



# Lập trình Android

## Bài 1. *View & Các điều khiển cơ bản*

*Ngành Mạng & Thiết bị di động*





# Nội dung

---

## 1. Khái niệm View

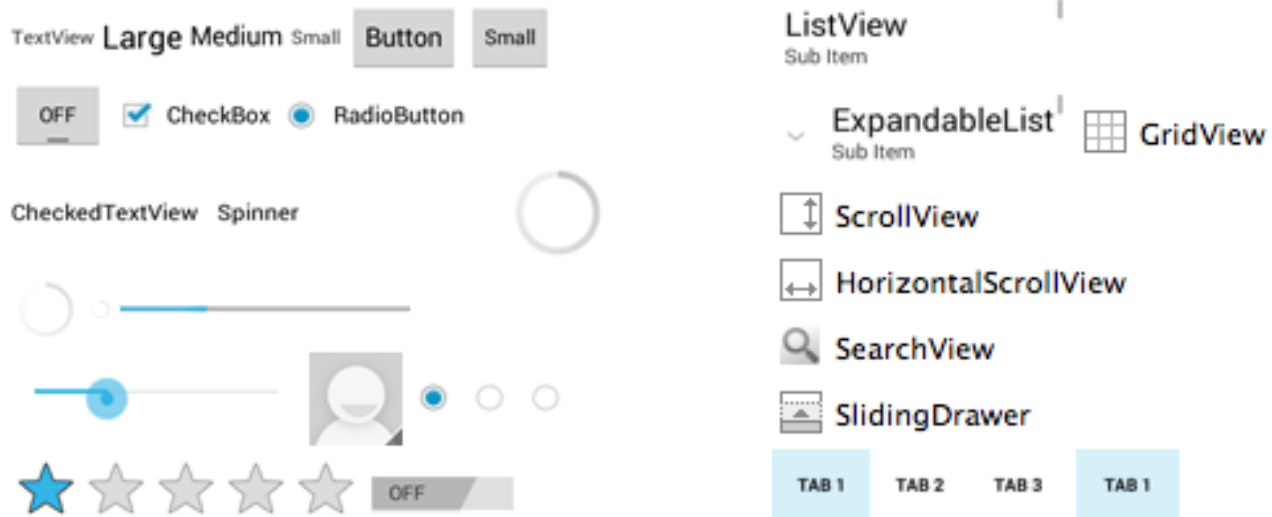
- View & View Group
- Thể hiện & Thao tác
- Thuộc tính

## 2. Các điều khiển cơ bản



# 1.1 View & ViewGroup

- ❑ View được sử dụng để tạo ra các điều khiển trên màn hình cho phép nhận các tương tác từ người dùng cũng như hiển thị các thông tin cần thiết.
- ❑ View bao gồm hai dạng:
  - View: các điều khiển đơn lẻ
  - ViewGroup: tập hợp nhiều điều khiển đơn lẻ





# 1.2 Thể hiện và Thao tác

---

## ❑ Thể hiện:

- Các đối tượng View được thể hiện trên màn hình giao diện như một hình chữ nhật tùy thuộc vị trí, kích thước, màu sắc và nhận vào cũng như xử lý các tương tác có liên quan.
- Một số thể hiện của lớp View: TextView, ImageView, SurfaceView...
- ViewGroup cũng là một thể hiện của View.

## ❑ Có thể xây dựng đối tượng View theo 2 cách:

- Kéo thả và tùy chỉnh thuộc tính trong XML.
- Thiết lập thông số và truy xuất trực tiếp trong Java Code.



# 1.2 Thể hiện và Thao tác

## ❑ Thao tác:

- Các đối tượng View được xây dựng và thiết lập với bốn thao tác chính:
  - Hiển thị nội dung thông qua phương thức `set<TT>(TS)`.
    - Ví dụ: TextView hiển thị văn bản, ImageView hiển thị hình ảnh...
  - Yêu cầu tương tác
    - Ví dụ: sử dụng `requestFocus` để yêu cầu tương tác với điều khiển.
  - Thiết lập chế độ hiển thị thông qua phương thức `setVisibility` (hoặc thuộc tính `visibility`: trong XML)
    - VISIBLE
    - INVISIBLE
    - GONE
  - Xây dựng phương thức “lắng nghe”
    - Ví dụ: bắt lại các sự kiện xảy ra trên điều khiển.



# 1.2 Thể hiện và Thao tác

## ❑ Thao tác:

- Một số sự kiện trên đối tượng View:

- OnClickListener
- onTouchListener
- OnLongClickListener
- OnDragListener
- OnKeyListener

- Ví dụ: lắng nghe sự kiện nhấn

```
view.setOnClickListener(new OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Log.i("HTSI", "onClick");  
    }  
});
```



# 1.3 Thuộc tính

---

## ❑ Id:

- Khai báo kiểu số nguyên **int**, đánh dấu vùng nhớ của đối tượng View.
- Id có thể giống nhau cho các điều khiển khác nhau trong cùng một tập tin giao diện.
- Phương thức thiết lập
  - `setId`
- Phương thức truy xuất
  - `getId`



# 1.3 Thuộc tính

- ❑ Thuộc tính Id được đi kèm với đối tượng View khi khai báo trong XML cho phép truy xuất trong Java Code khi cần.

- Ví dụ:

- Khai báo id trong XML

```
<Button
```

```
    android:id="@+id/my_button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/my_button_text"/>
```

- Truy xuất trong JavaCode

```
Button myBtn = (Button)findViewById(R.id.my_button);
```





# 1.3 Thuộc tính

---

- ❑ **Vị trí: cho biết tọa độ hiển thị cho View trên giao diện.**
  - Phương thức thiết lập
    - layout
    - setLeft
    - setTop
    - setRight
    - setBottom
  - Phương thức truy xuất:
    - getLeft
    - getTop
    - getRight
    - getBottom
  
- ❑ **Vị trí của View tùy thuộc vào thuộc tính của đối tượng Layout.**





## 1.3 Thuộc tính

---

- ❑ **Kích thước: bao gồm chiều ngang và chiều cao của một đối tượng View.**
  - Kích thước của đối tượng View có thể thiết lập qua 3 thông số:
    - `WRAP_CONTENT`
    - `MATCH_PARENT` (API 8 trở lên)
    - `FILL_PARENT`
    - Một con số bất kỳ (tính theo dp/px/dip).



# 1.3 Thuộc tính

---

- ❑ **Kích thước: bao gồm chiều ngang và chiều cao của một đối tượng View.**
  - Phương thức thiết lập trong Java Code:
    - Thiết lập thông qua đối tượng LayoutParams
  - Thuộc tính thiết lập trong XML:
    - `layout_width`
    - `layout_height`
  - Phương thức truy xuất:
    - `getWidth`
    - `getHeight`
    - `getMeasuredWidth`
    - `getMeasureHeight`



## 1.3 Thuộc tính

---

### ☐ Canh lề nội dung trong JavaCode:

- Phương thức thiết lập:

- `setPadding`

- Phương thức truy xuất:

- `getPaddingTop`

- `getPaddingLeft`

- `getPaddingRight`

- `getPaddingBottom`



# Nội dung

---

## 1. Khái niệm View

## 2. Các điều khiển cơ bản

- TextView
- Button



## 2.1 TextView

---

### ❑ Đối tượng 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("Đối tượng TextView");`
- Trong XML:
  - `android:text="Đối tượng TextView"`



## 2.1 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"`



## 2.1 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"`





## 2.1 TextView

---

❑ Một số phương thức quan trọng:

- `setTextColor` – `android:textColor`
- `setTextSize` – `android:textSize`
- `setTypeFace` – `android:typeface`



## 2.2 Button

---

- ❑ **Đối tượng Button được xây dựng từ TextView, cho phép thể hiện các 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



## 2.2 Button

### ❑ Button:

- Lắng nghe sự kiện nhấn trong Java-Code:

```
button.setOnClickListener(new OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Log.i("HTSI", "onClick");  
    }  
});
```

- Lắng nghe sự kiện nhấn trong XML:

```
android:onClick="tenPhuongThuc"
```



## 2.2 Button

### ❑ **CompoundButton:**

- Checkbox: đối tượng nút bấm hai trạng thái “được chọn” và “bỏ chọn”.
- Phương thức lắng nghe sự kiện thay đổi trạng thái:

```
checkBox.setOnCheckedChangeListener(new OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChangeListener(CompoundButton v, boolean isChecked) {  
        Log.i("HTSI", "onChecked");  
    }  
});
```



## 2.2 Button

### ❑ CompoundButton:

- RadioButton: đối tượng nút bấm hai trạng thái “được chọn” và “bỏ chọn”, không thể “bỏ chọn” khi đã “được chọn”.
- Thường xử lý trên nhóm nút nhiều trạng thái.
- Phương thức lắng nghe sự kiện thay đổi trạng thái trên nhóm nút:

```
radioGroup.setOnCheckedChangeListener(new OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChangeListener(RadioGroup group, int checkedId) {  
        Log.i("HTSI", "Checked at id:" + checkedId);  
    }  
});
```



## 2.2 Button

### ❑ **CompoundButton:**

- **ToggleButton:** đối tượng nút bấm hai trạng thái “bật” và “tắt”, thể hiện trạng thái trên đối tượng.

- Thuộc tính quan trọng:

- `textOn`: trạng thái nút đang bật
- `textOff`: trạng thái nút đang tắt

- Phương thức lắng nghe sự kiện thay đổi trạng thái trên nhóm nút:

```
toggleButton.setOnCheckedChangeListener(new OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChangeListener(CompoundButton v, boolean isChecked) {  
        Log.i("HTSI", "onChecked");  
    }  
});
```



## 2.2 Button

### ❑ CompoundButton:

- Switch: đối tượng nút bấm hai trạng thái “bật” và “tắt”, có thể thao tác bằng cách trượt ngón tay trên đối tượng.
- Thuộc tính quan trọng:
  - `textOn`: trạng thái nút đang bật
  - `textOff`: trạng thái nút đang tắt
- Phương thức lắng nghe sự kiện thay đổi trạng thái trên nhóm nút:

```
switchButton.setOnCheckedChangeListener(new OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChangeListener(CompoundButton v, boolean isChecked) {  
        Log.i("HTSI", "onChecked");  
    }  
});
```

