Lecture 3 - Layout

- LinearLayout
    - lay children horizontally/vertically (orientation=”…”)
    - layout_width/layout_height:
        - fixed size: 0dp,10dp,50dp…
        - wrap_content: large just enough to fit the contents inside
        - match_parent: as large as the one contains it
    - layout_weight: determine the proportion of space for a view (default=0; can be 1,2,3…)
    - gravity/layout_gravity:
        - arrange position on screen
            - gravity: contents inside View
            - layout_gravity: View
        - default = top|left
        - possible values: left,right,top,bottom,center,center_horizontal,center_vertical,…
    - layout_margin: outer space
    - padding: inner space

- RelativeLayout
    - lay children relatively to parent/siblings
    - layout_(toLeftOf/toRightOf/above/below)=”id”: on the direction of element
    - layout_align(Left/Right/Top/Bottom)=”id”: side on side (same direction)
    - layout_alignParent(Left/Right/Top/Below)=bool: true → align to parent
    - layout_center(InParent/Horizontal/Vertical): directional center, align to parent
    - *Note: Left=Start, Right=End*

- AbsoluteLayout → set position through coordination (x,y)

- TableLayout
  - lay children into rows & cols
  - stretchColumns: make cols occupy remaining space
  - shrinkColumns: avoid overflow
  - layout_span: merge continuous cells in one row (make a fat cell)
  - layout_column: starting position of the cell in row

- ConstraintLayout → similar to RelativeLayout, but has good drag-n-drop (*recommended*)