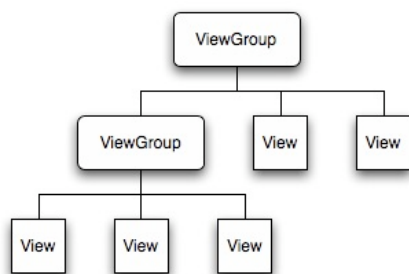


USER INTERFACES

69

69

View Hierarchy



- **View groups** define hierarchies of **views**.
- The layout on the screen is described in an XML document (elements are views and view groups).
- **Input controls** are views that support interaction e.g. buttons, text entry, etc.

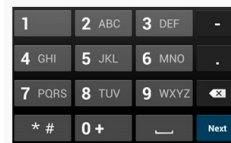
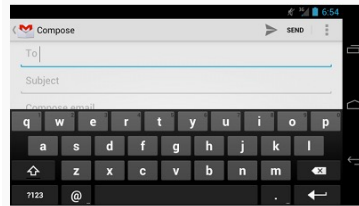
70

70

Views

- TextView
- **EditText**
- Spinner
- Button
- CheckBox
- RadioButton
- Etc...

```
<EditText
    android:id="@+id/email_address"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="@string/email_hint"
    android:inputType="textEmailAddress" />
```



71

71

View Groups

- LinearLayout
- AbsoluteLayout
- TableLayout
- RelativeLayout
- FrameLayout
- WebView
- ScrollView
- ListView
- GridView

Linear Layout



Relative Layout



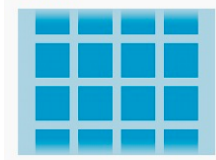
Web View



List View



Grid View



72

72

Attributes

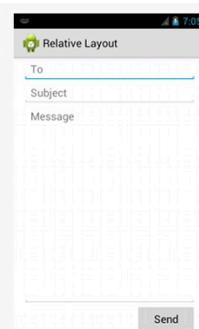
- layout_width, layout_height
 - match_parent
 - wrap_content
- layout_marginTop, layout_marginBottom
- layout_marginLeft, layout_marginRight
- layout_gravity
- layout_weight
- layout_x
- layout_y

For LinearLayout
or TableLayout

73

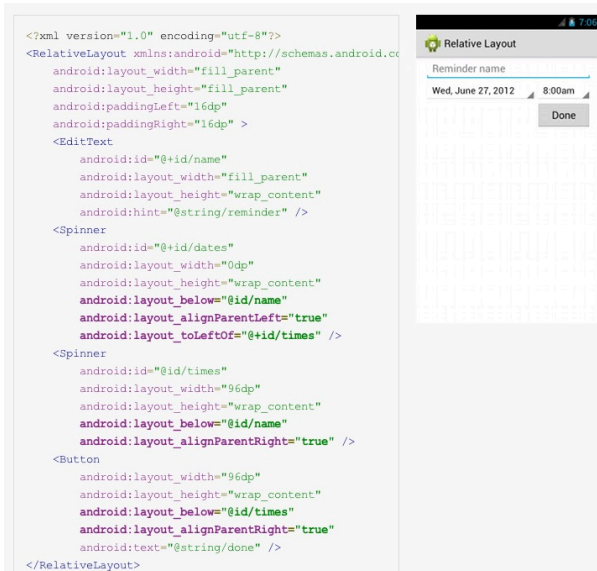
Linear Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```



74

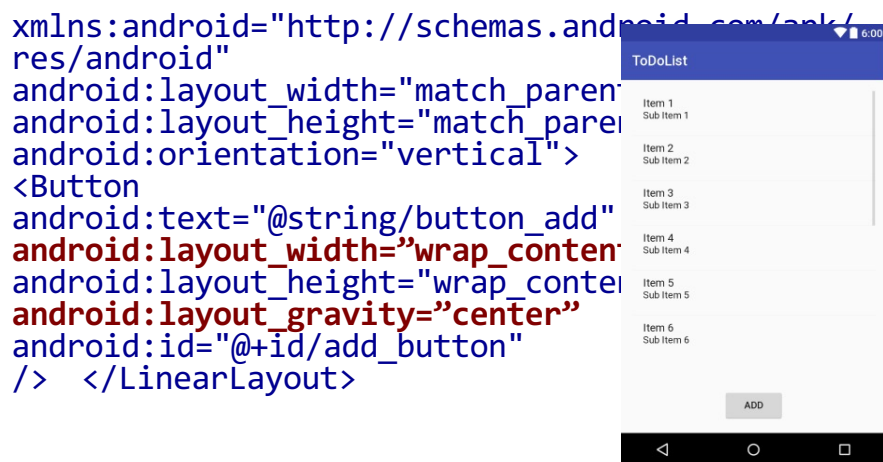
Relative Layout



75

Layout Parameters

<LinearLayout

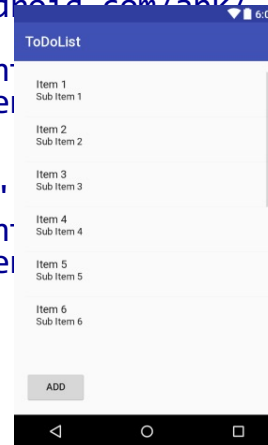


76

Layout Parameters

<LinearLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
  <Button
    android:text="@string/button_add"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:id="@+id/add_button"
  /> </LinearLayout>
```

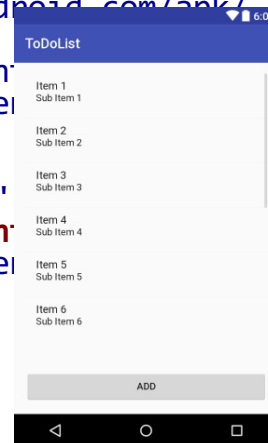


77

Layout Parameters

<LinearLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
  <Button
    android:text="@string/button_add"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:id="@+id/add_button"
  /> </LinearLayout>
```

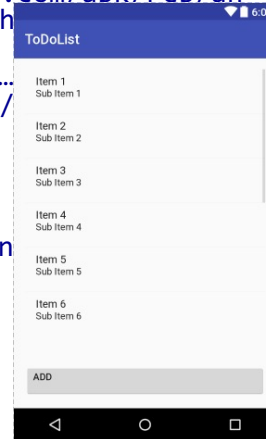


78

Layout Parameters

<LinearLayout

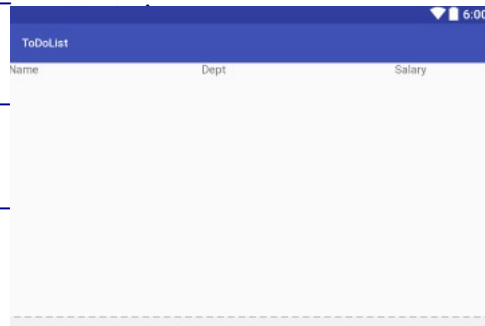
```
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
...
<Button
    android:text="@string/..."
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:gravity="left"
    android:id="@+id/add_button" /> </Lin
```



79

Layout Weight

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:text="Name"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:id="@+id/name" />
        android:layout_weight="0.4"
    <TextView
        android:text="Dept"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:id="@+id/dept" />
        android:layout_weight="0.4"
    <TextView
        android:text="Salary"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:id="@+id/salary" />
    </LinearLayout>
```



80

Event Listener API

- `onClick()`
 - from `View.OnClickListener`
- `onLongClick()`
 - from `View.OnLongClickListener`
- `onFocusChange()`
 - from `View.onFocusChange`
- `onKey()`
 - from `View.OnKeyListener`
 - *Hardware key*
- `onTouch()`
 - from `View.OnTouchListener`
- `onCreateContextMenu()`
 - from `View.OnCreateContextMenuListener`

81

Example: Button

```
private OnClickListener btnListener =
    new OnClickListener() {
        public void onClick(View v) {
            // do something when the button is clicked
        }
    };

protected void onCreate(Bundle savedInstanceState) {
    ...
    // Capture button from layout
    Button okButton = (Button)findViewById(R.id.ok_button);
    // Register the onClick listener
    okButton.setOnClickListener(btnListener);
    ...
}
```

82

Example: Button

```
public class ExampleActivity extends Activity
    implements OnClickListener {
    protected void onCreate(Bundle savedInstanceState) {
        ...
        Button okButton =
            (Button)findViewById(R.id.ok_button);
        okButton.setOnClickListener(this);
    }

    // Implement the OnClickListener callback
    public void onClick(View v) {
        // do something when the button is clicked
    }
    ...
}
```

83

Example: Edit Text

```
final EditText myEditText =
    (EditText)findViewById(R.id.myEditText);

// Assign a listener to the button
OnClickListener btnListener = new OnClickListener() {
    public void onClick(View v) {
        ... myEditText.getText().toString() ...
        myEditText.setText("");
    }
}

okButton.setOnClickListener(btnListener);
```

84

Example: Edit Text

```
myEditText = (EditText)findViewById(R.id.myEditText);

// Process input when textbox loses input focus
OnFocusChangeListener textListener = new OnFocusChangeListener() {
    private boolean hadFocus = false;
    public void onFocusChange(View v, boolean hasFocus) {
        if (hasFocus) {
            hadFocus = true;
        } else {
            ... myEditText.getText().toString() ...
            myEditText.setText("");
            hadFocus = false;
        }
    }
}

myEditText.setOnFocusChangeListener(textListener);
```

In button click listener:
button.requestFocusFromTouch()

85

Event Handlers

- Defined for View API
- Modify for customized view
- onKeyDown (int, KeyEvent)
- onKeyUp (int, KeyEvent)
- onTrackballEvent (MotionEvent)
- onTouchEvent (MotionEvent)
- onFocusChanged (boolean, int, Rect)

86

Creating New Views

- Modify an existing view
 - Override event handlers e.g. `onDraw()`, `onMeasure()`, ...
- Compound Control: Combine controls
 - Example: dropdown box combining `TextView` and `Button`
 - Best approach: Extend `Layout` class
- Custom View: Create an entirely new control
- Custom key press handling: e.g. for games

87

87