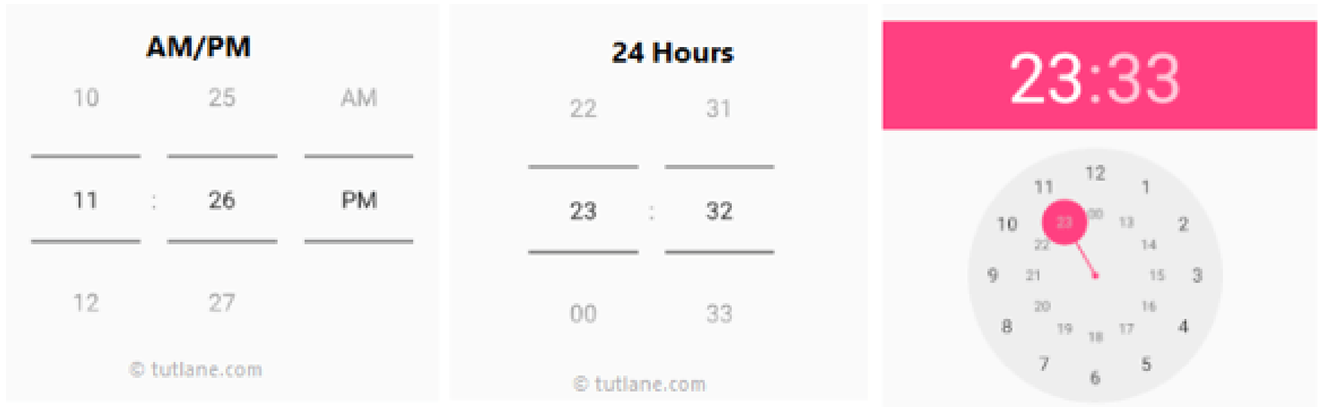


Month/Day/Year : 3/8/2019

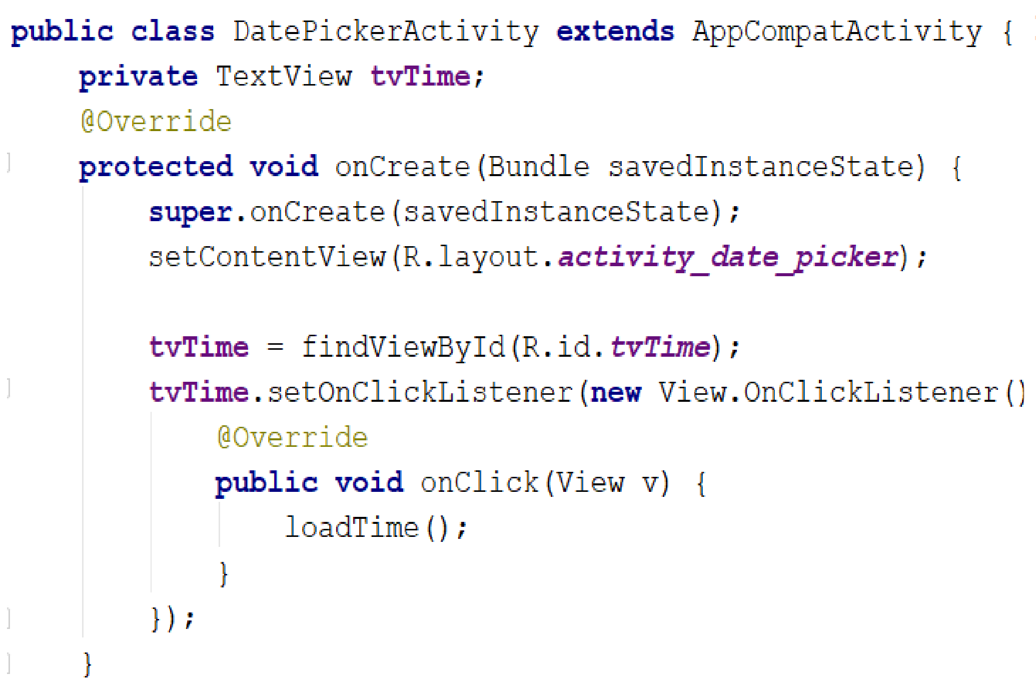
# Android Time Picker

**TimePicker** is a widget for selecting the time of day, in either 24-hour or AM/PM mode.

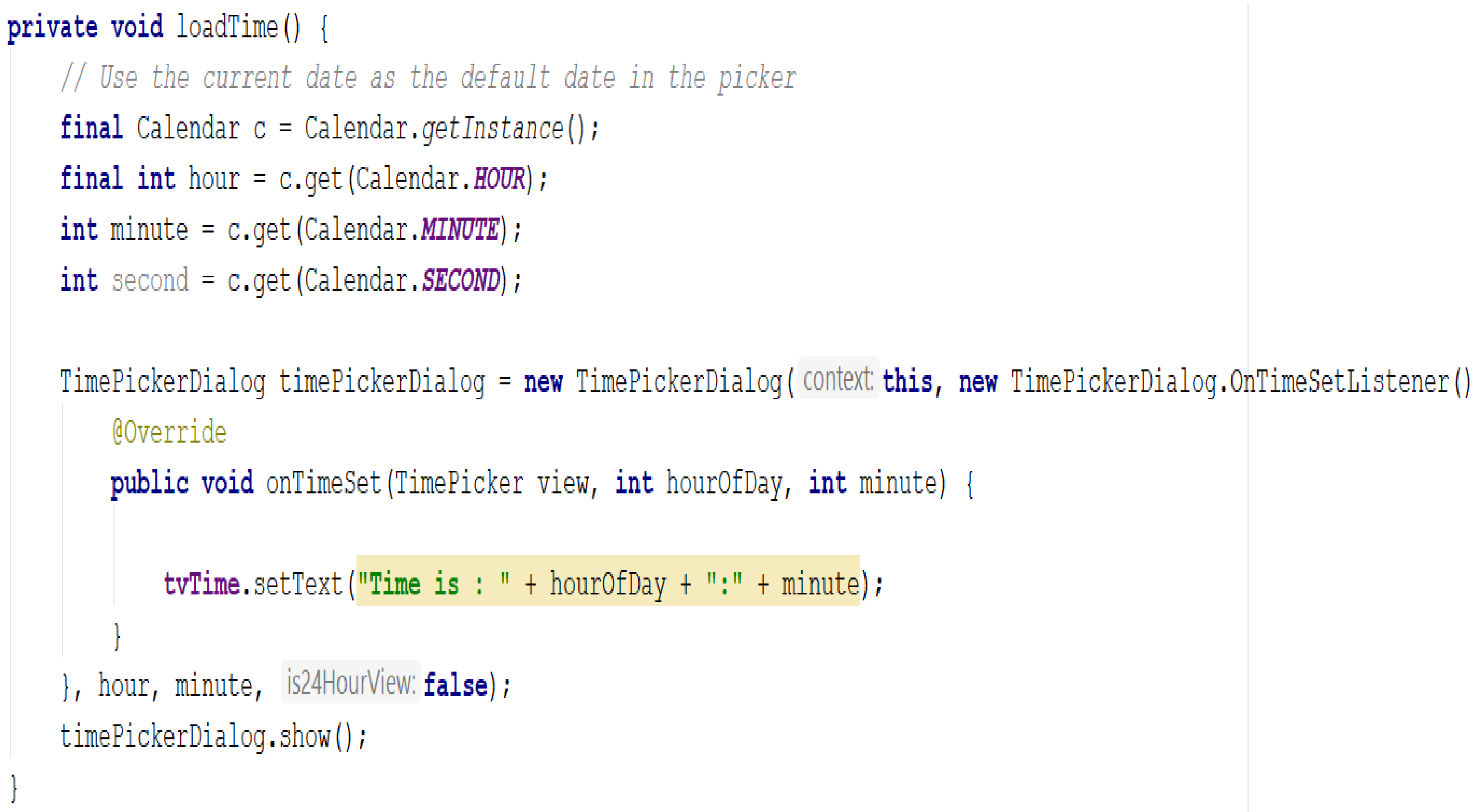
If we use **TimePicker** in our application, it will ensure that the users will select a valid time for the day.



# Android Time Picker

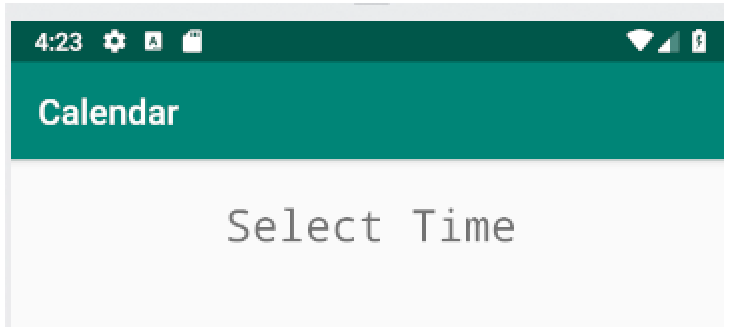


```java
public class DatePickerActivity extends AppCompatActivity {
    private TextView tvTime;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_date_picker);

        tvTime = findViewById(R.id.tvTime);
        tvTime.setOnClickListener(new View.OnClickListener()
            @Override
            public void onClick(View v) {
                loadTime();
            }
        });
    }
```



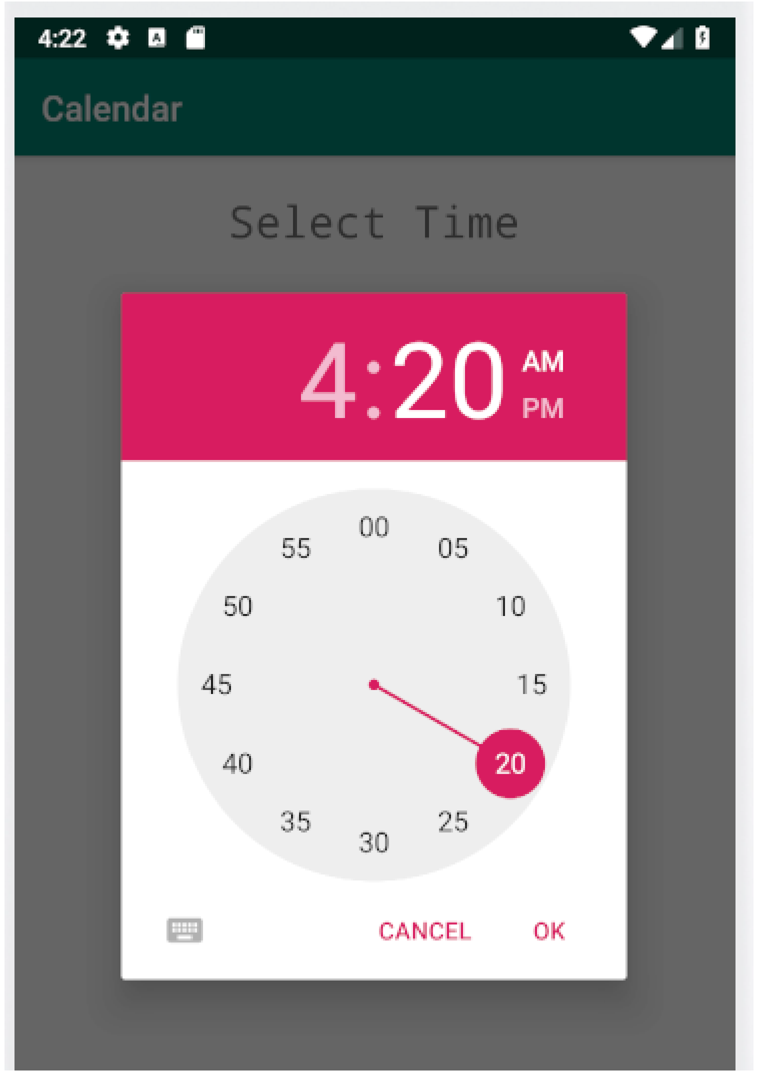
```java
private void loadTime() {
    // Use the current date as the default date in the picker
    final Calendar c = Calendar.getInstance();
    final int hour = c.get(Calendar.HOUR);
    int minute = c.get(Calendar.MINUTE);
    int second = c.get(Calendar.SECOND);

    TimePickerDialog timePickerDialog = new TimePickerDialog( context: this, new TimePickerDialog.OnTimeSetListener()
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

            tvTime.setText("Time is : " + hourOfDay + ":" + minute);
        }
    }, hour, minute,  is24HourView: false);
    timePickerDialog.show();
}
```

# Output


4:23

Calendar

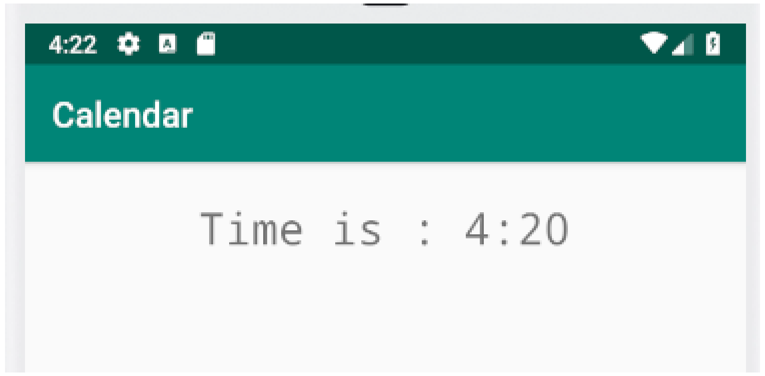
Select Time


4:22

Calendar

Select Time

4:20 AM PM

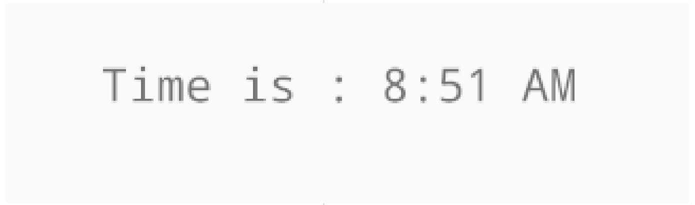

4:22

Calendar

Time is : 4:20

# 12 Hour format

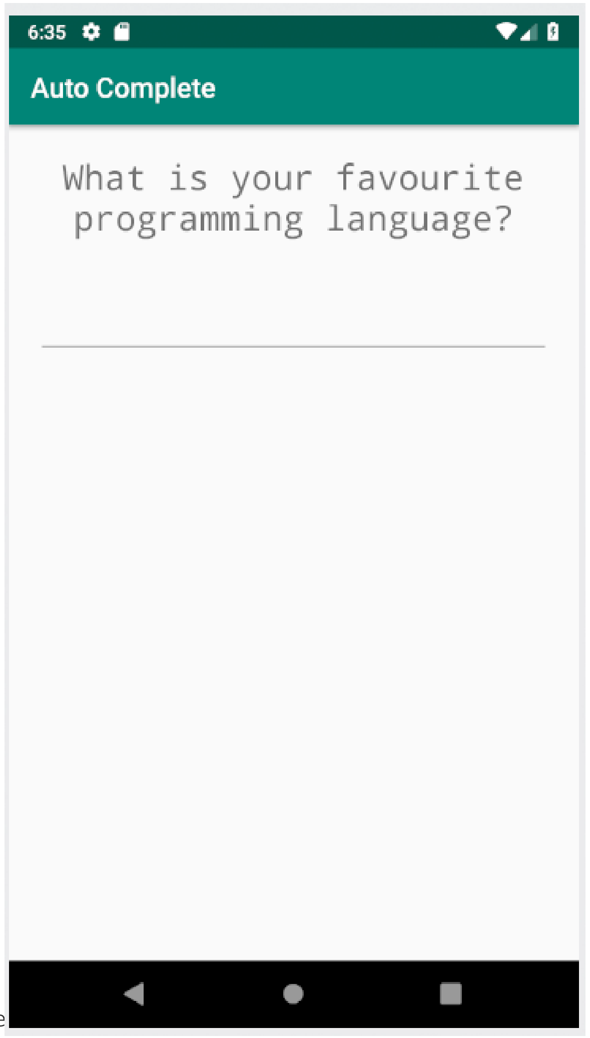
```
TimePickerDialog timePickerDialog = new TimePickerDialog( context: this, new TimePickerDialog.OnTimeSetListener()
    @Override
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
        String amPm;
        if(hourOfDay>=12)
        {
            hourOfDay-=12;
            amPm = "PM";
        }
        else
        {
            amPm="AM";
        }
        tvTime.setText(("Time is : " + hourOfDay + ":" + minute + " " + amPm).toString());
    }
}, hour, minute,  is24HourView: false);
timePickerDialog.show();
```



```
Time is : 8:51 AM
```

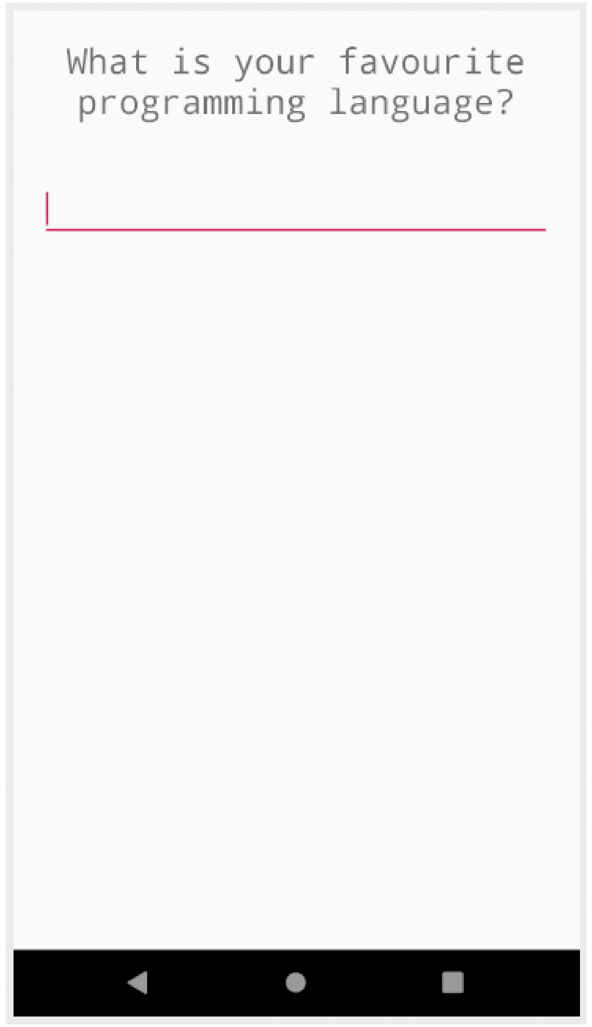
Friday, June 7, 2024

## Hiding Title Bar and Full Screen

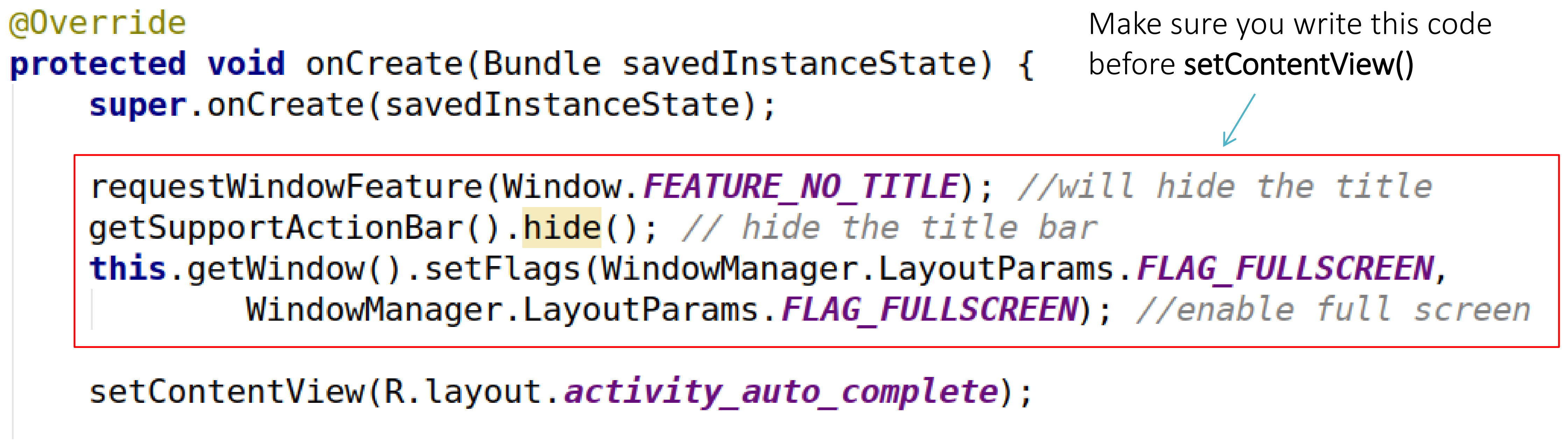

6:35

Auto Complete

What is your favourite programming language?


What is your favourite programming language?

# Code



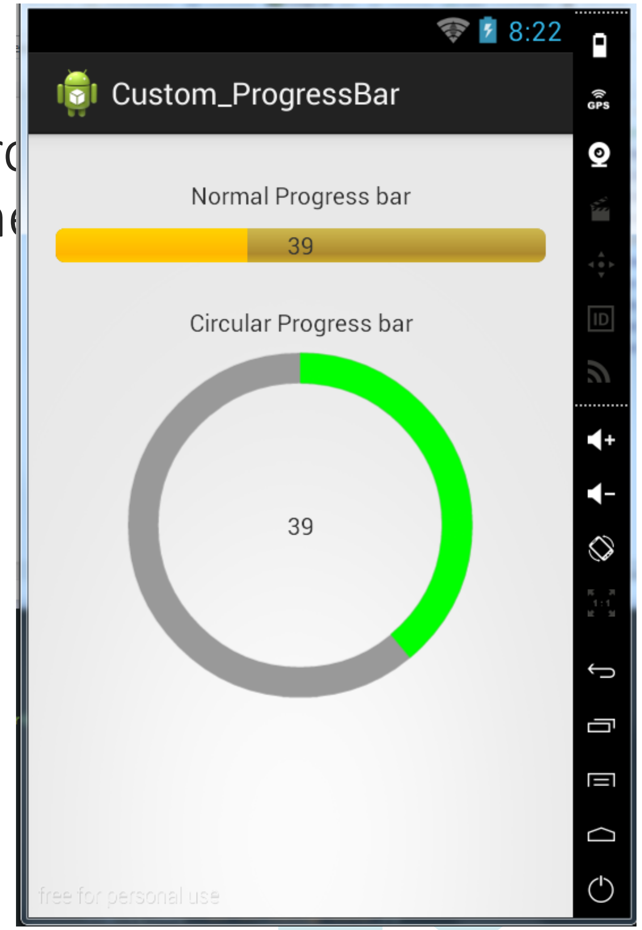
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    requestWindowFeature(Window.FEATURE_NO_TITLE); //will hide the title
    getSupportActionBar().hide(); // hide the title bar
    this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN); //enable full screen

    setContentView(R.layout.activity_auto_complete);
```

Make sure you write this code before **setContentView()**

# Progress Bar

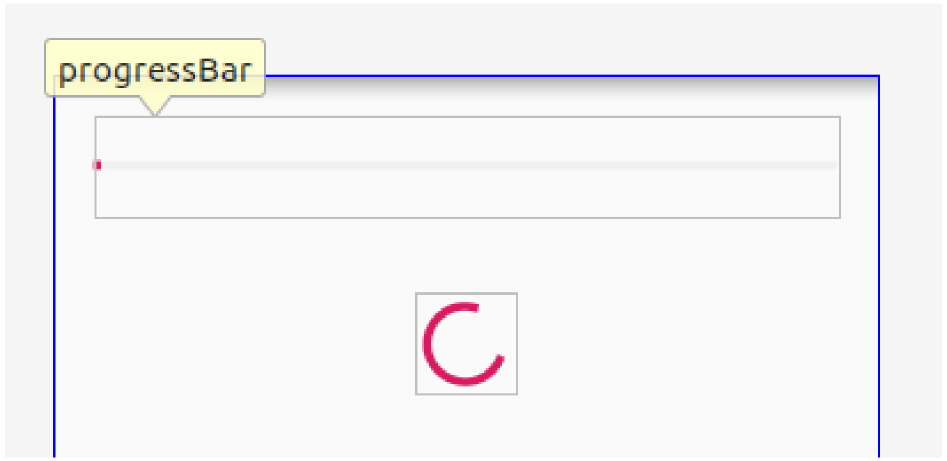
*Progress bars are used to show progress of a task.*

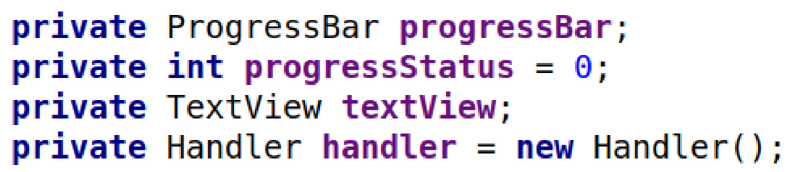
For example, when you are uploading or downloading something fro
internet, it is better to show the progress of download/upload to the



# Progress Bar - Design

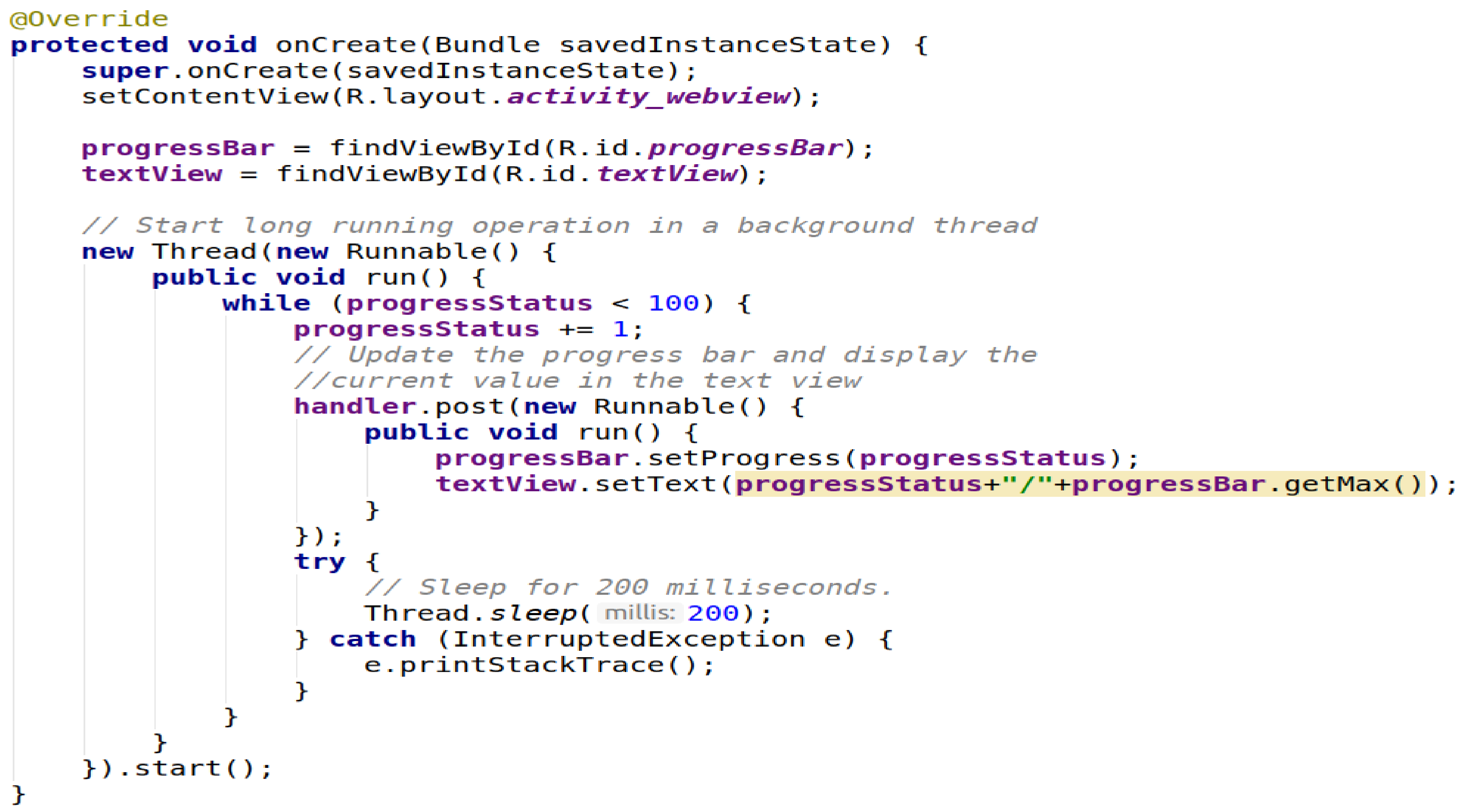
```
<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:indeterminate="false"
    android:max="100"
    android:minHeight="50dp"
    android:minWidth="200dp"
    android:progress="1"
    style="?android:attr/progressBarStyleHorizontal"
    />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
 />

<ProgressBar
    android:id="@+id/progressBar_cyclic"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:minHeight="50dp"
    android:minWidth="50dp"
    android:layout_gravity="center"
   />
```





```java
private ProgressBar progressBar;
private int progressStatus = 0;
private TextView textView;
private Handler handler = new Handler();
```

A **Handler** allows communicating back with UI thread from other background thread



```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_webview);

    progressBar = findViewById(R.id.progressBar);
    textView = findViewById(R.id.textView);

    // Start long running operation in a background thread
    new Thread(new Runnable() {
        public void run() {
            while (progressStatus < 100) {
                progressStatus += 1;
                // Update the progress bar and display the
                //current value in the text view
                handler.post(new Runnable() {
                    public void run() {
                        progressBar.setProgress(progressStatus);
                        textView.setText(progressStatus+"/"+progressBar.getMax());
                    }
                });
                try {
                    // Sleep for 200 milliseconds.
                    Thread.sleep( millis: 200);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }).start();
}
```