



**FPT POLYTECHNIC**

THIẾT KẾ GIAO DIỆN TRÊN  
ANDROID

## **Bài 3: Các widget cơ bản**

[www.poly.edu.vn](http://www.poly.edu.vn)

## Nội dung bài học

1. TextView
2. Buttons
3. Images
4. EditText
5. Checkboxes
6. RadioButtons
7. ToggleButtons



# 1. Basic Widget: TextView

- Trong Android, một label được sử dụng là **TextView**.
- Một **TextView** dùng để hiển thị thông tin và không cho phép người dùng chỉnh sửa.
- Một số thuộc tính của control:
  - Cần thiết lập id cho control.
  - layout\_width, layout\_height cũng nên thiết lập cho control (bắt buộc)
  - Để thay đổi màu nền dùng background, thay đổi màu chữ dùng textColor...

PHONE

1 650-123 MOBILE

EMAIL

Email HOME

ADDRESS

Address HOME

# 1. Basic Widget: TextView

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
  android:id="@+id/absLayout"
  android:layout_width="fill_parent" android:layout_height="fill_parent"
  xmlns:android="http://schemas.android.com/apk/res/android"
>
<TextView
  android:id="@+id/myTextView1"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:background="#ff0000ff"
  android:padding="3px"
  android:text="Enter User Name"
  android:textSize="16sp"
  android:textStyle="bold"
  android:gravity="center"
  android:layout_x="20px"
  android:layout_y="22px" >
</TextView>
</AbsoluteLayout>
```

## 1. Basic Widget: TextView

- Để hiển thị thông tin lên control TextView ta dùng lệnh dưới đây:

**`txt1.setText("Hello world");`**

- Để lấy thông tin bên trong control TextView ta dùng lệnh dưới đây:

**`String msg=txt1.getText().toString();`**

# 1. Basic Widget: TextView

## ● Tham khảo một số thuộc tính của Textview:

Attribute Name	Related Method	Description
android:autoLink	setAutoLinkMask(int)	Controls whether links such as urls and email addresses are automatically found and converted to clickable links.
android:autoText	setKeyListener(KeyListener)	If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.
android:bufferType	setText(CharSequence,TextView.BufferType)	Determines the minimum type that getText() will return.
android:capitalize	setKeyListener(KeyListener)	If set, specifies that this TextView has a textual input method and should automatically capitalize what the user types.
android:cursorVisible	setCursorVisible(boolean)	Makes the cursor visible (the default) or invisible Must be a boolean value, either "true" or "false".
android:digits	setKeyListener(KeyListener)	If set, specifies that this TextView has a numeric input method and that these specific characters are the ones that it will accept.
android:drawableBottom	setCompoundDrawablesWithIntrinsicBounds(Drawable,Drawable,Drawable,Drawable)	The drawable to be drawn below the text.
android:drawableLeft	setCompoundDrawablesWithIntrinsicBounds(Drawable,Drawable,Drawable,Drawable)	The drawable to be drawn to the left of the text.
android:drawablePadding	setCompoundDrawablePadding(int)	The padding between the drawables and the text.
android:drawableRight	setCompoundDrawablesWithIntrinsicBounds(Drawable,Drawable,Drawable,Drawable)	The drawable to be drawn to the right of the text.
android:drawableTop	setCompoundDrawablesWithIntrinsicBounds(Drawable,Drawable,Drawable,Drawable)	The drawable to be drawn above the text.
android:editable		If set, specifies that this TextView has an input method.
android:editorExtras	setInputExtras(int)	Reference to an <input-extras> XML resource containing additional data to supply to an input method, which is private to the implementation of the input method.
android:ellipsize	setEllipsize(TextUtils.TruncateAt)	If set, causes words that are longer than the view is wide to be ellipsized instead of broken in the middle.
android:ems	setEms(int)	Makes the TextView be exactly this many ems wide Must be an integer value, such as "100".
android:freezesText	setFreezesText(boolean)	If set, the text view will include its current complete text inside of its frozen icicle in addition to meta-data such as the current cursor position.

# 1. Basic Widget: TextView

Attribute Name	Related Method	Description
android:gravity	setGravity(int)	Specifies how to align the text by the view's x and/or y axis when the text is smaller than the view.
android:height	setHeight(int)	Makes the TextView be exactly this many pixels tall.
android:hint	setHint(int)	Hint text to display when the text is empty.
android:imeActionId	setImeActionLabel(CharSequence,int)	Supply a value for <a href="#">EditorInfo.actionId</a> used when an input method is connected to the text view.
android:imeActionLabel	setImeActionLabel(CharSequence,int)	Supply a value for <a href="#">EditorInfo.actionLabel</a> used when an input method is connected to the text view.
android:imeOptions	setImeOptions(int)	Additional features you can enable in an IME associated with an editor, to improve the integration with your application.
android:includeFontPadding	setIncludeFontPadding(boolean)	Leave enough room for ascenders and descenders instead of using the font ascent and descent strictly.
android:inputMethod	setKeyListener(KeyListener)	If set, specifies that this TextView should use the specified input method (specified by fully-qualified class name).
android:inputType	setRawInputType(int)	The type of data being placed in a text field, used to help an input method decide how to let the user enter text.
android:lineSpacingExtra	setLineSpacing(float,float)	Extra spacing between lines of text.
android:lineSpacingMultiplier	setLineSpacing(float,float)	Extra spacing between lines of text, as a multiplier.
android:lines	setLines(int)	Makes the TextView be exactly this many lines tall Must be an integer value, such as "100".
android:linksClickable	setLinksClickable(boolean)	If set to false, keeps the movement method from being set to the link movement method even if <a href="#">autoLink</a> causes links to be found.
android:marqueeRepeatLimit	setMarqueeRepeatLimit(int)	The number of times to repeat the marquee animation.
android:maxEms	setMaxEms(int)	Makes the TextView be at most this many ems wide Must be an integer value, such as "100".
android:maxHeight	setMaxHeight(int)	Makes the TextView be at most this many pixels tall Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".
android:maxLength	setFilters(InputFilter)	Set an input filter to constrain the text length to the specified number.
android:maxLines	setMaxLines(int)	Makes the TextView be at most this many lines tall Must be an integer value, such as "100".

# 1. Basic Widget: TextView

Attribute Name	Related Method	Description
android:maxLength	setMaxLength(int)	Makes the TextView be at most this many pixels wide Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".
android:minEms	setMinEms(int)	Makes the TextView be at least this many ems wide Must be an integer value, such as "100".
android:minHeight	setMinHeight(int)	Makes the TextView be at least this many pixels tall Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".
android:minLines	setMinLines(int)	Makes the TextView be at least this many lines tall Must be an integer value, such as "100".
android:minWidth	setMinWidth(int)	Makes the TextView be at least this many pixels wide Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".
android:numeric	setKeyListener(KeyListener)	If set, specifies that this TextView has a numeric input method.
android:password	setTransformationMethod(TransformationMethod)	Whether the characters of the field are displayed as password dots instead of themselves.
android:phoneNumber	setKeyListener(KeyListener)	If set, specifies that this TextView has a phone number input method.
android:privateImeOptions	setPrivateImeOptions(String)	An addition content type description to supply to the input method attached to the text view, which is private to the implementation of the input method.
android:scrollHorizontally	setHorizontallyScrolling(boolean)	Whether the text is allowed to be wider than the view (and therefore can be scrolled horizontally).
android:selectAllOnFocus	setSelectAllOnFocus(boolean)	If the text is selectable, select it all when the view takes focus instead of moving the cursor to the start or end.
android:shadowColor	setShadowLayer(float,float,float,int)	Place a shadow of the specified color behind the text.
android:shadowDx	setShadowLayer(float,float,float,int)	Horizontal offset of the shadow.
android:shadowDy	setShadowLayer(float,float,float,int)	Vertical offset of the shadow.
android:shadowRadius	setShadowLayer(float,float,float,int)	Radius of the shadow.



# 1. Basic Widget: TextView

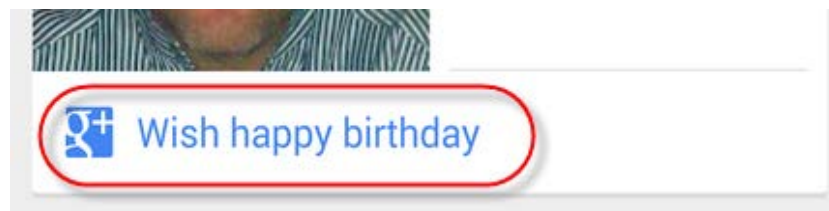
<b>Attribute Name</b>	<b>Related Method</b>	<b>Description</b>
android:singleLine	setTransformationMethod(TransformationMethod)	Constrains the text to a single horizontally scrolling line instead of letting it wrap onto multiple lines, and advances focus instead of inserting a newline when you press the enter key.
android:text	setText(CharSequence)	Text to display.
android:textColor	setTextColor(ColorStateList)	Text color.
android:textColorHighlight	setHighlightColor(int)	Color of the text selection highlight.
android:textColorHint	setHintTextColor(int)	Color of the hint text.
android:textColorLink	setLinkTextColor(int)	Text color for links.
android:textScaleX	setTextScaleX(float)	Sets the horizontal scaling factor for the text Must be a floating point value, such as "1.2".
android:textSize	setTextSize(float)	Size of the text.
android:textStyle	setTypeface(Typeface)	Style (bold, italic, bolditalic) for the text.
android:typeface	setTypeface(Typeface)	Typeface (normal, sans, serif, monospace) for the text.
android:width	setWidth(int)	Makes the TextView be exactly this many pixels wide.

## 2. Basic Widget: Buttons

- Thẻ Button cho phép người dùng thực hiện một hành động click vào giao diện.
- Là một lớp con của TextView, do đó về cơ bản các thuộc tính cũng tương tự như TextView được liệt kê trang trước.
- Một button có thể chứa text, hoặc icon, hoặc cả text cả icon.
- Với text, sử dụng thẻ button:

```
<Button
```

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



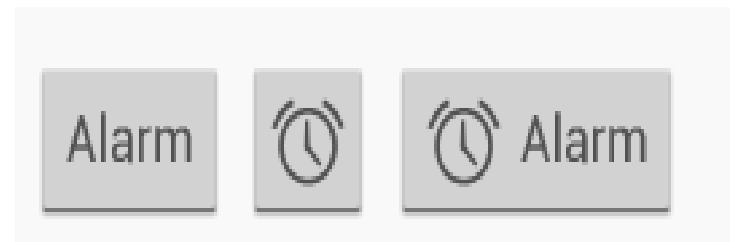
## 2. Basic Widget: Buttons

- Với text kết hợp với icon ta khai báo thêm thuộc tính `android:drawableLeft` trong thẻ button

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

- Với icon, sử dụng thẻ `ImageButton`

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



## 2. Basic Widget: Buttons

### Thực hiện sự kiện click:

- Khi người dùng click vào button, đối tượng button sẽ nhận một sự kiện on-click.
- Để xác định sự kiện này, trong thẻ <Button> phải khai báo thuộc tính android: onClick, giá trị thuộc tính phải trùng với tên của phương thức muốn gọi trong Activity.
- Phương thức được khai báo phải:
  - Public
  - Trả lại hàm void
  - Kế thừa hàm View

## 2. Basic Widget: Buttons

- Ví dụ trong thẻ button ta sử dụng android:onClick

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<Button
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:id="@+id/button_send"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="@string/button_send"
```

```
android:onClick="sendMessage" />
```

- Trong Activity, ta viết phương thức sendMessage()

```
/** Called when the user touches the button */
```

```
public void sendMessage(View view) {
```

```
    // Do something in response to button click
```

```
}
```

## 2. Basic Widget: Buttons

### Sử dụng OnClickListener:

- Để khai báo sự kiện, tạo một đối tượng **View.OnClickListener**, sau đó gán cho button bằng cách gọi **setOnClickListener(new View.OnClickListener())**.
- Ví dụ:

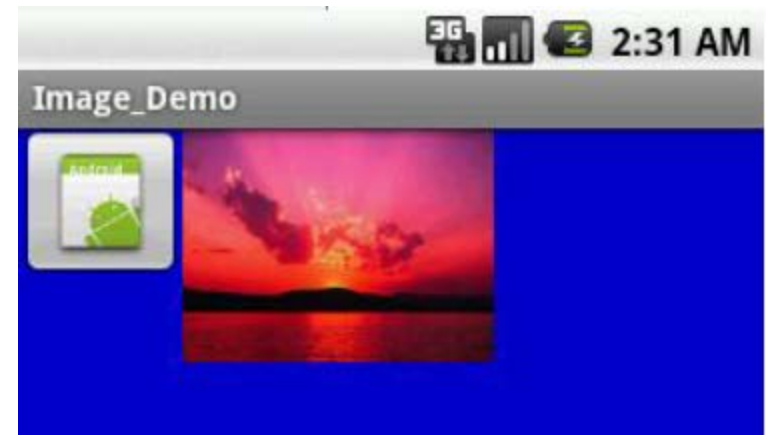
```
Button button = (Button) findViewById(R.id.button_send);
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        // Do something in response to button click
    }
});
```

### 3. Basic Widget: Images

- **ImageView** và **ImageButton** thường được dùng để nhúng hình ảnh cho ứng dụng.
- Cả hai thẻ này đều tương tự như **TextView** và **Button**.
- Các thẻ này đều sử dụng thuộc tính **android:src** hoặc **android:background** để hiển thị ảnh.
- Ảnh được sử dụng sẽ được tham chiếu tới thư mục */res/drawable/*
- **ImageButton** là một lớp con của **ImageView** và thêm một số chuẩn của **Button** để thực hiện các sự kiện Click

### 3. Basic Widget: Images (cont)

```
<ImageButton  
    android:id="@+id/myImageBtn1"  
    android:src="@drawable/icon"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" >  
</ImageButton>  
<ImageView  
    android:id="@+id/myImageView1"  
    android:src="@drawable/microsoft_sunset"  
    android:layout_width="150px"  
    android:layout_height="120px"  
    android:scaleType="fitXY">  
</ImageView>
```

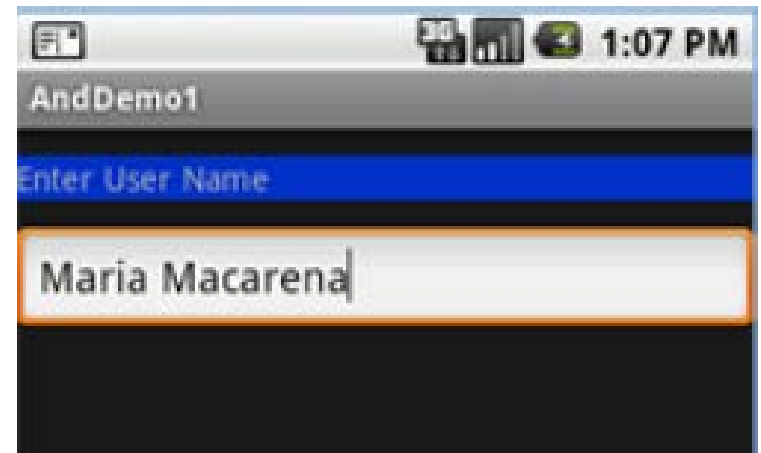




## 4. Basic Widget: EditText

- Thẻ EditText ( hoặc textBox) là một trường hợp mở rộng của TextView, cho phép cập nhật, chỉnh sửa dữ liệu

```
<EditText  
  android:id="@+id/txtUserName"  
  android:layout_width="fill_parent"  
  android:layout_height="wrap_content"  
  android:textSize="18sp"  
  android:autoText="true"  
  android:capitalize="words"  
  android:hint="First Last Name"  
>  
</EditText>
```

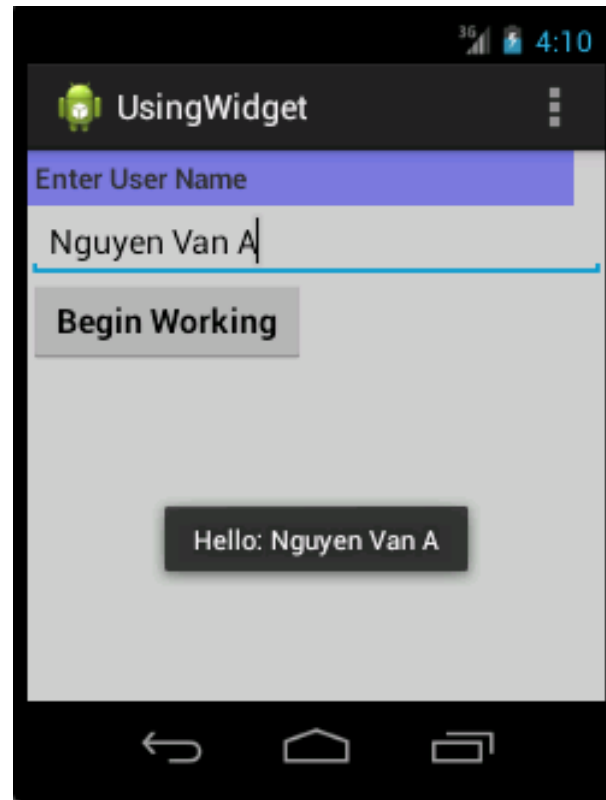


## 4. Basic Widget: EditText

- Một vài thuộc tính của thẻ EditText cần chú ý:
  - **Android: hint** để hiển thị thông tin gợi ý trong vùng nhập dữ liệu khi bạn chưa nhập bất kỳ dữ liệu nào vào, chỉ cần có dữ liệu là phần hint sẽ tự động mất đi.
  - **android:textSize** để thiết lập kích cỡ font chữ cho EditText
  - **android:autoText**, (true/false) Tự động sửa đúng chính tả, giả sử bạn nhập “teh” thì nó sẽ tự động sửa thành “the”
  - **android:capitalize**, (words/sentences) tự động viết hoa đầu câu, ký tự
  - **android:digits**, để cấu hình trường chỉ cho phép nhập dạng số
  - **android:singleLine**, is the field for single-line / multiple-line input: cho phép nhập một hay nhiều dòng.
  - **android:password**, (true/false) trường nhập vào sẽ được ẩn.
  - **android:numeric**, (integer, decimal, signed) định dạng kiểu số
  - **android:phoneNumber**, (true/false) định dạng số điện thoại

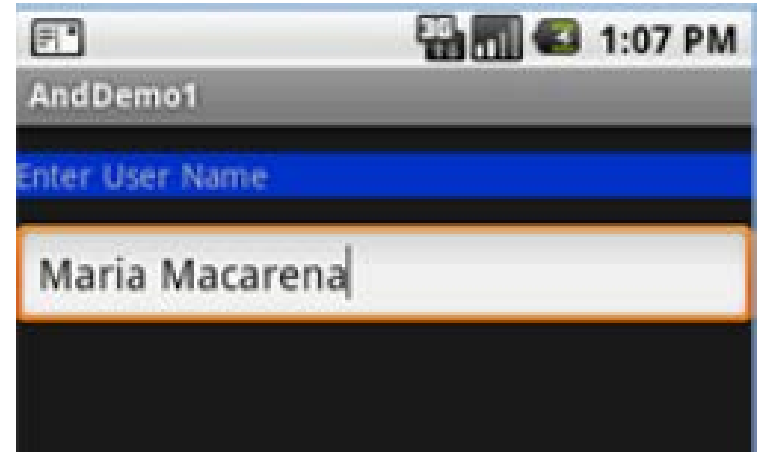
## 2. Các layout cơ bản

- Demo → Các widget



## 4. Basic Widget: EditText

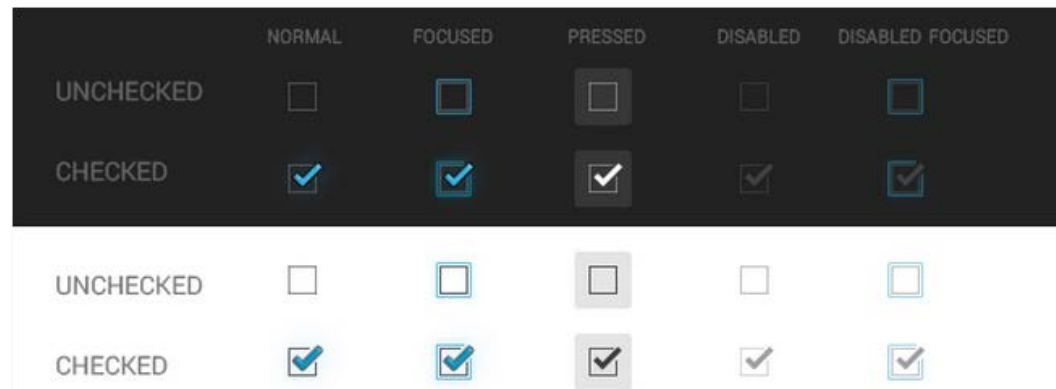
```
<EditText  
    android:id="@+id/txtUserName"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:textSize="18sp"  
    android:autoText="true"  
    android:capitalize="words"  
    android:hint="First Last Name"  
>  
</EditText>
```



- Lấy control theo Id:  
**EditText txtbox=(EditText) findViewById(R.id.editText1);**
- Thiết lập giá trị cho EditText  
**txtBox.setText("some text")**
- Lấy dữ liệu bên trong EditText:  
**String msg=txtBox.getText().toString()**

## 5. Basic Widget: CheckBox

- Checkboxes cho phép người sử dụng chọn một hoặc nhiều lựa chọn.
- Thẻ **checkbox** là một loại button đặc biệt có hai trạng thái : **checked** hoặc **unchecked**.
- CheckBox sử dụng chung 2 phương thức :
  - 1) Phương thức **setChecked**, dùng để thiết lập **checked**. Nếu ta gọi **setChecked(true)** tức là cho phép control **được checked**, còn gọi **setChecked(false)** thì control sẽ bị **unchecked**.
  - 2) Phương thức **isChecked**, kiểm tra xem control có được checked hay không. Nếu có checked thì trả về **true** ngược lại trả về **false**



## 5. Basic Widget: CheckBox

- Trong coding để kiểm tra xem CheckBox đó có được checked hay không thì làm như sau:

```
CheckBox chk = (CheckBox) findViewById(R.id.checkBox1);  
    If(chk.isChecked())  
    { xử lý checked}  
    Else  
    {xử lý unchecked}
```

- Để thiết lập checked:

```
chk.setChecked(true);
```

- Để loại bỏ checked

```
chk.setChecked(false);
```

## 5. Basic Widget: CheckBox

### Xử lý sự kiện Click

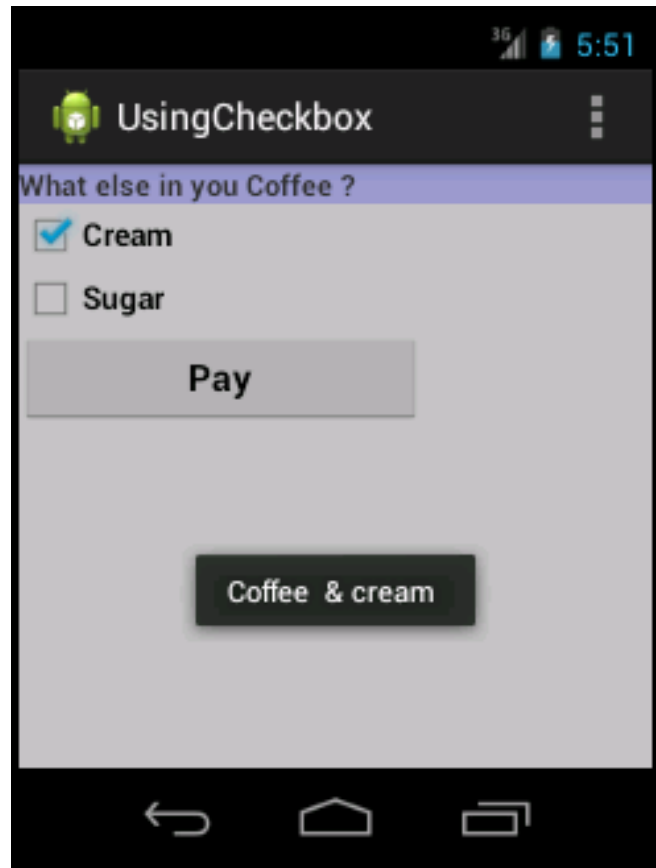
- Khai báo thuộc tính android:onClick trong xml
- Viết hàm có tên trùng với giá trị của thuộc tính onClick ở trên

```
<?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">
    <CheckBox android:id="@+id/checkbox_meat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/meat"
        android:onClick="onCheckboxClicked"/>
    <CheckBox android:id="@+id/checkbox_cheese"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/cheese"
        android:onClick="onCheckboxClicked"/>
</LinearLayout>
```

```
public void onCheckboxClicked(View view) {
    // Is the view now checked?
    boolean checked = ((CheckBox) view).isChecked();
    // Check which checkbox was clicked
    switch(view.getId()) {
        case R.id.checkbox_meat:
            if (checked)
                // Put some meat on the sandwich
            else
                // Remove the meat
                break;
        case R.id.checkbox_cheese:
            if (checked)
                // Cheese me
            else
                // I'm lactose intolerant
                break;
        // TODO: Veggie sandwich
    }
}
```

## 5. Basic Widget: CheckBox

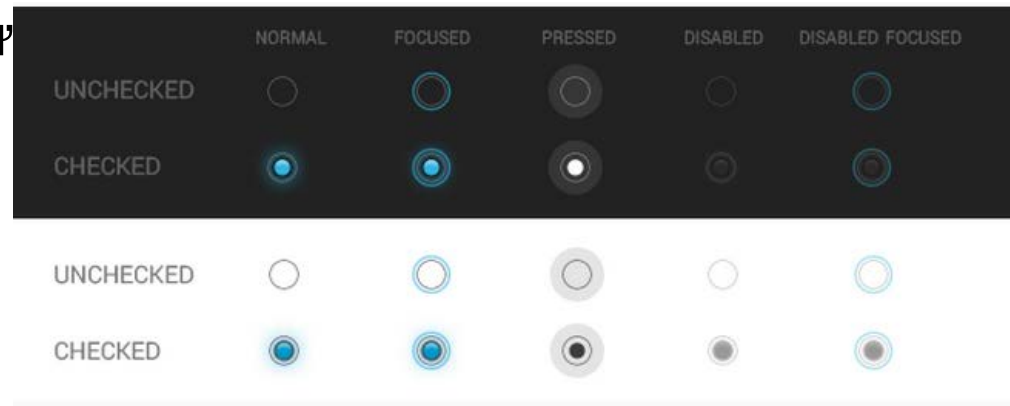
- Demo → CheckBox





## 6. Basic Widget: Radio Button

- Nút radio button cũng có hai trạng thái: **checked** hoặc **unchecked**.
- Để chọn 1 lựa chọn trong nhiều chọn lựa bạn đưa ra thì nên sử dụng RadioButton,
- Sử dụng **RadioGroup** để gom nhóm các RadioButton lại cùng một nhóm nào đó, những RadioButton mà cùng một nhóm thì tại 1 thời điểm chỉ có 1 RadioButton được checked mà thôi.
- RadioButton cũng kế thừa các thuộc tính của thẻ TextView như: font, style, color...
- Hàm **isChecked()** được sử dụng để kiểm tra trạng thái của nút radio button, tương tự với thẻ Checkbox



## 6. Basic Widget: RadioButton

- 2 cách xử lý tìm RadioButton được checked như sau:
  - **Cách 1**: Dựa vào RadioGroup để biết chính xác Id của RadioButton nào được checked. Dựa vào Id này ta sẽ xử lý:

```
RadioGroup group= (RadioGroup) findViewById(R.id.radioGroup1);
int idChecked = group.getCheckedRadioButtonId();
Switch (idCheckek)
{
Case R.id.radioButton1:
Break;
Case R.id.radioButton2:
Break;
...}
```
  - Như hình trên, tathấy hàm **getCheckedRadioButtonId()** : hàm này trả về Id của RadioButton nằm trong RadioGroup 1 được checked. Dựa vào Id này bạn so sánh để biết được trên giao diện người sử dụng đang checked lựa chọn nào.

## 6. Basic Widget: RadioButton

- **Cách 2:** Kiểm tra trực tiếp RadioButton đó có được checked hay không.

```
RadioButton rad=(RadioButton) findViewById(R.id.radio1);  
If(rad.isChecked())  
{  
}
```

- Để xóa bỏ checked trong group, ta dùng lệnh **clearChecked()**

## 6. Basic Widget: RadioButton

### Xử lý sự kiện Click

- Khai báo thuộc tính android:onClick trong xml
- Viết hàm có tên trùng với giá trị của thuộc tính onClick ở trên

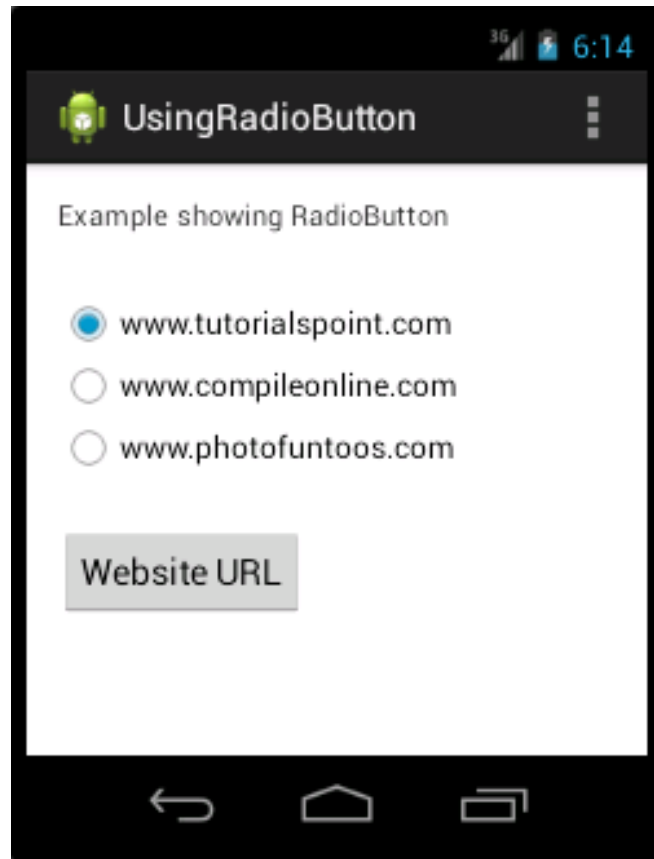
```
<?xml version="1.0" encoding="utf-8"?>
<RadioGroup
xmlns:android="http://schemas.android.com/apk/re
s/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <RadioButton android:id="@+id/radio_pirates"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/pirates"
        android:onClick="onRadioButtonClicked"/>
    <RadioButton android:id="@+id/radio_ninjas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ninjas"
        android:onClick="onRadioButtonClicked"/>
</RadioGroup>
```

```
public void onRadioButtonClicked(View view) {
    // Is the button now checked?
    boolean checked = ((RadioButton) view).isChecked();

    // Check which radio button was clicked
    switch(view.getId()) {
        case R.id.radio_pirates:
            if (checked)
                // Pirates are the best
                break;
        case R.id.radio_ninjas:
            if (checked)
                // Ninjas rule
                break;
    }
}
```

## 6. Basic Widget: RadioButton

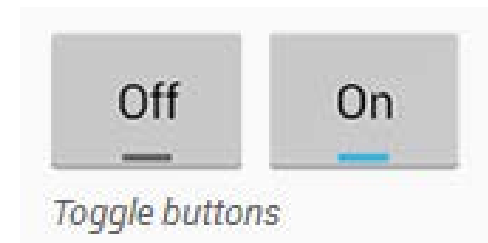
- Demo → RadioButton



## 7. Basic Widget: ToggleButton

- Toggle button cho phép người sử dụng thay đổi thiết lập giữa hai trạng thái.
- Sử dụng **OnCheckedChangeListener**

```
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
            } else {
                // The toggle is disabled
            }
        }
    });
```



## 7. Basic Widget: ToggleButton

### Xử lý sự kiện Click

- Khai báo thuộc tính android:onClick trong xml
- Viết hàm có tên trùng với giá trị của thuộc tính onClick ở trên

```
<ToggleButton
    android:id="@+id/togglebutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="Vibrate on"
    android:textOff="Vibrate off"
    android:onClick="onToggleClicked"/>
```

```
public void onToggleClicked(View view) {
    // Is the toggle on?
    boolean on = ((ToggleButton)
        view).isChecked();

    if (on) {
        // Enable vibrate
    } else {
        // Disable vibrate
    }
}
```

# Tổng kết bài học

## Tìm hiểu về các widget:

- TextView
- Buttons
- Images
- EditText
- Checkboxes
- RadioButtons
- ToggleButtons



**Kết thúc!**

