

# Thiết kế giao diện trên Android

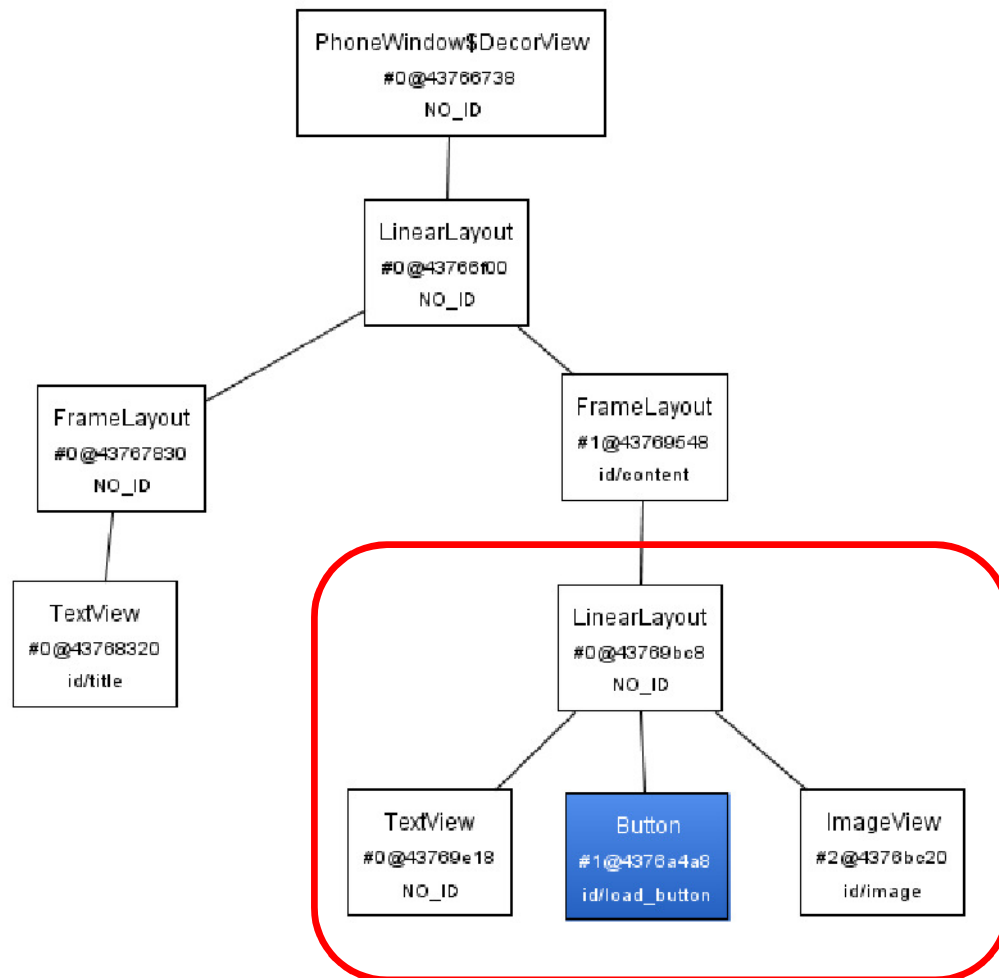
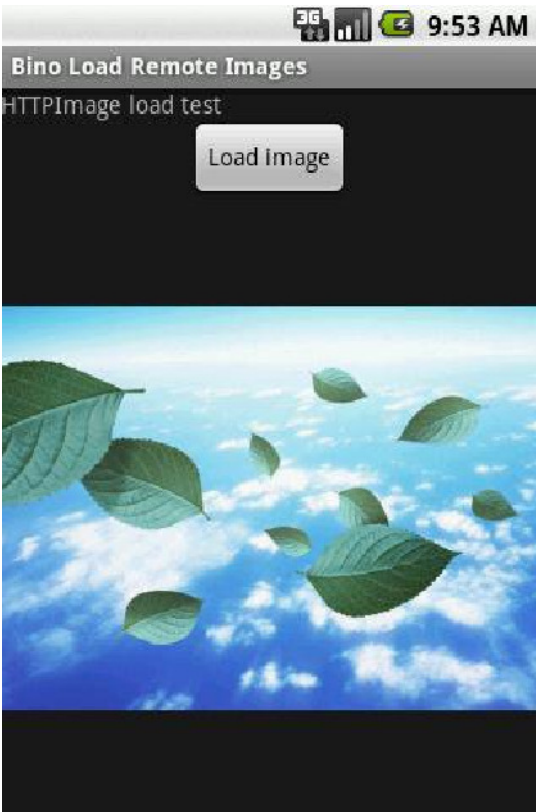
MultiUni

Trần Vũ Tất Bình

# Tổng quan

- Trong Android, dùng Activity để hiển thị màn hình.
- Mỗi activity sẽ chứa các View theo dạng cấu trúc cây, nghĩa là một Layout gốc chứa các view/layout con bên trong hoặc chỉ có 1 view duy nhất. (lưu ý Layout cũng là một view nhé)
- Có thể thiết kế giao diện trong code java hoặc trong file xml trong thư mục layout.

# Tree view



# Layout mẫu của helloworld

```
<?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">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>
```

# Một số thuộc tính cơ bản

- **Layout\_width, layout\_height**: chiều rộng của view (fill\_parent là to bằng kích thước của layout chứa view này, wrap\_content là vừa đủ nội dung cần hiển thị của view)
- **Orientation**: với LinearLayout, việc sắp xếp các view là nằm kề nhau theo hàng ngang hoặc hàng dọc, ta khai báo orientation để chọn sắp theo kiểu nào (horizontal/vertical)

# Một số thuộc tính cơ bản

- Gravity: thuộc tính này qui định các view nằm bên trong layout sẽ đặt theo vị trí nào so với layout(trung tâm, trái , phải, trên dưới...)
- Weight: để các view phân chia tỉ lệ diện tích hiển thị trên màn hình (tỉ lệ tính theo weight của từng view trên tổng số weight, các view ko khai báo weight thì sẽ xem qua width và height)

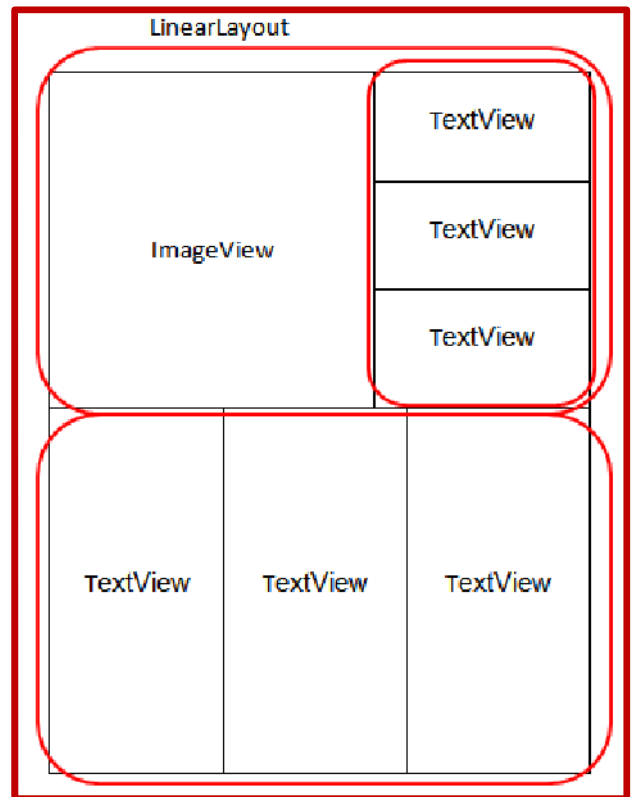
# Ví dụ cơ bản

- Các bạn vào link này lấy code cho vào ứng dụng, lưu ý đọc thêm các dòng giải thích tiếng Anh nhé

<http://developer.android.com/resources/tutorials/views/hello-linearlayout.html>

# Giao diện với LinearLayout

- Giả sử bạn cần thiết kế một màn hình như sau:



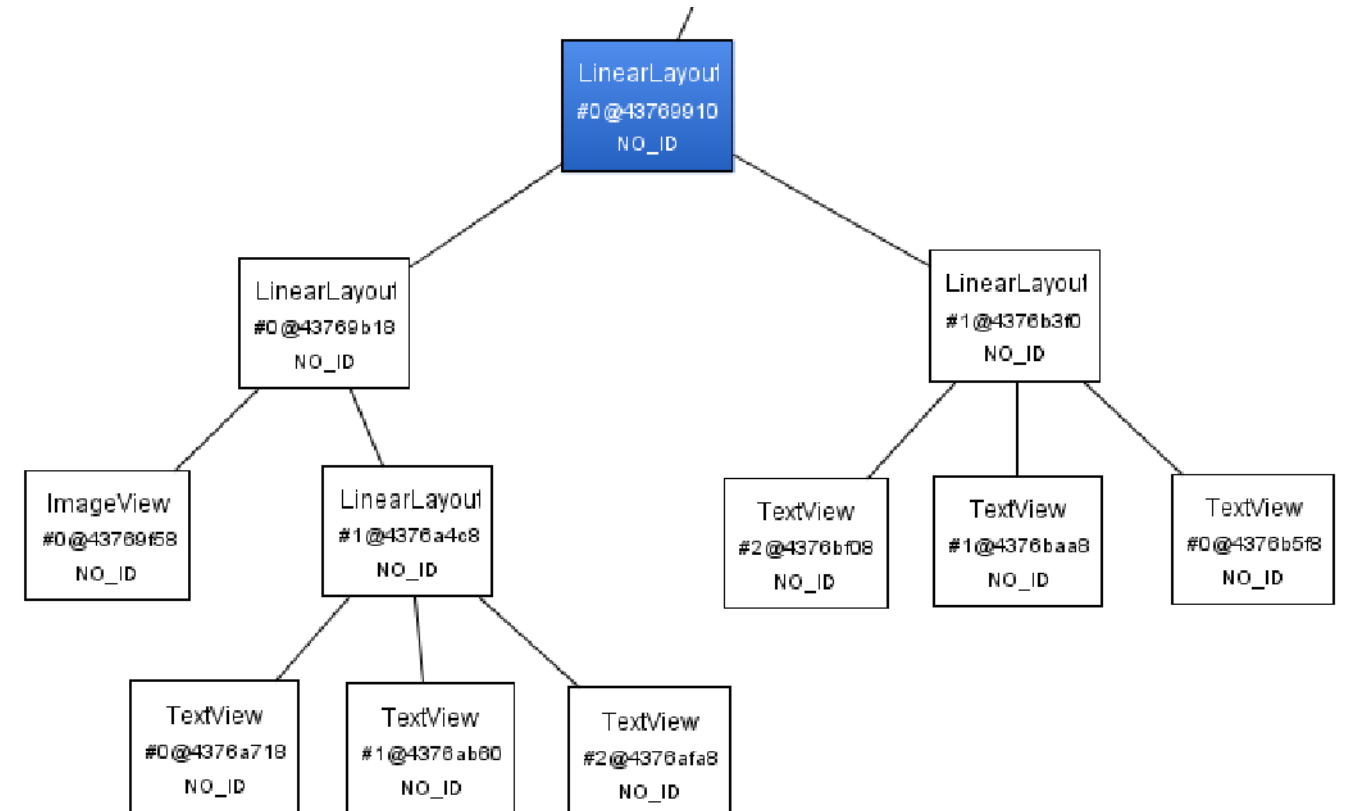


# Giao diện với LinearLayout

- Với LinearLayout (LL), các view bên trong nó được đặt kề nhau theo hàng ngang hoặc hàng dọc (cần lưu ý đặc điểm này)
- Với ví dụ vừa rồi, ta thấy cách phân tích như sau:
  - Nguyên tắc chủ yếu là phân nhóm các View liên tiếp kề nhau (như 3 TextView kề nhau theo hàng dọc hoặc hàng ngang trong ví dụ trên) vào trong một LL, phân rã từ lớn đến nhỏ.
  - Như vậy màn hình gồm 1 LL lớn bao bên ngoài, nhìn thấy bên trong chia thành 2 phần trên dưới rõ ràng vậy thuộc tính của LL này là dạng dọc, sau đó chia đôi ra và phân tích tiếp.
  - Phần bên trên lại chia thành 2 nửa theo hàng ngang → là một LL dạng ngang, lại chia đôi: một bên là 1 ImageView (vì chỉ có 1 view nên ko cần bỏ vào trong LL), một bên lại là 1 LL chứa 3 TextView theo hàng dọc.
  - Nửa bên dưới ta thấy rõ ràng chứa 3 TextView kề nhau theo hàng ngang → cho vào 1 LL dạng ngang là xong. 😊

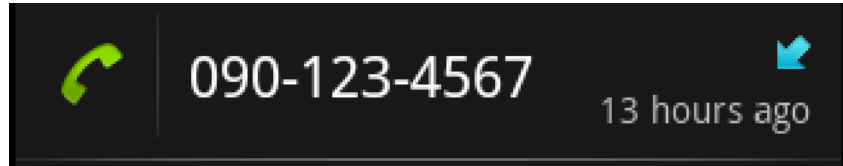
# Giao diện với LinearLayout

- Xem cây:



# Thực hành tại chỗ

- Làm một layout hiển thị như trong hình:



- Các bước như sau:
  - Phân tích thành phần layout trên giấy (thảo luận)
  - Phân tích đặc điểm các view
  - Add thêm resource ảnh
  - Thử trước với với tab layout (khi view file xml trong eclipse)
  - Đưa vào thực thi trên máy, đánh giá.

# Một số loại layout khác

- `FrameLayout`: các view bên trong được qui định vị trí bằng khoảng cách so với biên trái và trên so với layout, các view có thể đè lên nhau.
- `RelativeLayout`: các view được thiết kế dựa trên quan hệ giữa chúng với nhau và với layout chứa chúng.
- `AbsoluteLayout`: dành cho bạn nào làm nhiều với C#, nhưng ko khuyến khích với Android nhé.

...

# Lưu ý khi thiết kế giao diện

- Hạn chế độ sâu của cây
- Với các Layout phức tạp, đừng dùng RelativeLayout
- Nên chèn vào dữ liệu tạm để xem trước layout hiển thị ra sao bên tab layout (trong eclipse), nhưng xong rồi thì nhớ xóa dữ liệu tạm đi.
- FrameLayout có vấn đề với background
- Muốn tìm thuộc tính gì, bấm “android:” rồi đợi suggestion sô ra xem.

# Code nhanh một số ví dụ

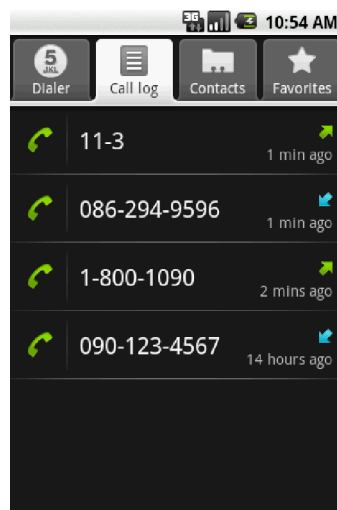
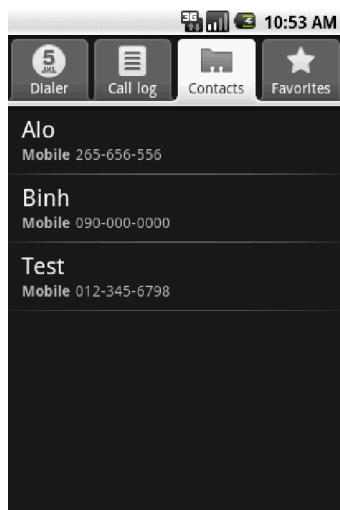
- Thử cùng nhau ngồi code nhanh nhé (khi đã hiểu rồi thì làm sẽ rất nhanh thôi)

<http://developer.android.com/resources/tutorials/views/hello-formstuff.html>

<http://developer.android.com/resources/tutorials/views/hello-webview.html>

# List

- Rất hay dùng trong Android. Đặc biệt các ứng dụng cần lưu trữ và hiển thị nhiều dữ liệu.
- List là một danh sách các view thông thường có cùng dạng layout đặt liên nhau.



# ApiDemos

- Mở ứng dụng ApiDemos đã có sẵn:
  - New android project → Create project from existing source → Browse → mở thư mục SDK → platforms → android-1.6 → samples → ApiDemos → OK → Finish.
  - Mở thư mục project trong eclipse → src → mở com.example.android.apis.view → mở file List1.java



# List1

- Trong code của List1, các bạn tách bạch phần tạo mới ArrayAdapter ra nhé

```
ArrayAdapter adapter = new ArrayAdapter<String>(this,  
        android.R.layout.simple_list_item_1, mStrings);  
setListAdapter(adapter);
```

- Adapter dùng để kết nối dữ liệu cần đưa vào list với layout hiển thị và cung cấp cho ListView hiển thị.
- Sau khi tạo xong adapter, truyền vào cho list hiển thị (dòng code bên dưới)

# List1

- Với đoạn tạo một adapter, ta phân tích ra:
  - **this**: context, lúc nào làm view cũng cần rồi
  - **android.R.layout.simple\_list\_item\_1**: layout để hiển thị mỗi item trong list, ở đây chỉ có một TextView
  - **mStrings**: mảng các chuỗi kí tự, là dữ liệu cần hiển thị
- Giải thích đoạn tạo adapter như sau: tạo một adapter mới với context của activity hiện tại, dùng layout **simple\_list\_item\_1** để hiển thị mỗi item trong list, dữ liệu đưa vào lấy từ mảng các chuỗi **mStrings**

# List1

- Tạo một project mới, copy code List1 bỏ qua, rồi làm thêm các thao tác sau:
  - Tạo 1 file layout có 1 TextView và 1 ImageView
  - Trong TextView thêm dòng này  
`android:id="@+id/text"`
  - Sửa lại đoạn tạo mới adapter cho nó dùng cái layout vừa tạo. (tìm xem có cái constructor nào giúp chỉ chính xác đến cái TextView trong layout hay ko)  
Dạng này: [ArrayAdapter](#)(Context context, int resource, int textViewResourceId, T[] objects)
  - Chạy ứng dụng

# List1

- Giải thích:
  - Với code mẫu ban đầu, layout chỉ có một view duy nhất là TextView nên bạn ko cần khai báo gì khác ngoài layout và đưa dữ liệu dạng mảng String vào, adapter sẽ tự hiểu và xử lý.
  - Với yêu cầu sau này, layout của bạ có nhiều hơn 1 view, bạn cần truyền thêm id của TextView vào (ở đây là **R.id.text**) để adapter có thể hiểu bạn muốn dữ liệu của bạn được truyền vào view nào trong layout.

# List14

- Mở List14 ra, đọc và chạy thử.

Với mục đích giúp cho các bạn không đến tham dự lớp vẫn có thể hiểu được bài giảng kỹ càng nên trong slide sẽ có nhiều đoạn chú thích hơi dài.

