Layouts

- Control layout of widgets
- Are laid out within screen or another layout
- Layout attributes
  - specified for child (widget)
  - vary by layout manager
  - fine tune layout of children within container
  - setLayoutParams(...)
• gravity
  - layout of content within widget
  - center_vertical, center_horizontal, left, right, top, bottom, ...
  - multiple values separated by | (e.g. top|right)
  - also setGravity(…)
• layout_gravity
  - layout of widget within container
  - as above
  - also setLayoutParams(…)

**RelativeLayout**

• Widgets laid out relative to container and each other
• Relative to container (boolean)
  - layout_alignParentTop (Left, Right, Bottom)
  - layout_centerHorizontal (Vertical, InParent)
• Relative to another widget (widget reference)
  - layout_above (below, toLeftOf, toRightOf)
  - layout_alignTop (Bottom, Left, Right, Baseline)
• Margin (dimension)
  - space between control and other control or container
  - layout_margin, layout_marginTop (Bottom, Left, Right)
• Padding (dimension)
  - space inside control between content and border
  - padding, paddingTop (Bottom, Left, Right)
  - setPadding(…)

**FrameLayout**

• Widget aligned top-left corner
  - layout can be affected by layout parameters
  - widgets overlap
• ImageView
  - widget in which an image is displayed
  - widget attributes
    * src – image source location
    * scaleType – fitting image within view
      - fitXY, center, centerInside, fitCenter
    * adjustViewBounds – adjusts view size to keep aspect ratio
    * resizeMode – allow user to resize view
      - horizontal, vertical, both, none
Example – image viewer
- FrameLayout
- overlapping ImageView
  - image scaled to fit view (doesn’t retain aspect ratio)
  - source in res/drawable-
  - folders for differing resolutions
  - runtime automatically chooses
- TextView
  - overlaps images
  - centered at bottom
  - text and background color set
- activity

TableLayout
- Widgets arranged in rows & columns
- TableRow
  - row of a table
  - 1 or more cells (columns)
  - table has max
  - columns aligned
  - any View can replace a TableRow
- Column widths
  - stretchColumns (column number(s) or *)
  - indicated columns stretch to fill parent
  - shrinkColumns (column number(s) or *)
  - indicated columns word-wrap to shrink
  - collapseColumns (column number(s) or *)
  - indicated columns reduced to 0 width
- Spanning columns
  - layout_span (>=1)

GridLayout
- New in API 14
- Widgets laid out by grid position
- rowCount, columnCount
- Attributes
  - layout_row, layout_column
  - next column (row if vertical) by default
  - layout_columnSpan, layout_rowSpan
- Spacing for empty row
  - Space View
Screen Orientation

- Two orientations
  - portrait
  - landscape
- Layout often needs to change when change orientation
  - Activity destroyed and recreated on screen orientation change
    - onPause, onResume
    - save, restore state
    - onCreate
      - recreate UI
- RelativeLayout can adapt (see RelativeLayout example)
  - relative to screen borders
- Can have two layouts and runtime selects one for onCreate
  - res/layout-land