

Native Mobile Applications

TalTech, Andres Käver, 2020-2021, Fall semester

Skype: akaver Email: akaver@itcollege.ee



Layout fundamentals

2

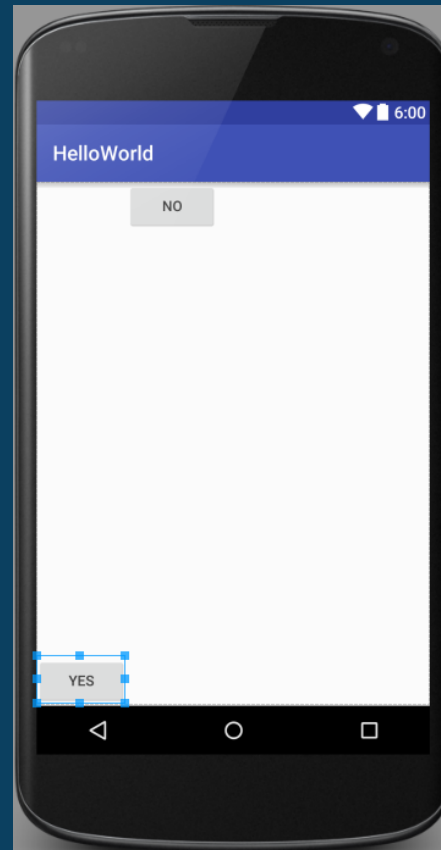
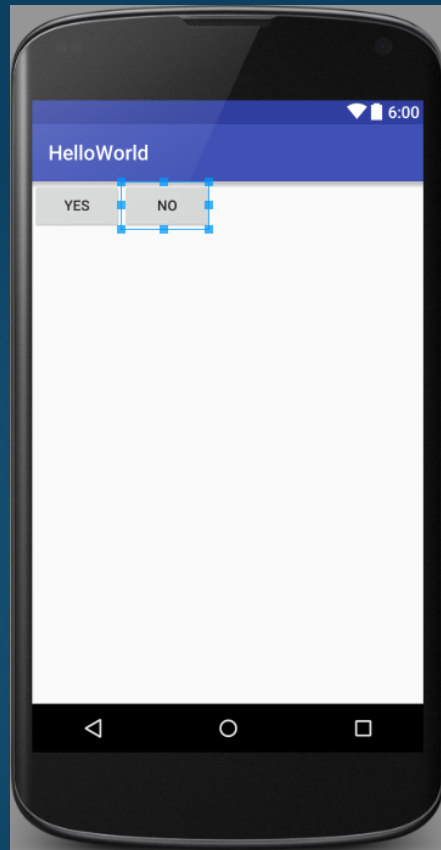
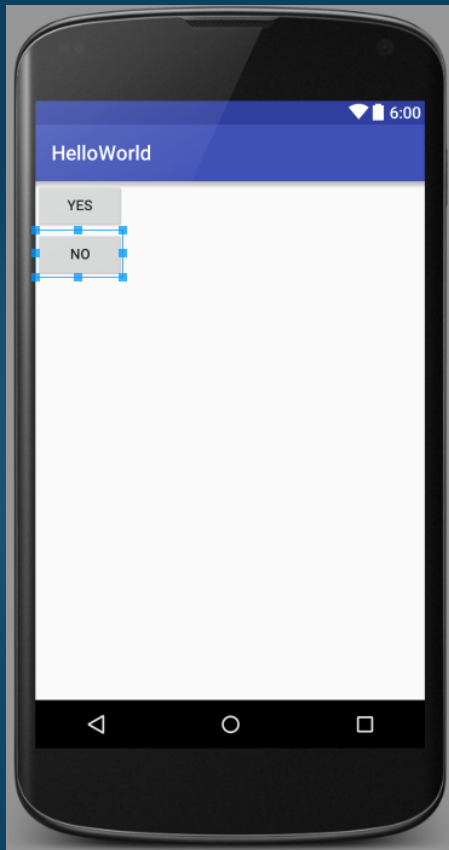
- ▶ Basic layout
 - ▶ Linear
 - ▶ Constraint
 - ▶ Grid
- ▶ Basic attributes
 - ▶ Size
 - ▶ Margin vs padding
 - ▶ Gravity

Layout

- ▶ Defines declaratively visual structure for app
- ▶ Takes into consideration screen properties
 - ▶ size
 - ▶ pixel density
- ▶ System calculates sizes and position for all UI elements

Layout

4



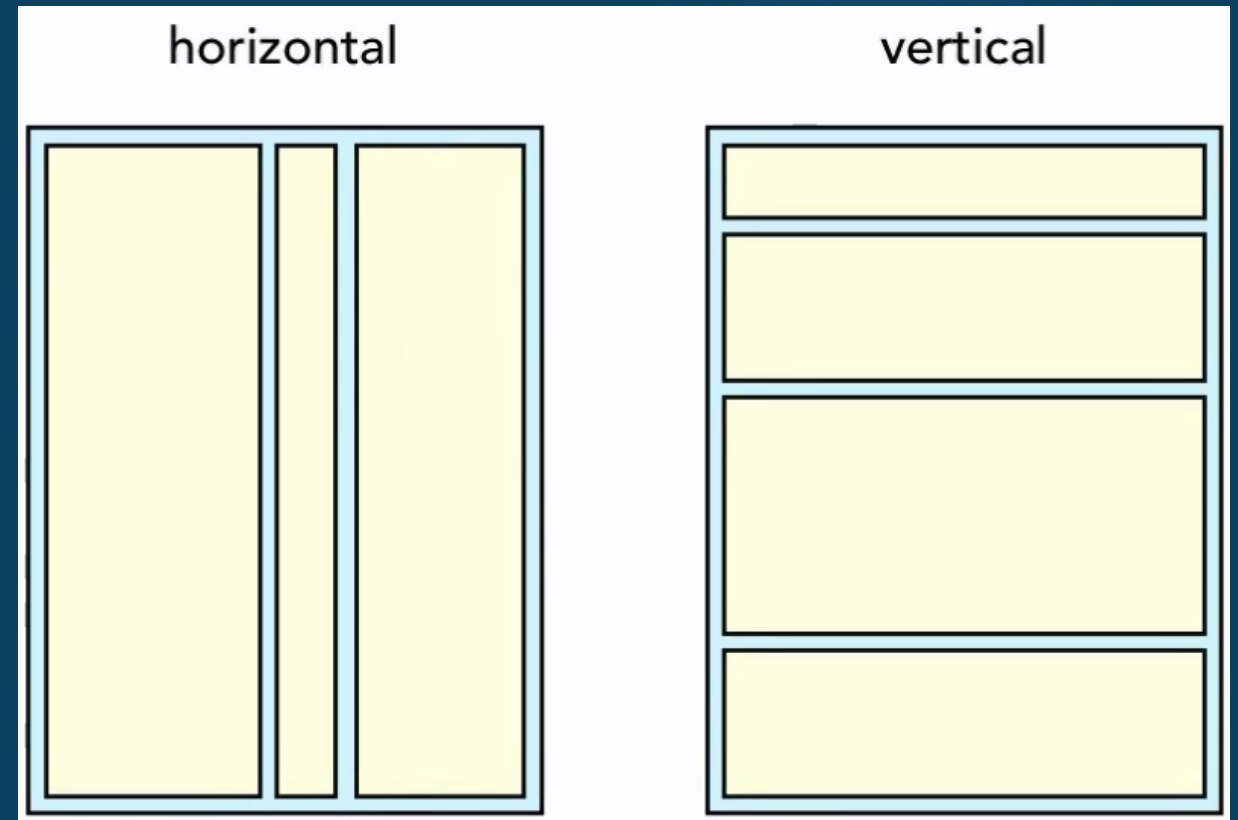
Layout - LinearLayout

5

- ▶ Arranges its children in single direction
 - ▶ Orientation="horizontal"
 - ▶ Orientation="vertical"

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    ...ui elements...
</LinearLayout>
```



Basic attributes

6

- ▶ Used across all layouts
 - ▶ Size
 - ▶ Margin vs padding
 - ▶ Gravity

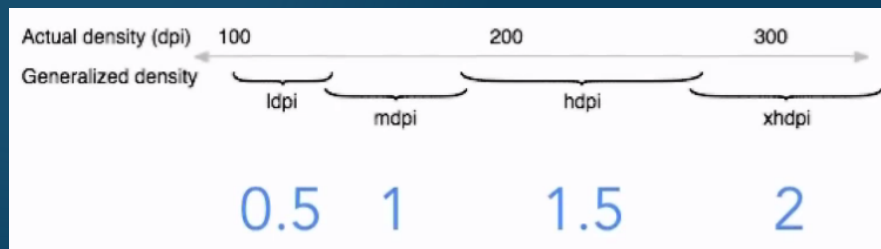
Size

7

- ▶ Match_parent
- ▶ Wrap_content
- ▶ Use background tint for visual cues

```
android:background="#ff0000"
```

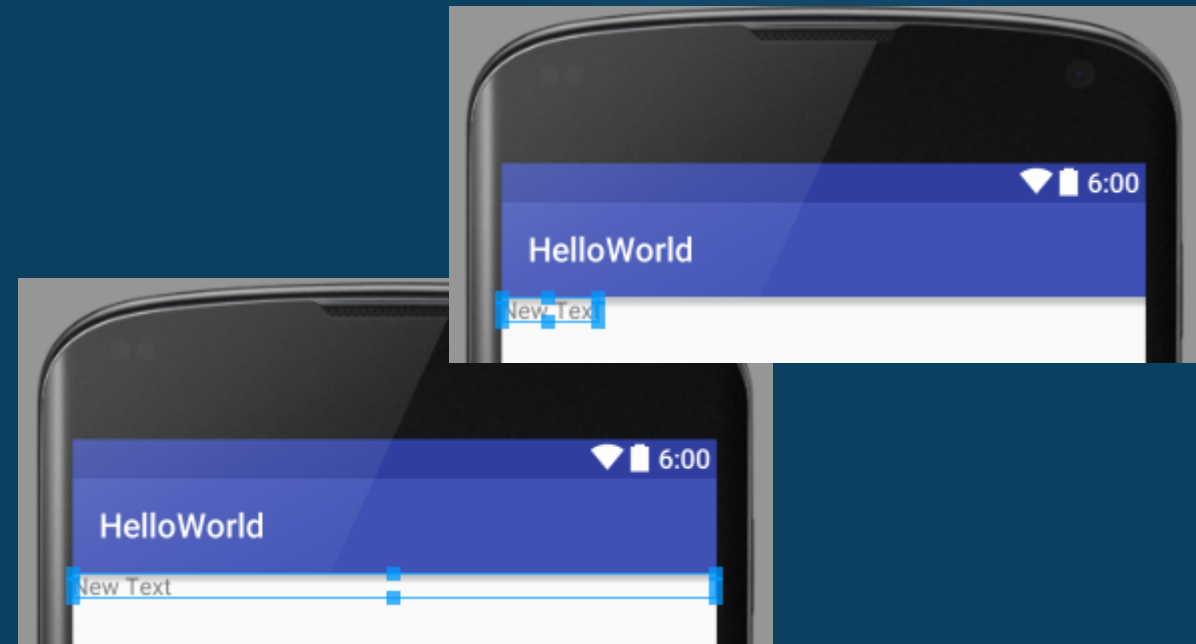
- ▶ Xxdp
 - ▶ Density-independent pixel
 - ▶ $px = dp * density$
 - ▶ $density = bucket(dpi/160)$



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:id="@+id/textView2" />

</LinearLayout>
```



Size - dp

8

Normal density

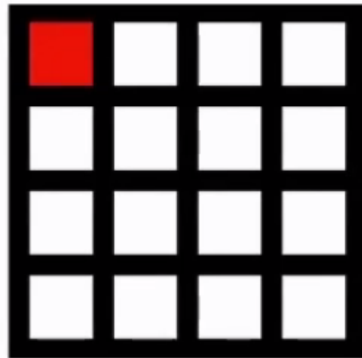


1px

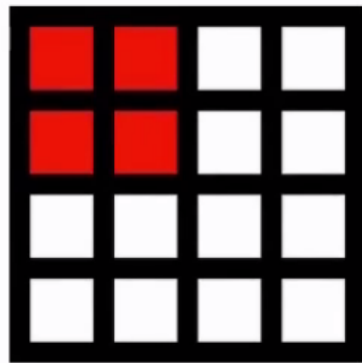


1dp

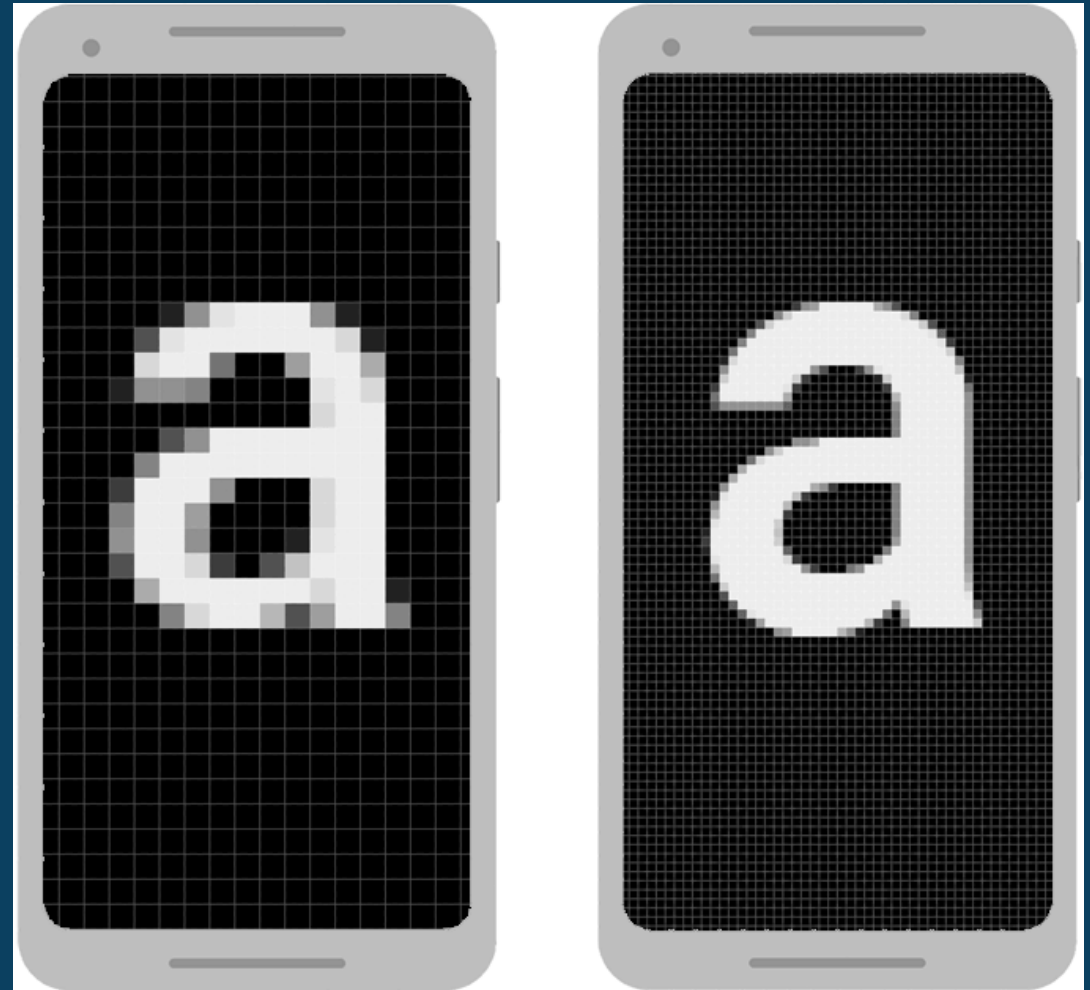
Double density



1px

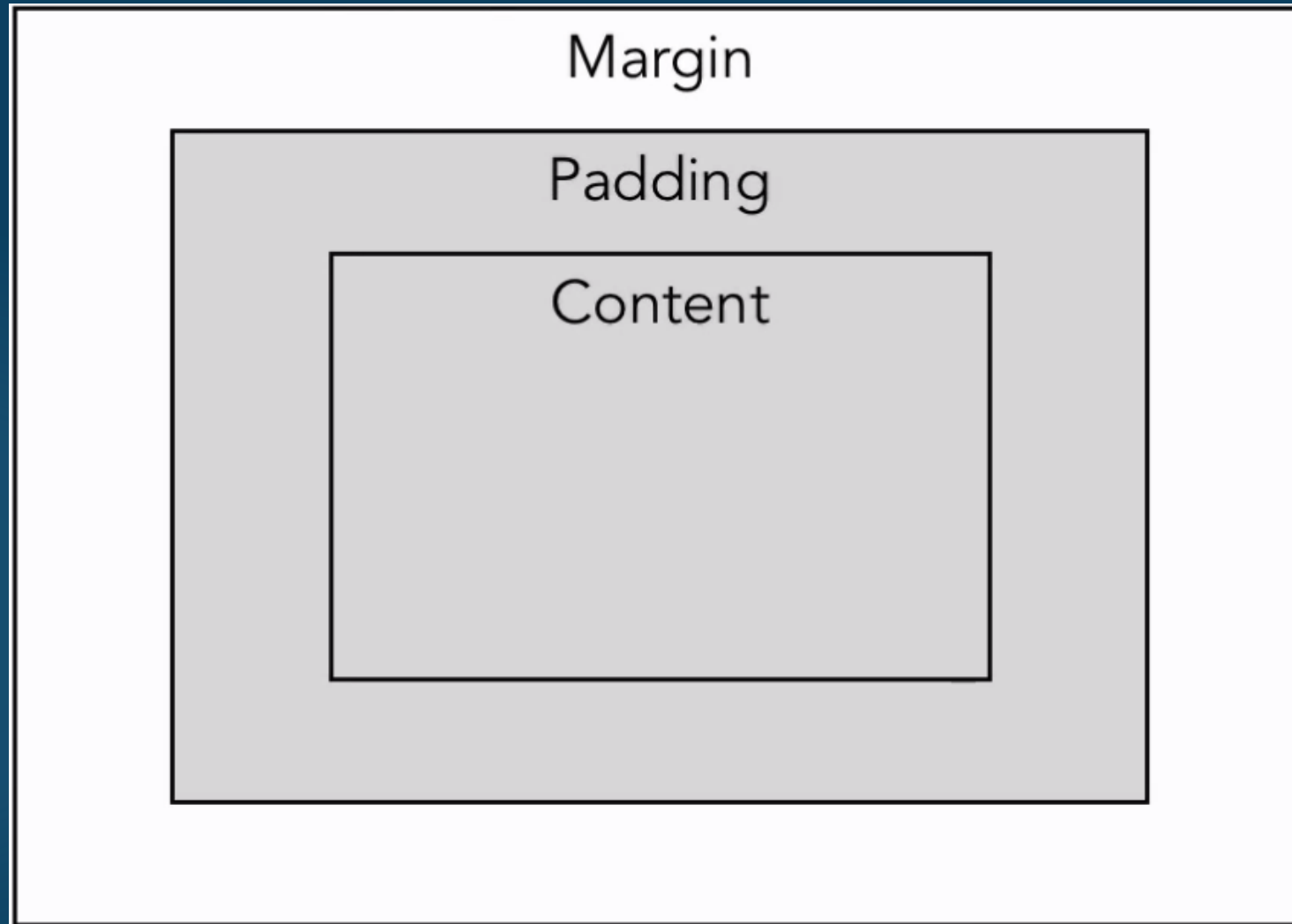


2px



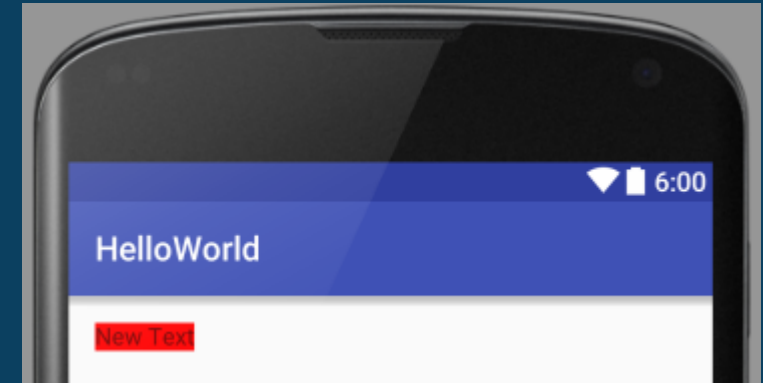
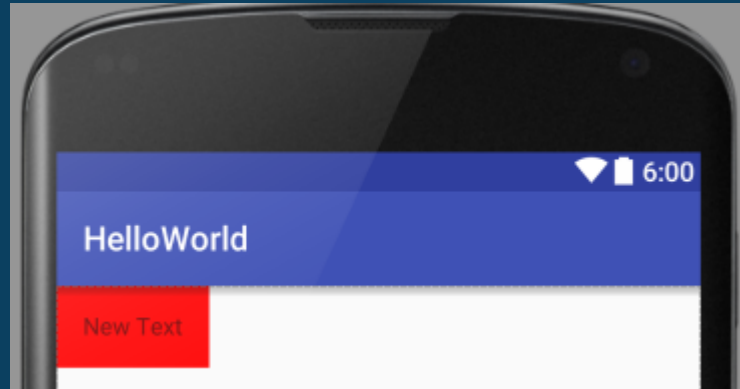
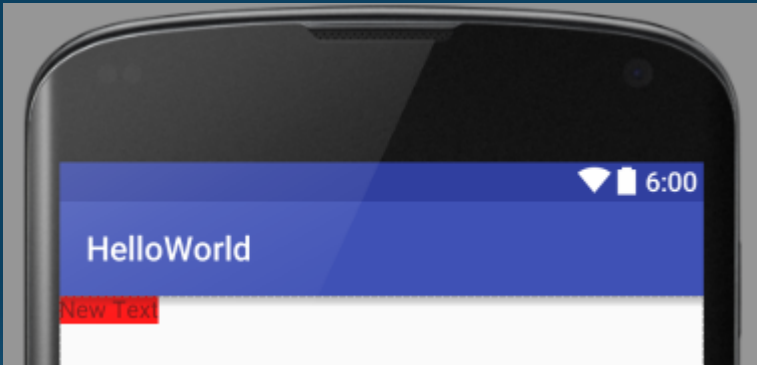
Margin vs padding

9



Margin vs padding

10



- ▶ No margin
- ▶ `android:padding`
- ▶ `android:layout_margin`
- ▶ `layout_xxx` prefix – deals with data from outside the view
- ▶ padding – inside the view

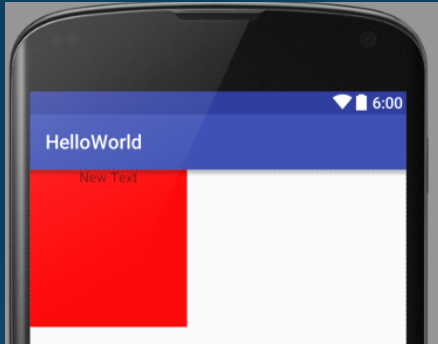
Gravity

11

- ▶ `Android:layout_gravity`
 - ▶ Position of the view, regarding its parent
- ▶ `Android:gravity`
 - ▶ Position of the content inside the view

Gravity

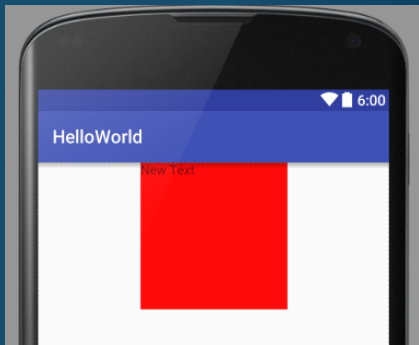
12



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <TextView
    android:layout_width="160dp"
    android:layout_height="160dp"
    android:gravity="center_horizontal"
    android:background="#ff0000"
    android:text="New Text"
    android:id="@+id/textView2" />

</LinearLayout>
```



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <TextView
    android:layout_width="160dp"
    android:layout_height="160dp"
    android:layout_gravity="center_horizontal"
    android:background="#ff0000"
    android:text="New Text"
    android:id="@+id/textView2" />

</LinearLayout>
```

Gravity

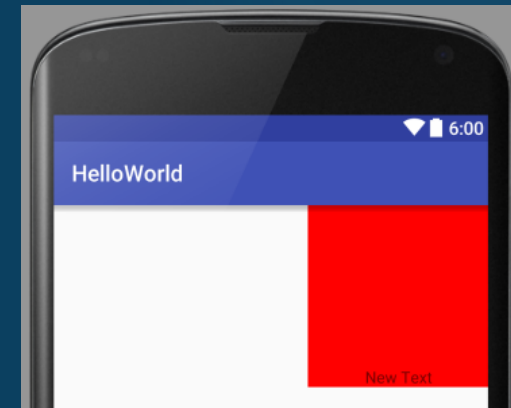
13

- Can be mixed and combined

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <TextView
    android:layout_width="160dp"
    android:layout_height="160dp"
    android:gravity="center_horizontal|bottom"
    android:layout_gravity="right"
    android:background="#ff0000"
    android:text="New Text"
    android:id="@+id/textView2" />

</LinearLayout>
```



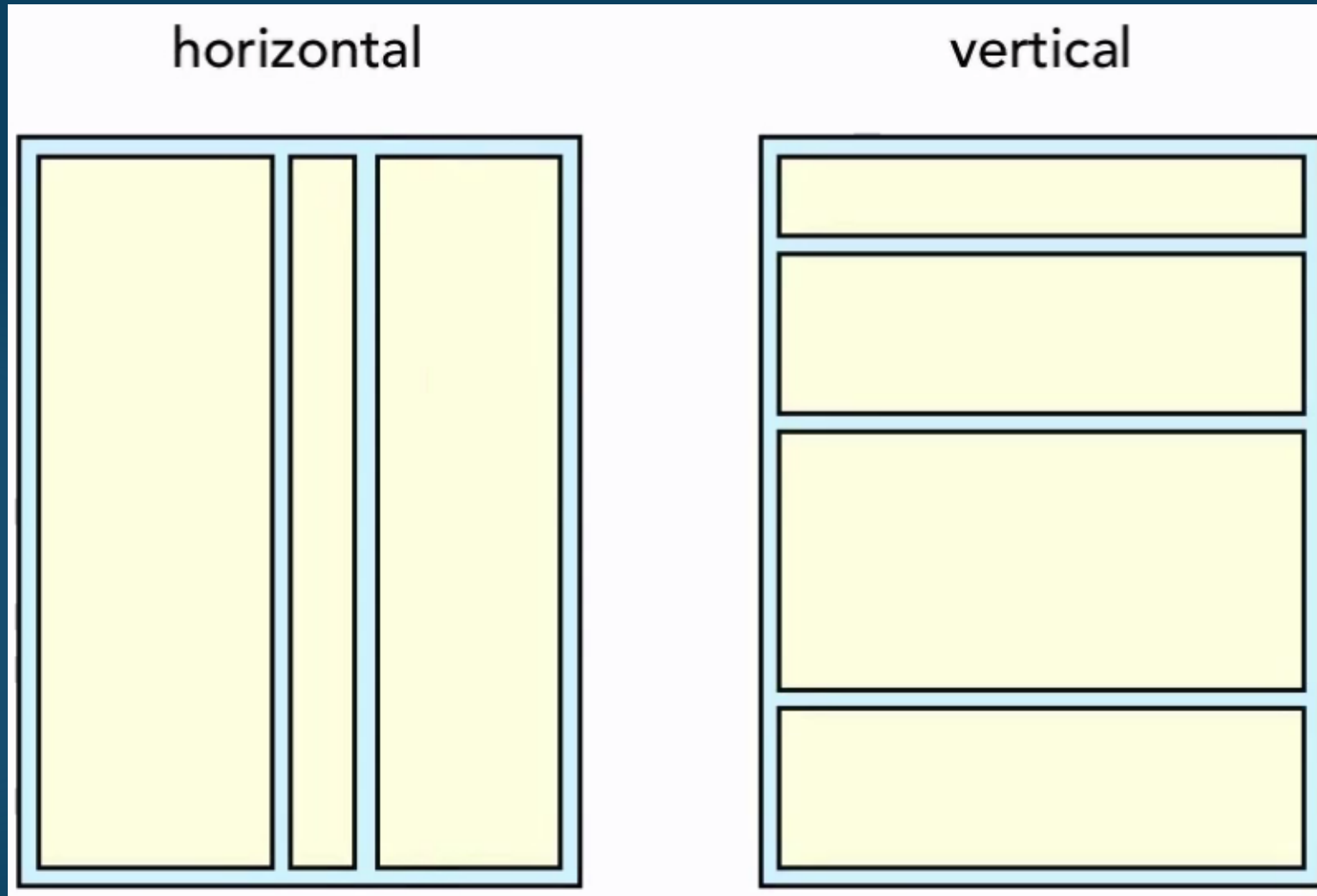
LinearLayout

14

- ▶ What is LinearLayout?
- ▶ Gravity
- ▶ Weight
- ▶ Nested layout

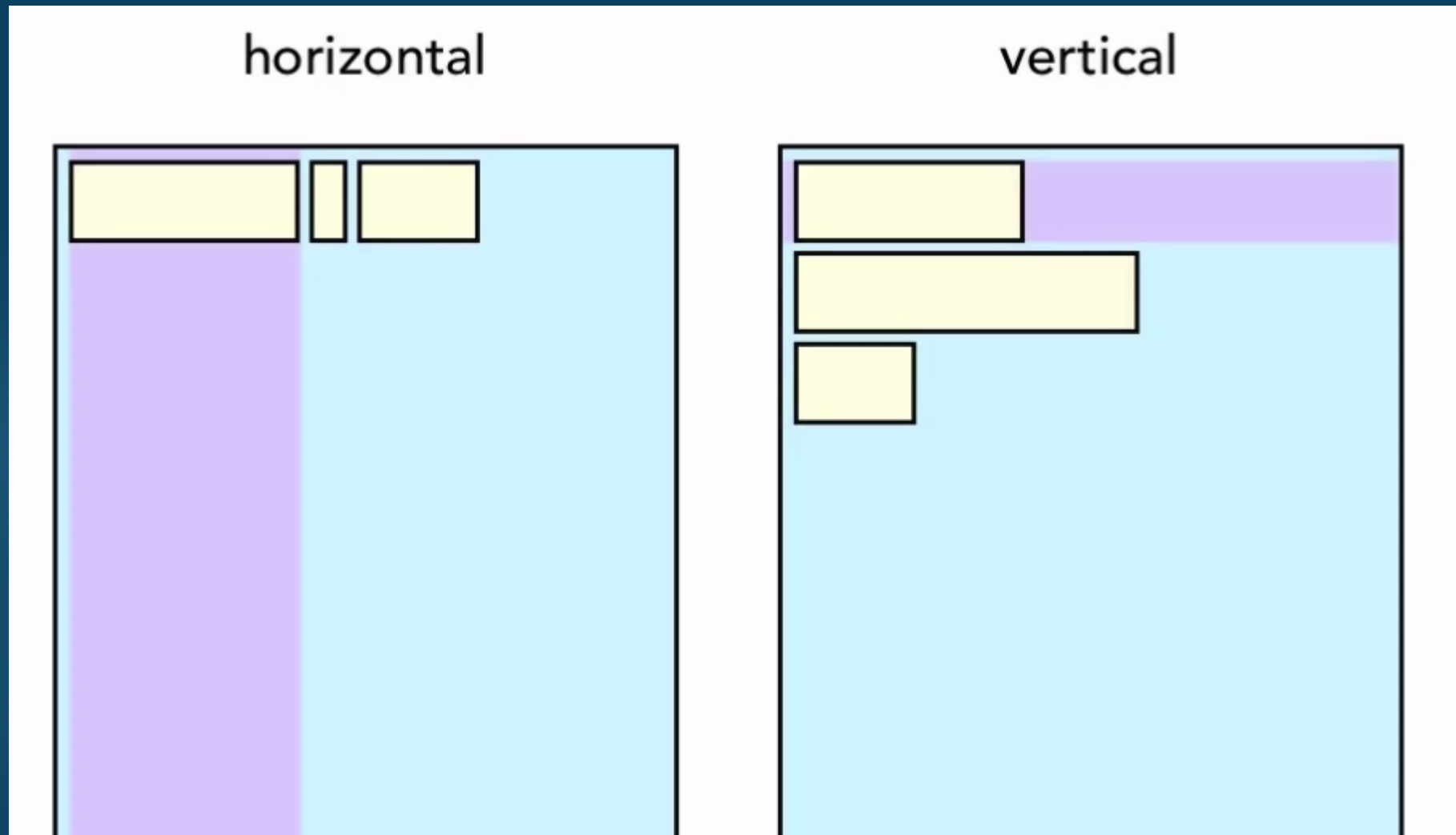
LinearLayout

15



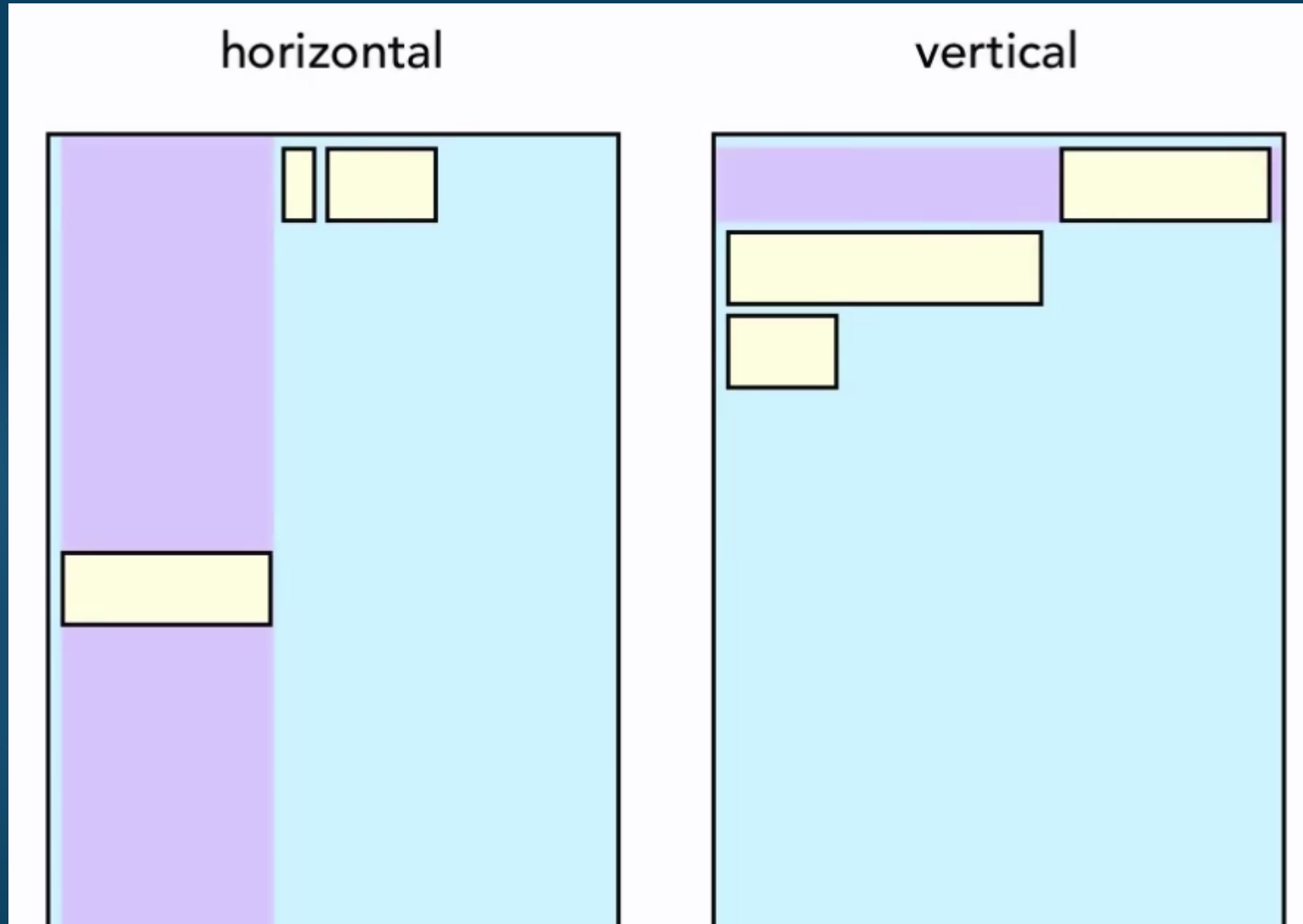
LinearLayout - Gravity

16



LinearLayout - Gravity

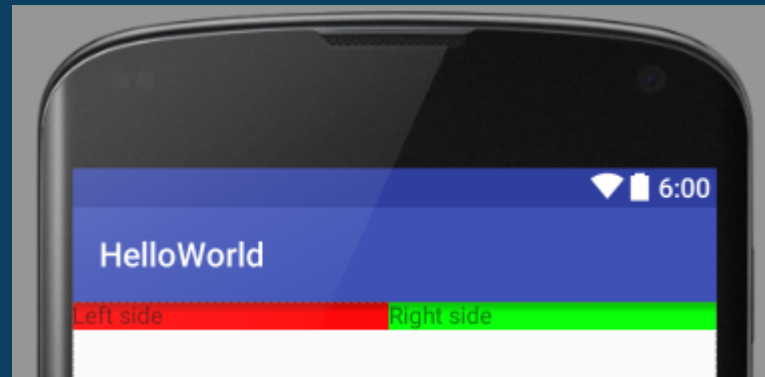
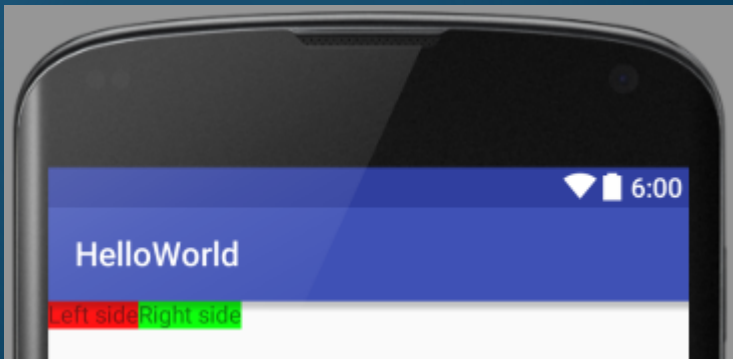
17



LinearLayout - Weight

18

- ▶ Additional way to specify dimensions (match_parent, wrap_content, xxxdp)
- ▶ Adds weights together
- ▶ Width is based on ratios ($1+1=2 \Rightarrow \frac{1}{2}$ and $\frac{1}{2}$)
- ▶ Can be mixed every way



LinearLayout - weightSum

19

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:weightSum="4">

    <TextView
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:background="#ff0000"
        android:text="Left side"
        android:id="@+id/textView2" />

    <TextView
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:background="#00ff00"
        android:text="Right side"
        android:id="@+id/textView3" />

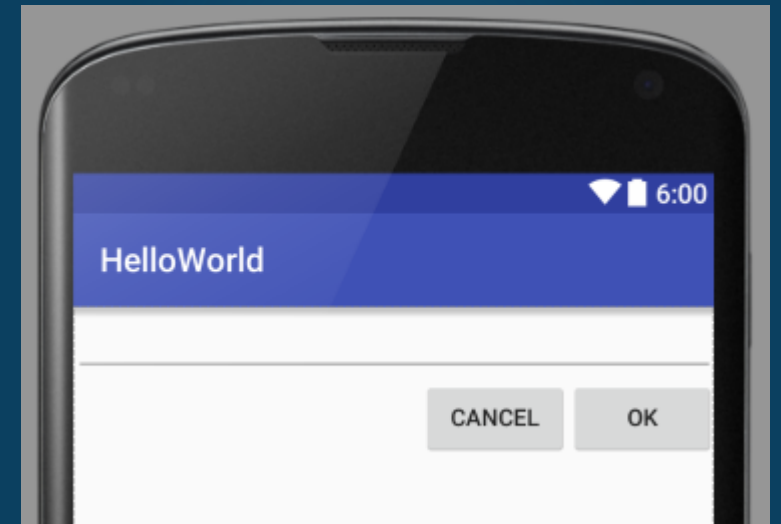
</LinearLayout>
```



LinearLayout - nested

20

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText3" />
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="right">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Cancel"
            android:id="@+id/button6" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="OK"
            android:id="@+id/button7" />
    </LinearLayout>
</LinearLayout>
```



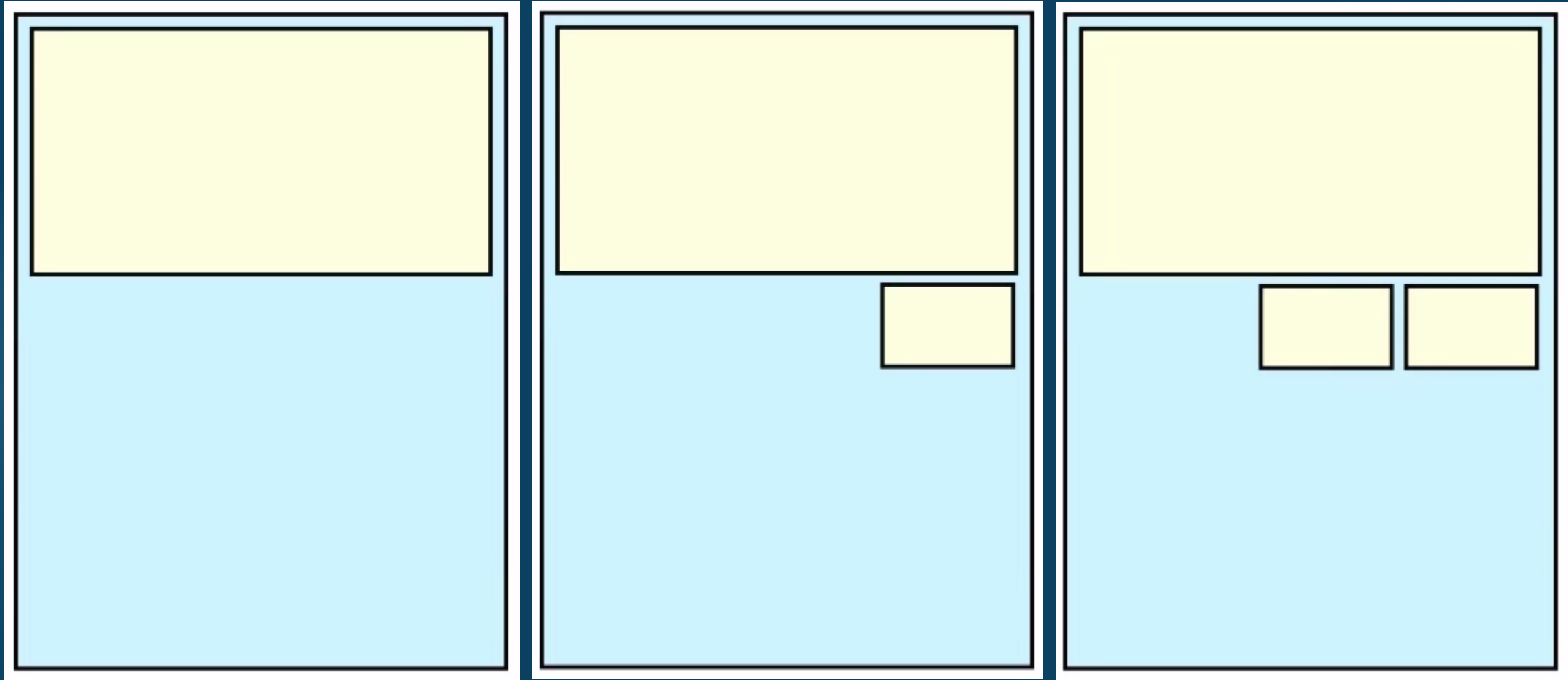
ConstraintLayout

21

- ▶ What it is?
- ▶ Relative Positioning
- ▶ Relative Alignement
- ▶ Missing Views

ConstraintLayout

22



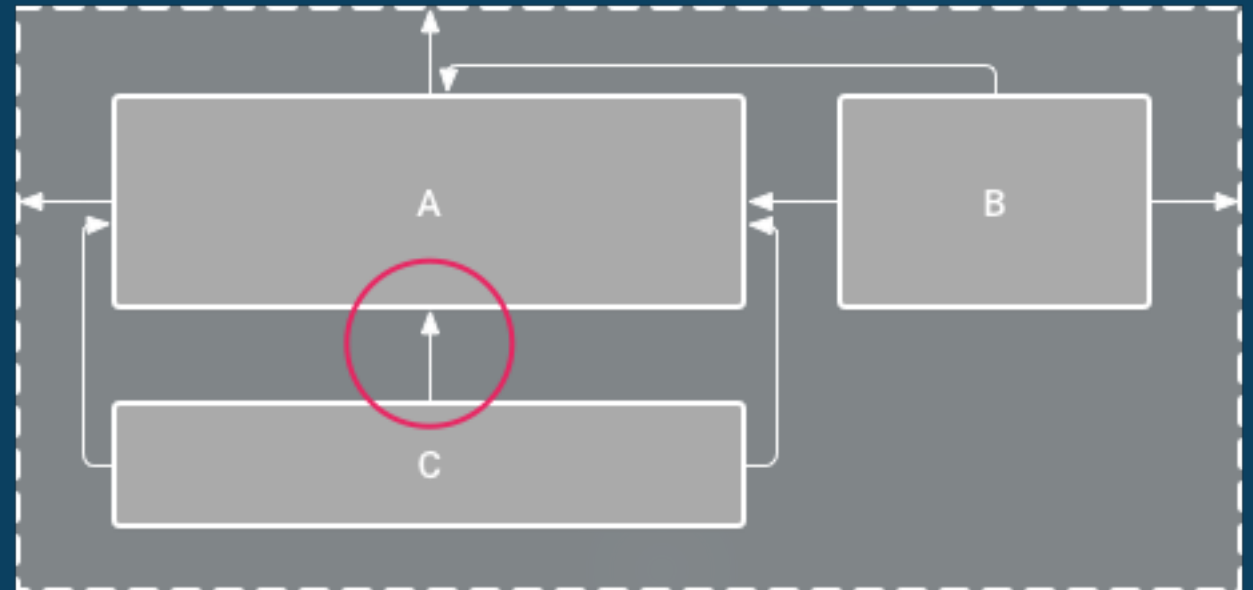
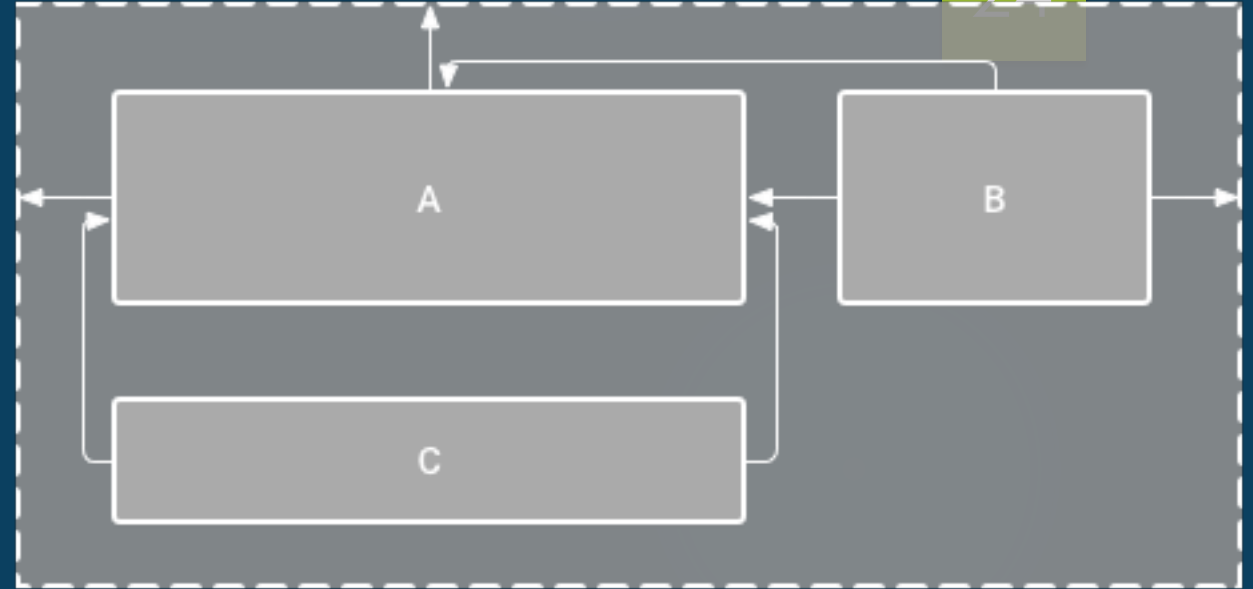
ConstraintLayout

23

- ▶ ConstraintLayout allows you to create large and complex layouts with a flat view hierarchy (no nested view groups) - performant.
- ▶ Relative
 - ▶ Position
 - ▶ Alignment
- ▶ To
 - ▶ Parent
 - ▶ Sibling

ConstraintLayout

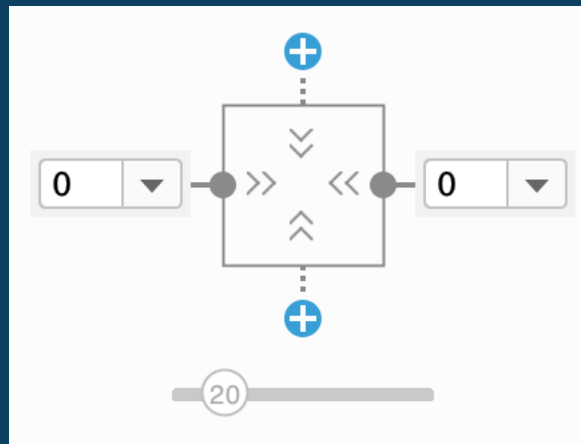
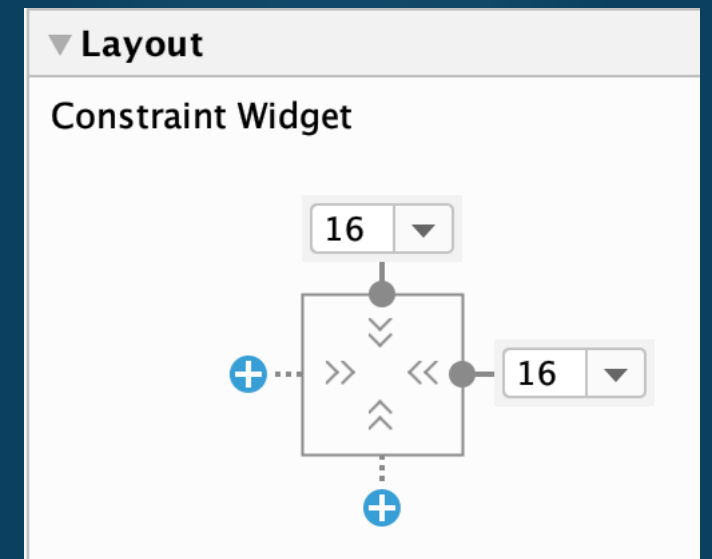
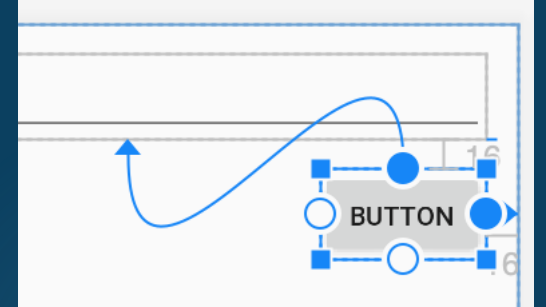
- ▶ Every element needs at least one horizontal and one vertical constraint.
- ▶ Constraint can connect to
 - ▶ Another element
 - ▶ Parent layout
 - ▶ Guideline



ConstraintLayout

25

- ▶ Add constraint – drag from the connection handle (the circle)
- ▶ Or click the + sign in Attributes panel
- ▶ Each constraint handle can be used for just one constraint
- ▶ When you add opposing constraints to fixed size element, element will be centered. Change bias via slider.



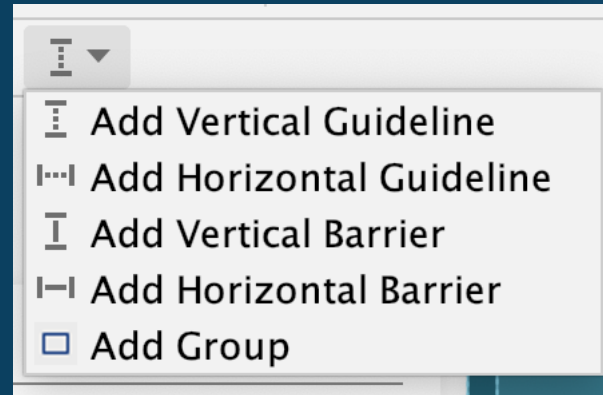
ConstraintLayout

► Guideline

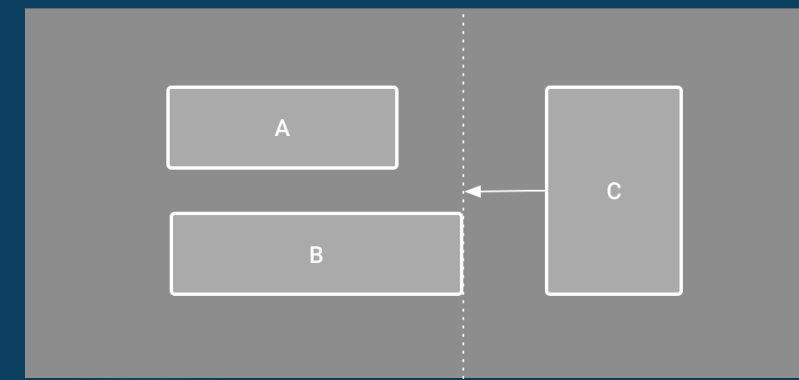
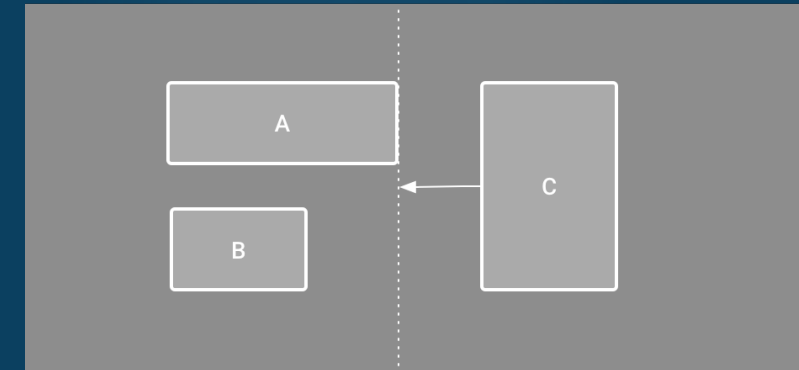
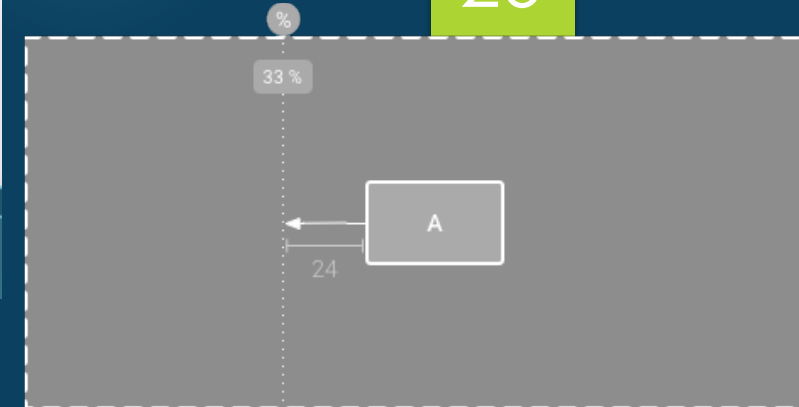
- You can add a vertical or horizontal guideline to which you can constrain views, and the guideline will be invisible to app users.
- You can position the guideline within the layout based on either dp units or percent, relative to the layout's edge.

► Barrier

- Invisible line that you can constrain views to.
- Except a barrier does not define its own position; instead, the barrier position moves based on the position of views contained within it.



26

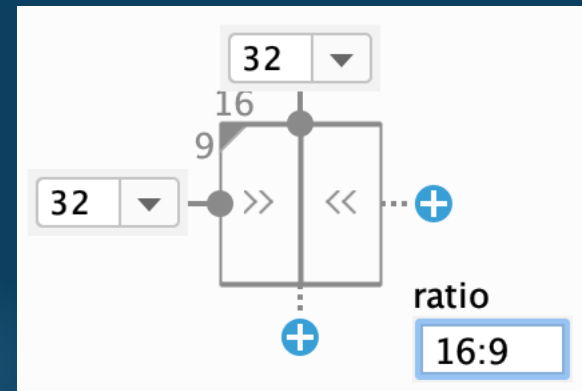
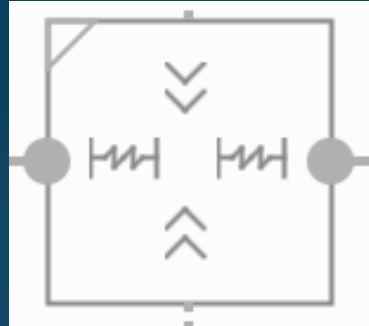


ConstraintLayout

27



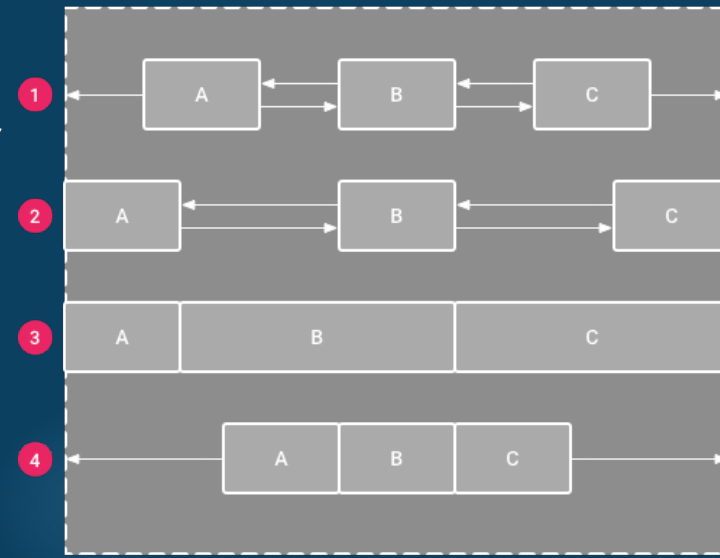
- ▶ View size
- ▶ Fixed: You specify a specific dimension in the text box below or by resizing the view in the editor.
- ▶ Wrap Content: The view expands only as much as needed to fit its contents.
- ▶ Match Constraints: The view expands as much as possible to meet the constraints on each side (after accounting for the view's margins).
- ▶ Size as ratio – (click the upper left corner)
- ▶ You can set the view size to a ratio such as 16:9 if at least one of the view dimensions is set to "match constraints" (0dp).



ConstraintLayout

28

- ▶ A chain is a group of views that are linked to each other with bi-directional position constraints.
- ▶ Select elements to be in chain, right click, select Chains. Chain head sets the style: spread, spread inside, weighted, packed
- ▶ Spread: The views are evenly distributed (after margins are accounted for). This is the default.
- ▶ Spread inside: The first and last view are affixed to the constraints on each end of the chain and the rest are evenly distributed.
- ▶ Weighted: When the chain is set to either spread or spread inside, you can fill the remaining space by setting one or more views to "match constraints" and use `layout_constraintHorizontal_weight` and `layout_constraintVertical_weight`
- ▶ Packed: The views are packed together (after margins are accounted for)



ConstraintLayout

29

- ▶ Missing views - Visibility
 - ▶ Invisible – view is hidden, layout stays the same
 - ▶ Gone – view is semi-removed, layout falls apart

Reusing layouts

30

- ▶ To efficiently re-use complete layouts, you can use the `<include/>` and `<merge/>` tags to embed another layout inside the current layout.
`<include layout="@layout/game_board"/>`
- ▶ Override layout parameters (any `android:layout_*` attributes) of the included layout's root view by specifying them in the `<include/>` tag.
- ▶ You must override both `android:layout_height` and `android:layout_width` in order for other layout attributes to take effect.

Reusing layouts

31

- ▶ The `<merge />` tag helps eliminate redundant view groups in your view hierarchy when including one layout within another.
- ▶ Include this layout in another layout (using the `<include/>` tag) - the system ignores the `<merge>` element and places the two buttons directly in the layout, in place of the `<include/>` tag.

```
<merge xmlns:android="http://schemas.android.com/apk/res/android">
  <Button
    android:layout_width "fill_parent"
    android:layout_height "wrap_content"
    android:text "@string/add"/>
  <Button
    android:layout_width "fill_parent"
    android:layout_height "wrap_content"
    android:text "@string/delete"/>
</merge>
```

Android

32

▶ The END!