

LẬP TRÌNH DI ĐỘNG

Các widgets cơ bản và
các kiểu xử lý sự kiện

Nội dung



- Các widgets cơ bản
- Các kiểu xử lý sự kiện

Các widgets cơ bản



- TextView
- EditText
- Button

TextView



- Cho phép hiển thị các nội dung văn bản ở 4 dạng:
 - Normal
 - SmallText
 - MediumText
 - LargeText
- Thiết lập nội dung hiển thị:
 - Trong Java code:

```
textView.setText("Nội dung trong TextView");
```
 - Trong XML:

```
android:text="Nội dung trong TextView"
```

TextView



- Cơ chế tự động thiết lập hành động cho các siêu liên kết, bao gồm:
 - Web
 - Email
 - Phone
 - Map
- Phương thức thiết lập
 - Trong Java code:
 - `textView.setAutoLinkMask(Linkify.PHONE_NUMBERS);`
 - Trong XML:
 - `android:autoLink="phone"`

TextView



- Cho phép hiển thị hình ảnh theo văn bản ở hai dạng:
 - Theo bố cục văn bản: Left, Top, Right, Bottom
 - Theo đoạn văn bản: Start, End
- Phương thức thiết lập
 - Trong Java code:
 - `textView.setCompoundDrawables(Left, Top, Right, Bottom);`
 - Trong XML:
 - `android:drawableLeft="@drawable/ic_launcher"`
- Một số phương thức quan trọng
 - `setTextColor` – `android:textColor`
 - `setTextSize` – `android:textSize`
 - `setTypeFace` – `android:typeFace`

TextView



- Một số thuộc tính của TextView thường dùng

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#ff0000ff"
    android:textStyle="bold"
    android:textSize="25sp"
    android:padding="5dp"
    android:fontFamily="tahoma"
    android:textColor="@android:color/holo_red_dark"
    android:text="Nhập tên:" />
```

- Lưu ý: nên thiết lập **id** cho view để tiện xử lý

TextView

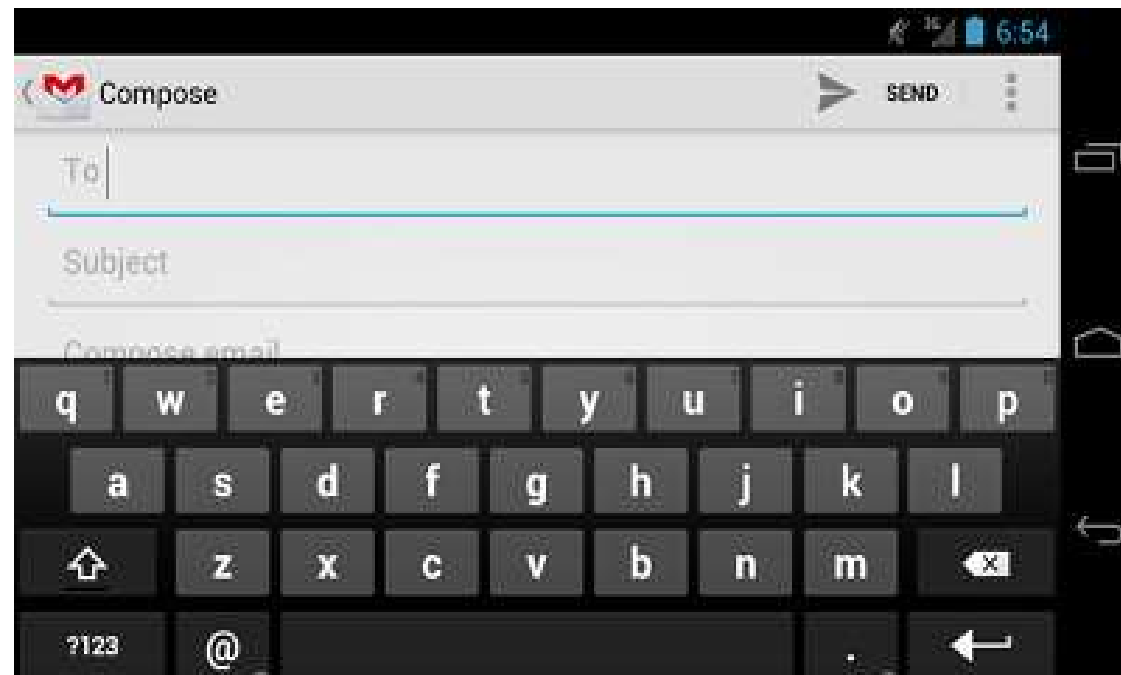


- Cách lấy view theo Id:
 - `TextView txt1= (TextView) findViewById(R.id.textView1);`
- Hàm **findViewById** trả về 1 View theo đúng Id truyền vào, và ta ép kiểu cho đúng với View đang dùng.
- Hiển thị thông tin lên TextView, dùng lệnh sau:
 - `txt1.setText("Hello");`
- Lấy thông tin bên trong TextView, dùng lệnh:
 - `String msg=txt1.getText().toString();`

EditText



- View này kế thừa từ TextView, cho phép nhập dữ liệu



EditText



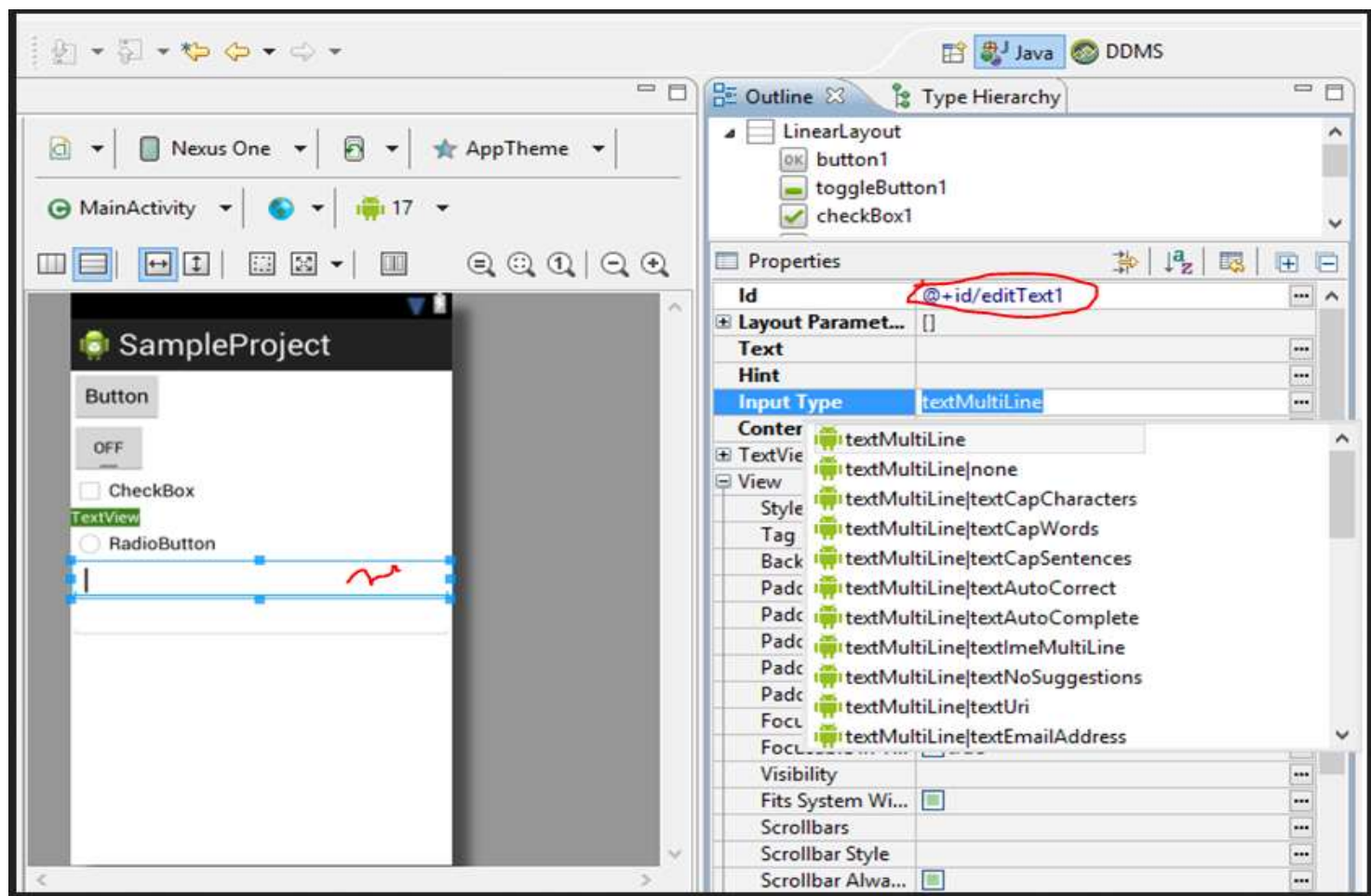
- Một số thuộc tính
 - Thuộc tính **hint**: hiển thị thông tin gợi ý trong vùng nhập dữ liệu khi chưa nhập dữ liệu nào, khi có dữ liệu thì dữ liệu trong **hint** sẽ tự động mất.
 - **textSize** để thiết lập kích cỡ font chữ

```
<EditText
    android:id="@+id/editText2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="text|textAutoCorrect|textCapWords"
    android:hint="Nhập tài khoản"
    android:textSize="18sp"
    android:ems="10" />
```

EditText



- Có thể dùng cửa sổ Properties để thiết lập thuộc tính



EditText



- Chỉ định kiểu dữ liệu nhập

- Sử dụng thuộc tính `android:inputType`

- Ví dụ: Nhập vào địa chỉ email:

- `android:inputType="textEmailAddress".`

- `<EditText`

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



# EditText



- Chỉ định kiểu dữ liệu nhập

| Kiểu nhập          | Chức năng                              |
|--------------------|----------------------------------------|
| "text"             | Kiểu bàn phím thông thường             |
| "textEmailAddress" | Kiểu bàn phím thông thường với ký tự @ |
| "textUri"          | Kiểu bàn phím thông thường với ký tự / |
| "number"           | Bàn phím với các ký tự số              |
| "phone"            | Bàn phím số điện thoại                 |

# EditText



- Chỉ định cách hành xử của bàn phím
  - Thuộc tính **android:inputType** cũng cho phép chỉ định các cách hành xử của bàn phím
  - Chẳng hạn viết hoa tất cả các ký tự hoặc auto-complete, gợi ý sửa lỗi chính tả. Chúng ta có thể kết hợp nhiều giá trị input-type bằng phép toán OR (ký hiệu '|')

# EditText



- Một số giá trị input-type để định nghĩa cách hành xử của bàn phím

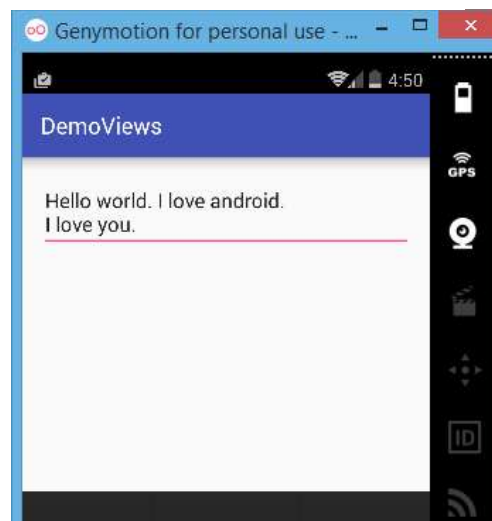
| Kiểu nhập          | Chức năng                                                                                        |
|--------------------|--------------------------------------------------------------------------------------------------|
| "textCapSentences" | Bàn phím thông thường với tính năng viết hoa ký tự đầu tiên của mỗi câu.                         |
| "textCapWords"     | Bàn phím thông thường với tính năng viết hoa mỗi từ. Thường dùng để nhập tên riêng hoặc tiêu đề. |
| "textAutoCorrect"  | Bàn phím thông thường với tính năng sửa lỗi chính tả.                                            |
| "textPassword"     | Bàn phím thông thường với tính năng các ký tự nhập vào sẽ được hiển thị thành dấu chấm.          |
| "textMultiLine"    | Bàn phím thông thường với tính năng cho phép nhập văn bản có nhiều dòng (có dấu xuống dòng)      |

# EditText



- Ví dụ: Hiển thị trường văn bản cho phép nhập nhiều dòng, tự động viết hoa ký tự đầu mỗi câu.

```
<EditText
 android:id="@+id/email_address1"
 android:layout_width="fill_parent"
 android:layout_height="wrap_content"
 android:inputType="textCapSentences|textMultiLine" />
```





# EditText



- Lấy view theo Id:
  - `EditText txtbox=(EditText) findViewById(R.id.editText1);`
- Thiết lập giá trị cho EditText
  - `txtbox.setText("nhập nội dung")`
- Lấy dữ liệu bên trong EditText:
  - `String msg=txtbox.getText().toString()`

# Button



- Cho phép thể hiện nội dung văn bản, hình ảnh
- Nhận và phản hồi tương tác nhấn từ người dùng
- **Các dạng Button**
  - Button
  - CompoundButton
    - CheckBox
    - RadioButton
    - ToggleButton
    - Switch



# Button



## ■ Tạo Button

```
<Button
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="@string/button_text"
 ... />
```

Để tạo một Button chứa hình ảnh (icon), chúng ta dùng ImageButton:

```
<ImageButton
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:src="@drawable/button_icon"
 ... />
```

# Button



- Tạo một Button chứa cả chữ và hình ảnh
  - Đặt thuộc tính `android:drawableLeft` (hoặc `android:drawableRight`, `android:drawableTop`, `android:drawableBottom`)

```
<Button
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="@string/button_text"
 android:drawableLeft="@drawable/button_icon"
 ... />
```

# Button



- **Sự kiện**

- Có 2 sự kiện thường dùng: OnClick và OnLongClick

```
Button btnok=(Button) findViewById(R.id.btnOk);
btnok.setOnClickListener(new View.OnClickListener() {
 public void onClick(View arg0) {
 //perform click here
 }
});
btnok.setOnLongClickListener(new View.OnLongClickListener() {
 public boolean onLongClick(View arg0) {
 //perform long click here
 return false;
 }
});
```

# Các kiểu lập trình sự kiện

---



## ■ Listener

- Các Listener được sử dụng để bắt một sự kiện nào đó
  - OnKeyListener: bắt sự kiện khi nhấn phím
  - OnClickListener: bắt sự kiện khi chạm vào một view trên màn hình
- Mỗi view đều phải set Listener riêng để xử lý cho kiện tương ứng với view đó
- Mỗi loại view có Listener dành riêng cho nó
- VD:
  - Button: setOnKeyListener
  - Button: setOnClickListener
  - CheckBox: setOnCheckedChangeListener

# Các kiểu lập trình sự kiện

---



- Onclick in XML
- Inline anonymous listener
- Activity is listener
- Listener in variable
- Explicit listener class
- View Subclassing

# Các kiểu lập trình sự kiện



- Onclick in XML

| Các kiểu lập trình sự kiện |    |
|----------------------------|----|
| Số a:                      | 80 |
| Số b:                      | 33 |
| 113                        |    |
| Tổng 2 số                  |    |

```
<Button
 android:id="@+id/btntong2so"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:onClick="btn_tong2so"
 android:text="Tổng 2 số" />
```

- Sử dụng thuộc tính **android:onClick="btn\_tong2so"**, tức là gán sự kiện click cho Button này, sự kiện này tên là **btn\_tong2so**.
- Rồi khai báo tường minh phương thức **btn\_tong2so** trong MainActivity.java



# Các kiểu lập trình sự kiện



- Onclick in XML
  - Khai báo phương thức btn\_tong2so trong MainActivity.java

```
public class MainActivity extends Activity {
 protected void onCreate(Bundle savedInstanceState) {
 public void btn_tong2so(View v)
 {
 EditText edita=(EditText)
 findViewById(R.id.editsoa);
 int a=Integer.parseInt(edita.getText()+"");

 EditText editb=(EditText) findViewById(R.id.editsob);

 int b=Integer.parseInt(editb.getText()+"");

 TextView txtkq=(TextView) findViewById(R.id.txtkq);
 txtkq.setText((a+b)+"");
 }
 public boolean onCreateOptionsMenu(Menu menu) {
 }
```

# Các kiểu lập trình sự kiện



- Inline anonymous listener



```
<Button
 android:id="@+id/btnTong"
 android:layout_width="fill_parent"
 android:layout_height="wrap_content"
 android:layout_span="3"
 android:text="Tính tổng" />
```

# Các kiểu lập trình sự kiện



- **Inline anonymous listener**

- Xử lý sự kiện trong MainActivity.java

```
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 Button btnTong = (Button)findViewById(R.id.btnTong);
 btnTong.setOnClickListener(new
View.OnClickListener() {
 @Override
 public void onClick(View v) {
 //B1: lấy dữ liệu từ các EditText
 //B2: chuyển dữ liệu sang kiểu số
 //B3: tính tổng và hiển thị kết quả
 }
}
...
}
```

# Các kiểu lập trình sự kiện



- Activity is Listener

- File xml vẫn như ví dụ trên
- File MainActivity

```
public class MainActivity extends Activity implements
OnClickListener {
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 Button btn_clickme = (Button)
 findViewById(R.id.btnTong);
 btn_clickme.setOnClickListener(this);
 }
 @Override
 public void onClick(View arg0) {
 // Xử lý khi button được click
 }
}
```

# Các kiểu lập trình sự kiện

---



- Tìm hiểu thêm các xử lý sự kiện
    - Listener in variable
    - Explicit listener class
    - View Subclassing
- => So sánh

# Một số view thường dùng khác

---



- `ToggleButton`
- `Checkbox`
- `RadioGroup` & `RadioButton`
- `Picker`

# Toggle Button và Switch



- Cho phép người dùng thay đổi thiết đặt giữa hai trạng thái (bật hoặc tắt)



- Android 4.0 (API level 14) giới thiệu một loại khác của Toggle Button, đó là Switch.



# Toggle Button và Switch



```
<Switch
```



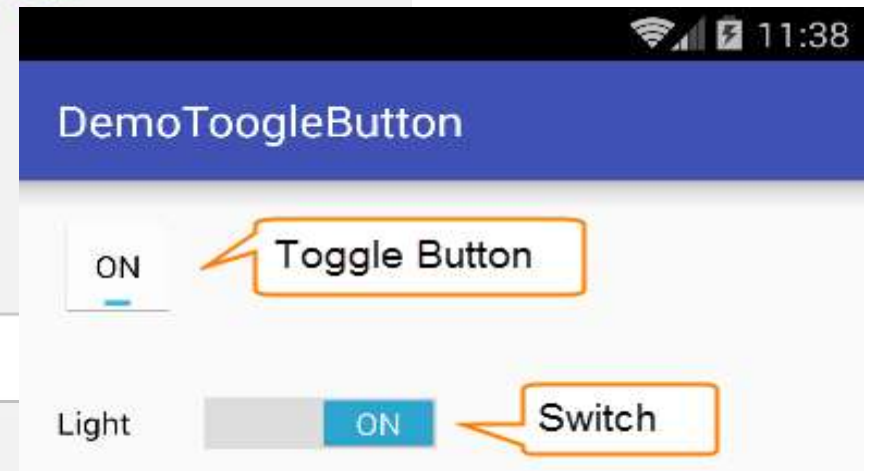
```
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Light"
 android:id="@+id/switch1"
 android:checked="true"
```

```
/>
```

```
<ToggleButton
```

```
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="New ToggleButton"
 android:id="@+id/toggleButton"
```

```
/>
```





# Toggle Button và Switch

---



- Xử lý sự kiện
  - Toggle Button nhận sự kiện **OnCheckedChangeListener**.
  - Thiết lập view Toggle Button nhận sự kiện:  
**setOnCheckedChangeListener**

# Toggle Button và Switch



- Xử lý sự kiện đối với Toggle Button

```
ToggleButton toggle = (ToggleButton) findViewById(R.id.toggleButton);
toggle.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
 public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
 if (isChecked) {
 // The toggle is enabled
 Toast.makeText(MainActivity.this, "TOGGLE IS ENABLED",
Toast.LENGTH_SHORT).show();
 } else {
 // The toggle is disabled
 Toast.makeText(MainActivity.this, "TOGGLE IS DISABLED",
Toast.LENGTH_SHORT).show();
 }
 }
});
```

# Toggle Button và Switch



- Xử lý sự kiện đối với Switch

```
Switch sw = (Switch) findViewById(R.id.switch1);
sw.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
 public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
 if (isChecked) {
 Toast.makeText(MainActivity.this, "SWITCH IS ON",
Toast.LENGTH_SHORT).show();
 } else {
 Toast.makeText(MainActivity.this, "SWITCH IS OFF",
Toast.LENGTH_SHORT).show();
 }
 }
});|
```

# Checkbox



- Nút bấm hai trạng thái “chọn” và “bỏ chọn”.
- Sử dụng 2 phương thức
  - **setChecked**: dùng để thiết lập checked.
    - `setChecked(true)`: chọn
    - `setChecked(false)`: bỏ chọn
  - **isChecked**: kiểm tra view có được chọn hay không. Trả về giá trị **true/false**

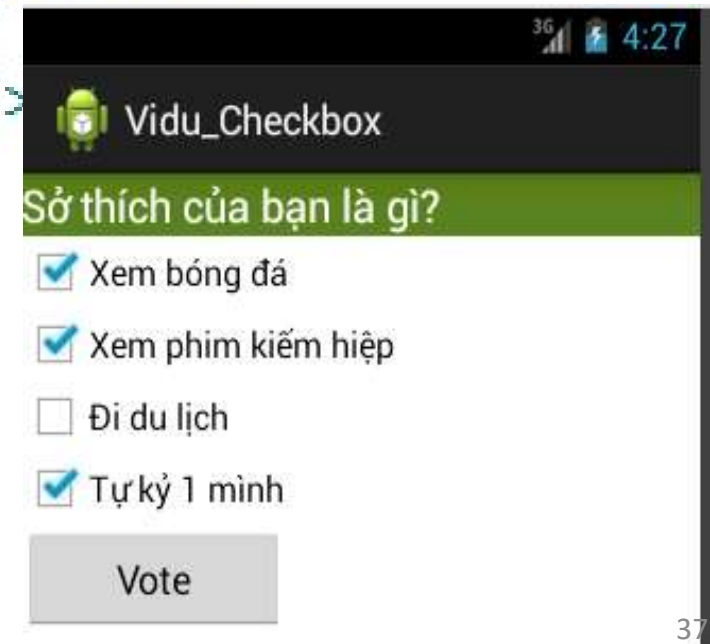
# Checkbox



## ■ Ví dụ

```
<CheckBox
```

```
 android:id="@+id/checkBox1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:checked="true"
 android:text="Xem bóng đá" />
```



# Checkbox



## ■ Xử lý

```
CheckBox chk=(CheckBox) findViewById(R.id.checkbox1);
if(chk.isChecked())
{
 //xử lý checked
}
else
{
 //xử lý unchecked
}
//Muốn thiết lập checked:
chk.setChecked(true);
//Muốn clear checked:
chk.setChecked(false);
```

# RadioGroup & RadioButton

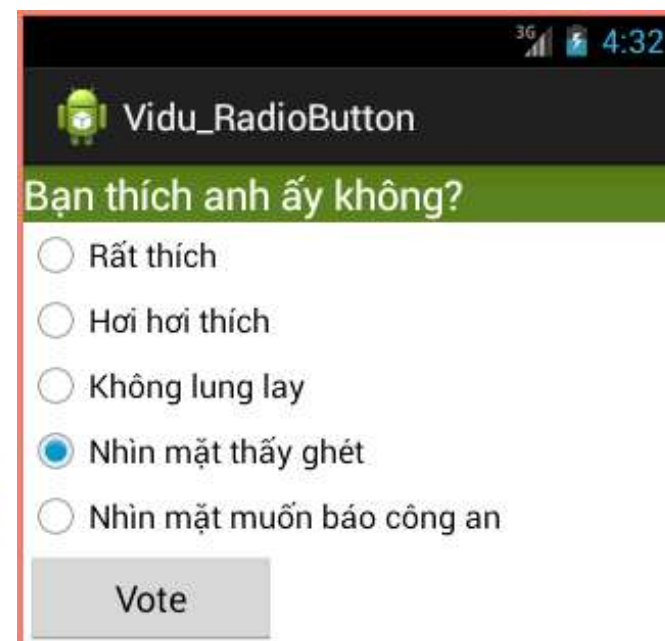


## ■ Ví dụ

```
<RadioGroup
 android:id="@+id/radioGroup1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content" >

 <RadioButton
 android:id="@+id/radio0"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Rất thích" />

 <RadioButton
 android:id="@+id/radio1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:checked="true"
 android:text="Hơi hơi thích" />
</RadioGroup>
```



# RadioGroup & RadioButton



- Xử lý RadioButton nào được chọn:
  - **Cách 1:** Dựa vào RadioGroup để biết Id của RadioButton nào được checked.

```
RadioGroup group=(RadioGroup) findViewById(R.id.radioGroup1);
int idChecked=group.getCheckedRadioButtonId();
switch(idChecked)
{
case R.id.radrathailong:
 break;
case R.id.radhailong:
 break;
case R.id.radtamchapnhan:
 break;
case R.id.radthayghe:
 break;
}
```



# RadioGroup & RadioButton



- Xử lý RadioButton nào được chọn:
  - **Cách 2:** Kiểm tra trực tiếp RadioButton đó có được checked hay không

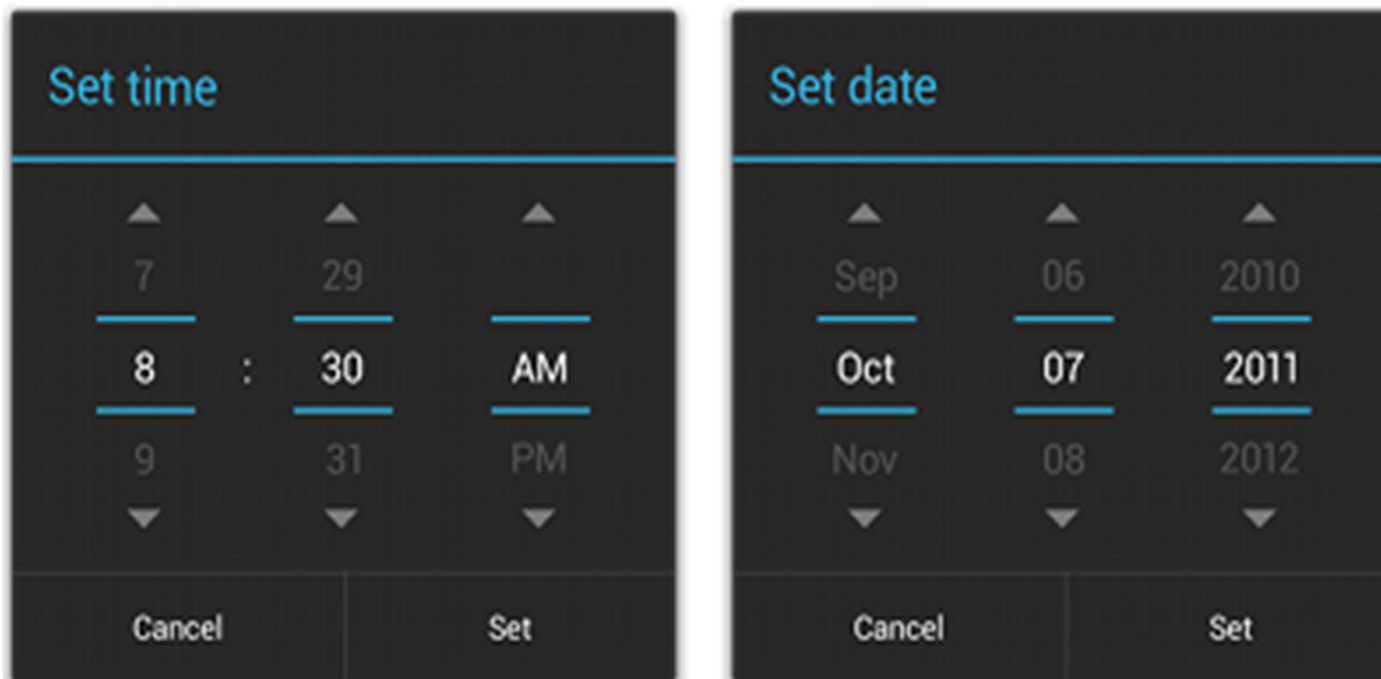
```
RadioButton rad=(RadioButton) findViewById(R.id.radrathailong);
if(rad.isChecked())
{

}
```

# Picker



- View cho phép chọn thời gian, ngày tháng
  - TimePicker
  - DatePicker



# Picker



- Tạo TimePicker và DatePicker

```
<TimePicker
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:id="@+id/timePicker" />
<DatePicker
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:id="@+id/datePicker" />
```

# Picker



- Lấy/gán giá trị từ TimePicker và DatePicker
  - Hàm `setCurrentHour(h)` và `setCurrentMinute(m)` để chỉnh thời gian hiển thị
  - Hàm `getCurrentHour()` và `getCurrentMinute()` để lấy thời gian do người dùng nhập
  - Không có cách thiết lập thời gian bằng design

```
TimePicker tp = (TimePicker) findViewById(R.id.timePicker);
String t = tp.getCurrentHour() + ":" + tp.getCurrentMinute();
```

- **TimePickerDialog** hiển thị dialog cho phép thiết lập giờ ban đầu và người dùng hiệu chỉnh sau

# Picker



- **OnDateSetListener**

```
final Calendar c = Calendar.getInstance();
```

```
mYear = c.get(Calendar.YEAR);
```

```
mMonth = c.get(Calendar.MONTH);
```

```
mDay = c.get(Calendar.DAY_OF_MONTH);
```

```
DatePickerDialog dpd = new DatePickerDialog(this,
```

```
 new DatePickerDialog.OnDateSetListener() {
```

```
 @Override
```

```
 public void onDateSet(DatePicker v, int y, int m, int d) {
```

```
 txtDate.setText(d + "-" + (m + 1) + "-" + y);
```

```
 }
```

```
 }, mYear, mMonth, mDay);
```

```
dpd.show();
```

# Picker



- OnTimeSetListener

```
TimePickerDialog tpd = new TimePickerDialog(this,
 new TimePickerDialog.OnTimeSetListener() {
 @Override
 public void onTimeSet(TimePicker view, int hour,
 int minute) {
 txtTime.setText(hour + ":" + minute);
 }
 }, mHour, mMinute, false);
tpd.show();
```