Pictures and Menus with Views

part3

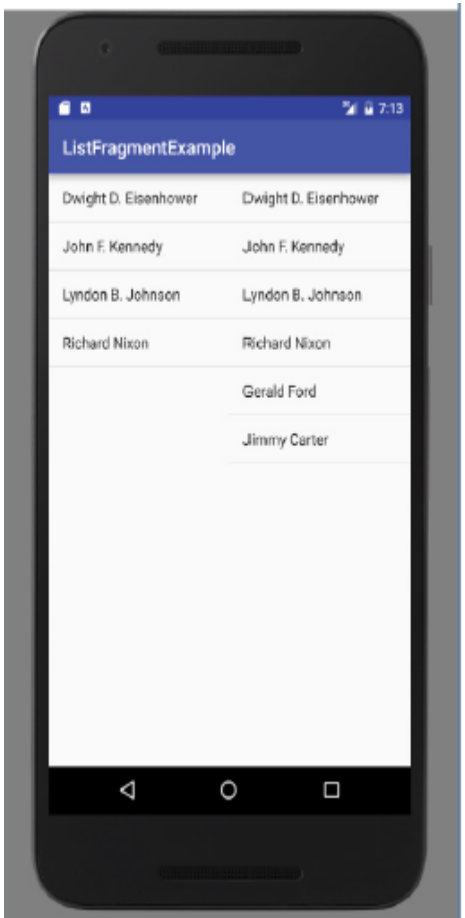
5.6 Understanding Specialized Fragments

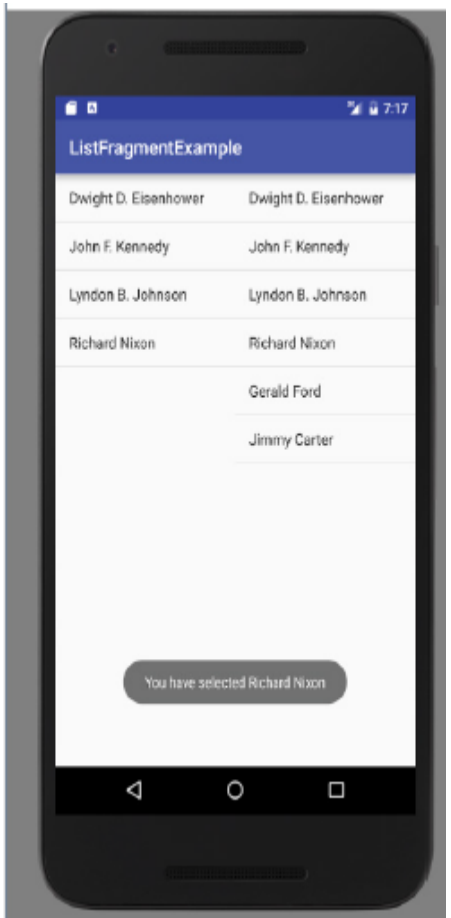
- Fragments allow you to customize the user interface of your Android application by dynamically rearranging fragments to fit within an activity.
- This enables you to build applications that run on devices with different screen sizes.
- As you have learned, fragments are really “mini-activities” that have their own life cycles. To create a fragment, you need a class that extends the Fragment base class.

- subclasses of Fragment:
  - ListFragment
  - DialogFragment

Using a ListFragment

- A list fragment is a fragment that contains a ListView, which displays a list of items from a data source, such as an array





```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >
    <fragment
        android:name="com.jfdimarzio.listfragmentexample.Fragment1"
        android:id="@+id/fragment1"
        android:layout_weight="0.5"
        android:layout_width="0dp"
        android:layout_height="200dp" />
    <fragment
        android:name="com.jfdimarzio.listfragmentexample.Fragment1"
        android:id="@+id/fragment2"
        android:layout_weight="0.5"
        android:layout_width="0dp"
        android:layout_height="300dp" />
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ListView
        android:id="@id/android:list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:drawSelectorOnTop="false"/>
</LinearLayout>
```

```
import android.app.ListFragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
public class Fragment1 extends ListFragment {
    String[] presidents = {
        "Dwight D. Eisenhower",
        "John F. Kennedy",
        "Lyndon B. Johnson",
        "Richard Nixon",
        "Gerald Ford",
        "Jimmy Carter",
        "Ronald Reagan",
        "George H. W. Bush",
        "Bill Clinton",
        "George W. Bush",
        "Barack Obama"
    };
    @Override
    public View onCreateView(LayoutInflater inflater,
    ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment1, container, false);
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setListAdapter(new ArrayAdapter<String>(getActivity(),
            android.R.layout.simple_list_item_1, presidents));
    }

    public void onListItemClick(ListView parent, View v,
    int position, long id)
    {
        Toast.makeText(getActivity(),
            "You have selected " + presidents[position],
            Toast.LENGTH_SHORT).show();
    }
}
```

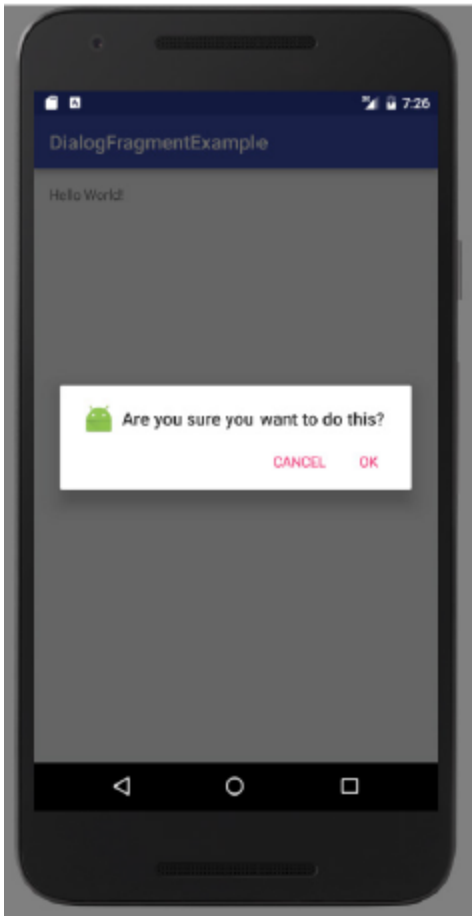
Using a DialogFragment

- A dialog fragment floats on top of an activity and is displayed modally.
- Dialog fragments are useful when you need to obtain the user's response before continuing with execution.
- To create a dialog fragment, you must extend the DialogFragment base class.

```
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.DialogFragment;
import android.content.DialogInterface;
import android.os.Bundle;

public class Fragment1 extends DialogFragment {
    static Fragment1 newInstance(String title) {
        Fragment1 fragment = new Fragment1();
        Bundle args = new Bundle();
        args.putString("title", title);
        fragment.setArguments(args);
        return fragment;
    }
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        String title = getArguments().getString("title");
        return new AlertDialog.Builder(getActivity())
                .setIcon(R.mipmap.ic_launcher)
                .setTitle(title)
                .setPositiveButton("OK",
                        new DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog,
                                                int whichButton) {
                                ((MainActivity)
                                        getActivity()).doPositiveClick();

                            }
                        })
                .setNegativeButton("Cancel",
                        new DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog,
                                                int whichButton) {
                                ((MainActivity)
                                        getActivity()).doNegativeClick();
                            }
                        }).create();
    }
}
```



```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Fragment1 dialogFragment = Fragment1.newInstance(
                "Are you sure you want to do this?");
        dialogFragment.show(getFragmentManager(), "dialog");
    }

    public void doPositiveClick() {
        //---perform steps when user clicks on OK---
        Log.d("DialogFragmentExample", "User clicks on OK");
    }
    public void doNegativeClick() {
        //---perform steps when user clicks on Cancel---
        Log.d("DialogFragmentExample", "User clicks on Cancel");
    }
}
```

In This chapter

- you find out how to work with views that enable you to display images.
- you see how to create option and context menus in your Android application.
- And, see how to create web content

Using Image Views to Display Pictures

- All the views you have seen are used to display text information.
- you can use the ImageView, ImageSwitcher, and GridView views for displaying images

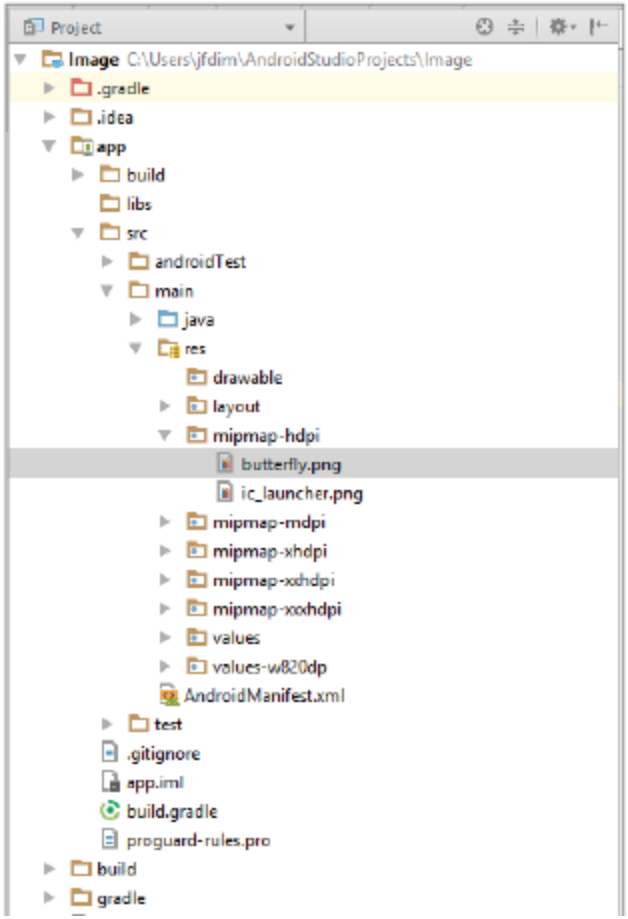
ImageView View

- The ImageView is a view that shows images on the device screen

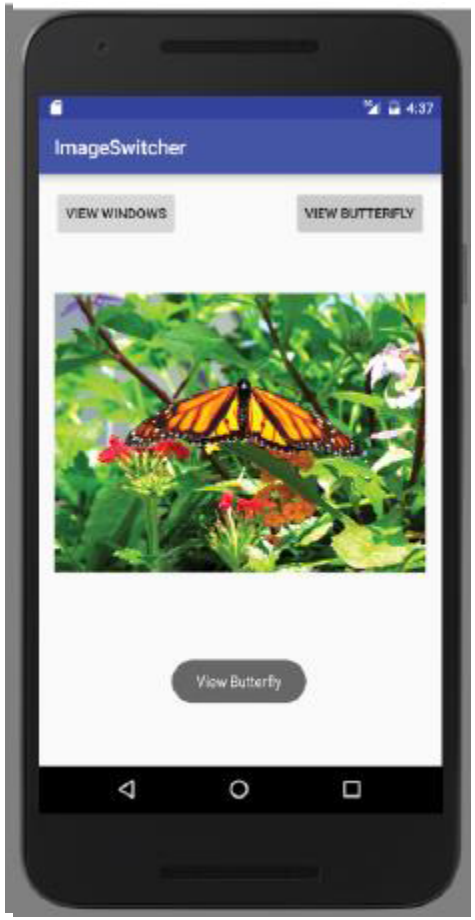
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    tools:context="com.jfdimarzio.image.MainActivity">

    <android.support.v7.widget.AppCompatImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@mipmap/butterfly" />

</LinearLayout>
```



ImageSwitcher



```
<?xml version="1.0" encoding="utf-8"?>
<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="com.jfdimarzio.imageswitcher.MainActivity">
    <Button
        android:text="View Windows"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button2"
        />

    <ImageSwitcher
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/button2"
        android:id="@+id/imageSwitcher">
    </ImageSwitcher>

    <Button
        android:text="View Butterfly"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"


        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        />
</RelativeLayout>
```

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.ImageSwitcher;
import android.widget.ImageView;
import android.widget.Toast;
import android.widget.ViewSwitcher;

public class MainActivity extends AppCompatActivity {
    private ImageSwitcher imgSwitcher;
    private Button btnViewWindows,btnViewButterfly;

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

    imgSwitcher = (ImageSwitcher) findViewById(R.id.imageSwitcher);
    imgSwitcher.setInAnimation(AnimationUtils.loadAnimation(this,
            android.R.anim.fade_in));
    imgSwitcher.setOutAnimation(AnimationUtils.loadAnimation(this,
            android.R.anim.fade_out));

     btnViewWindows = (Button) findViewById(R.id.button2);
     btnViewButterfly = (Button) findViewById(R.id.button);

     imgSwitcher.setFactory(new ViewSwitcher.ViewFactory() {
         @Override
         public View makeView() {
             ImageView myView = new ImageView(getApplicationContext());
             myView.setScaleType(ImageView.ScaleType.FIT_CENTER);
             myView.setLayoutParams(new ImageSwitcher.LayoutParams(
 ActionBar.LayoutParams.WRAP_CONTENT, ActionBar.LayoutParams.WRAP_CONTENT));
             return myView;
         }
     });
     btnViewWindows.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
             Toast.makeText(getApplicationContext()
    , "View Windows",Toast.LENGTH_LONG).show();
                 imgSwitcher.setImageResource(R.mipmap.windows);
             }
         });

         btnViewButterfly.setOnClickListener(new View.OnClickListener() {
             @Override
             public void onClick(View v) {
                 Toast.makeText(getApplicationContext(), "View Butterfly"
    ,Toast.LENGTH_LONG).show();
                 imgSwitcher.setImageResource(R.mipmap.butterfly);
             }
         });
    }
}
```

GridView

- The GridView shows items in a two-dimensional scrolling grid. You can use the GridView together with an ImageView to display a series of images.

FIGURE 6-6

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.jfdimarzio.grid.MainActivity"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="81dp">

        <GridView
        android:layout_width="384dp"
        android:layout_height="511dp"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="0dp"
        app:layout_constraintLeft_toLeftOf="@+id/activity_main"
        tools:layout_constraintLeft_creator="0"
        app:layout_constraintTop_toTopOf="@+id/activity_main"
        tools:layout_constraintTop_creator="0"
        app:layout_constraintRight_toRightOf="@+id/activity_main"
        tools:layout_constraintRight_creator="0"
        app:layout_constraintBottom_toBottomOf="@+id/activity_main"
        tools:layout_constraintBottom_creator="0"
        android:id="@+id/gridview" />

</android.support.constraint.ConstraintLayout>
```

```
import android.content.Context;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    //---the images to display---
    Integer[] imageIDs = {
            R.mipmap.butterfly,
            R.mipmap.windows,
            R.mipmap.ic_launcher
    };

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

        GridView gridView = (GridView) findViewById(R.id.gridview);
        gridView.setAdapter(new ImageAdapter(this));
        gridView.setOnItemClickListener(new AdapterView.OnItemClickListener()
        {
            public void onItemClick(AdapterView parent,
                                    View v, int position, long id)
            {
                Toast.makeText(getBaseContext(),
                        "pic" + (position + 1) + " selected",
                    Toast.LENGTH_SHORT).show();
            }
        }
    });
}

public class ImageAdapter extends BaseAdapter
```

```
{
    private Context context;
    public ImageAdapter(Context c)
    {
        context = c;
    }
    //---returns the number of images---
    public int getCount() {
        return imageIDs.length;
    }
    //---returns the item---
    public Object getItem(int position) {
        return position;
    }
    //---returns the ID of an item---
    public long getItemId(int position) {
        return position;
    }
    //---returns an ImageView view---
    public View getView(int position, View convertView,
                        ViewGroup parent)
    {



            ImageView imageView;
            if (convertView == null) {
                imageView = new ImageView(context);
                imageView.setLayoutParams(new
                        GridView.LayoutParams(150, 150));
                imageView.setScaleType(
                        ImageView.ScaleType.CENTER_CROP);
                imageView.setPadding(5, 5, 5, 5);
            } else {
                imageView = (ImageView) convertView;
            }
            imageView.setImageResource(imageIDs[position]);
            return imageView;
        }
    }
}
```

Using Menus with Views

- Menus are useful for displaying additional options that are not directly visible on the main (UI) of an application.
- There are two main types of menus in Android:
  - **Options menu**—This menu displays information related to the current activity.
  - **Context menu**—This menu displays information related to a particular view on an activity

Creating the Helper Methods

- Before you go ahead and create your options and context menus, you need to create two helper methods.
  - One creates a list of items to show inside a menu,
  - the other handles the event that is fired when the user selects an item inside the menu.

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
private void createMenu(Menu menu) {
    MenuItem mnu1 = menu.add(0, 0, 0, "Item 1");
    {
        mnu1.setAlphabeticShortcut('a');
    }
    MenuItem mnu2 = menu.add(0, 1, 1, "Item 2");
    {
        mnu2.setAlphabeticShortcut('b');
    }
    MenuItem mnu3 = menu.add(0, 2, 2, "Item 3");
    {
        mnu3.setAlphabeticShortcut('c');
    }
    MenuItem mnu4 = menu.add(0, 3, 3, "Item 4");
    {
        mnu4.setAlphabeticShortcut('d');
    }
    menu.add(0, 4, 4, "Item 5");
    menu.add(0, 5, 5, "Item 6");
    menu.add(0, 6, 6, "Item 7");
}
```

```
    private boolean MenuChoice(MenuItem item) {
        switch (item.getItemId()) {
            case 0:
                Toast.makeText(this, "You clicked on Item 1",
                        Toast.LENGTH_LONG).show();
                return true;
            case 1:
                Toast.makeText(this, "You clicked on Item 2",
                        Toast.LENGTH_LONG).show();
                return true;
            case 2:
                Toast.makeText(this, "You clicked on Item 3",
                        Toast.LENGTH_LONG).show();
                return true;
            case 3:
                Toast.makeText(this, "You clicked on Item 4",
                        Toast.LENGTH_LONG).show();
                return true;
            case 4:
                Toast.makeText(this, "You clicked on Item 5",
                        Toast.LENGTH_LONG).show();
                return true;
            case 5:
                Toast.makeText(this, "You clicked on Item 6",
                        Toast.LENGTH_LONG).show();
                return true;
            case 6:
                Toast.makeText(this, "You clicked on Item 7",
                        Toast.LENGTH_LONG).show();
                return true;
        }
        return false;
    }
}
```

Options Menu

FIGURE 6-7

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        createMenu(menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        return menuChoice(item);
    }

  private void createMenu(Menu menu) {
      MenuItem mnu1 = menu.add(0, 0, 0, "Item 1");
      {
          mnu1.setAlphabeticShortcut('a');
      }
      MenuItem mnu2 = menu.add(0, 1, 1, "Item 2");
      {
          mnu2.setAlphabeticShortcut('b');
      }
      MenuItem mnu3 = menu.add(0, 2, 2, "Item 3");
      {
          mnu3.setAlphabeticShortcut('c');
      }
      MenuItem mnu4 = menu.add(0, 3, 3, "Item 4");
      {
          mnu4.setAlphabeticShortcut('d');
      }
      menu.add(0, 4, 4, "Item 5");
      menu.add(0, 5, 5, "Item 6");
      menu.add(0, 6, 6, "Item 7");
  }

  private boolean menuChoice(MenuItem item) {
      switch (item.getItemId()) {
          case 0:
              Toast.makeText(this, "You clicked on Item 1",
                      Toast.LENGTH_LONG).show();
```

```
            return true;
        case 1:
            Toast.makeText(this, "You clicked on Item 2",
                    Toast.LENGTH_LONG).show();
            return true;
        case 2:
            Toast.makeText(this, "You clicked on Item 3",
                    Toast.LENGTH_LONG).show();
            return true;
        case 3:
            Toast.makeText(this, "You clicked on Item 4",
                    Toast.LENGTH_LONG).show();
            return true;
        case 4:
            Toast.makeText(this, "You clicked on Item 5",
                    Toast.LENGTH_LONG).show();
            return true;
        case 5:
            Toast.makeText(this, "You clicked on Item 6",
                    Toast.LENGTH_LONG).show();
            return true;
        case 6:
            Toast.makeText(this, "You clicked on Item 7",
                    Toast.LENGTH_LONG).show();
            return true;
    }
    return false;
  }
}
```

Context Menu

- A context menu is usually associated with a view on an activity.
- A context menu is displayed when the user taps and holds an item.
- For example, if the user taps a Button view and holds it for a few seconds, a context menu can be displayed.
- If you want to associate a context menu with a view on an activity, you need to call the setOnCreateContextMenuListener() method of that particular view.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.jfdimarzio.menus.MainActivity"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="81dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        tools:layout_editor_absoluteX="154dp"
        tools:layout_editor_absoluteY="247dp"
        app:layout_constraintLeft_toLeftOf="@+id/activity_main"
        tools:layout_constraintLeft_creator="0"
        app:layout_constraintTop_toTopOf="@+id/activity_main"
        tools:layout_constraintTop_creator="0"
        app:layout_constraintRight_toRightOf="@+id/activity_main"
        tools:layout_constraintRight_creator="0"
        app:layout_constraintBottom_toBottomOf="@+id/activity_main"
        tools:layout_constraintBottom_creator="0" />

    <Button
        android:text="Button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:layout_editor_absoluteX="148dp"
        tools:layout_editor_absoluteY="102dp"
        android:id="@+id/button"
        app:layout_constraintLeft_toLeftOf="@+id/activity_main"
        tools:layout_constraintLeft_creator="0"
        app:layout_constraintRight_toRightOf="@+id/activity_main"
        tools:layout_constraintRight_creator="0" />

</android.support.constraint.ConstraintLayout>
```

```
import android.os.Bundle;
import android.view.ContextMenu;
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 {

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

        Button btn = (Button) findViewById(R.id.button);
        btn.setOnCreateContextMenuListener(this);
    }

    @Override
    public void onCreateContextMenu(ContextMenu menu, View view,
                                    ContextMenu.ContextMenuInfo menuInfo)
    {
        super.onCreateContextMenu(menu, view, menuInfo);
        createMenu(menu);
    }
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    createMenu(menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    return menuChoice(item);
}


private void createMenu(Menu menu) {
    MenuItem mnu1 = menu.add(0, 0, 0, "Item 1");
    {
        mnu1.setAlphabeticShortcut('a');
    }
    MenuItem mnu2 = menu.add(0, 1, 1, "Item 2");
    {
        mnu2.setAlphabeticShortcut('b');
```

```
    }
    MenuItem mnu3 = menu.add(0, 2, 2, "Item 3");
    {
        mnu3.setAlphabeticShortcut('c');
    }
    MenuItem mnu4 = menu.add(0, 3, 3, "Item 4");
    {
        mnu4.setAlphabeticShortcut('d');
    }
    menu.add(0, 4, 4, "Item 5");
    menu.add(0, 5, 5, "Item 6");
    menu.add(0, 6, 6, "Item 7");
}
    private boolean menuChoice(MenuItem item) {
        switch (item.getItemId()) {
            case 0:
                Toast.makeText(this, "You clicked on Item 1",
                        Toast.LENGTH_LONG).show();
                return true;
            case 1:
                Toast.makeText(this, "You clicked on Item 2",
                        Toast.LENGTH_LONG).show();
                return true;
            case 2:
                Toast.makeText(this, "You clicked on Item 3",
                        Toast.LENGTH_LONG).show();
                return true;
            case 3:
                Toast.makeText(this, "You clicked on Item 4",
                        Toast.LENGTH_LONG).show();
                return true;
            case 4:
                Toast.makeText(this, "You clicked on Item 5",
                        Toast.LENGTH_LONG).show();
                return true;
            case 5:
                Toast.makeText(this, "You clicked on Item 6",
                        Toast.LENGTH_LONG).show();
                return true;
            case 6:
                Toast.makeText(this, "You clicked on Item 7",
                        Toast.LENGTH_LONG).show();
                return true;
        }
        return false;
    }
}
```

Using WebView

- Aside from the standard views that you have seen up to this point, the Android SDK provides some additional views that make your applications much more interesting. This section explains more about the WebView.

WebView

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.jfdimarzio.webview.MainActivity"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="81dp">

    <WebView
        android:layout_width="384dp"
        android:layout_height="511dp"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="0dp"
        app:layout_constraintLeft_toLeftOf="@+id/activity_main"
        tools:layout_constraintLeft_creator="0"
        app:layout_constraintTop_toTopOf="@+id/activity_main"
        tools:layout_constraintTop_creator="0"

      app:layout_constraintRight_toRightOf="@+id/activity_main"
      tools:layout_constraintRight_creator="0"
      app:layout_constraintBottom_toBottomOf="@+id/activity_main"
      tools:layout_constraintBottom_creator="0"
      android:id="@+id/webview" />

</android.support.constraint.ConstraintLayout>
```

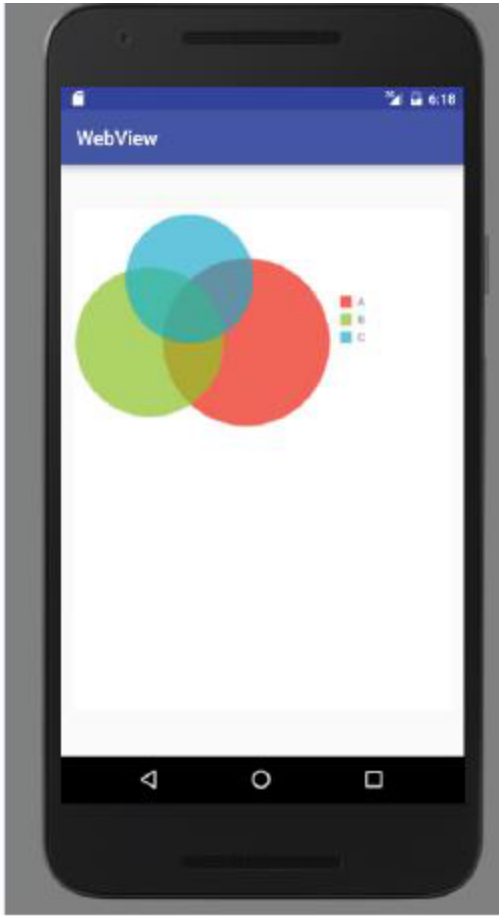


FIGURE 6-9

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;

public class MainActivity extends AppCompatActivity {

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

        WebView wv = (WebView) findViewById(R.id.webview);

        WebSettings webSettings = wv.getSettings();
        webSettings.setBuiltInZoomControls(true);
        wv.loadUrl(
                "http://chart.apis.google.com/chart" +
                        "?chs=300x225" +
                        "&cht=v" +
                        "&chco=FF6342,ADDE63,63C6DE" +
                        "&chd=t:100,80,60,30,30,30,10" +
                        "&chdl=A|B|C");
    }
}
```

- In the app/manifests/AndroidManifest.xml file, add the following permission (shown in bold):

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.jfdimarzio.webview">
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                  <category android:name="android.intent.category.LAUNCHER" />
              </intent-filter>
          </activity>
      </application>

  </manifest>
```