

Google Android

Các thành phần giao diện



View



ViewGroup



Button



ImageButton



ImageView



ListView



TextView



EditText



CheckBox



MenuOptions



ContextMenu



Quick Search Box

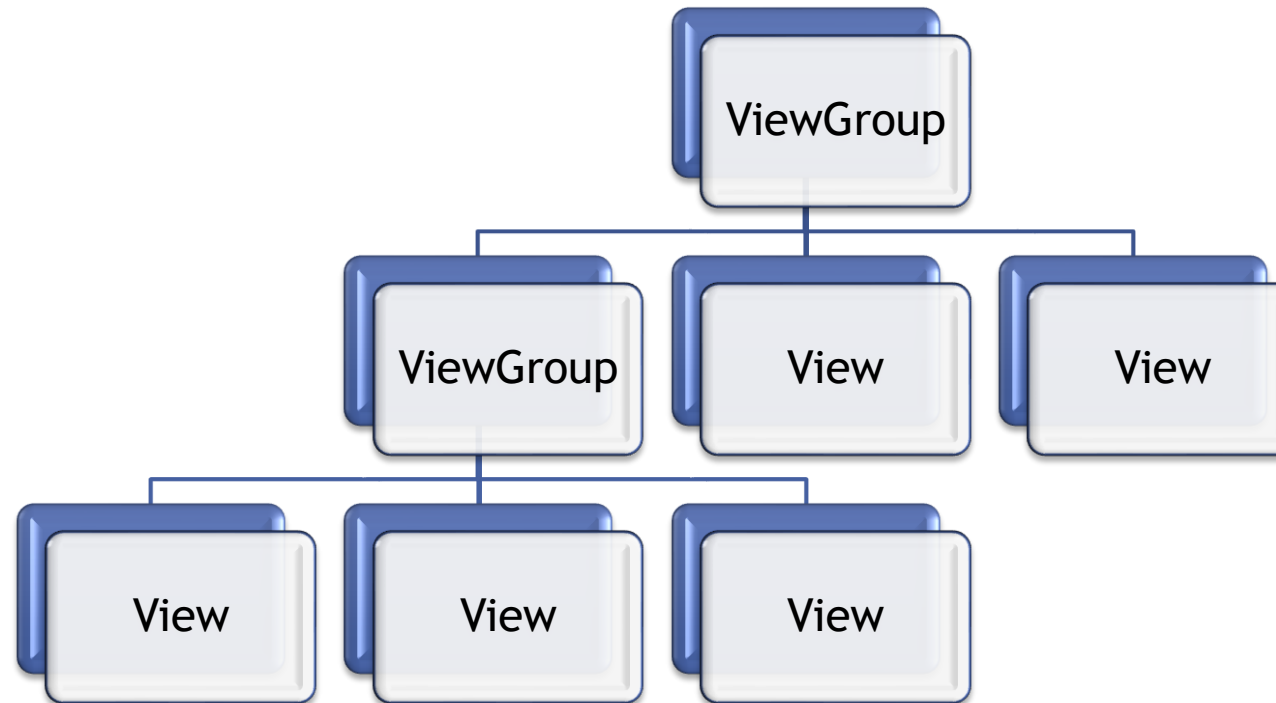


Activity & Intent

View

- ✔ Là đối tượng xây dựng giao diện người dùng.
- ✔ Có nhiều kiểu View và ViewGroup, và đều gọi là Widget.
- ✔ Mọi Widget đều có chung các thuộc tính cơ bản: cách trình bày vị trí, background, kích thước, lề,...
- ✔ Trong Android Platform, các screen luôn được bố trí theo một kiểu cấu trúc phân cấp.
- ✔ Một screen là một tập hợp các Layout và Widget được bố trí có thứ tự.
- ✔ Hàm thể hiện một screen:
`setContentView(R.layout.main)`

Thành phần giao diện



Cấu trúc phân cấp một giao diện ứng dụng Android

ViewGroup

- ✔ Là các Widget Layout được dùng để bố trí các đối tượng khác trong một screen.
- ✔ Có một số loại ViewGroup như sau:
 - ✔ LinearLayout
 - ✔ FrameLayout
 - ✔ AbsoluteLayout
 - ✔ RelativeLayout
 - ✔ TableLayout

Cách thiết kế 1 Layout



1

Trong file XML layout
(Thông thường và
cho giao diện ban
đầu)
- Cần hiểu biết sơ
lược về XML

2

Trong đoạn mã
chương trình
- Tránh phức tạp
trong các giao diện

Các thuộc tính cơ bản

- ▶ *Layout_width*: Chiều rộng của View.
- ▶ *Layout_height*: Chiều cao của View.
- ▶ *Layout_marginTop*: Khoảng cách với biên trên của View.
- ▶ *Layout_marginBottom*: Khoảng cách với biên dưới cùng của View.
- ▶ *Layout_left*: Khoảng cách với biên trái của View.
- ▶ *Layout_right*: Khoảng cách với biên phải của View.
- ▶ *Layout_gravity*: Xác định vị trí của View.
- ▶ *Layout_weight*: Phân chia tỉ lệ hiển thị diện tích trên màn hình (tỉ lệ tính theo weight của từng view trên tổng số weight, các view không khai báo weight thì sẽ xem qua width và height).
- ▶ *Layout_x*: Tọa độ x của View.
- ▶ *Layout_y*: Tọa độ y của View.

Layout

- Layout được sử dụng nhằm mục đích thiết kế giao diện cho nhiều độ phân giải. Thường khi lập trình nên kết hợp nhiều layout với nhau để tạo ra giao diện bạn mong muốn.



FrameLayout - Thêm phần con vào góc bên trái màn hình



LinearLayout - Thêm phần con theo 1 chiều nhất định (ngang hoặc dọc)



RelativeLayout - Thêm phần con dựa trên quan hệ với các tp khác hoặc biên layout



TableLayout - Thêm các thành phần con dựa trên 1 lưới các ô ngang và dọc

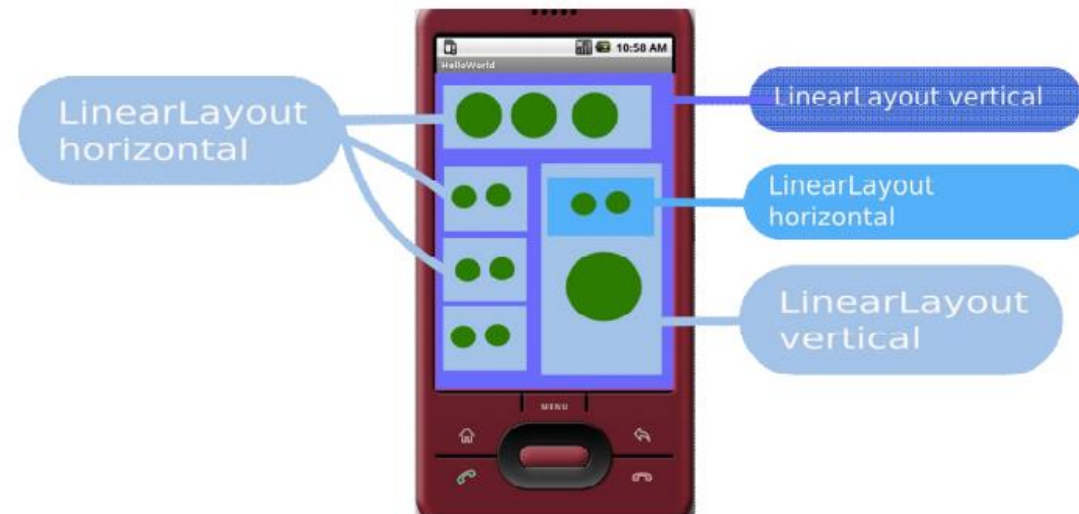


AbsoluteLayout - Thêm các thành phần con dựa theo tọa độ x, y.

Layout

✔ LinearLayout

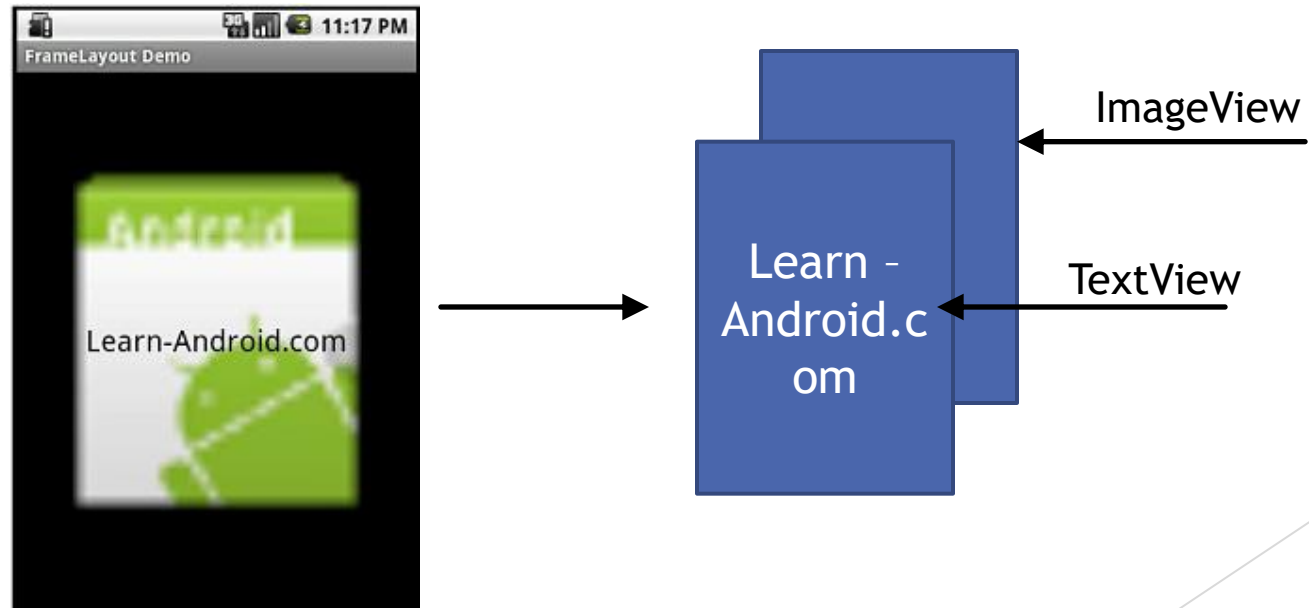
- Bố trí các thành phần giao diện theo chiều ngang, chiều dọc nhưng trên một line duy nhất mà không xuống dòng.
- Không phụ thuộc vào kích thước màn hình.



Layout

✔ FrameLayout

- Bố trí các đối tượng theo kiểu những đối tượng thuộc Layer bên dưới sẽ bị che khuất bởi các đối tượng thuộc Layer nằm trên.



Demo

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    android:id="@+id/widget68"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    >
    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="40px"
        android:layout_y="35px"
        >
        <ImageView
            android:src="@drawable/Logo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            />
        <Button
            android:layout_width="124px"
            android:layout_height="wrap_content"
            android:text="Print Picture"
            />
        </FrameLayout>
    </AbsoluteLayout>
```



Layout

✔ AbsoluteLayout

- Bố trí các Widget vào một vị trí bất kỳ trong Layout dựa vào 2 thuộc tính tọa độ x, y.
- Ít được dùng vì tọa độ của các đối tượng luôn cố định và không tự điều chỉnh được tỉ lệ khoảng cách giữa các đối tượng.

✔ RelativeLayout

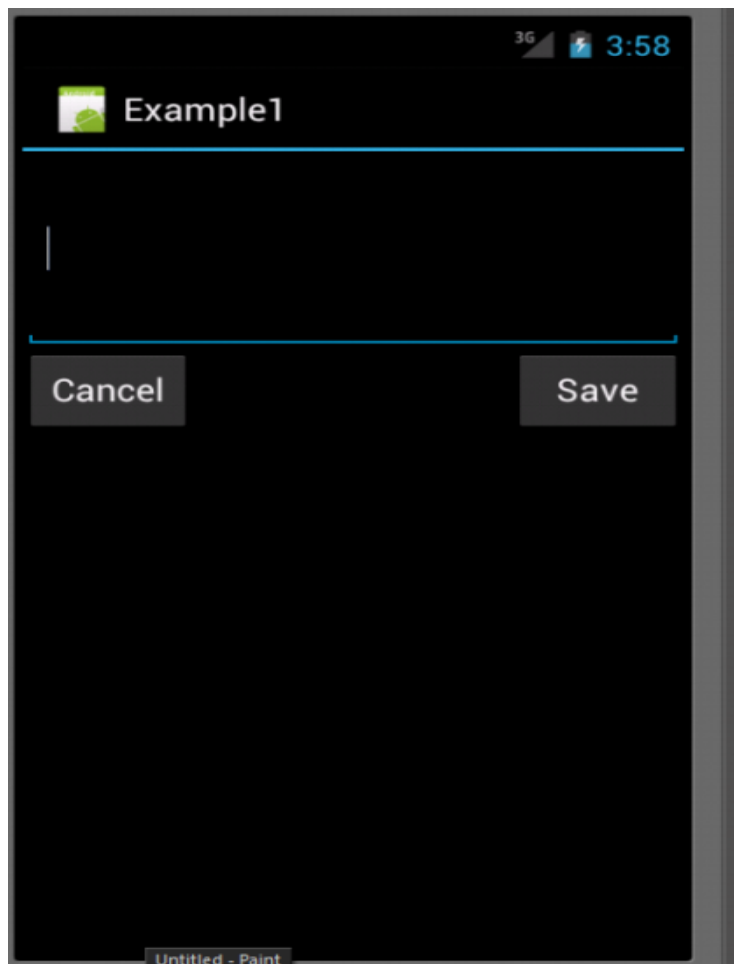
• Bố trí các Widget theo trục đối

xúng ngang hoặc dọc.

- Không phụ thuộc kích thước screen thiết bị.
- Giúp tiết kiệm Layout sử dụng, đẩy nhanh quá trình xử lý.



Thuộc tính RelativeLayout



`Layout_alignParentTop`: Vị trí trên cùng, giá trị là true hoặc false.

`Layout_alignParentLeft`: Vị trí biên trái, giá trị là true hoặc false.

`Layout_alignLeft`: căn biên trái so với View con.

`Layout_alignRight`: căn biên phải so với View con.

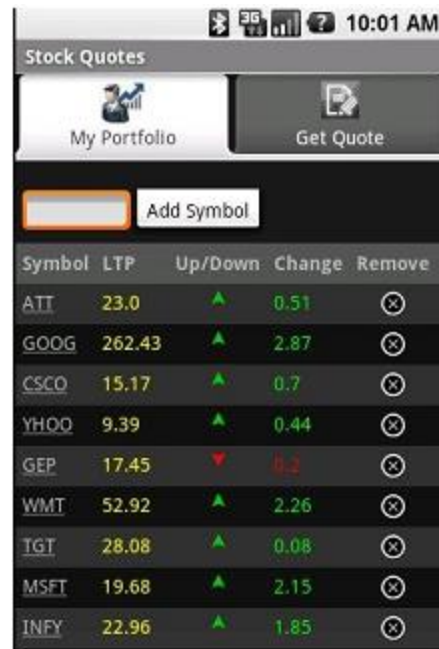
`Layout_below`: nằm dưới 1 View con nào đó.

`Layout_centerHorizontal`: căn giữa theo phương ngang.

Layout

✔ TableLayout

- Được sử dụng khi cần thiết kế một table chứa dữ liệu hoặc cần bố trí các Widget theo các row và column.



Button

- ✓ Được sử dụng nhiều trong hầu hết các ứng dụng Android.
- ✓ Có 2 cách thiết kế giao diện một Button như sau:
 - Thiết kế bằng XML

<Button

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/cmdButton1"  
    android:text="Touch me!"  
    android:onClick="touchMe"/>
```

Thuộc tính android:onClick = "touchMe" được dùng để nắm bắt sự kiện click vào Button

- Thiết kế bằng code: mục đích là để cho phù hợp với hoàn cảnh, ví dụ các nút game, các menu hay các nút điều khiển,...

Button

- Thiết kế bằng code:

Ví dụ:

Để khai báo m

```
Button cmdButton = new Button(this);  
cmdButton.setText("Touch Me!");  
cmdButon.setOnClickListener(...);
```

Để custom một Widget nào đó ta phải tạo một class kế thừa từ class Widget muốn custom. sau đó sử dụng hàm draw để vẽ lại Widget đó như một Canvas.
`canvas.drawPicture(Picture.createFromStream(...));`

ImageButton

- ✔ Tương tự Button, có thêm thuộc tính `android:src = "@drawable/icon"` để thêm hình ảnh vào và không có thẻ `text`

```
<ImageButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/cmdButton1"  
    android:src="@drawable/icon"  
    android:onClick="touchMe"/>
```



ImageButton

ImageView

- ✓ Được dùng để thể hiện một hình ảnh, giống ImageButton, chỉ khác là không có hình dáng một Button.

```
ImageView iv = new ImageView(this);  
iv.setImageResource(R.drawable.icon);
```



```
<ImageView  
    android:id="@+id/ImageView01"  
    android:layout_width="wrap_content"
```

```
<ImageButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/cmdButton1"
```

ListView

- ✔ Được sử dụng để thể hiện một danh sách các thông tin theo từng cell.
- ✔ Mỗi cell thường được load lên từ một file XML.
- ✔ Để thể hiện một list thông tin lên một screen cần 3 yếu tố chính:
 - ✔ Data Source: có thể là Array, HashMap hoặc bất kỳ một cấu trúc dữ liệu kiểu danh sách nào.
 - ✔ Adapter: là một class trung gian giúp ánh xạ dữ liệu trong DataSource vào đúng vị trí hiển thị trong ListView.
 - ✔ ListView: hiển thị thông tin trong DataSource một cách trực quan, người dùng có thể thao tác trực tiếp trên đó.

List View

Code phần listview.java trong folder src:

```
package com.android.listviewexample;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class listview extends Activity {
    private ListView lv1;
    private String lst_array[]={ "CNPM", "KHMT", "HTT", "MMT", "HTTT" };

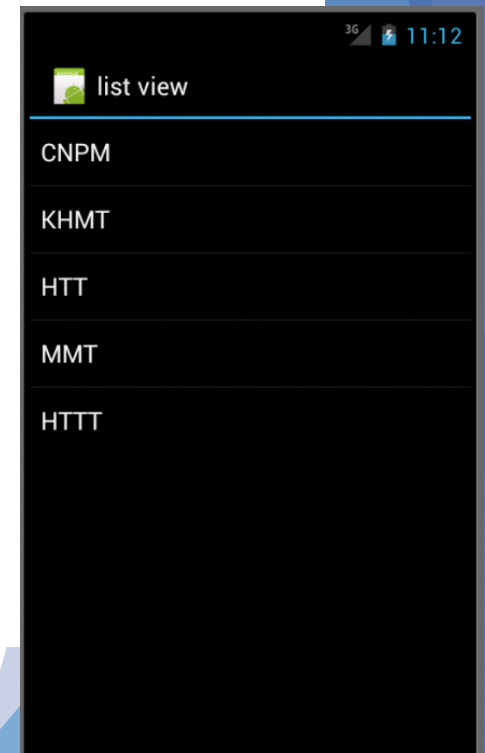
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        lv1=(ListView)findViewById(R.id.listView1);
        lv1.setAdapter(new
ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, lst_array));
    }
}
```

Code phần mail.xml:

```
<?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="vertical" >

    <ListView
        android:id="@+id/listView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

    </ListView>
</LinearLayout>
```



TextView

- ✔ Hiển thị văn bản và cho phép định dạng nội dung bằng thẻ html.

- ✔ Ví dụ:

```
TextView textView = (TextView)findViewById(R.id.textView);  
CharSequence styledText =  
    Html.fromHtml("<i>This</i> is some <b>styled</b> <s>text</s>");  
textView.setText(styledText);
```

- ✔ Nội dung TextView cũng có thể được định dạng bằng thẻ html ngay trong XML.

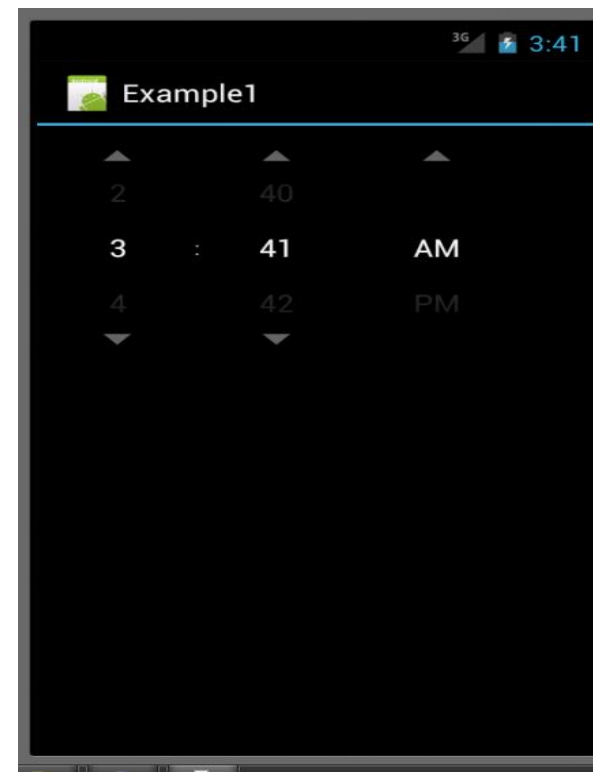
Picker Views:

- ▶ View này giúp bạn lựa chọn thời gian và ngày tháng trong hiển thị trong Android. Android hỗ trợ việc này thông qua 2 view là:
 - + TimePicker
 - + DatePicker.

TIMEPICKER VIEW

- View TimePicker cho phép người dùng lựa chọn thời điểm trong ngày, có thể là chế độ 24h hoặc là chế độ 12h cùng với AM/PM.

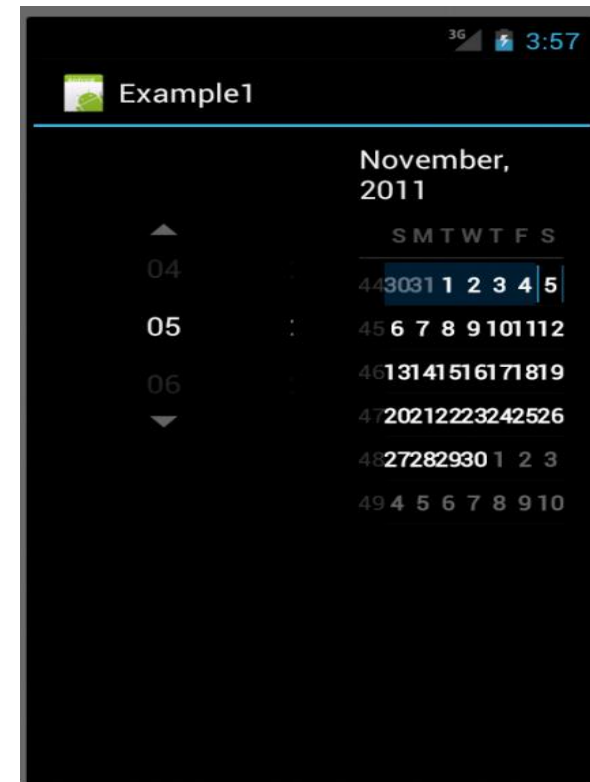
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TimePicker
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```



DATEPICKER VIEW

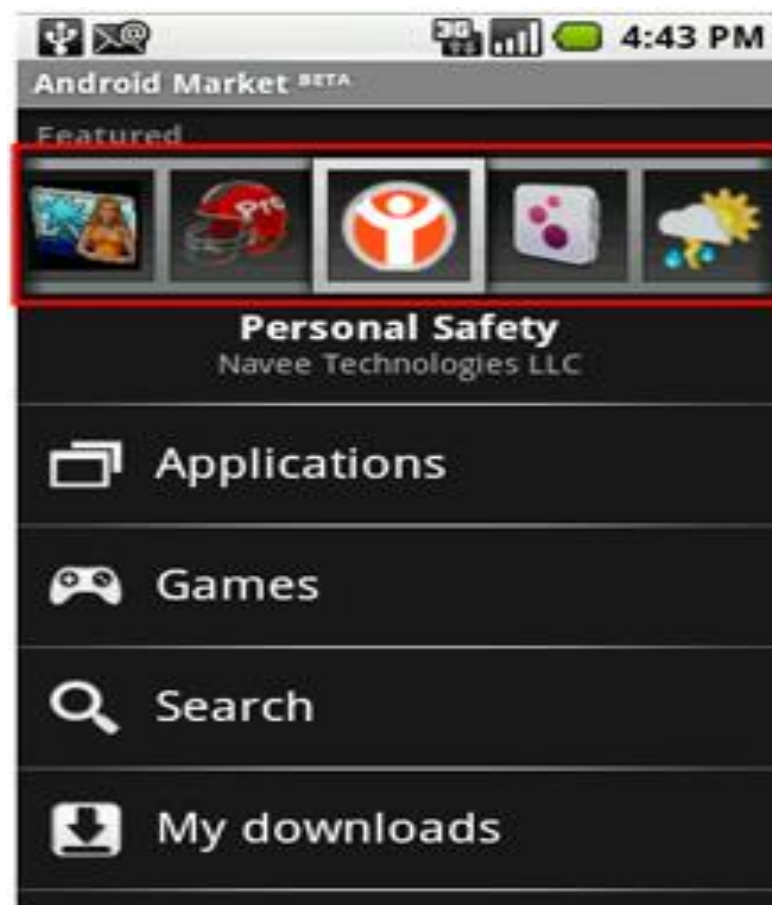
- Cũng giống như TimePicker View. DatePicker View giúp cho người dùng Android có thể chọn và thay đổi ngày.

```
<?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"
    >
    <DatePicker
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```



Display Views

- ▶ Views này dùng để hiển thị hình ảnh trong một vùng đóng, danh sách hình ảnh dịch chuyển theo chiều ngang.



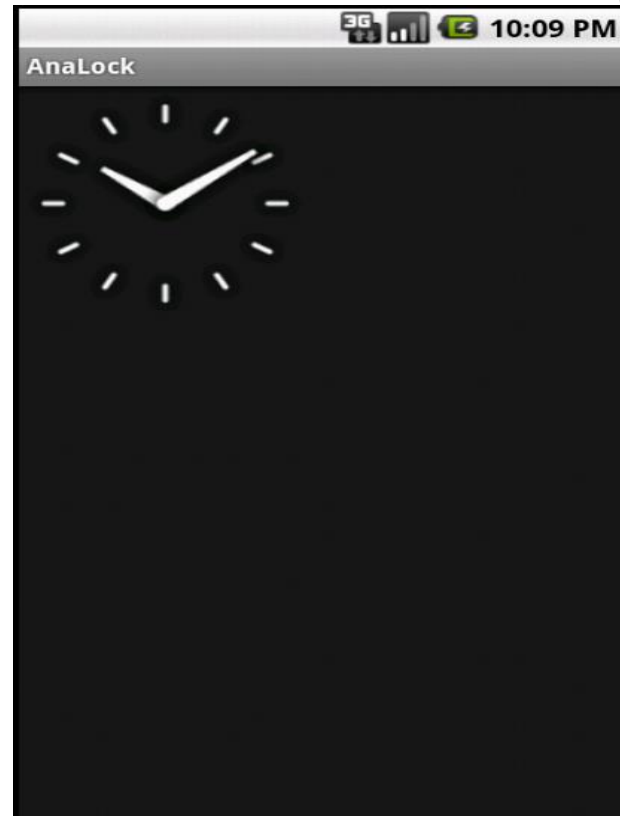
Addition View

- ▶ AnalogClock Views: hiển thị một đồng hồ kim treo tường

```
<?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"
    >

    <AnalogClock android:id="@+id/clock1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        />

</LinearLayout>
```



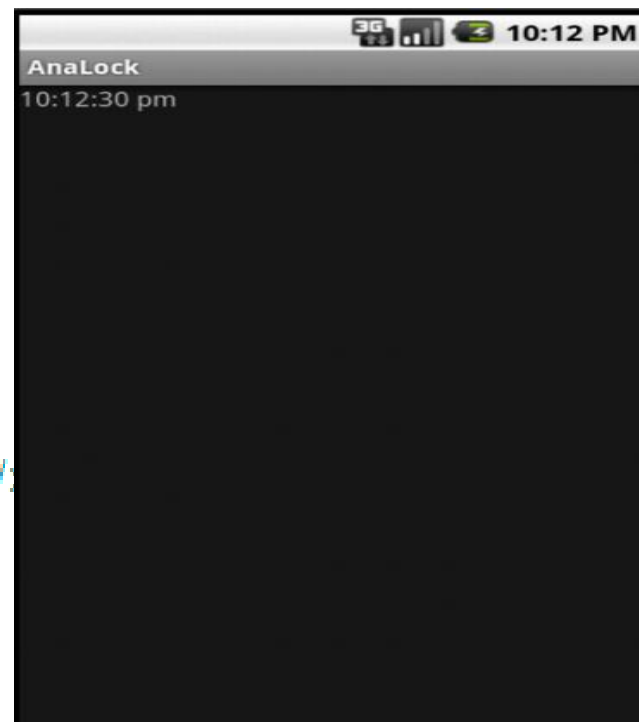
DigitalClock Views

- Views dùng để hiển thị đồng hồ số

```
<?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"
    >

    <DigitalClock android:id="@+id/clock2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```



CheckBox

- ✔ Nhận 2 giá trị true hoặc false, cho phép chọn nhiều item cùng một lúc.
- ✔ Khai báo: `CheckBox cb =new CheckBox(Context...)::`

```
<CheckBox
    android:id="@+id/CheckBox01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Check me"
    android:checked="true"/>
```

EditText

- ✓ Được sử dụng như một TextField hoặc một TextBox.
- ✓ Ví dụ:

```
<EditText  
    android:id="@+id/EditText01"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textStyle="bold"  
    android:textSize="20dip"  
    android:textColor="#000000"  
    android:text="Hello Android!"  
    android:singleLine="true"  
    android:inputType="textCapWords"/>
```

android:singleLine="true" EditText trở thành TextField, ngược lại là TextBox

android:inputType="..." sử dụng để xác định phương thức nhập cho EditText, như Password, Email...

MenuOptions

✔ Có 2 cách tạo một MenuOptions:

• Tạo bằng code:

```
public class Main extends Activity {
    private int searchBtnId = Menu.FIRST;
    private int scheduleBtnId = Menu.FIRST + 1;
    private int playBtnId = Menu.FIRST + 2;
    private int stopBtnId = Menu.FIRST + 3;
    private int group1Id = 1;
    private int group2Id = 2;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this setContentView(R.layout.main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        menu.add(group1Id, searchBtnId, searchBtnId, "Search");
        menu.add(group2Id, scheduleBtnId, scheduleBtnId, R.string.schedule);
        menu.add(group2Id, playBtnId, playBtnId, "Play");
        menu.add(group2Id, stopBtnId, stopBtnId, R.string.stop);
        // the following line will hide search
        // when we turn the 2nd parameter to false
        menu.setGroupVisible(1, false);
        return super.onCreateOptionsMenu(menu);
    }
}
```

