

[سؤال وجواب] ما هو الفرق بين الLinearLayout والRelativeLayout ؟

مقالة , مناقشة , عام

في ١٨ مايو

OmarAlbelbaisy
Tutor

السؤال: أحد أكثر الأسئلة شيوعا في المنتدى، ما هو الفرق بين النسق الخطي LinearLayout والنسق النسبي RelativeLayout ؟

الإجابة:

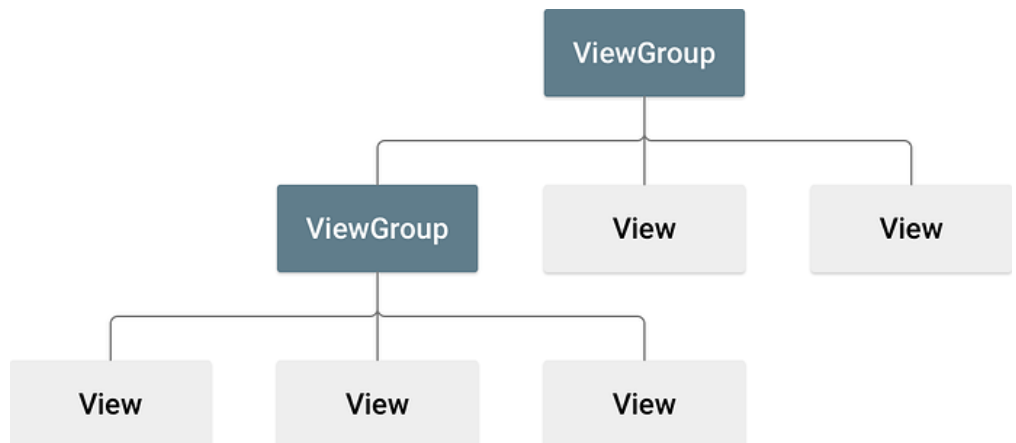
قبل أن نشرح الفرق بينهم لا بد من معرفة المصطلحات التالية

1- ال **View** هي الوحدة الأساسية لمكونات واجهة المستخدم، وتحتل ال **View** مساحة مستطيلة على الشاشة، يعتبر كلاس **View** هو ال **base class** لكل ال **Widgets** و ال **ViewGroups** في أندرويد.

2- ال **ViewGroup** هو عبارة عن نوع خاص من ال **Views** يمكنه أن يحتوى على **Views** أخرى بداخله وعندها نطلق عليهم **child views**! يعتبر كلاس **ViewGroup** هو ال **base class** لل **layouts** و ال **view containers** الأخرى.

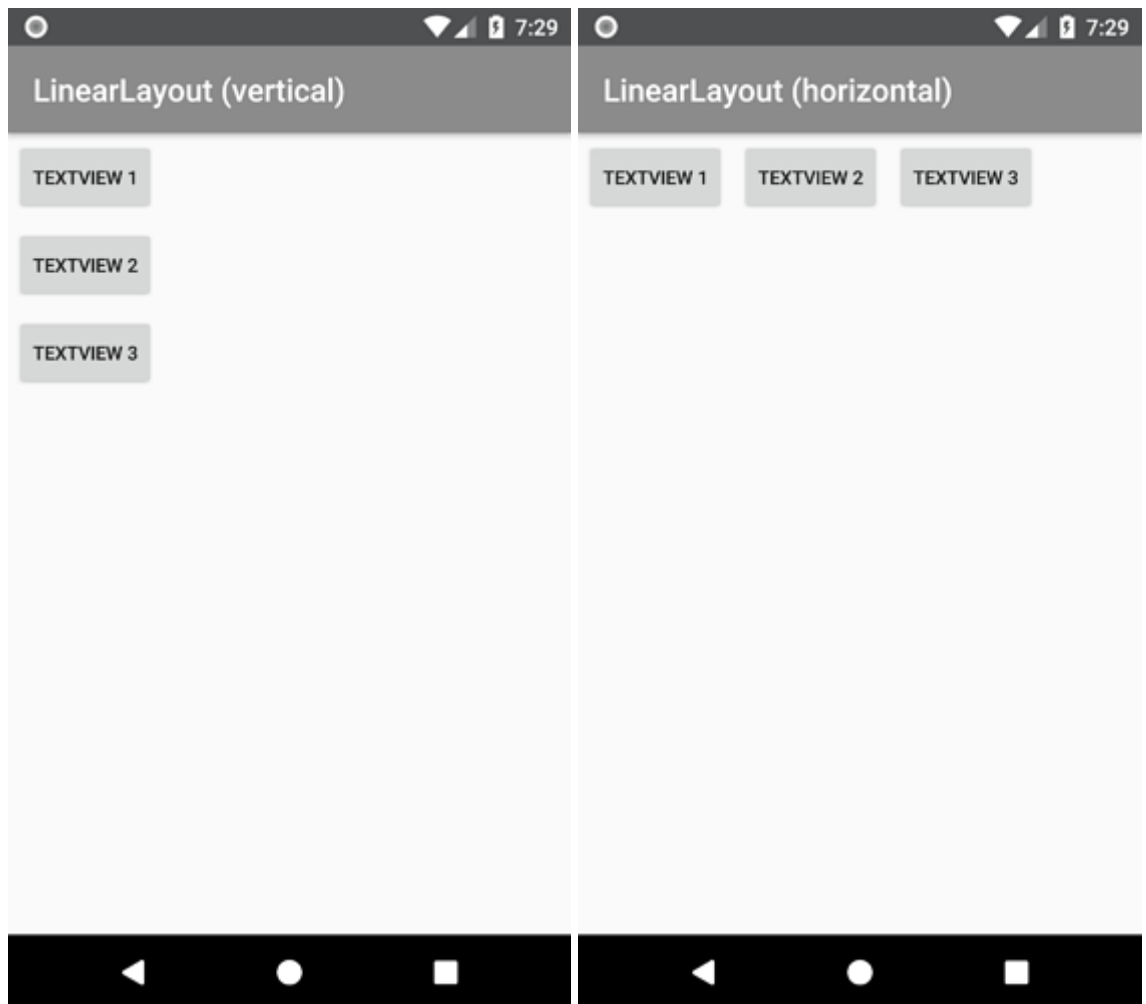
3- ال **Layout** هو نوع من ال **ViewGroups** يمكننا من خلاله تعريف تركيبة واجهة المستخدم في التطبيق حيث يمكن أن يحتوي ال **layout** على عدد من ال **Views** و ال **ViewGroups** (والتي بدورها قد تحتوى على **Views** و **ViewGroups** أخرى) ويتم ترتيب أماكنها وكيفية ظهورها على الشاشة حسب نوع هذا ال **layout** والخصائص المستخدمة ومن أشهر أنواعه ال **LinearLayout** و ال **RelativeLayout**.

4- ال **View Containers** هو نوع من ال **ViewGroups** ولكنها لا تستخدم لتحديد تركيبة وتوزيع أماكن ال **Views** على الشاشة وإنما هي إما أن تحتوي على مجموعة من ال **Views** مضافة مسبقا لتشكل وحدة واحدة تخدم وظيفة أو هدف معين مثل ال **DatePicker** و ال **CalenderView** وإما أن تكون **AdapterView** يتم إدارة إضافة وإزالة ال **Views** الخاصة بها من خلال ما يسمى بال **Adapter** ومن الأمثلة عليها ال **ListView** و ال **GridView**.

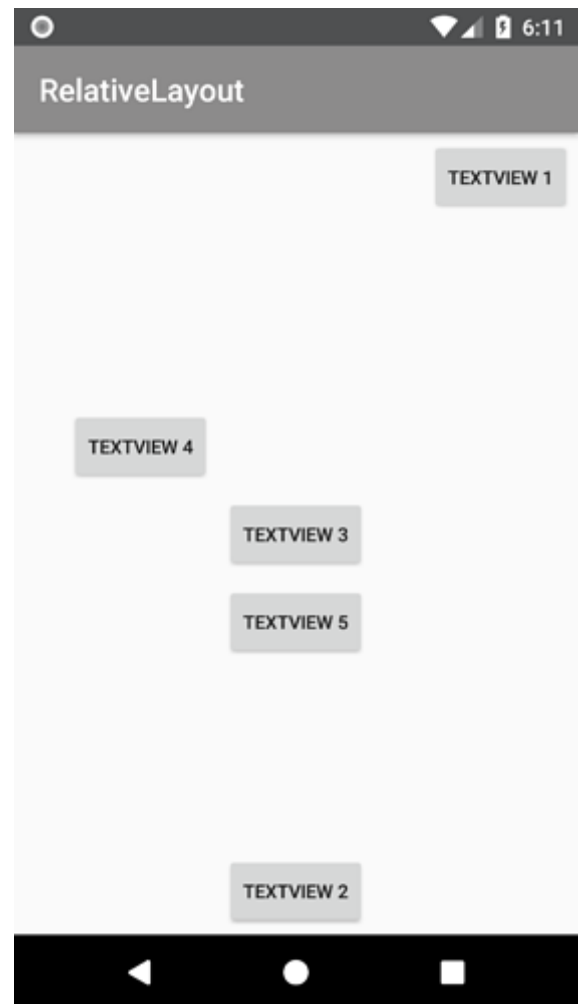


والآن لنأتي لتعريف كل من ال **LinearLayout** و ال **RelativeLayout**

ال **LinearLayout** هو عبارة عن **layout** يقوم بترتيب ال **views** التي بداخله إما عموديا في عمود واحد فقط أو أفقيا في صف واحد فقط.



الRelativeLayout هو عبارة عن layout يتم تحديد أماكن ال views بداخله إما نسبة إلى بعضها البعض أو نسبة إلى ال RelativeLayout نفسه.



والآن سنقوم بشرح كل منهما بشيء من التفصيل:-

أولاً: ال LinearLayout

وهو كما ذكرنا يستخدم لمحاذاة وصف كل ال view بداخله باتجاه واحد إما أفقي أو عمودي واحدا تلو الآخر، ويتم تحديد اتجاه ال LinearLayout باستخدام الخاصية orientation فإذا كان الاتجاه عمودي فهذا يعني أنه سيتم إضافة View واحدة في كل صف بغض النظر عن عرض هذه ال View وأما إذا كان الاتجاه أفقي فهذا يعني إضافة View واحدة في كل عمود بغض النظر عن طول هذه ال View.

من أهم مزايا ال LinearLayout أنه يلتزم بال margins بين ال Views وبخاصية ال gravity والتي تحدد محاذاة ال View بحيث إن كان اتجاه ال LinearLayout أفقي يمكن تحديد محاذاة ال View إما في الوسط أو في الأعلى أو في الأسفل أما إذا كان اتجاه ال LinearLayout عمودي فهذا يعني أنه يمكن تحديد محاذاة ال View إما في الوسط أو على اليمين أو على اليسار.

من أشهر خواص ال LinearLayout هي:-

- 1- ال orientation والتي يمكن من خلالها تحديد اتجاه صف العناصر في ال LinearLayout إما أفقياً `android:orientation="horizontal"` أو عمودياً `android:orientation="vertical"`

- 2- ال weight والتي يمكن من خلالها توزيع أحد أبعاد ال Layout على شكل نسب بين ال child views التي يحتويها بحيث يكون لكل ال View عدد وزن أو عدد أسهم يتم بناء عليه تحديد مقدار نسبته من الحجم الإجمالي الخاص بال parent layout مطروحاً منه حجم ال Views التي تم تحديد ذلك البعد الخاص بها بقيمة غير الصفر، مثال:-

في حال كان لدينا ال LinearLayout ارتفاعه 600dp ويحتوي على خمسة Views وكان ارتفاع ال View الأولي هو 100dp والثانية 100dp وأما الثالثة والرابعة والخامسة كل منهم قيمة ارتفاعها هي 0dp ولكن تم إعطاء قيم للخاصية weight للثالثة بمقدار 2 وأما الرابعة 1 والخامسة 1 عندها سيتم توزيع الارتفاع 600dp بعد طرح قيم الارتفاع الثابتة وهي 100 للأولى و 100 للثانية وبالتالي يتبقى 400dp يتم توزيعهم إلى 4 حصص وهي إجمالي قيم ال weight لل Views المتبقية وعندها ستحصل ال View الثالثة على حصتين أي $200dp = 2 \times 100$ وأما الرابعة والخامسة فستحصل كل منهما على 100dp مع العلم أنه يمكن أن تكون قيمة ال weight عبارة عن رقم عشري مثل 1.5 وعندها ستحصل ال View على ما قيمته سهم ونصف من إجمالي الارتفاع.

ثانياً: ال RelativeLayout

وهو كما ذكرنا يقوم بتحديد أماكن نسبية لل Views، بحيث يتم تحديد مكان كل View بداخله نسبة إلى حدود ال layout نفسه (مثل محاذاة

view إلى أسفل الـ layout أو إلى يسار الـ layout) أو نسبة إلى أماكن Views أخرى بداخله بناء على الـ id الخاص بالـ views الأخرى (مثل على يمين view أخرى أو تحت view أخرى وما إلى ذلك) مع العلم أن كل الـ Views التي يتم إضافتها إلى الـ RelativeLayout يتم إضافتها إلى أعلى يسار الـ layout بشكل افتراضي في حال لم يتم تحديد الخواص المكانية لهذا الـ View يعتبر الـ RelativeLayout من أقوى أنواع الـ Layouts والذي يمكن من خلاله تقليل استخدام الـ Nested ViewGroups (أي وضع ViewGroup داخل ViewGroup أخرى) وبالتالي يمكن تحسين أداء التطبيق حيث أن الـ nested viewgroups تؤدي إلى تراجع في أداء التطبيق وتستهلك الكثير من الموارد ووقت أطول لعرض الـ layout.

من المهم جدا عند العمل مع الـ RelativeLayout الانتباه لموضوع الـ circular dependency بين حجم الـ layout ومكان الـ view التي بداخله بحيث لا يمكن تحديد ارتفاع الـ layout بـ wrap_content وتحديد مكان الـ child view بـ alignParentBottom على سبيل المثال!

هناك الكثير من الخصائص التي يمكن استخدامها لتحديد مكان الـ View في الـ RelativeLayout يمكنك الإطلاع عليها مع شرح كل خاصية من خلال زيارة الرابط التالي

<https://developer.android.com/reference/android/widget/RelativeLayout.LayoutParams>

ملاحظة هامة:- تم إطلاق layout جديد لتطبيقات أندرويد منذ حوالي عامين من خلال تضمينه ضمن مكتبات الدعم support libraries حتى يمكن استخدامه في الإصدارات القديمة من أندرويد وهو الـ **ConstraintLayout** والذي يتميز بأنه يمكن ترتيب أماكن الـ Views بداخله بشكل مرن جدا اعتمادا على تحديد قيود خاصة لكل View تربطها بالـ parent views الأخرى المحيطة بها من خلال العديد من الخصائص التي جعلت منه الـ layout الأفضل حتى الآن والذي جعل من الـ RelativeLayout بمثابة إصدار قديم. للتعرف على إمكانيات الـ ConstraintLayout وكيفية استخدامه يمكنكم زيارة الروابط التالية

<https://developer.android.com/reference/android/support/constraint/ConstraintLayout>
<https://codelabs.developers.google.com/codelabs/constraint-layout/#0>

وفي النهاية أرجوا أن يكون الموضوع قد أظهر لكم الفرق بشكل واضح أو أضاف لكم معلومة جديدة. ولا تنسوا أنه بإمكانكم مشاركة الموضوع مع زملائكم الذين يسألون هذا السؤال وما يزال لديهم مشكلة في معرفة الفرق.

تحياتي للجميع

✎ إحتراف الأندرويد ستوديو

✎ RelativeLayout LinearLayout or ConstraintLayout

✎ متى نستخدم النسق الخطي linearlayout والنسق النسبي relativelayout

✎ Linear & relative layouts

✎ Constraint Layout ازاى استخدم

5 أخرى