**I. Layout là gì?**

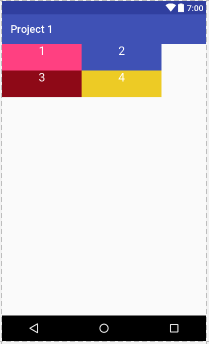
Nếu bạn nào đã từng đã từng học thiết web thi hẳn đã biết muốn xây dựng một giao diện website thì ta sẽ phải dựng các layout bố cục cho trang web đó và Android cũng vậy ta cũng phải dùng các layout được Android cung cấp sẵn để chứa đựng và sắp xếp bố cục các view để tạo nên giao diện cho ứng dụng Android.

**II. Các loại Layout.**

Vậy Android có những loại layout nào. Android hỗ trợ 6 loại layout là RelativeLayout, LinearLayout,GridLayout, TableLayout, FrameLayout, ConstraintLayout.

A. RelativeLayout.

RelativeLayout là loại Layout mà trong Layout vị trí của mỗi view con sẽ được xác định so với view khác hoặc so với thành phần cha của chúng thông qua ID. Bạn có thể sắp xếp 1 view ở bên trái, bên phải view khác hoặc ở giữa màn hình. Ví dụ tôi có một màn hình như sau.

**

*Ví dụ về Relativelayout*

và đây là code của ví dụ trên.

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

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

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.tuandc.project1.project1.MainActivity">

<TextView

android:layout\_width="150dp"

android:layout\_height="50dp"

android:text="1"

android:textColor="#fff"

android:id="@+id/hop1"

android:textSize="10pt"

android:textAlignment="center"

android:background="@color/colorAccent"

/>

<TextView

android:layout\_width="150dp"

android:layout\_height="50dp"

android:text="2"

android:textColor="#fff"

android:id="@+id/hop2"

android:textSize="10pt"

android:textAlignment="center"

android:layout\_toRightOf="@+id/hop1"

android:background="@color/colorPrimary"

/>

<TextView

android:layout\_width="150dp"

android:layout\_height="50dp"

android:text="3"

android:textColor="#fff"

android:id="@+id/hop3"

android:background="#8e0917"

android:textSize="10pt"

android:textAlignment="center"

android:layout\_below="@+id/hop1"

/>

<TextView

android:layout\_width="150dp"

android:layout\_height="50dp"

android:text="4"

android:textColor="#fff"

android:id="@+id/hop4"

android:textSize="10pt"

android:textAlignment="center"

android:background="#edcb25"

android:layout\_below="@+id/hop2"

android:layout\_toRightOf="@+id/hop3"

/>

</RelativeLayout>

Như bạn thấy trong ví dụ trên tôi có 4 cái hộp với màu sắc khác nhau được sắp xếp cạnh nhau và mỗi hộp đều được gán id từ hop1->hop4 như bạn thấy hộp 2 ở bên phải hộp 1 nên tôi dùng thuộc tính android:layout\_toRightOf="@+id/hop1" để đưa hộp 2 về bên phải hộp 1 và bạn thấy hộp 4 nằm bên phải hộp 3 và dưới hộp 2 nên tôi sử dụng hai thuộc tính

android:layout\_below="@+id/hop2" android:layout\_toRightOf="@+id/hop3"

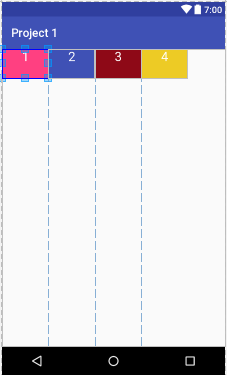
để đưa hộp 4 về bên trái hộp 3 và dưới hộp hai. Bạn thấy trong thuộc tính android:layout\_toRightOf có giá trị là 1 id của một view vì Relative Layout sử dụng ID để xác định các vị trí của view với nhau mà.

Như vậy là tôi đã giới thiệu xong Relative Layout.

B. LinearLayout

LinearLayout là loại layout sẽ sắp xếp các view theo chiều dọc hoặc ngang theo thứ tự của các view.

**Ví dụ**: LinearLayout nằm ngang

**

*LinearLayout horizontal*

**Code**

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

<LinearLayout

xmlns:android1="http://schemas.android.com/apk/res/android"

android1:layout\_width="match\_parent"

android1:layout\_height="match\_parent"

android1:orientation="horizontal"

>

<TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:text="1"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textAlignment="center"

android1:textStyle="bold"

android1:background="@color/colorAccent"

/><TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:text="1"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textAlignment="center"

android1:textStyle="bold"

android1:background="@color/colorPrimary"

/><TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:text="3"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textAlignment="center"

android1:textStyle="bold"

android1:background="#8c0520"

/>

<TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:background="#efcd21"

android1:text="4"

android1:textAlignment="center"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textStyle="bold" />

</LinearLayout>

Như bạn thấy các view sẽ được sắp sếp theo chiều ngang bởi thuộc tính android:orientation="horizontal" của Linearlayout là chiều ngang và các view được sắp xếp theo đúng thứ tự view trước đứng trước từ trái qua phải.

**Ví dụ**: LinearLayout nằm dọc

**

*relative layout vertical*

**Code:**

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

<LinearLayout

xmlns:android1="http://schemas.android.com/apk/res/android"

android1:layout\_width="match\_parent"

android1:layout\_height="match\_parent"

android1:orientation="vertical"

>

<TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:text="1"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textAlignment="center"

android1:textStyle="bold"

android1:background="@color/colorAccent"

/><TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:text="1"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textAlignment="center"

android1:textStyle="bold"

android1:background="@color/colorPrimary"

/><TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:text="3"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textAlignment="center"

android1:textStyle="bold"

android1:background="#8c0520"

/>

<TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:background="#efcd21"

android1:text="4"

android1:textAlignment="center"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textStyle="bold" />

</LinearLayout>

Như các bạn thấy các view trong LinearLayout sẽ được hiển thị theo chiều dọc từ trên xuống dưới.

Vậy là tôi đã giới thiệu với các bạn về LinearLayout.

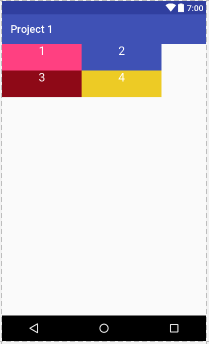
C. GridLayout

Bạn biết framework bootstrap nổi tiếng chứ.

*Bootstrap*

Thiết kế của Framework này là chia màn hình thành 12 cột và các dòng và gọi đây là giao diện dạng lưới hay để dẽ hiều hơn thì bạn hãy liên tưởng đến các lưới đánh cá của các ngư dân ấy. GridLayout của Android cũng tương tự chỉ là 1 layout dạng lưới và ta có thể chia các cột và dòng cho cái lưới đó, các view sẽ được dặt vào các ô trong cái lưới này.

****Ví dụ**

**

*Ví dụ về GridLayout*

**Code**

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

<GridLayout

xmlns:android1="http://schemas.android.com/apk/res/android"

android1:layout\_width="match\_parent"

android1:layout\_height="match\_parent"

android1:columnCount="2"

android1:rowCount="2"

>

<TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:text="1"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textAlignment="center"

android1:textStyle="bold"

android1:background="@color/colorAccent"

/><TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:text="2"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textAlignment="center"

android1:textStyle="bold"

android1:background="@color/colorPrimary"

/><TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:text="3"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textAlignment="center"

android1:textStyle="bold"

android1:background="#8c0520"

/>

<TextView

android1:layout\_width="80dp"

android1:layout\_height="80dp"

android1:background="#efcd21"

android1:text="4"

android1:textAlignment="center"

android1:textColor="#fff"

android1:textSize="15pt"

android1:textStyle="bold" />

</GridLayout>

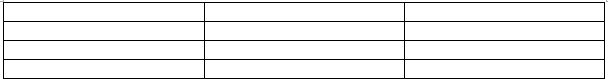
Đấy là GridLayout các view được đặt trong các ộ của Gridlayout và số cột cũng như dòng được qui định bằng hai thuộc tinh:

**1**android:columnCount="2" android:rowCount="2"

Vậy là tôi đã giới thiệu với các bạn về GridLayout

D. TableLayout

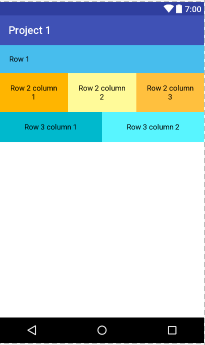
Nghe tên là bạn biết nó là dạng layout gì rồi đúng không nào nó là layout dạng bảng mà thôi cái này quá quen thuộc với chúng ta rồi.

**

*Chỉ là cái bảng thôi mà*

Nhìn gần giống như GridLayout nhưng chúng ta không thể quy định số cột và dòng ngay từ đầu được.

**Ví dụ**

**

*TableLayout*

**Code**

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

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/TableLayout1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:background="#ffffff"

android:shrinkColumns="\*"

android:stretchColumns="\*"

tools:context=".MainActivity" >

<!-- Row 1 with single column -->

<TableRow

android:id="@+id/tableRow1"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal" >

<TextView

android:id="@+id/tv1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_span="3"

android:background="#47BDED"

android:padding="18dip"

android:text="Row 1"

android:textColor="#000000" />

</TableRow>

<!-- Row 2 with 3 columns -->

<TableRow

android:id="@+id/tableRow2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:id="@+id/tv21"

android:layout\_weight="1"

android:background="#FFB500"

android:gravity="center"

android:padding="20dip"

android:text="Row 2 column 1"

android:textColor="#000000" />

<TextView

android:id="@+id/tv22"

android:layout\_weight="1"

android:background="#FFFA99"

android:gravity="center"

android:padding="20dip"

android:text="Row 2 column 2"

android:textColor="#000000" />

<TextView

android:id="@+id/tv23"

android:layout\_weight="1"

android:background="#FFC03E"

android:gravity="center"

android:padding="20dip"

android:text="Row 2 column 3"

android:textColor="#000000" />

</TableRow>

<!-- Row 3 with 2 columns -->

<TableRow

android:id="@+id/tableRow3"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal" >

<TextView

android:id="@+id/tv31"

android:layout\_weight="1"

android:background="#00B9CD"

android:gravity="center"

android:padding="20dip"

android:text="Row 3 column 1"

android:textColor="#000000" />

<TextView

android:id="@+id/tv32"

android:layout\_weight="1"

android:background="#58F5FF"

android:gravity="center"

android:padding="20dip"

android:text="Row 3 column 2"

android:textColor="#000000" />

</TableRow>

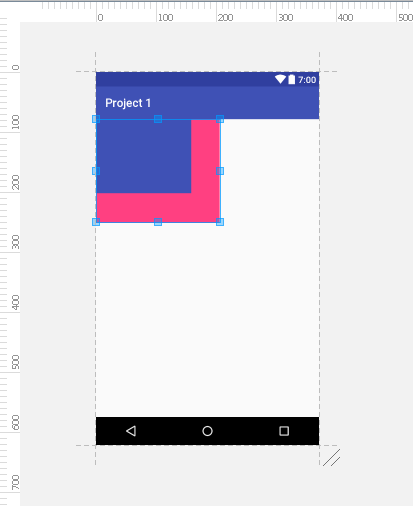
</TableLayout>

Đây là một ví dụ về TableLayout bạn sẽ sử dụng các thẻ <TableRow></TableRow> để tạo ra các dòng cho bảng và các view sẽ nằm trong các dòng này.

D. Framelayout

Framelayout là dạng layout cơ bản nhất khi gắn các view lên layout này thì nó sẽ luôn giữ các view này ở phía góc trái màn hình và không cho chúng ta thay đồi vị trí của chúng, các view đưa vào sau sẽ đè lên view ở trước trừ khi bạn thiết lập transparent cho view sau đó.

**Ví dụ**

**

*FrameLayout*

**Code:**

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

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/TableLayout1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity" >

<TextView

android:id="@+id/textView"

android:layout\_width="254dp"

android:layout\_height="258dp"

android:background="@color/colorAccent"

android:text="TextView" />

<TextView

android:id="@+id/textView2"

android:layout\_width="218dp"

android:layout\_height="221dp"

android:background="@color/colorPrimary" />

</FrameLayout>

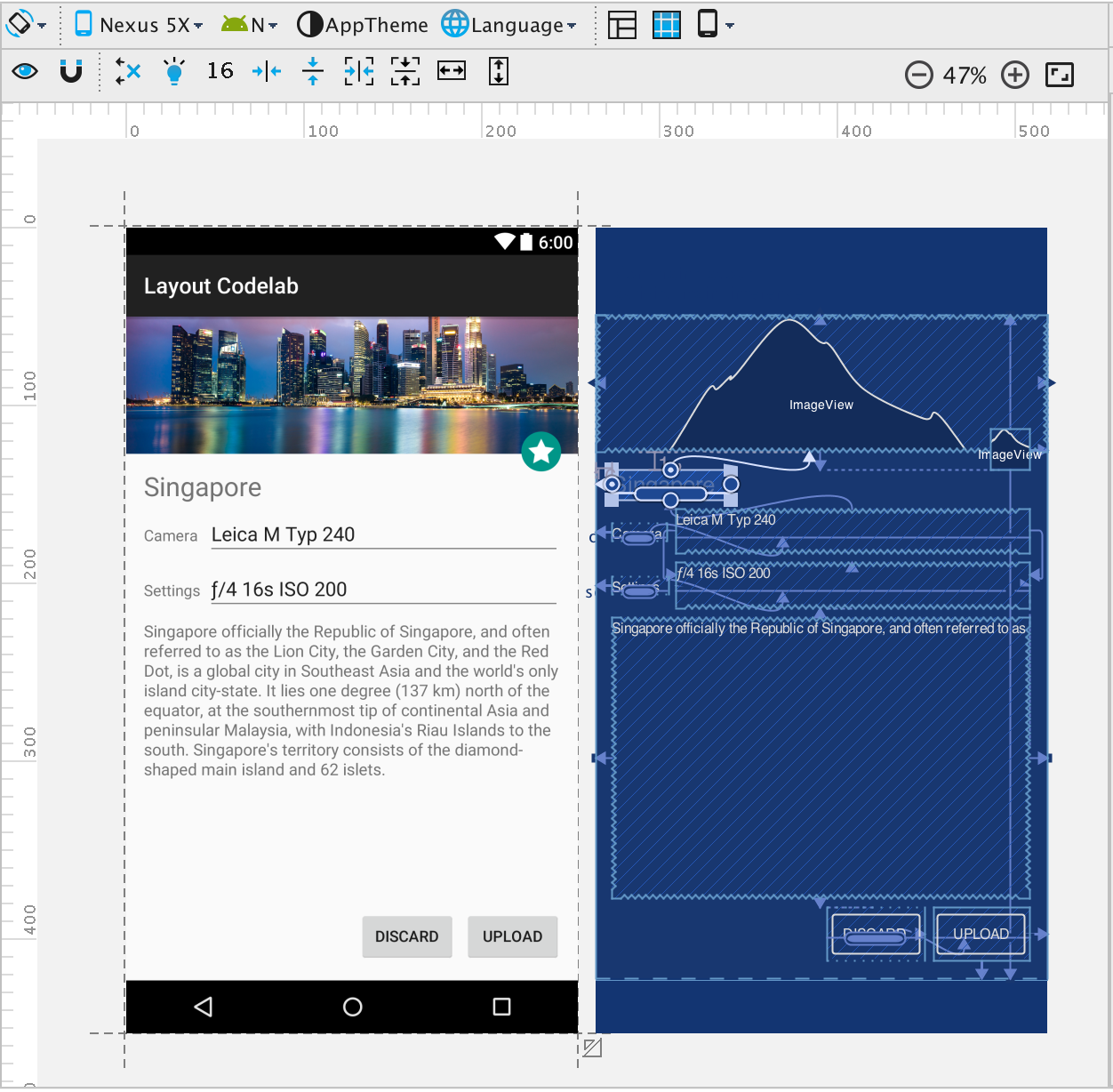
Trong ví dụ trên hình màu xanh đã nằm trên hình màu hồng và bạn sẽ chẳng thế thay đổi vị trí của các view này.

E. ConstraintLayout

Đây là dạng layout mà các view nằm trong đó sẽ được xác định vị trí tương đối với các view khác. Đây là dạng layout mà Google mới công bố và được thiết kế để sử dụng hoàn toàn trên công cụ Design của Android Studio.

**Ví dụ**

***ConstraintLayout***

**