

## Layouts

<http://developer.android.com/guide/topics/ui/declaring-layout.html>

Child views have “layout\_\*” attributes that specify how the containing layout should arrange them (see [layoutparams.png](#))

Universal layout attributes:

- layout\_width, layout\_height
- wrap\_content, match\_parent
- specific values (e.g., 30dp)

LinearLayout

- orientation
- LinearLayout.LayoutParams

RelativeLayout

- RelativeLayout.LayoutParams

GridLayout

- rowCount, columnCount
- GridLayout.LayoutParams

FrameLayout

- Used as placeholder for dynamically created fragments

## Layout Exercises

How would you create the following layouts (widgets and layouts)? [Widgets used are listed below]

Login Fragment

Map Fragment

Main Activity

Person Activity

Map Activity

## Widgets / Attributes / Events

<http://developer.android.com/guide/topics/ui/controls.html>

TextView (text labels)

- text, textAppearance
- .setClickable(boolean) – make clickable
- .setOnClickListener(View.OnClickListener)

EditText (text fields)

- inputType, ems

- .addTextChangedListener(TextWatcher)

Space (blank space)

- set layout\_width and layout\_height to specific values (e.g., 30dp)

Button

- text

- .setOnClickListener(View.OnClickListener)

ImageView (display image)

- .setClickable(boolean) – make clickable

- .setImageDrawable(Drawable) – set icon to display

- .setOnClickListener(View.OnClickListener)

Switch (on/off)

- .setChecked(boolean) – set check state

- .setOnCheckedChangeListener(CompoundButton.OnCheckedChangeListener)

Spinner (dropdown list)

- .setAdapter(ArrayAdapter) – specify list values

- .setSelection(int) – specify selected item

- onItemSelectedListener(AdapterView.OnItemSelectedListener)

SearchView

- .setFocusable(boolean) – accept key focus

- .setIconified(boolean) – make always visible

- .requestFocusFromTouch() – request focus when touched

- .setQueryTextListener(SearchView.OnQueryTextListener)

ScrollView

- Wrap around any view to make it scrollable

RecyclerView

- Dynamic lists of items

\* ListView / ExpandableListView