



មជ្ឈមណ្ឌលកូរ៉េ សហប្រតិបត្តិការ អេច អ ឌី

Korea Software HRD Center

Android

Android Layout

ណែនាំដោយ : បណ្ឌិត គឹម ថេឡុង



<http://www.kshrd.com.kh>



୧. Linear Layout

୨. Relative Layout

୩. Frame Layout

୪. Table Layout

୫. Grid Layout

୬. Constraint Layout

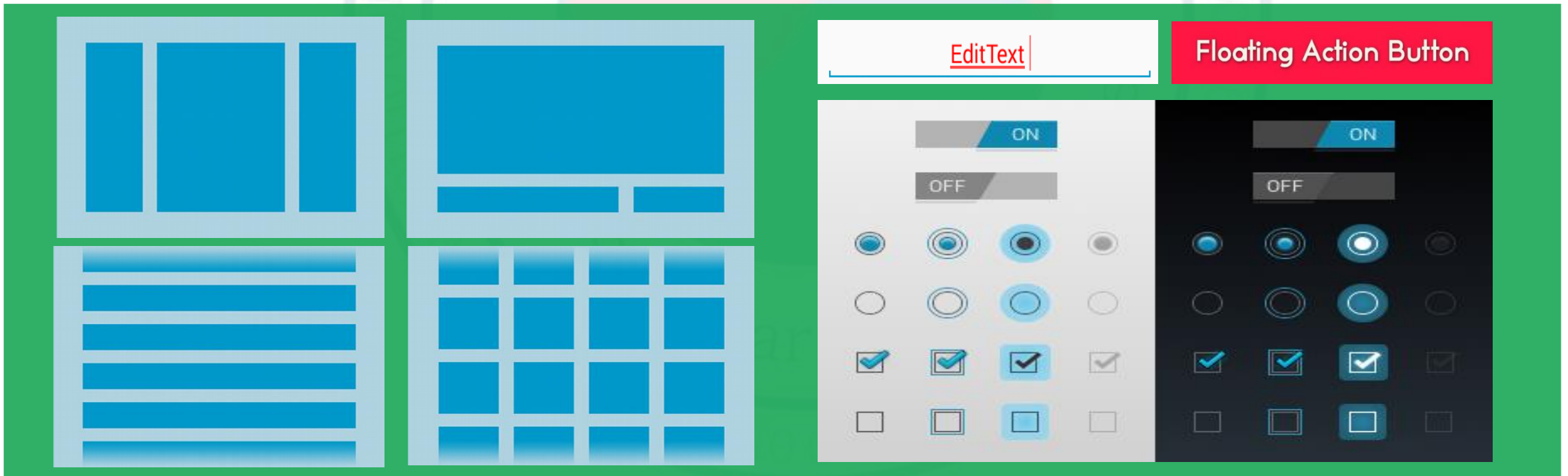
୭. Scroll View

୮. Horizontal Scroll View

୯. Nested Scroll View

១. Linear Layout

Layouts គឺជារូបសណ្ឋានធាតុរបស់កម្មវិធី នៃការរចនាឲ្យឃើញនៅលើ អេក្រង់ (Screen) ដែលប្រើសម្រាប់ បង្កើត user Interface នៅក្នុង android apps ឧទាហរណ៍ដូចជា UI activity ឬ app widget ជាដើម។



១. Linear Layout (ត)

យើងអាចបង្កើត Layouts បានតាមពីរ វិធី គឺ៖

- UI Elements in XML : ប្រើ XML vocabulary សរសេរបង្កើត UI, ដែល XML vocabulary ឆ្លើយតប ទៅនឹង View classes និង subclasses ដូចជា Layout ឬ widget ជាដើម។
- Instantiate layout Elements at runtime : បង្កើត Layout ដោយប្រើ java code ។

ការបង្កើត Layout ដោយប្រើ UI Elements របស់ XML គឺ ប្រសើរជាងប្រើ វិធី Instantiate layout Elements at runtime ព្រោះថា UI Elements របស់ XML មាន screen elements និង properties របស់វា ឲ្យមើលពេល Design។

១. Linear Layout (ត)

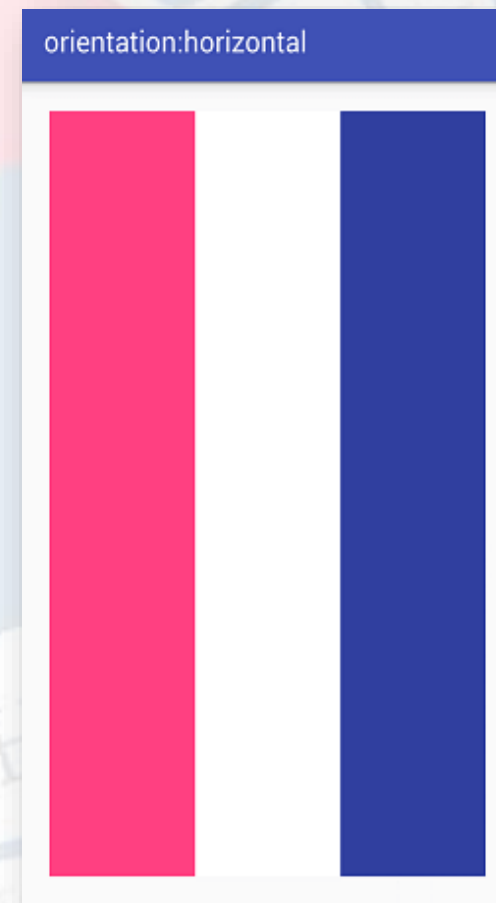
Parameter ដែលត្រូវការជាចាំបាច់នៅពេលប្រើ layout ៖

- រាល់នូវ Layout គឺតែងតែមាន width និង height
- layout_width និង layout_height គឺមាននូវ ២ values ដូចជា ៖
 1. Wrap_content: ទំហំប្រែប្រួលតាមអក្សរ
 2. Match_parent: ទំហំតាម Parent របស់វា។

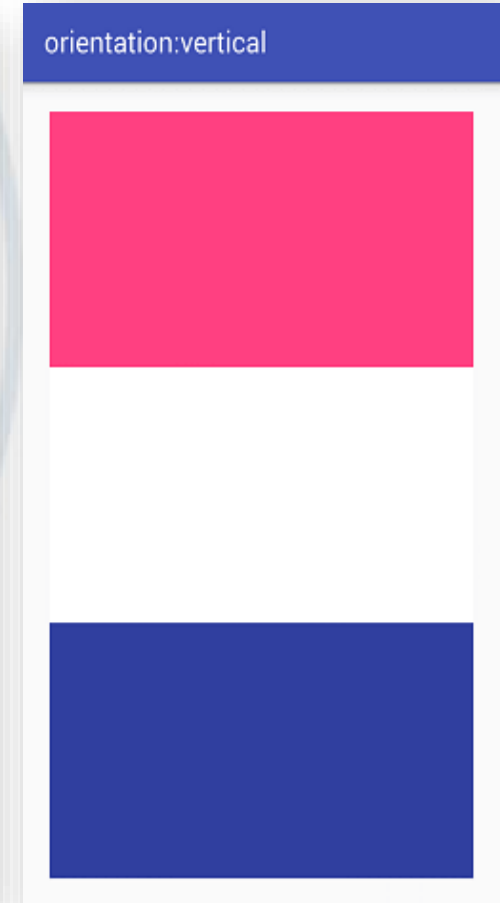
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
</RelativeLayout>
```

១. Linear Layout (ត)

- Views កូន ទាំងអស់ដែលនៅក្នុង Linear Layout ត្រូវបានតម្រៀបត្រួតត្រា ជាជួរដេក ឬជួរឈរ
- Linear Layout បែងចែកជាពីរដូចខាងក្រោម៖
 - **Horizontal**: ជា default layout របស់ Linear Layout និងមានទិសដៅឈរ។
 - **Vertical**: មានទិសដៅដេកហើយអាចកំណត់បានដោយប្រើ
`android:orientation="vertical"`។



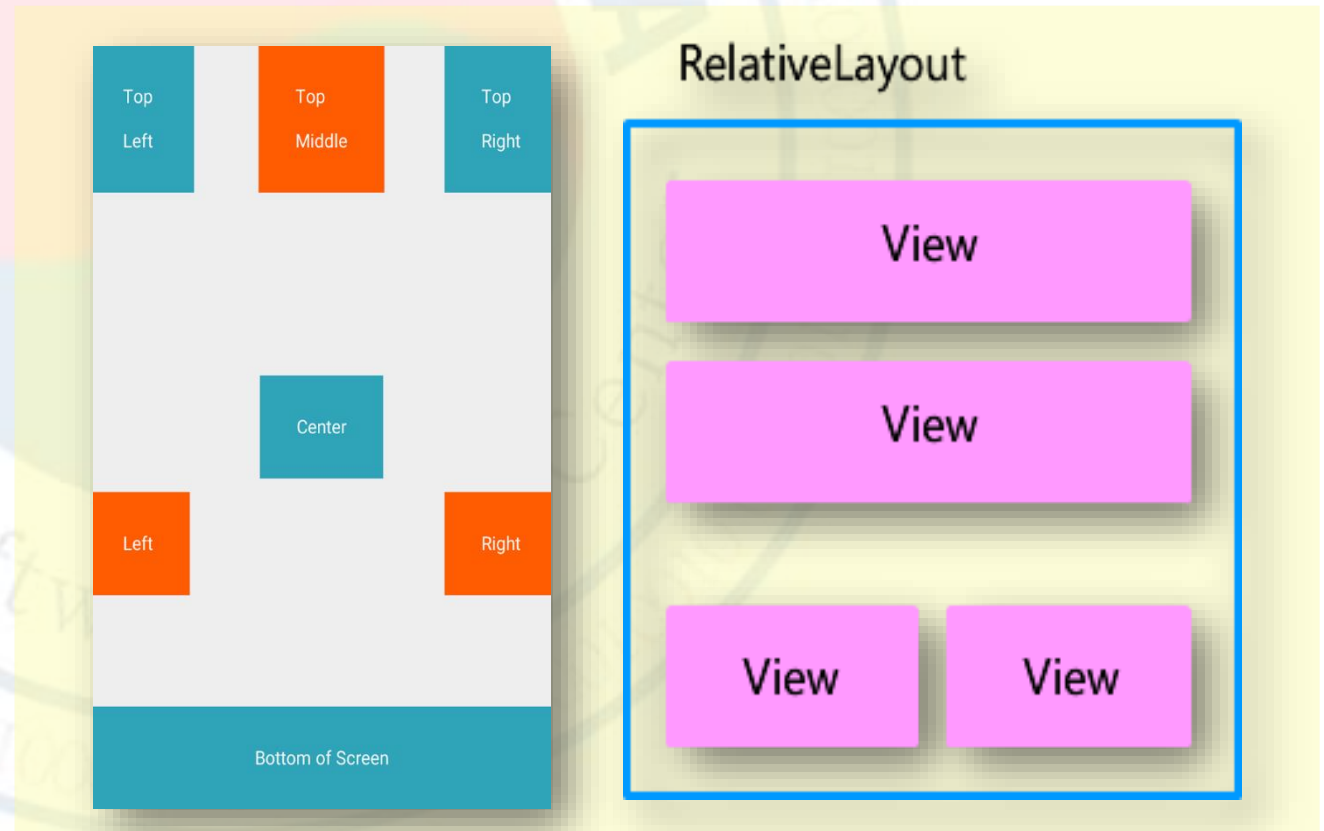
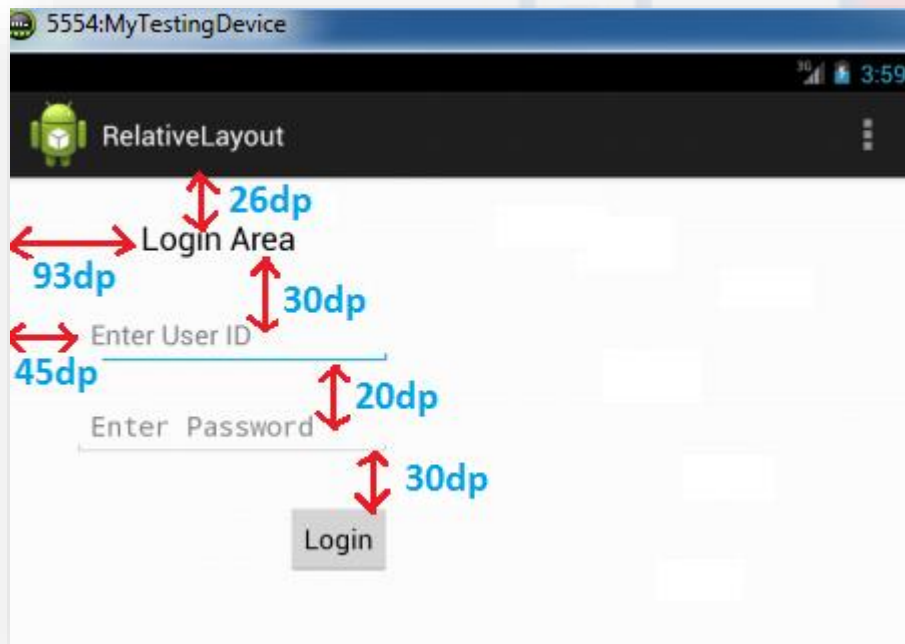
Horizontal



Vertical

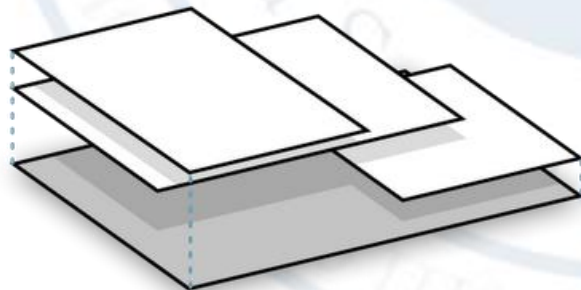
២. Relative Layout

Relative Layout គឺជាក្រុម View ដែលវាតំរៀប views កូនទៅតាមទីតាំងដែលមានទំនាក់ទំនងគ្នា រវាង view កូន និង view កូន ដែលជាបងប្អូនគ្នា ឬក៏ views កូនមានទំនាក់ទំនង ទៅនឹង parent របស់វា តាមរយៈ ID។



៣. Frame Layout

Frame Layout គឺ child views របស់វាតម្រៀបត្របដណ្តប់ ត្រួតលើគ្នា។ មានន័យថា child view ណាដែល add ចូល Frame Layout ចុងក្រោយ នោះទីតាំងរបស់វា និងត្របដណ្តប់ លើ child views ដែលនៅបន្តបន្ទាប់វា។ គេនិយមប្រើវា ជាមួយការងារ ដូចជា design ជា tabs ឬ gallery។



៣. Frame Layout (ត)

- យើងអាចបញ្ជា Child របស់វាតាម android:layout_gravity។
- Frame Layout មានលក្ខណៈស្រដៀងទៅនឹង Relative Layout ដែរ។

ចំណាំ៖

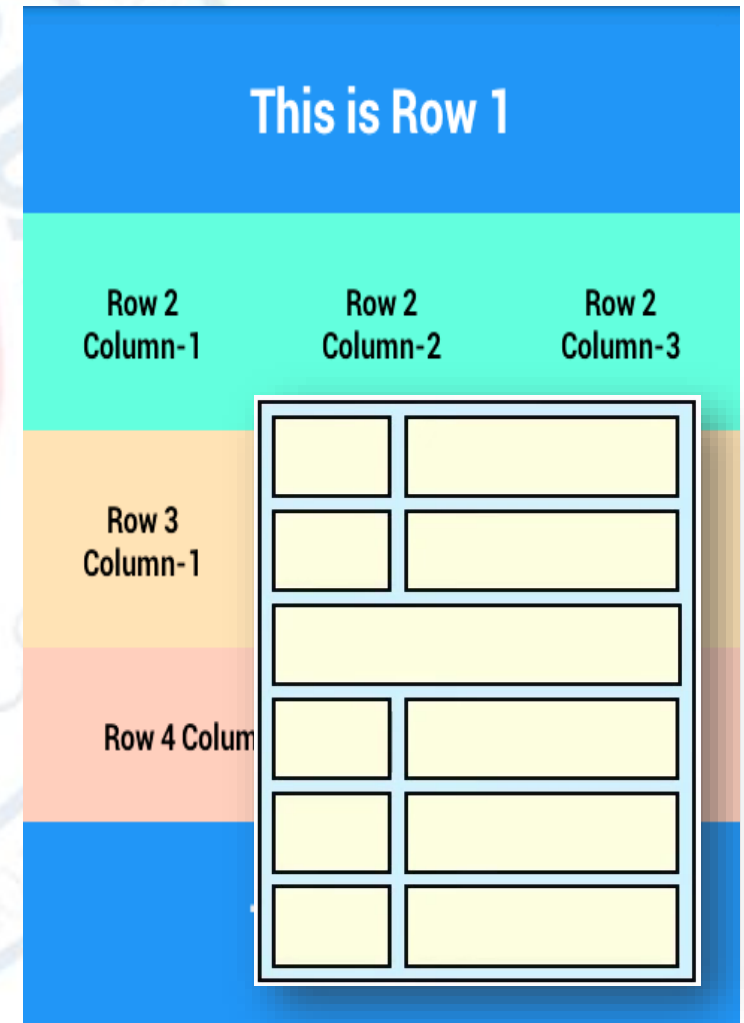
- android:layout_gravity: សម្រាប់ align ឬ កំណត់ទីតាំង View (element) ទៅកាន់ទីតាំងណា

មួយ។

- android:gravity: សម្រាប់ align ឬ កំណត់ទីតាំង Content របស់ View ទៅកាន់ទីតាំងណាមួយ។

៤. Table Layout

- Table Layout ជា groupView ដែលយើងប្រើដើម្បីរៀបចំ ឬ តម្រៀប child view elements ជា row, column ។
- Rows មួយអាចគ្មាន ឬមាន Cell ច្រើនបាន។
- ចំណាំ៖ Cell មួយអាច span បានច្រើន Columns ដូចនៅក្នុង HTML ដែរតែ មិនអាច span បានច្រើន Rows នោះទេ។
- យើងប្រើនូវ TableRow ដើម្បីផ្ទុកនូវ Control ផ្សេងៗ។



៤. Table Layout (ត)

លក្ខណរបស់ Table Layout

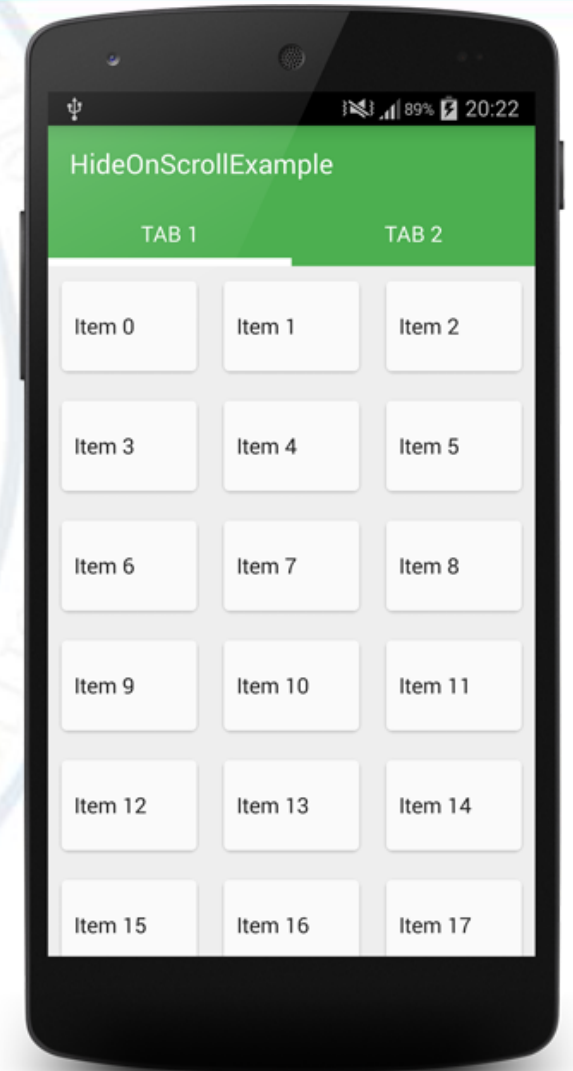
- មាន rows, column និង cells ។
- Containers របស់វា គឺ rows, column និង cells មិនមាន border ទេ។
- Table មួយមាន rows, column ជាច្រើន និង cells របស់វាផងដែរ។
- Cells របស់ table អាចពុះជា columns ជាច្រើន តែ មិនអាច ពុះជា rows បានទេ។
- យើងអាចពុះ cell មួយជា columns ជាច្រើនបាន ដោយប្រើ span field ក្នុង `TableRow.LayoutParams` class ឬក៏ប្រើ XML attributes `Android:Layout_span` ។

៥. Grid Layout

- Grid Layout គឺជា View Group ដែលបង្ហាញ items របស់វាជាទម្រង់ two-dimensional, scrollable grid ។
- Grid items ត្រូវបានដាក់បញ្ចូលដោយស្វ័យប្រវត្តិចូលទៅកាន់ Layout ដោយប្រើប្រាស់ ListAdapter។
- ចំនួន Rows និង Columns របស់វាអាចប្រកាសដោយប្រើ android:rowCount និង android:columnCount properties។

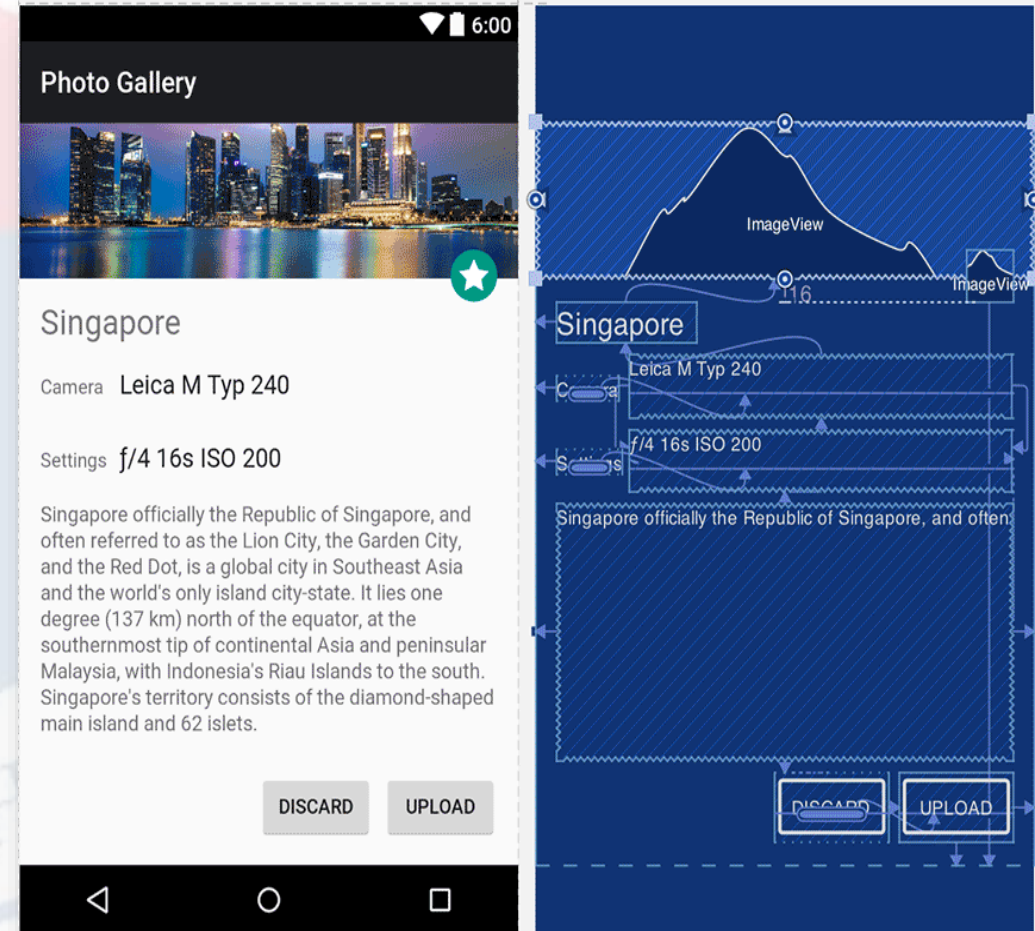
៥. Grid Layout (ត)

- មានចាប់តាំងពី Android v 4.0(Ice Cream Sandwich) ដែលវា Support rich user interface design។
- ធ្វើការរៀប Child របស់វាជាលក្ខណៈ Grid (Row and Column) ដែលស្រដៀងទៅនឹង Table Layout ដែរ ។



៦. Constraint layout

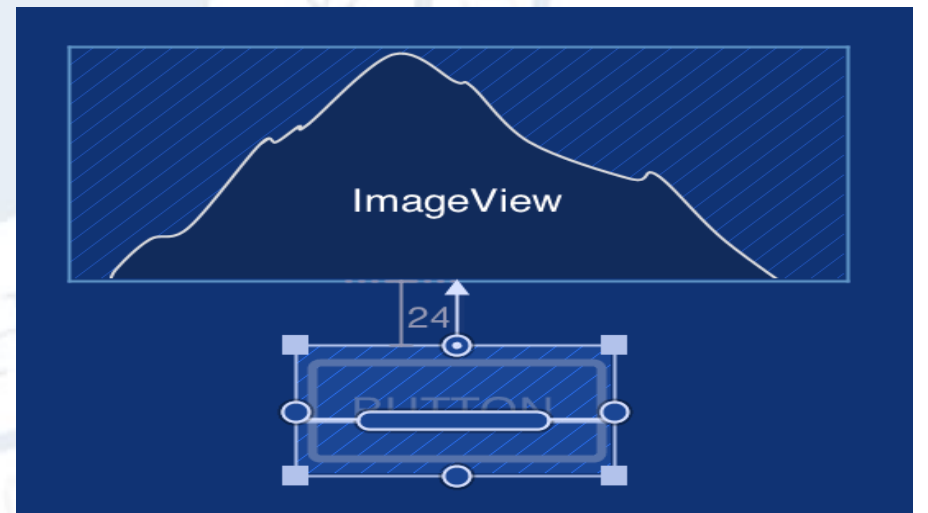
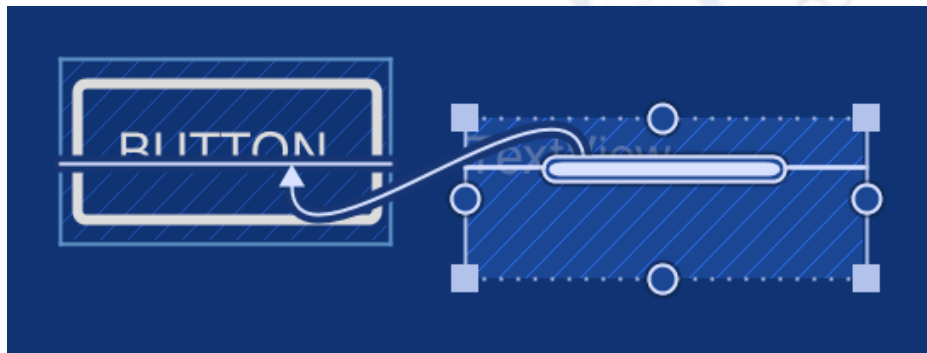
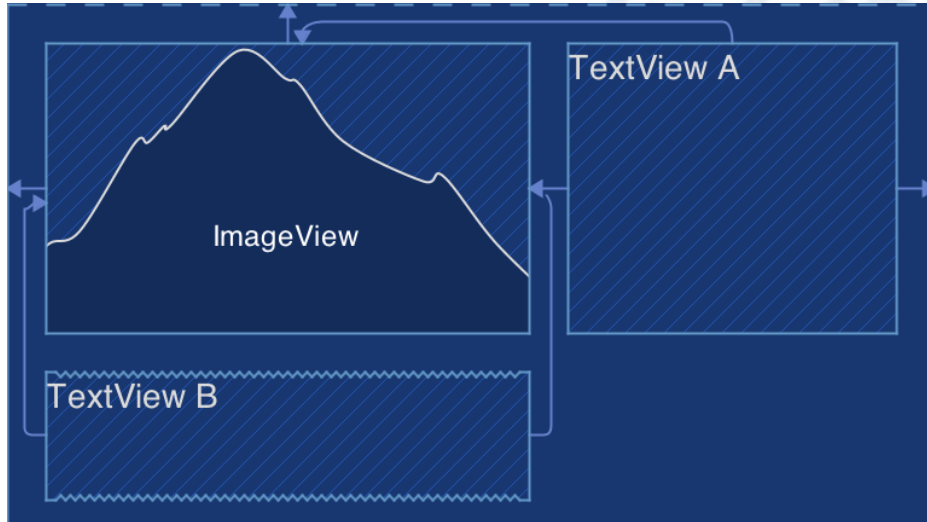
- មានតាំងពី Android Studio version 2.2 មក។
- Blue print mode
- ធ្វើការបម្លែង layout ដែលមានស្រាប់ទៅជា Constraint Layout
- មិនចាំបាច់ទៅ Edit ក្នុង XML file ដោយផ្ទាល់ក៏បាន។
- ធ្វើអោយការ Design មានលក្ខណៈលឿន។



៦. Constraint layout (ត)

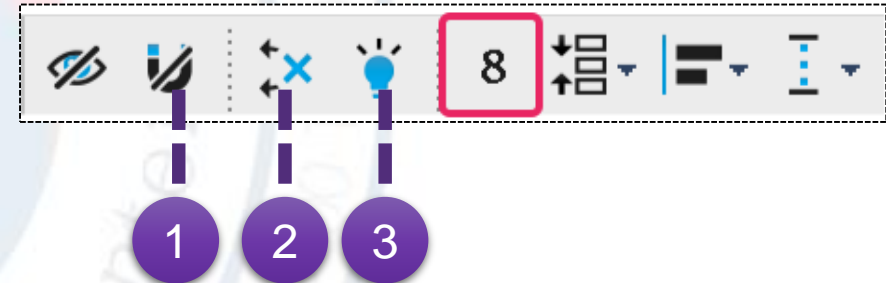
- Constraint Layout ជា Layout Editor ថ្មីដែលអនុញ្ញាតអោយយើងបង្កើត Layout ដែលមានទ្រង់ទ្រាយធំនិងមានលក្ខណៈស្មុកស្មាញបាន ជាមួយនឹង flat view hierarchy (no nested view groups) ។
- Constraint Layout មានលក្ខណៈស្រដៀងគ្នាទៅនឹង Relative Layout ដែរ ដោយវាមានធ្វើការដោយយើងទៅតាមទំនាក់ទំនងរវាង Sibling និង Parent របស់វា តែវាមានលក្ខណៈពិសេសជាងនិងងាយស្រួលប្រើប្រាស់ជាងជាមួយ Android Studio's Layout Editor ។

၁. Constraint layout (၈)



៦. Constraint layout (ត)

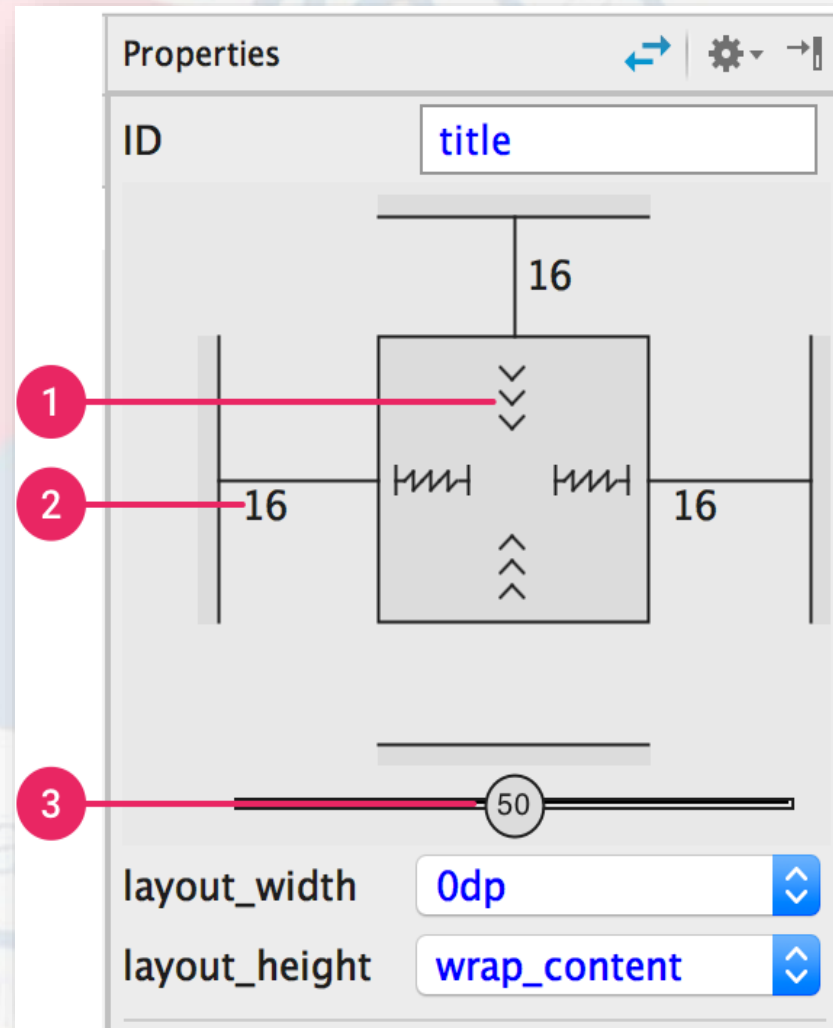
1. Autoconnect ពេល enable វានឹងបង្កើត constraint auto ឱ្យ widget នៅពេលយក widget នោះដាក់ក្នុង constraint layout។
2. លុប constraint ចេញពី View ។
3. ពេលយើង click វានឹងបង្កើត constraints ឱ្យគ្រប់ views ដែលមិនទាន់មាន constraints ។



៦. Constraint layout (ត)

និមិត្តសញ្ញា ប្រើនៅលើ Android Studio

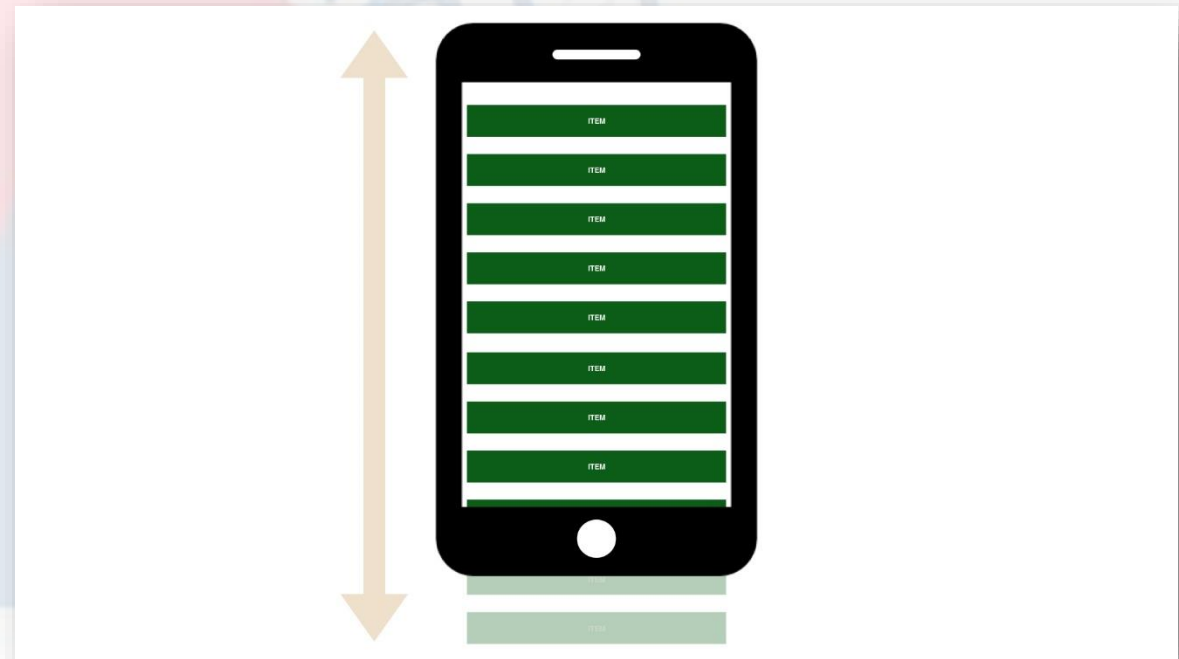
1. >>> Wrap_content
2. ៣៣៣ Any size
3. ៣ Fix size



៧. Scroll View

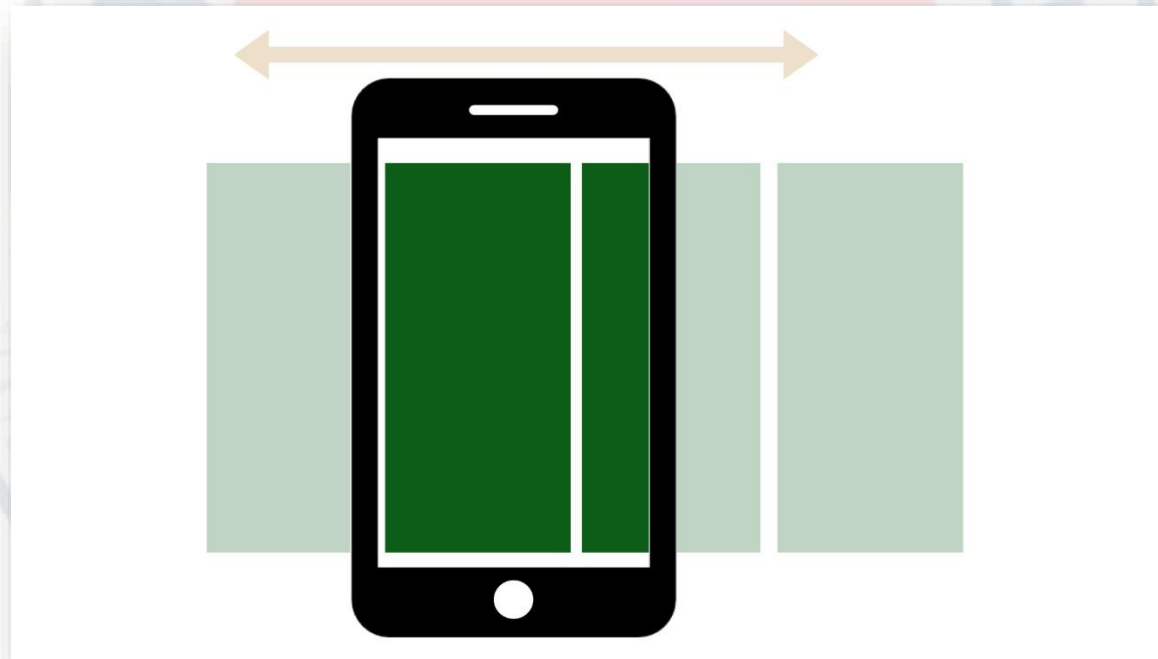
Scroll view ជា layout container អាចឱ្យ user scroll បាន នៅពេល views display ធំជាង physical display ។ លក្ខណរបស់ Scroll View ៖

- ផ្ទុក child តែមួយ ។
- Support តែ vertical scrolling ។
- គេនិយមប្រើ Linear Layout ជា child ។



៨. Horizontal Scroll View

Horizontal Scroll view ជា layout container អាចឱ្យ user scroll បាន នៅពេល child views ជា horizontal។



៩. Nested Scroll View

Nested Scroll View គឺជាការប្រើដាក់បញ្ចូលគ្នាទាំង Horizontal Scroll View និង Vertical Scroll View។

