Android Resources

Tran Giang Son, tran-giang.son@usth.edu.vn

ICT Department, USTH

Resources

- Things that are embedded (bundled) into the app
- Resources in res/ directory
- Several resource categories
- Accessible through code: `R.<category>.<resourceName>`
- Do NOT hard-code values inside codes

Contents

- Values
- Layouts
- Drawables
- Raw
- Styles, Design Guidelines

Layouts

Layouts

- **Remind**
- **Definition**
- **Layout in XML**
- Popular Layout classes

Remind

- Remind: HTML
- Old-school: `<div>` and `<table>`
- HTML5: `<header>`, `<nav>`, `<section>`, `<aside>`, `<footer>`
- CSS float and padding

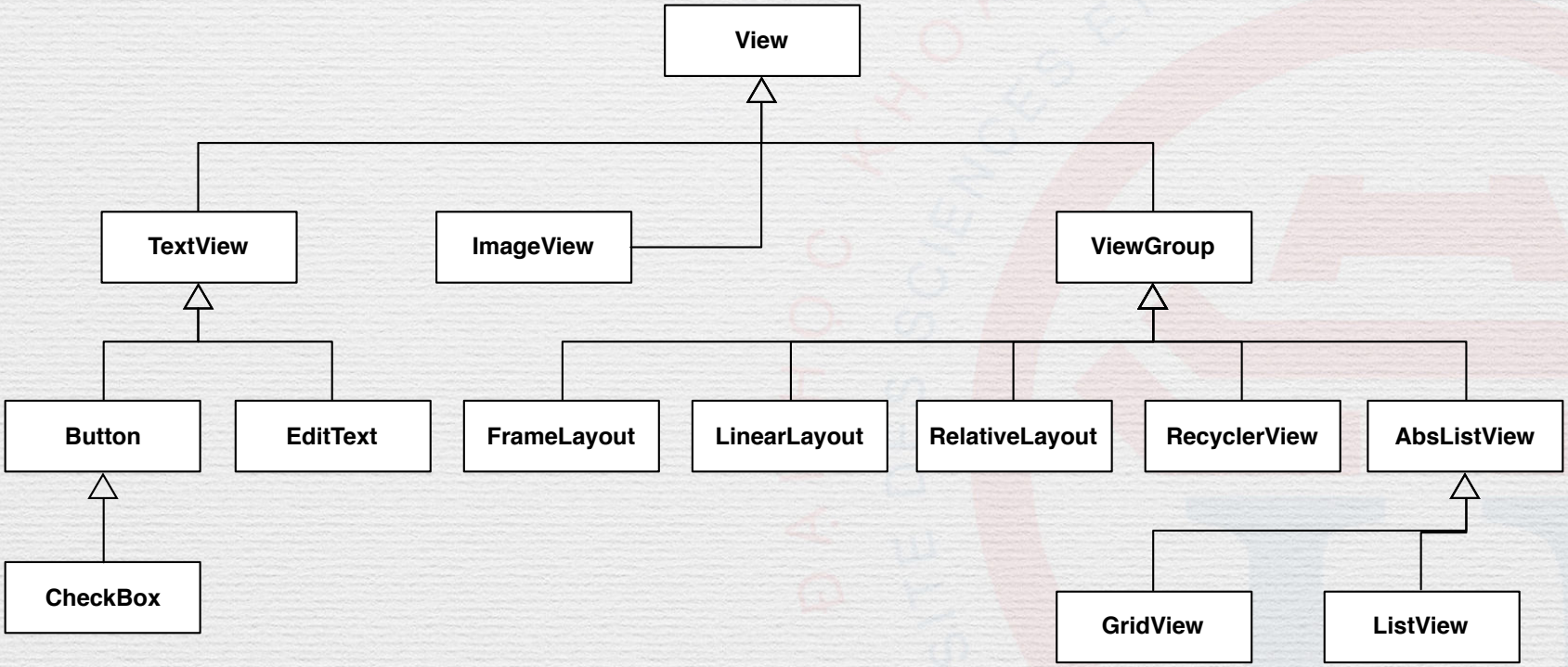
Remind

- Remind: what's the base View class for layouting in android?

Remind

- Remind: what's the base View class for layouting in android?
- ViewGroup
- FrameLayout, LinearLayout, RelativeLayout

Remind


View
TextView
ImageView
ViewGroup
Button
EditText
FrameLayout
LinearLayout
RelativeLayout
RecyclerView
AbsListView
CheckBox
GridView
ListView

Layout

- A way to organize Views inside an UI
- Can be created by code (see Practical Work #4)
- XML files in **res/layout**
- Hierarchical "structure" of one UI
- Can be nested
- WYSIWYG or manual editor

Layout XML

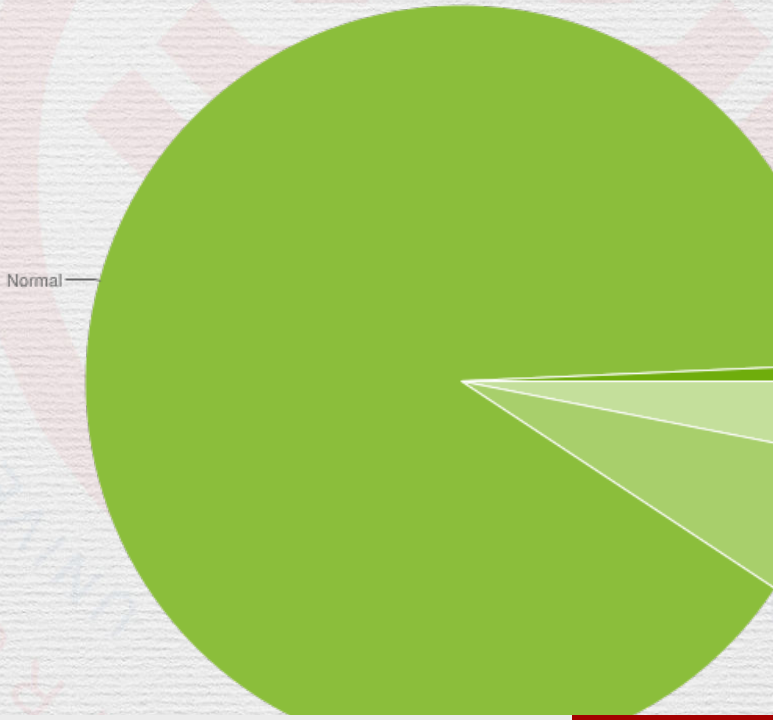
- Containers (ViewGroups) contain Views (TextView, ImageView, EditText, Button, ImageButton...)
- Required: layout_width, layout_height
- Optional: id (for later `findViewById()`)

Layout XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dip">

    <TextView
        android:id="@+id/question"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Is Android a Linux Distribution?"
        android:textAppearance=
            "?android:attr/textAppearanceMedium" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="True"
            android:id="@+id/btnTrue" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="False"
            android:id="@+id/btnFalse" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Next"
            android:id="@+id/btnNext" />
    </LinearLayout>
</LinearLayout>
```

«Adaptive» Layout on Android

- Use different layout XMLs in different directories
- Tablet: layout-large, layout-xlarge
- Phone: layout-normal
- Small: layout-small
- Orientation: -land, -port
- Examples



Layout XML

- How to load XML layout?
  - Activity: in `onCreate()`, with `setContentView()`
  - Fragment: in `onCreateView()`

```
// Activity
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_layout);
}

// Fragment
public View onCreateView(LayoutInflater inflater, ViewGroup container,...) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_layout, container, false);
}
```

Layouts

- Remind
- Definition
- Layout in XML
- **Popular Layout classes**

Popular Layout Classes

- FrameLayout
- LinearLayout
- RelativeLayout
- ViewPager

FrameLayout

- Can contain multiple children (Views)
- Multiple layers, Z-based order: like a Photoshop design
- First child will be at the bottom
- Support child margins
- Support gravity

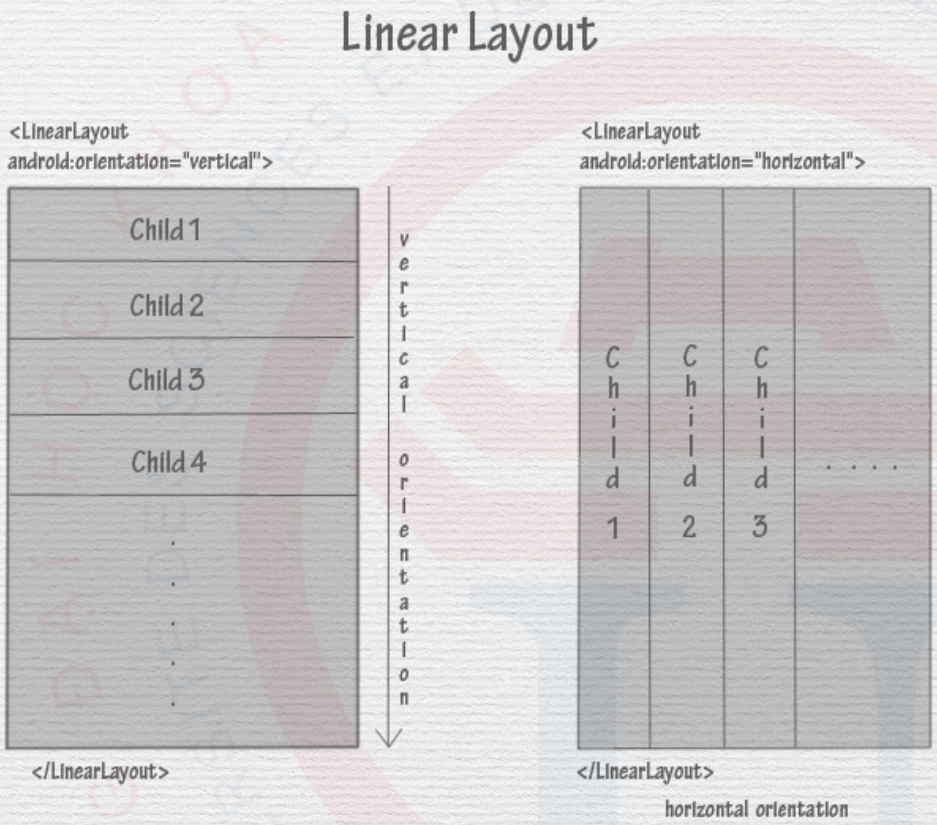
FrameLayout

```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:adjustViewBounds="true"
        android:background="#20FF0000"
        android:src="@drawable/usth" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#FF00FF00"
        android:text="Is USTH awesome?" />
</FrameLayout>
```

LinearLayout

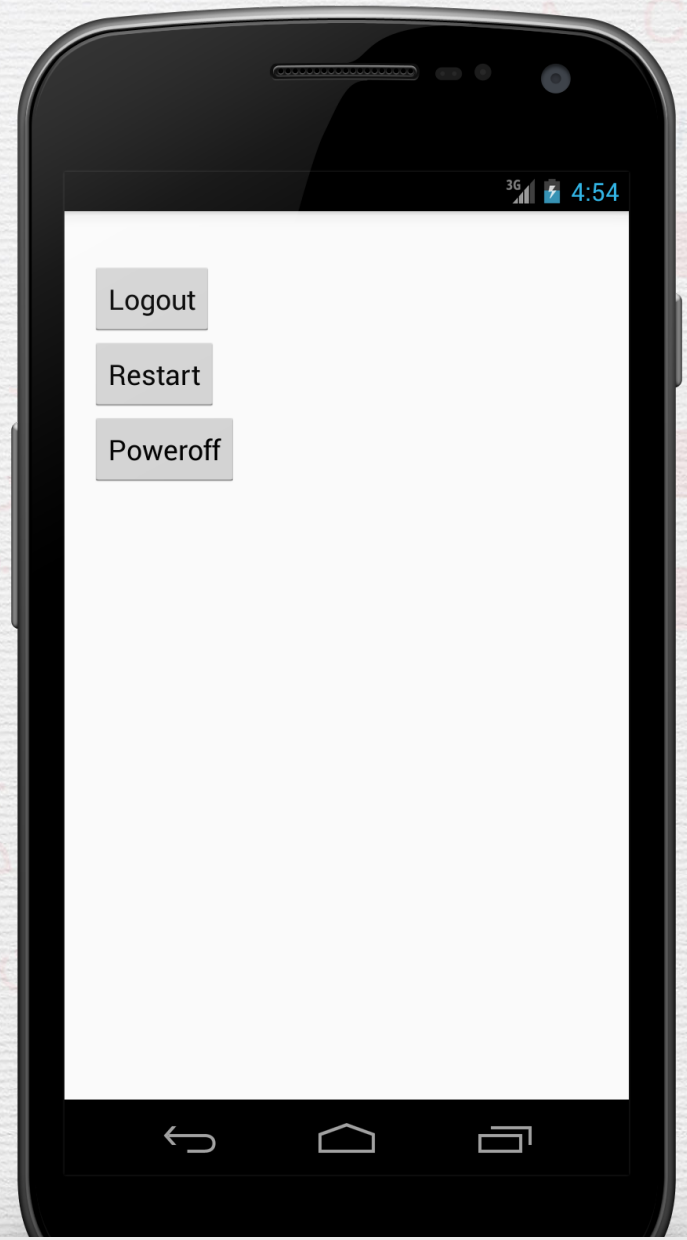
- One direction
- Horizontal or Vertical

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Logout" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Restart" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Poweroff" />
</LinearLayout>
```

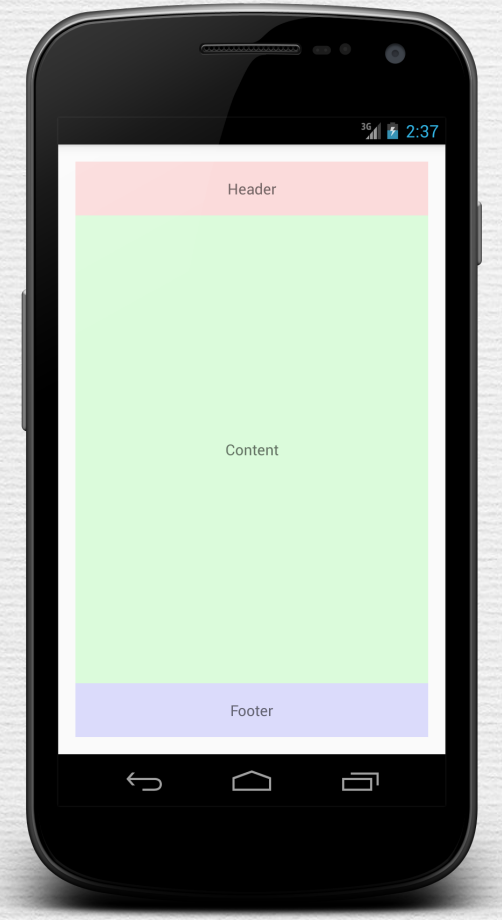


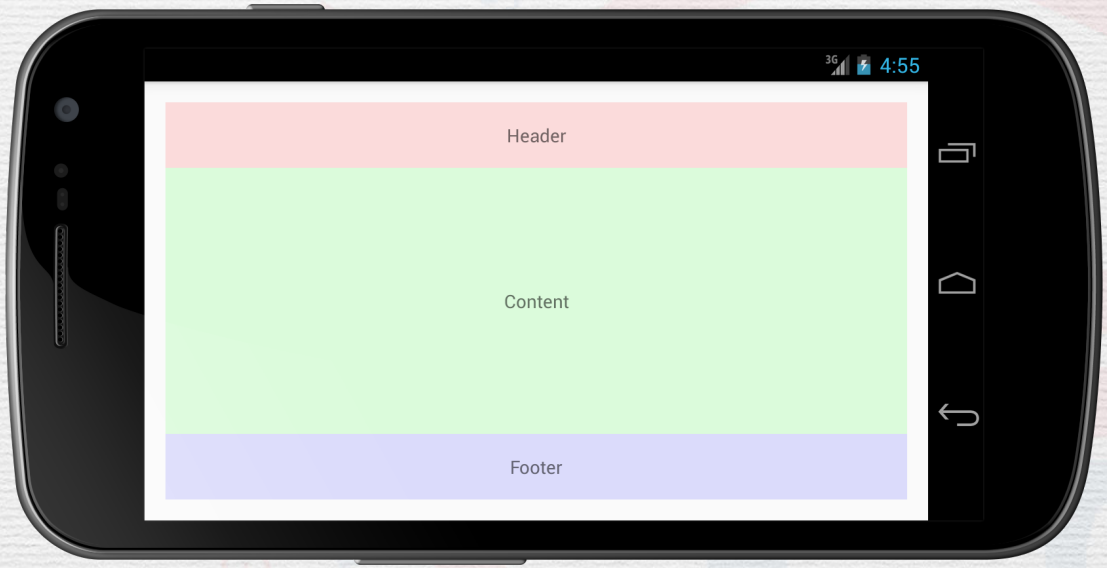
LinearLayout

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Logout" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Restart" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Poweroff" />
</LinearLayout>
```



LinearLayout





LinearLayout Stretching

- Use layout_weight
- Based on orientation
  - horizontal: stretch width
  - vertical: stretch height
- no layout_weight: no stretch
- width/height ω_i is calculated based on weight γ_i of child i as

$$\omega_i = \frac{\gamma_i}{\sum_{j=0}^{n-1} \gamma_j} \times \left(\omega_{parent} - \sum_{k=0}^{n-1} \omega_k | \gamma_k = 0\right)$$

LinearLayout: Exercise

```
<LinearLayout
    android:layout_width="720px"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:id="@+id/button"
        android:layout_width="0px"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Logout" />

    <Button
        android:id="@+id/button2"
        android:layout_width="100px"
        android:layout_height="wrap_content"
        android:text="Restart" />

    <Button
        android:id="@+id/button3"
        android:layout_width="0px"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Poweroff" />
</LinearLayout>
```

What's the width value of each child in this layout?

$$\omega_i = \frac{\gamma_i}{\sum_{j=0}^{n-1} \gamma_j} \times \left(\omega_{parent} - \sum_{k=0}^{n-1} \omega_k | \gamma_k = 0\right)$$

LinearLayout: Exercise

```
<LinearLayout
    android:id="@+id/container"
    android:layout_width="720px"
    android:layout_height="48px"
    android:orientation="horizontal"
    android:padding="4px">

    <View
        android:layout_width="0px"
        android:layout_height="1px"
        android:layout_weight="1" />

    <TextView
        android:id="@+id/item1"
        android:layout_width="100px"
        android:layout_height="match_parent"
        android:paddingLeft="8px"
        android:paddingRight="8px" />

    <View
        android:layout_width="1px"
        android:layout_height="match_parent"
        android:layout_marginLeft="8px"
        android:layout_marginRight="8px"
        android:background="@drawable/divider" />

    <TextView
        android:id="@+id/item2"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:paddingLeft="16px"
        android:paddingRight="16px" />

    <View
        android:layout_width="1px"
        android:layout_height="match_parent"
        android:layout_marginLeft="8px"
        android:layout_marginRight="8px"
        android:background="@drawable/divider" />

    <TextView
        android:id="@+id/item3"
        android:layout_width="120px"
        android:layout_height="match_parent"
        android:paddingLeft="8px"
        android:paddingRight="8px" />

    <View
        android:layout_width="0px"
        android:layout_height="1px"
        android:layout_weight="1" />
</LinearLayout>
```

$$\omega_i = \frac{\gamma_i}{\sum_{j=0}^{n-1} \gamma_j} \times (\omega_{parent} - \sum_{k=0}^{n-1} \omega_k | \gamma_k = 0)$$

Practical Work 5

- Modify your ForecastFragment layout
- Use LinearLayout to have the blue forecast area