

LẬP TRÌNH CHO CÁC THIẾT BỊ DI ĐỘNG

Bài 5: Layout trong Android

Giới thiệu chung

- ▶ 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.
- ▶ Layout là thành phần định nghĩa cấu trúc giao diện người dùng hay nói cách khác là thành phần quyết định đến giao diện của một màn hình trong ứng dụng Android. Layout hỗ trợ việc canh chỉnh các widget (Ví dụ: TextView, Button, hay EditText...) như chúng ta thấy trong các ứng dụng Android.



Các loại layout trong Android

- ▶ RelativeLayout
- ▶ LinearLayout
- ▶ GridLayout
- ▶ TableLayout
- ▶ Framelayout
- ▶ ConstraintLayout.



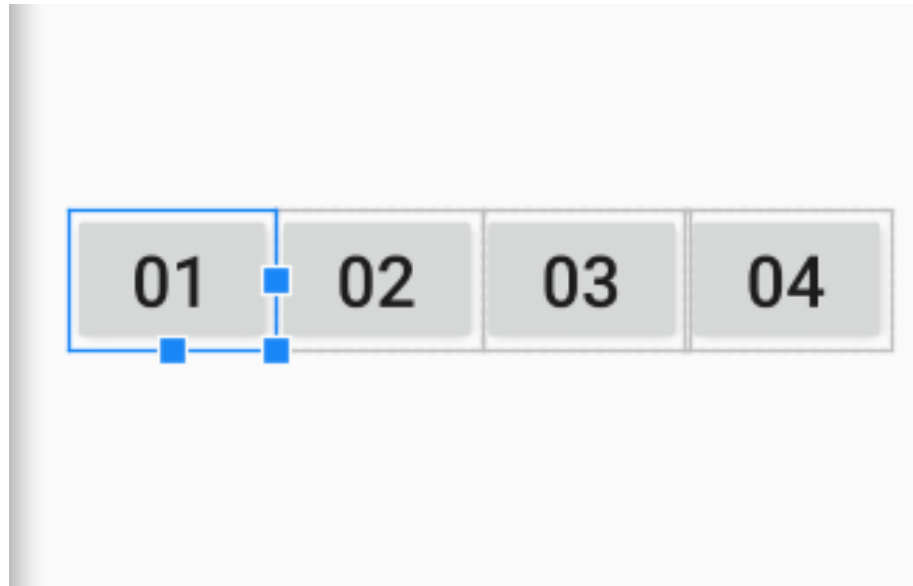
LinearLayout

- ▶ LinearLayout là loại layout sẽ sắp xếp các view theo chiều dọc hoặc ngang theo thứ tự của các view.
- ▶ Đây là ViewGroup sẽ giúp các bạn sắp xếp các view con chứa bên trong theo dạng hàng ngang hoặc hàng dọc với nhau.



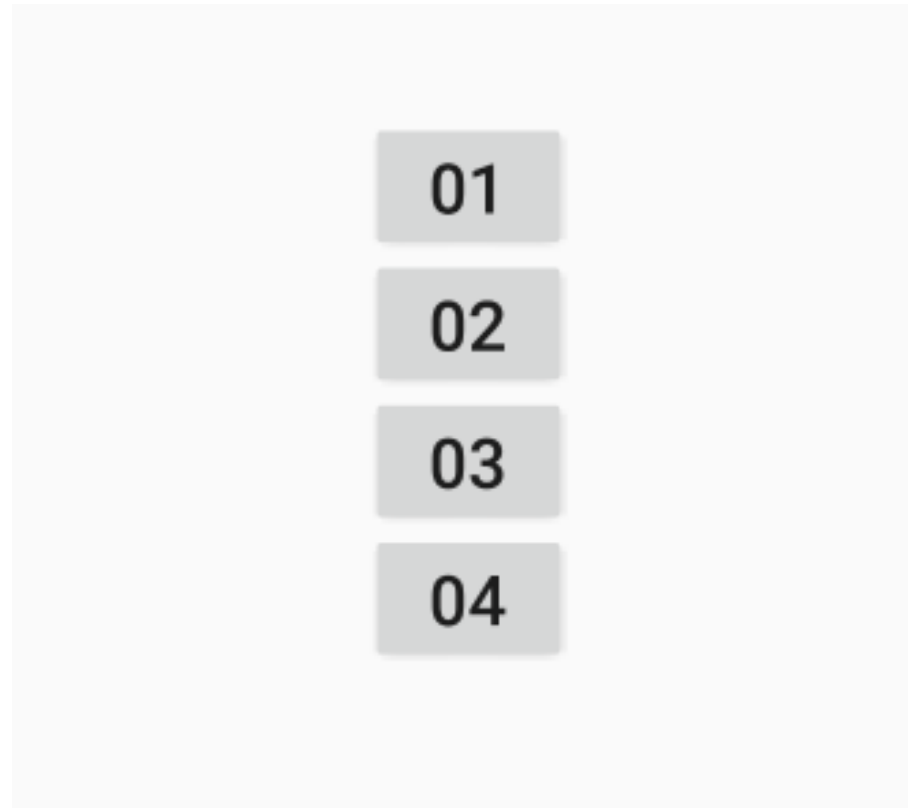
LinearLayout

- ▶ `android:orientation="horizontal"`



LinearLayout

- ▶ `android:orientation="vertical "`



LinearLayout

▶ Thuộc Tính Gravity

- ▶ Mặc định thì các thành phần con bên trong LinearLayout sẽ được “hút” về start-top theo “lực hấp dẫn” mặc định
- ▶ Thuộc tính `android:gravity` để canh chỉnh các View nằm ở vị trí nào trong LinearLayout, nó nhận các giá trị (có thể tổ hợp lại với ký hiệu |)



LinearLayout

| Giá trị | Thuộc tính |
|-------------------|-------------------------|
| center | Canh ở giữa |
| top | Ở phần trên |
| bottom | Phần dưới |
| center_horizontal | Ở giữa theo chiều ngang |
| center_vertical | Ở giữa theo chiều đứng |
| left | Theo cạnh trái |
| right | Theo cạnh phải |
| bottom | Cạnh dưới |



LinearLayout

▶ Thuộc Tính Trọng Số (Weight)

- ▶ Các View con trong LinearLayout có thể gán cho nó một giá trị trọng số bằng thuộc tính `android:layout_weight` ví dụ như: `android:layout_weight="2"`; `android:layout_weight="1.5"` Nếu View không gán giá trị này coi như nó có trọng số bằng không.
- ▶ Trường hợp LinearLayout không sử dụng đến thuộc tính `android:weightSum`.
- ▶ Khi chúng ta muốn các view tự động full màn hình thì sử dụng `android:weightSum`.
- ▶ Ví dụ như khi muốn chia màn hình thành 5 phần bằng nhau, ta sử dụng thuộc tính `weightSum` như hình dưới.



LinearLayout

01

02

03

04

05



LinearLayout

- ▶ `<?xml version="1.0" encoding="utf-8"?>`
- ▶ `<LinearLayout 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"`
- ▶ `android:layout_gravity="center"`
- ▶ `android:orientation="vertical"`
- ▶ `android:weightSum="5"`
- ▶ `android:gravity="center"`
- ▶ `tools:context=".View">`
- ▶ `<Button`
- ▶ `android:layout_weight="1"`
- ▶ `android:textSize="30dp"`
- ▶ `android:text="01"`
- ▶ `android:layout_width="match_parent"`
- ▶ `android:layout_height="match_parent"/>`



LinearLayout

- ▶ <Button
 - ▶ android:layout_weight="1"
 - ▶ android:textSize="30dp"
 - ▶ android:text="02"
 - ▶ android:layout_width="match_parent"
 - ▶ android:layout_height="match_parent"/>

 - ▶ <Button
 - ▶ android:layout_weight="1"
 - ▶ android:textSize="30dp"
 - ▶ android:text="03"
 - ▶ android:layout_width="match_parent"
 - ▶ android:layout_height="match_parent"/>
-



LinearLayout

- ▶ <Button
 - ▶ android:layout_weight="1"
 - ▶ android:textSize="30dp"
 - ▶ android:text="04"
 - ▶ android:layout_width="match_parent"
 - ▶ android:layout_height="match_parent"/>

 - ▶ <Button
 - ▶ android:layout_weight="1"
 - ▶ android:textSize="30dp"
 - ▶ android:text="05"
 - ▶ android:layout_width="match_parent"
 - ▶ android:layout_height="match_parent"/>

 - ▶ </LinearLayout>
-



RelativeLayout

- ▶ RelativeLayout là một ViewGroup có hiển thị các View con ở các vị trí tương đối. Vị trí của mỗi View có thể được quy định liên quan đến các View anh em (như bên trái của hoặc bên dưới một View khác) hoặc ở các vị trí tương đối với khu vực cha RelativeLayout (chẳng hạn như sắp xếp ngay phía dưới, bên trái hoặc giữa).



RelativeLayout

▶ Thuộc tính Gravity

- ▶ Các View con khi đã định vị xong trong RelativeLayout, giả sử coi như tất cả các View con nằm vừa trong một đường biên chữ nhật, thì cả khối các View con này có thể dịch chuyển tới những vị trí nhất định trong RelativeLayout bằng thuộc tính: `android:gravity`, nó nhận các giá trị (có thể tổ hợp lại với ký hiệu |)



Relativelayout

| Giá trị | Ý nghĩa |
|-------------------|-------------------------|
| center | Canh ở giữa |
| top | Ở phần trên |
| bottom | Phần dưới |
| center_horizontal | Ở giữa theo chiều ngang |
| center_vertical | Ở giữa theo chiều đứng |
| left | Theo cạnh trái |
| right | Theo cạnh phải |
| bottom | Cạnh dưới |



RelativeLayout

- ▶ Định vị view con bằng view cha
 - ▶ Vị trí của View con trong RelativeLayout có thể thiết lập bằng cách chỉ ra mối liên hệ vị trí với view cha, như canh thẳng cạnh trái View cha với View con, canh thẳng cạnh phải View cha với View con... Các thuộc tính thực hiện chức năng này như sau:



RelativeLayout

| Thuộc tính | Ý nghĩa |
|---|---|
| <code>android:layout_alignParentBottom</code> | true canh thẳng cạnh dưới view con với cạnh dưới View cha |
| <code>android:layout_alignParentLeft</code> | true canh thẳng cạnh trái view con với cạnh trái View cha |
| <code>android:layout_alignParentRight</code> | true canh thẳng cạnh phải view con với cạnh phải View cha |
| <code>android:layout_alignParentTop</code> | true canh thẳng cạnh trên view con với cạnh trên View cha |
| <code>android:layout_centerInParent</code> | true canh view con vào giữa View cha |
| <code>android:layout_centerHorizontal</code> | true canh view con vào giữa View cha theo chiều ngang |
| <code>android:layout_centerVertical</code> | true canh view con vào giữa View cha theo chiều đứng |



RelativeLayout

- ▶ Định vị các view con với nhau bằng thuộc tính liên hệ với nhau
- ▶ Tất cả các thuộc tính dưới đây cần phải truyền vào một ID @+id/
 - ▶ android:layout_alignTop – Chỉ định đỉnh của thành phần này sẽ được canh theo đỉnh của thành phần gọi đến bằng ID.
 - ▶ android:layout_alignBottom – Chỉ định đáy của thành phần này sẽ được canh theo đáy của thành phần gọi đến bằng ID.
 - ▶ android:layout_alignStart – Chỉ định cạnh trái của thành phần này sẽ được canh theo cạnh trái của thành phần gọi đến bằng ID.
 - ▶ android:layout_alignEnd – Chỉ định cạnh phải của thành phần này sẽ được canh theo cạnh phải của thành phần gọi đến bằng ID.



RelativeLayout

- ▶ `android:layout_alignBaseline` – Chỉ định baseline của thành phần này sẽ được canh theo baseline của thành phần gọi đến bằng ID. Baseline này bạn không nhìn thấy được, dùng để canh chỉnh cho text hiển thị bên trong widget, do đó sẽ hữu dụng khi canh chỉnh các TextView với nhau).
 - ▶ `android:layout_above` – Chỉ định thành phần này sẽ nằm ở trên so với thành phần gọi đến bằng ID.
 - ▶ `android:layout_below` – Chỉ định thành phần này sẽ nằm dưới so với thành phần gọi đến bằng ID.
 - ▶ `android:layout_toStartOf` – Chỉ định thành phần này sẽ nằm bên phía start so với thành phần gọi đến bằng ID.
 - ▶ `android:layout_toEndOf` – Chỉ định thành phần này sẽ nằm bên phía end so với thành phần gọi đến bằng ID.
 - ▶ `android:layout_toLeftOf` – Chỉ định thành phần này sẽ nằm bên phía trái so với thành phần gọi đến bằng ID.
 - ▶ `android:layout_toRightOf` – Chỉ định thành phần này sẽ nằm bên phía phải so với thành phần gọi đến bằng ID
-



Relativelayout



RelativeLayout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/androi
d"
    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"
    android:layout_gravity="center"
    android:gravity="center"
    tools:context=".View">
```



RelativeLayout

```
<TextView
```

```
    android:layout_width="150dp"
```

```
    android:layout_height="50dp"
```

```
    android:text=" Hop 1 "
```

```
    android:textColor="@android:color/black"
```

```
    android:id="@+id/hop1"
```

```
    android:textSize="10pt"
```

```
    android:textAlignment="center"
```

```
    android:background="#D32F2F"
```

```
/>
```



RelativeLayout

<TextView

android:layout_width="150dp"

android:layout_height="50dp"

android:text=" Hop 2"

android:textColor="@android:color/black"

android:id="@+id/hop2"

android:textSize="10pt"

android:textAlignment="center"

android:layout_toRightOf="@+id/hop1"

android:background="#FFEB3B"

/>



RelativeLayout

<TextView

android:layout_width="150dp"

android:layout_height="50dp"

android:text=" Hop 3"

android:textColor="@android:color/black"

android:id="@+id/hop3"

android:background="#FFAB40"

android:textSize="10pt"

android:textAlignment="center"

android:layout_below="@+id/hop1"

/>



RelativeLayout

<TextView

```
    android:layout_width="150dp"
    android:layout_height="50dp"
    android:text="Hop 4"
    android:textColor="@android:color/black"
    android:id="@+id/hop4"
    android:textSize="10pt"
    android:textAlignment="center"
    android:background="#E040FB"
    android:layout_below="@+id/hop2"
    android:layout_toRightOf="@+id/hop3"
/>
```

</RelativeLayout>



TableLayout

- ▶ Trong hệ thống layout của Android thì ta thường sử dụng TableLayout để bố trí các view dưới dạng table, mỗi hàng là một TableRow, và bạn có thể thay đổi các thuộc tính để chỉnh giao diện cho TableRow đó.
- ▶ Ta thường sử dụng TableLayout để bố trí các view dưới dạng table, mỗi hàng là một TableRow, và bạn có thể thay đổi các thuộc tính để chỉnh giao diện cho TableRow đó.



TableLayout

| column1 | column2 | column3 |
|---------|---------|---------|
| 1 | vncoder | vncoder |
| 2 | vncoder | vncoder |
| 3 | vncoder | vncoder |
| | | |



TableLayout

| Thuộc tính | Mô tả |
|--------------------------------------|--|
| <code>android:id</code> | ID là duy nhất để nhận diện Layout |
| <code>android:collapseColumns</code> | Xác định index của các cột (bắt đầu từ 0). Các index cột được phân biệt riêng rẽ bởi dấu phẩy. |
| <code>android:shrinkColumns</code> | Xác định index của các cột (bắt đầu từ 0) để co lại. Các index cột được phân biệt riêng rẽ bởi dấu phẩy. |
| <code>android:stretchColumns</code> | Xác định index của các cột (bắt đầu từ 0) để kéo ra. Các index cột được phân biệt riêng rẽ bởi dấu phẩy. |



ConstraintLayout

- ▶ ConstraintLayout là một layout mạnh, khuyến khích dùng nếu có thể vì nó giúp tạo ra các giao diện phức tạp, mềm dẻo (hạn chế tối đa sử dụng các layout lồng nhau).
- ▶ Nó giúp định vị, sắp xếp các View con dựa trên sự ràng buộc liên hệ của các View con với View cha và sự liên hệ ràng buộc giữa các View con với nhau, với cơ chế tạo xích các View, gán trọng số hay sử dụng trợ giúp giao diện với Guideline.



ConstraintLayout

- ▶ Mỗi view trong ConstraintLayout để định vị được chính xác cần tối thiểu 2 ràng buộc, một theo phương ngang (X) và một theo phương đứng (Y).
- ▶ Khái niệm ràng buộc giữa các phần tử ở đây ám chỉ sự liên kết với nhau của các phần tử với nhau (kể cả với phần tử cha ConstraintLayout), sự canh chỉnh phần tử theo phần tử khác, hoặc với những đường thẳng ẩn thêm vào. Mỗi ràng buộc của phần tử View sẽ hoặc hưởng đến vị trí của nó theo trục X hoặc trục Y. Các View không có ràng buộc sẽ định vị ở góc trái - trên (tọa độ 0,0).
- ▶ Trước tiên tham khảo bảng các thuộc tính về ràng buộc `layout_constraint ...`, các thuộc tính ràng buộc sử dụng với `namespace:app`, giá trị nó gán vào là một ID của phần tử khác để kết nối ràng buộc hoặc là phần tử ch bằng hằng số "parent"



ConstraintLayout

| Ràng buộc | Ý nghĩa ràng buộc |
|---|---|
| <code>layout_constraintLeft_toLeftOf</code> | Ràng buộc cạnh trái của phần tử tới phần tử chỉ ra trong giá trị (gán ID) |
| <code>layout_constraintLeft_toRightOf</code> | Bên trái với bên phải của phần tử chỉ ra |
| <code>layout_constraintRight_toLeftOf</code> | Bên phải với bên trái |
| <code>layout_constraintRight_toRightOf</code> | Phải với phải |
| <code>layout_constraintTop_toTopOf</code> | Cạnh trên với cạnh trên |
| <code>layout_constraintTop_toBottomOf</code> | Cạnh trên nối với cạnh dưới |



ConstraintLayout

Ràng buộc

Ý nghĩa ràng buộc

layout_constraintBottom_toTopOf

Dưới với trên

layout_constraintBottom_toBottomOf

Dưới với dưới

layout_constraintBaseline_toBaselineOf

Trùng Baseline

layout_constraintStart_toEndOf

Bắt đầu - Kết thúc

layout_constraintStart_toStartOf

Bắt đầu - Bắt đầu

layout_constraintEnd_toStartOf

Cuối với bắt đầu

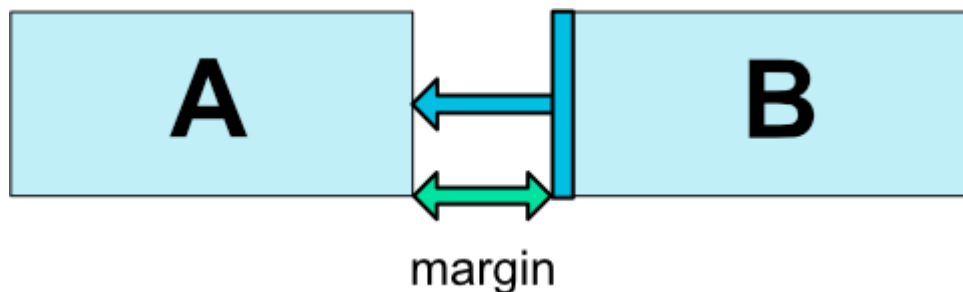
layout_constraintEnd_toEndOf

Cuối với cuối



ConstraintLayout

- ▶ Cạnh nào của View con có ràng buộc thì có thể thiết lập thêm thuộc tính Margin để điều chỉnh thêm khoảng cách các cạnh tới điểm nối ràng buộc.



- ▶ Các thuộc tính margin theo các cạnh:
android:layout_marginStart, android:layout_marginEnd,
android:layout_marginLeft, android:layout_marginTop,
android:layout_marginRight,
android:layout_marginBottom

ConstraintLayout

► Bias

Khi hai cạnh đối diện nhau của View có ràng buộc, thì hai ràng buộc này ứng xử như một liên kết lò xo mặc định nó có độ mềm (độ cứng) bằng nhau dẫn đến View sẽ nằm giữa 2 điểm neo của ràng buộc. Nếu muốn điều chỉnh độ cứng này thì sử dụng thuộc tính:

- `app:layout_constraintVertical_bias` thiết lập độ mềm của ràng buộc đầu (ngang). Với tổng độ mềm là 1 thì khi `app:layout_constraintVertical_bias="0.1"` thì độ mềm ràng buộc thứ hai sẽ là 0.9
- `layout_constraintHorizontal_bias` để thiết lập độ mềm hai ràng buộc theo phương đứng

