

ЛЕКЦИЯ I © 2021 Нет Ит

# ANDROID:

# Въведение. Views

## Теодор Костадинов



SOFTWARE  
ACADEMY



# Съдържание



1. Преговор
2. ViewGroups
3. Layouts
4. Demo
5. Homework



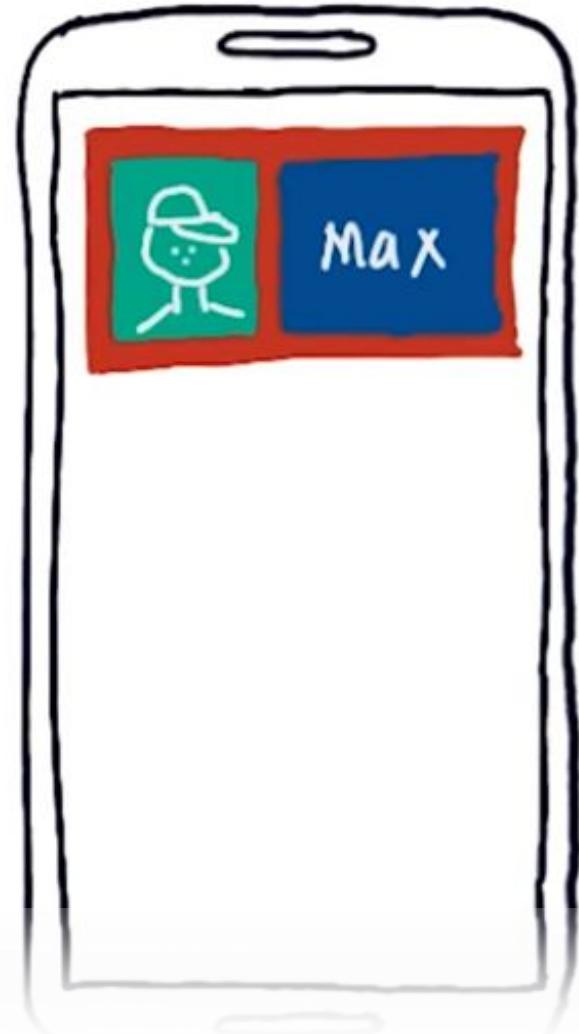
ПРЕГОВОР



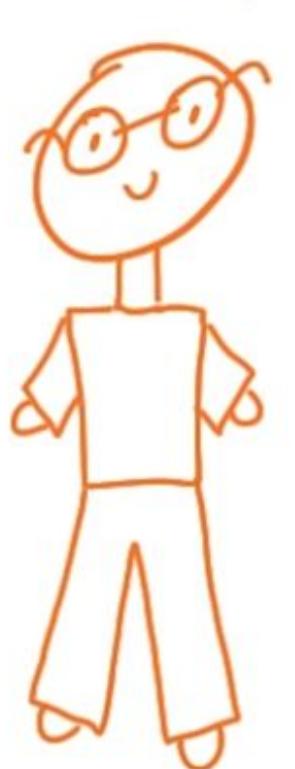
# VIEW GROUPS

- Използват се, за да групират други view-та и да им казват **как да се разполагат върху экрана**
- View-та, които са в даден ViewGroup, се наричат негови деца, а ViewGroup-ата е техен родител
- View-та, които са в един и същ ViewGroup, са **siblings**
- Използват се различни layout-и, за да се постигнат различни резултати

# VIEW GROUPS



# MEET THE VIEWS



Ray  
RelativeLayout  
ViewGroup



Laura  
LinearLayout  
ViewGroup



Tommy  
TextView



Ivy  
ImageView

Children Views

Parent Views



# LAYOUTS



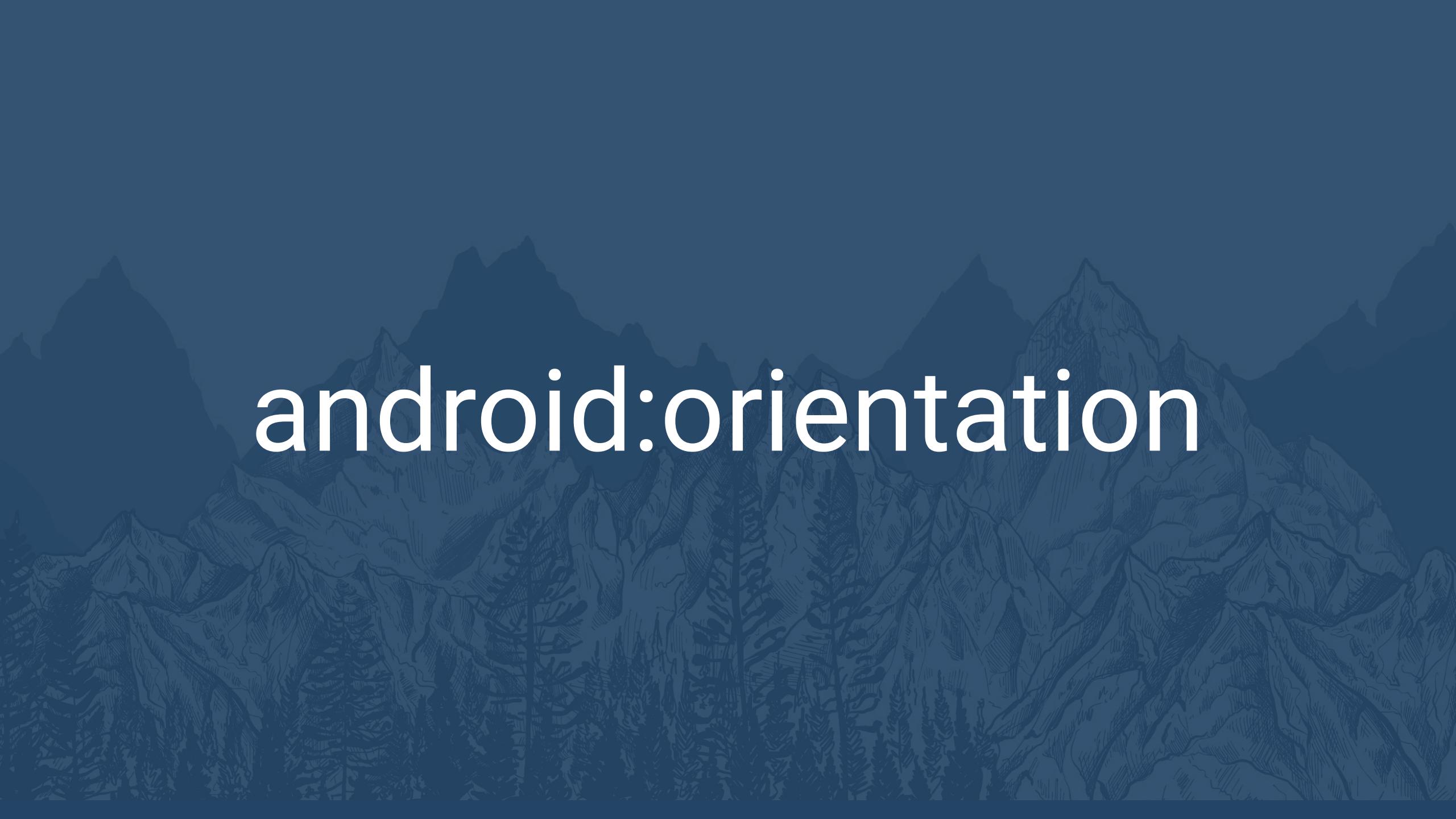
# LinearLayout

# Какво представлява?

- ViewGroup-а, която подравнява децата си с едно направление - вертикално или хоризонтално
- Децата в този layout се нареждат едно след друго/едно под друго в зависимост от зададената ориентация



децата са едно след друго



android:orientation

# PARENTAL SUPERVISION



Laura

LinearLayout



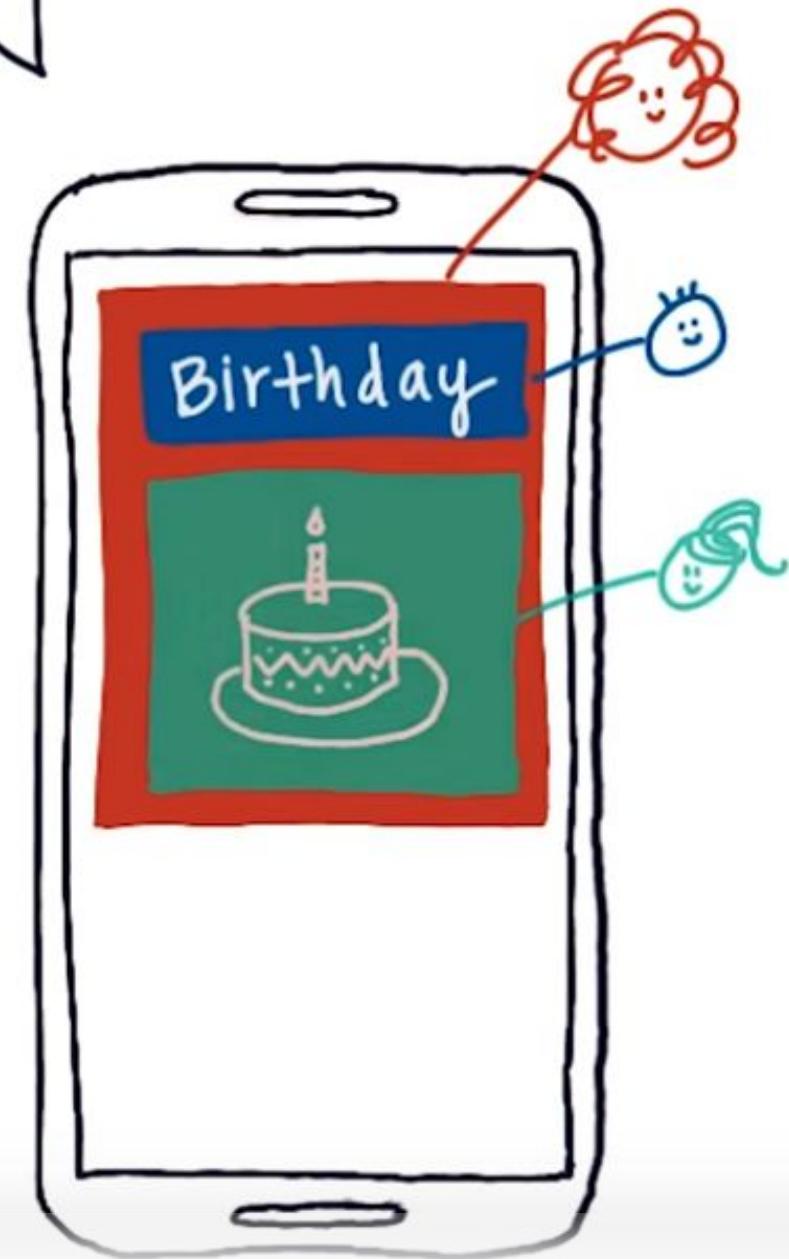
Tommy

TextView



Ivy

ImageView



# VIEW GROUPS

1. How many views are there? 4

2. The mountain image is a \_\_\_\_\_ of the red ViewGroup.

- child
- parent

3. The red ViewGroup is the \_\_\_\_\_ of the "Hiking" TextView.

- child
- parent

4. Which views are siblings of each other?

mountain ImageView, hiking TextView, we hiked up the mounta



# LINEAR LAYOUT

vertical column



Laura

LinearLayout



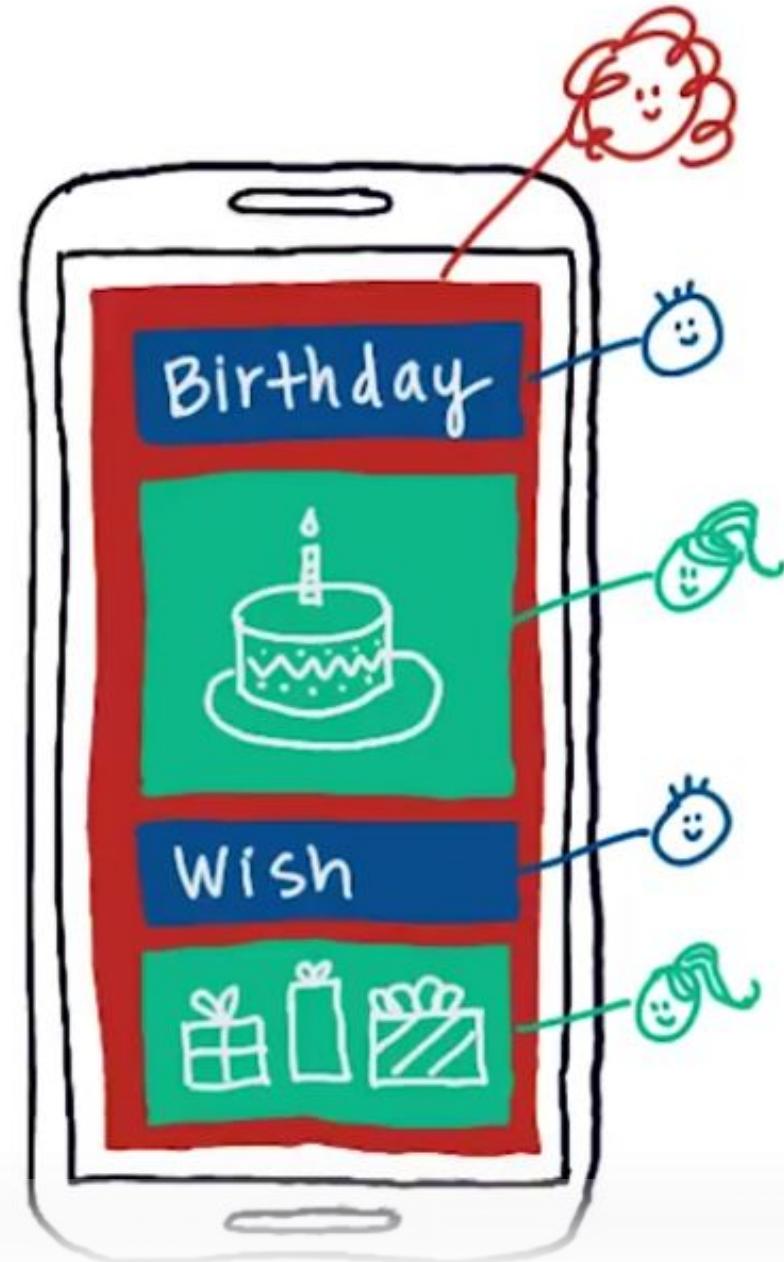
Tommy

TextView



Ivy

ImageView



# LINEAR LAYOUT



Laura

Linear Layout

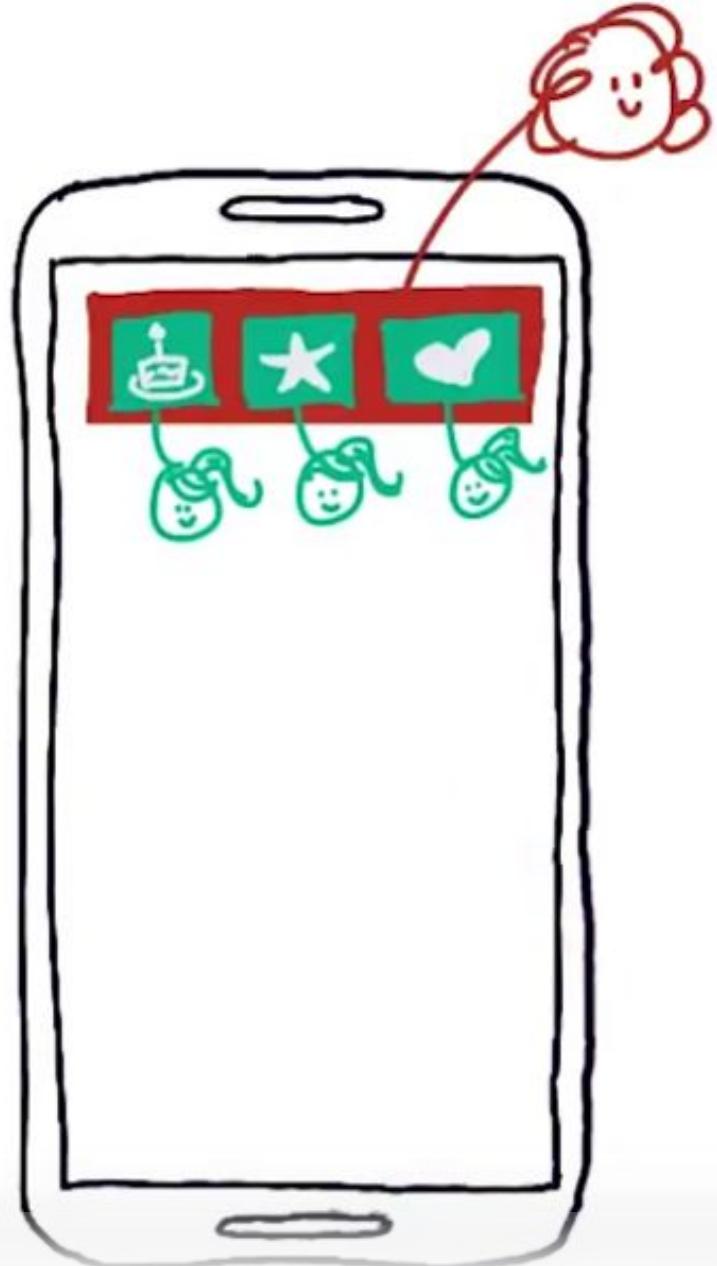


Tommy

TextView



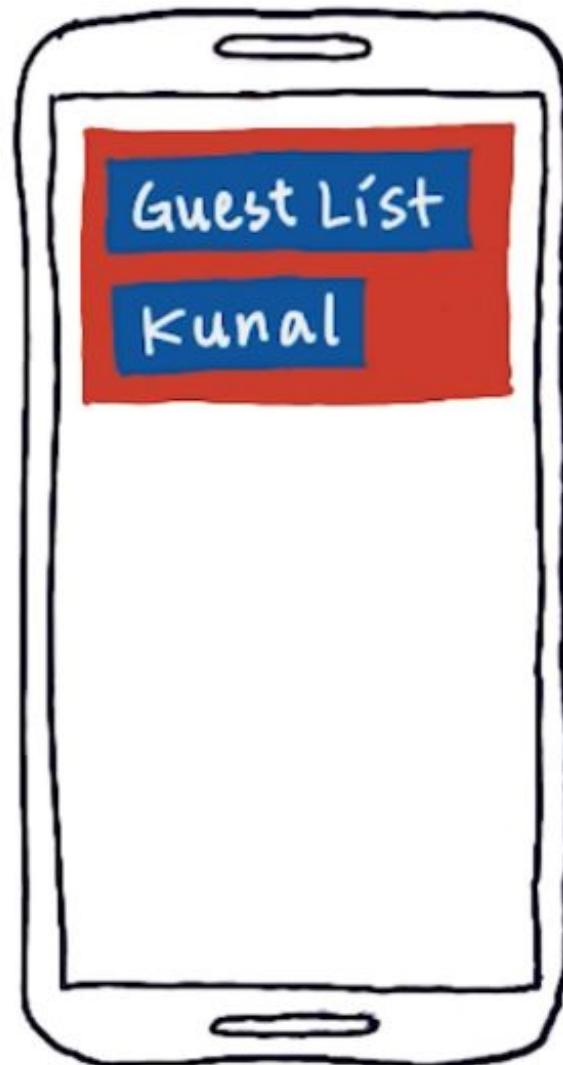
Ivy  
ImageView



# LINEAR LAYOUT

- Add more TextViews to the parent LinearLayout View Group
- Change the LinearLayout orientation  
attribute

Vertical      or      Horizontal

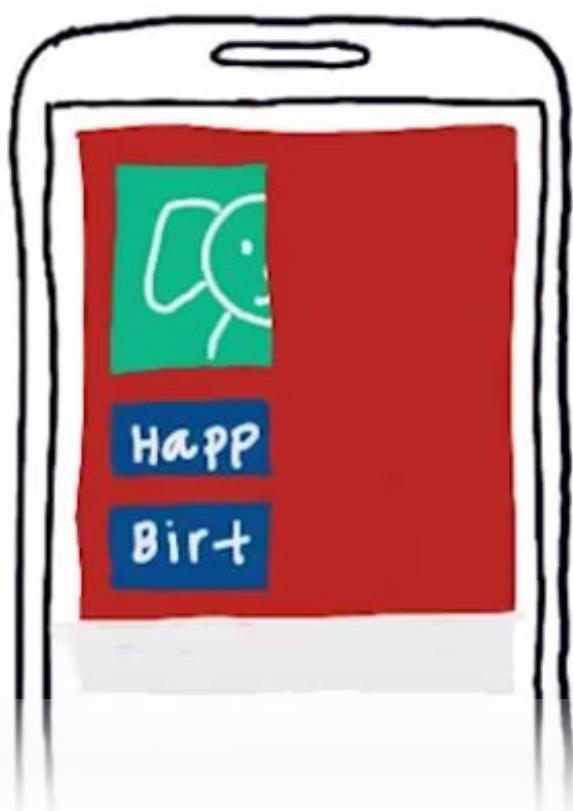


# size: match\_parent

за да изпълните родителската група по дадено измерение

# DIFFERENT WIDTHS ON CHILD VIEWS

200 dp



wrap-content



match-parent



# DIFFERENT HEIGHTS ON CHILD VIEWS

200 dp



wrap-content



match-parent

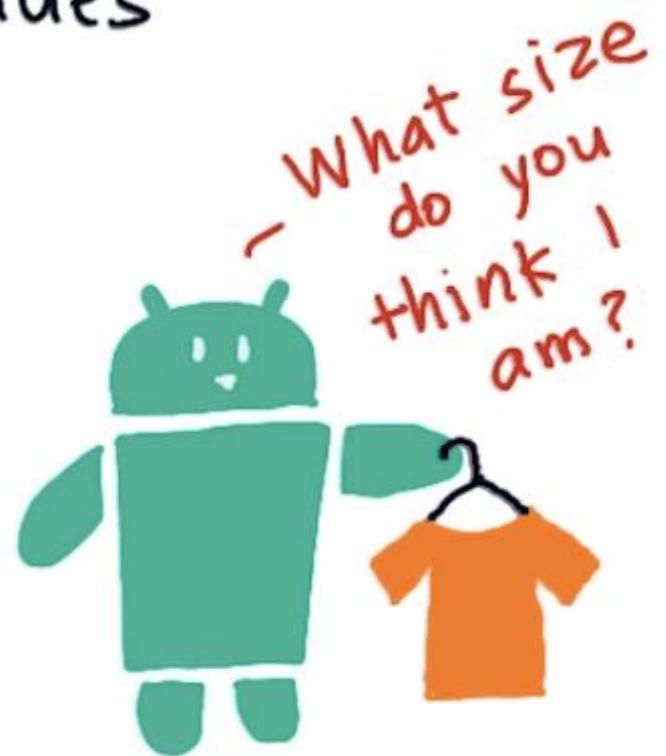


# WIDTH and HEIGHT

- ☐ Try different width and height values for each TextView

- fixed dp values
- wrap-content
- match-parent

- \* Remember that width and height \* don't have to match



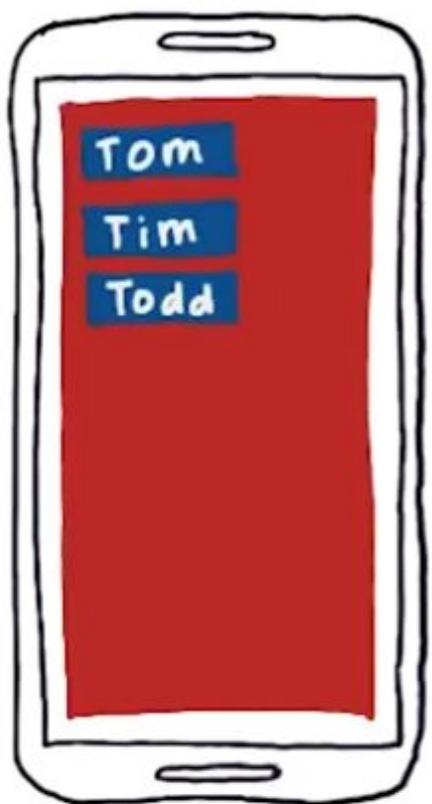
[http://labs.udacity.com/android-visualizer/#/  
android/match-parent](http://labs.udacity.com/android-visualizer/#/android/match-parent)



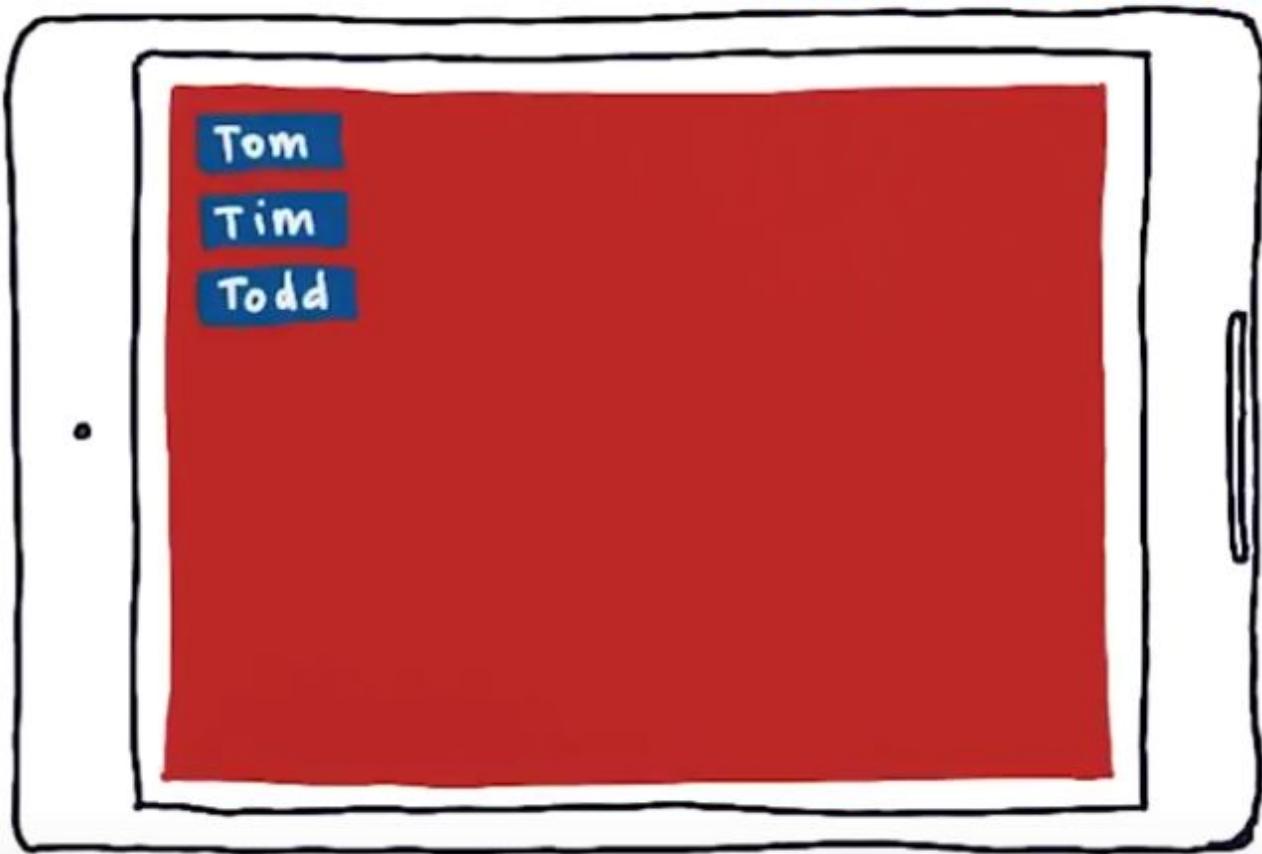
size: weight

## TAKING ADVANTAGE OF SCREEN REAL ESTATE

Portrait Mode



Tablet Landscape

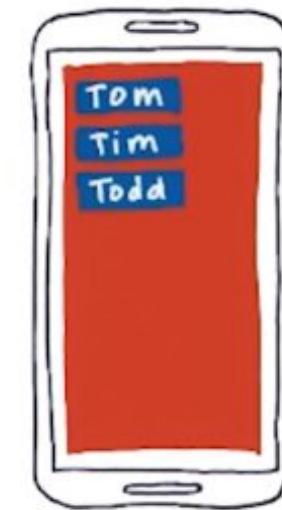


# LINEAR LAYOUT WEIGHT

What we have:

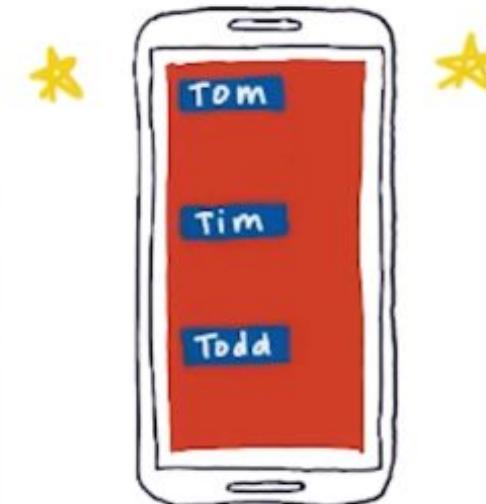
1. Read Linearlayout article linked below.

2. Provide some ideas on how we can build the desired layout in XML



What we want:

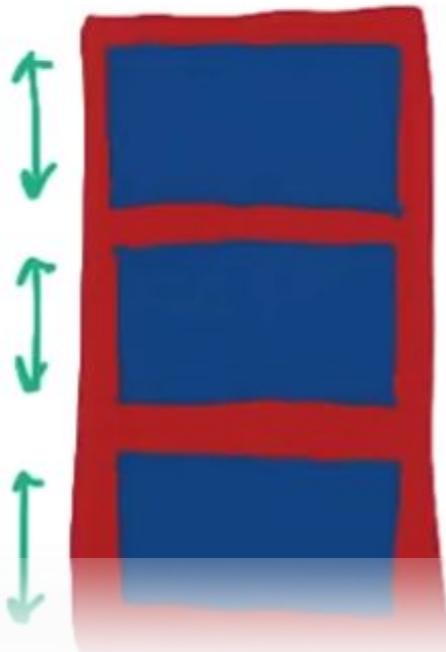
[http://developer.android.com/guide/topics/ui/layout/linear.html?utm\\_source=udacity&utm\\_medium=course&utm\\_campaign=android\\_basics](http://developer.android.com/guide/topics/ui/layout/linear.html?utm_source=udacity&utm_medium=course&utm_campaign=android_basics)



# LINEAR LAYOUT WEIGHT

## Vertical Linear Layout

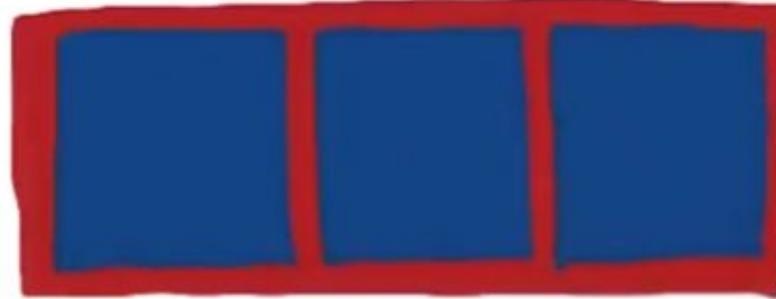
equal weights means equal heights



3 child views  
height: 0 dp  
weight: 1

## Horizontal Linear layout

equal weights means equal widths



3 child views  
width: 0 dp  
weight: 1

# HANGOUTS APP



Horizontal linear layout



ImageView

wrap\_content

wrap\_content

weight

EditText

wrap\_content

0dp

ImageView

wrap\_content

wrap\_content

0

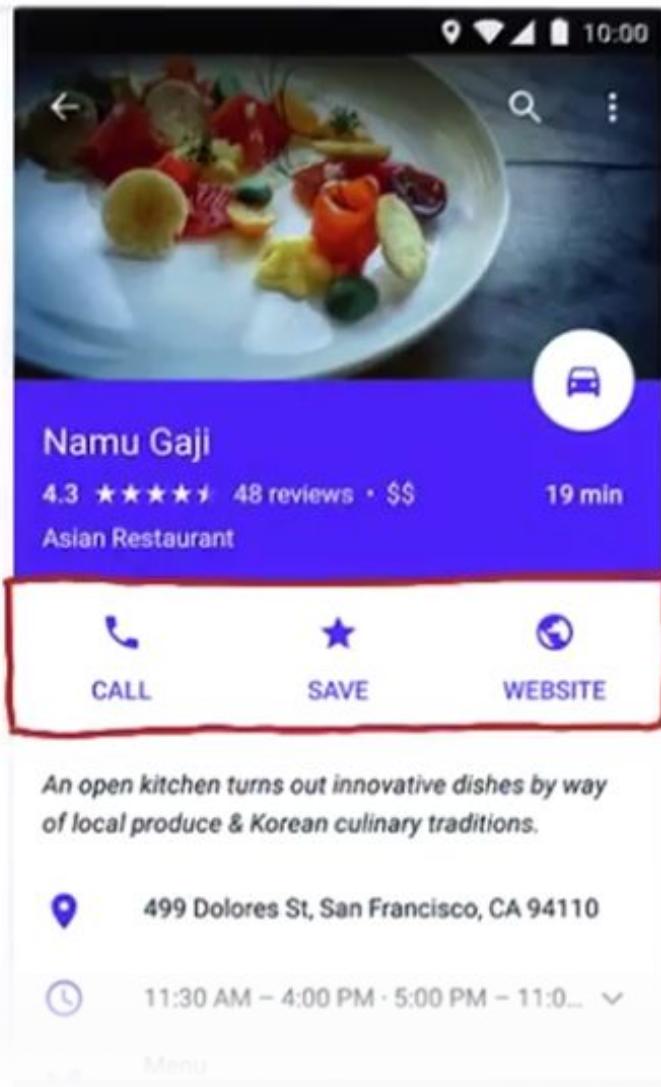
0

1

height

width

# MAPS APP



Horizontal linear layout



TextView

Height

width

weight

TextView

wrap-content

0dp

1

TextView

wrap-content

0dp

1

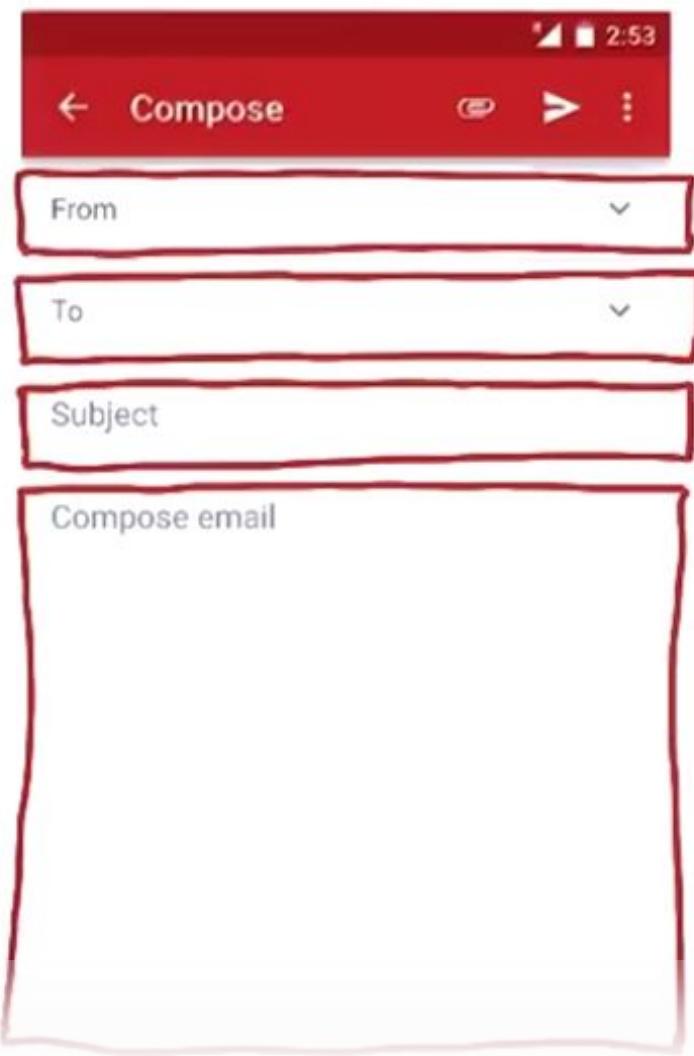
TextView

wrap-content

0dp

1

# EMAIL APP



Vertical Linear Layout

	Width	Height	Weight
Edit Text	match-parent	wrap-content	0
Edit Text	match-parent	wrap-content	0
Edit Text	match-parent	wrap-content	0
Edit Text	match-parent	0dp	1

[http://labs.udacity.com/android-visualizer/#/  
android/linear-layout-weight](http://labs.udacity.com/android-visualizer/#/android/linear-layout-weight)

# Почивка

## до 20:15



# RelativeLayout

# Какво представлява?

- ViewGroup-а, която подравнява децата си едно спрямо друго или спрямо себе си
- Обикновено, когато имаме няколко вложени едни в други LinearLayout-и, по-добър вариант е да използваме един RelativeLayout

децата подравнени по някакъв критерии



# RELATIVE LAYOUT

Relative to parent



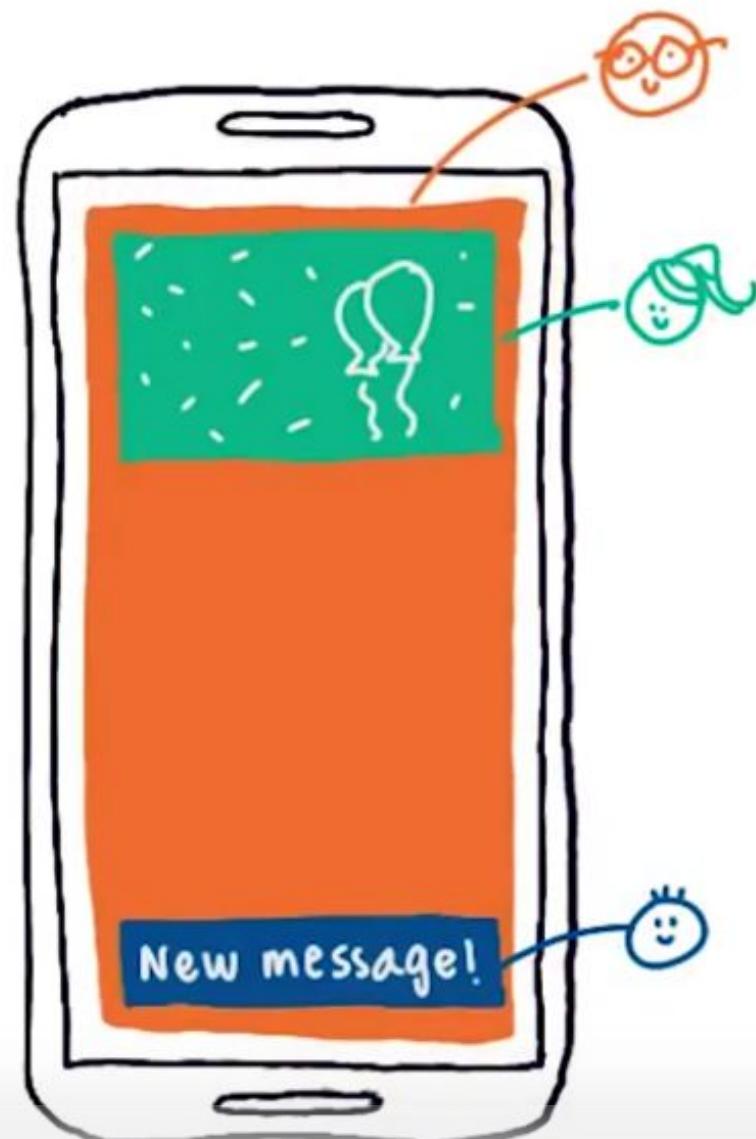
Ray  
RelativeLayout



Tommy  
TextView



Ivy  
ImageView



# RELATIVE LAYOUT

Relative to other children



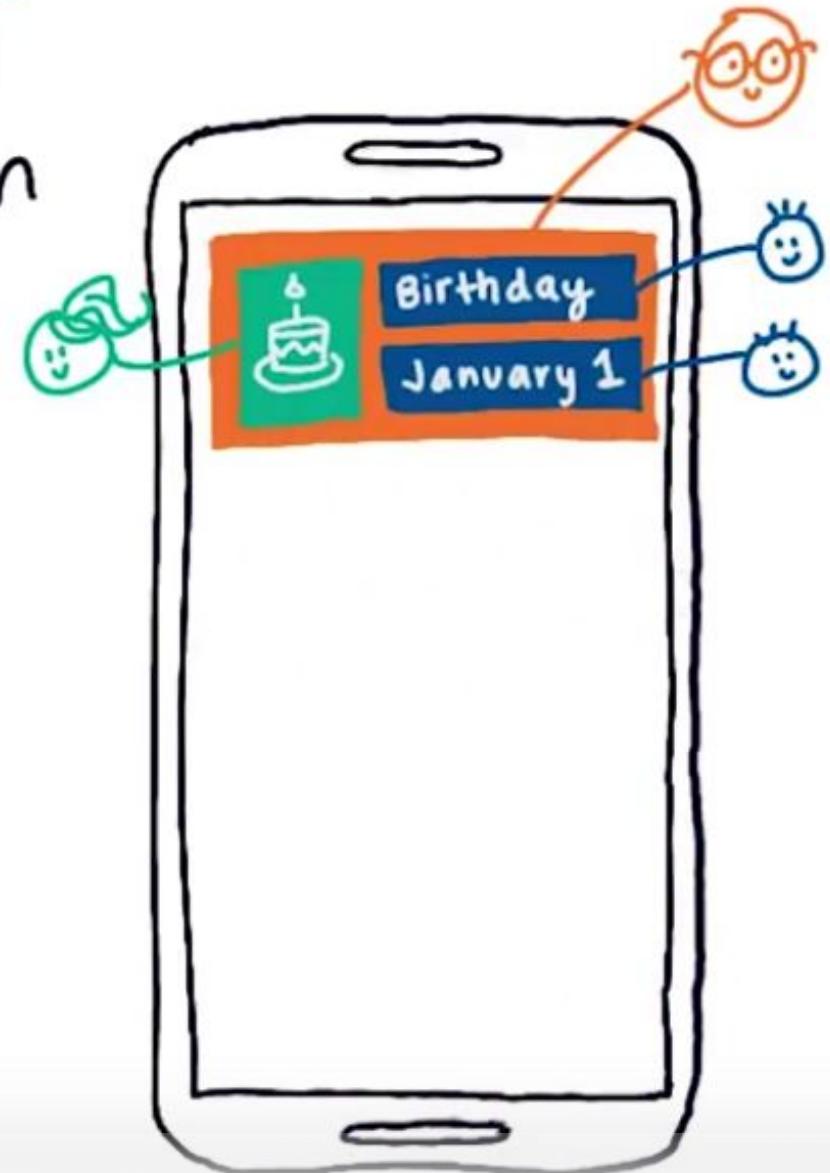
Ray  
Relative Layout



Tommy  
TextView

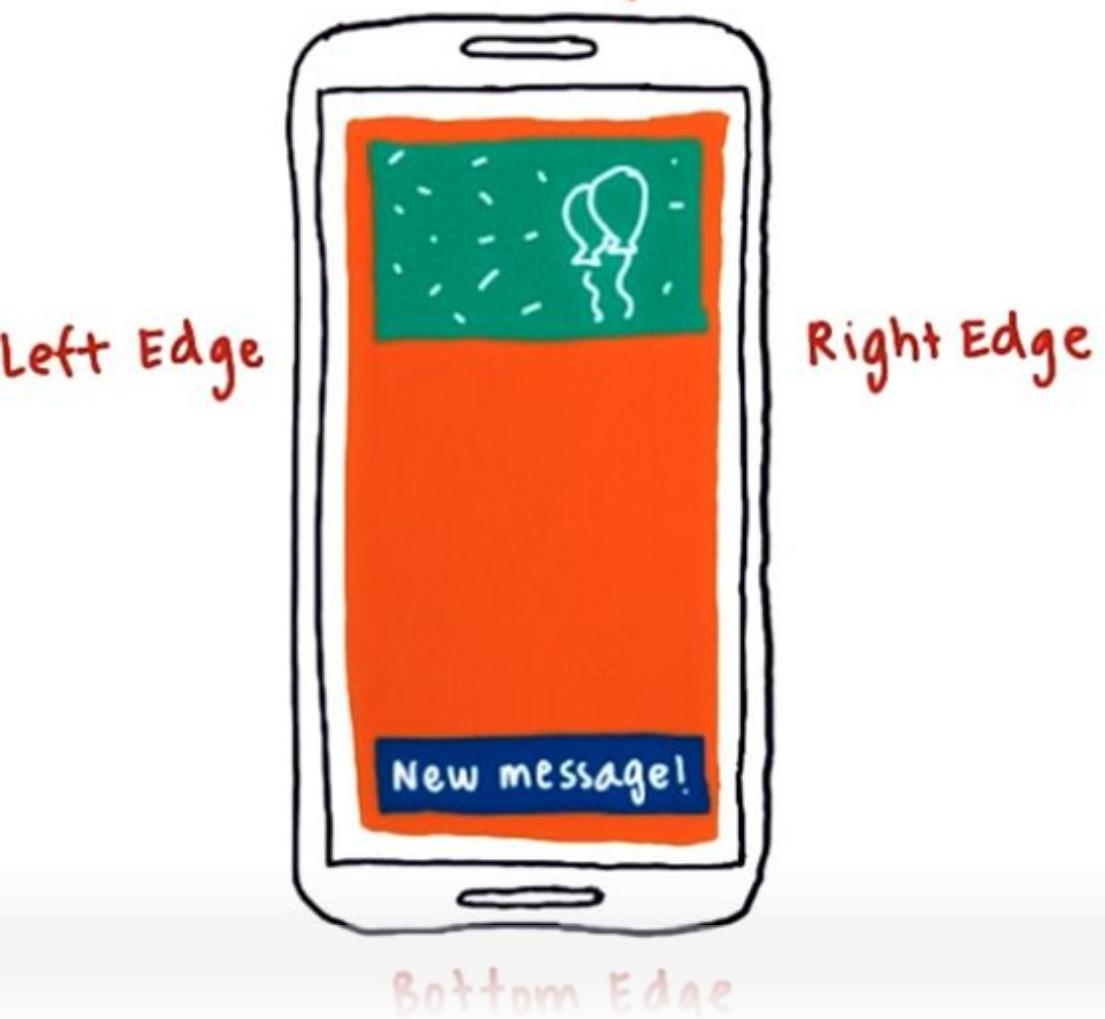


Ivy  
ImageView



# RELATIVE LAYOUT

## Relative to Parent



Child view attributes:

`android:layout_alignParentTop = "true"`  
OR  
`"false"`

`android:layout_alignParentBottom`

`android:layout_alignParentLeft`

`android:layout_alignParentRight`

# RELATIVE LAYOUT

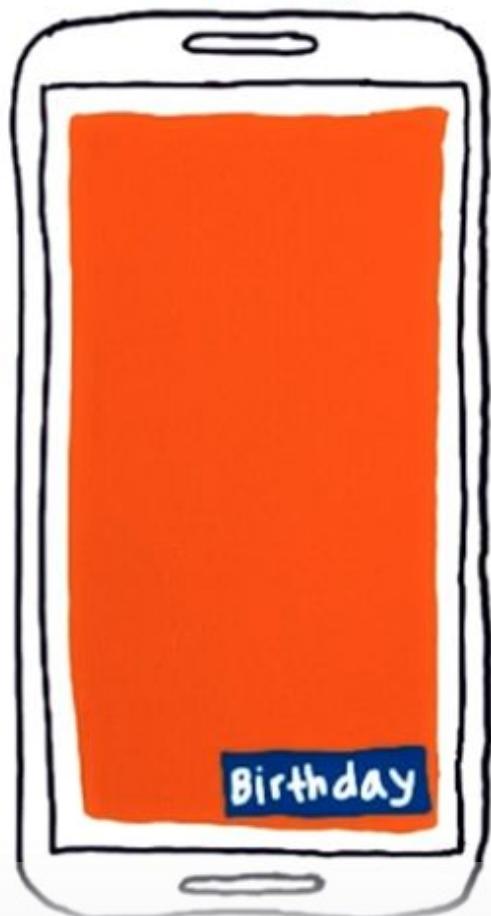
## Relative to Parent

Child view attributes:

`android:layout_alignParentTop`

Left Edge

Right Edge



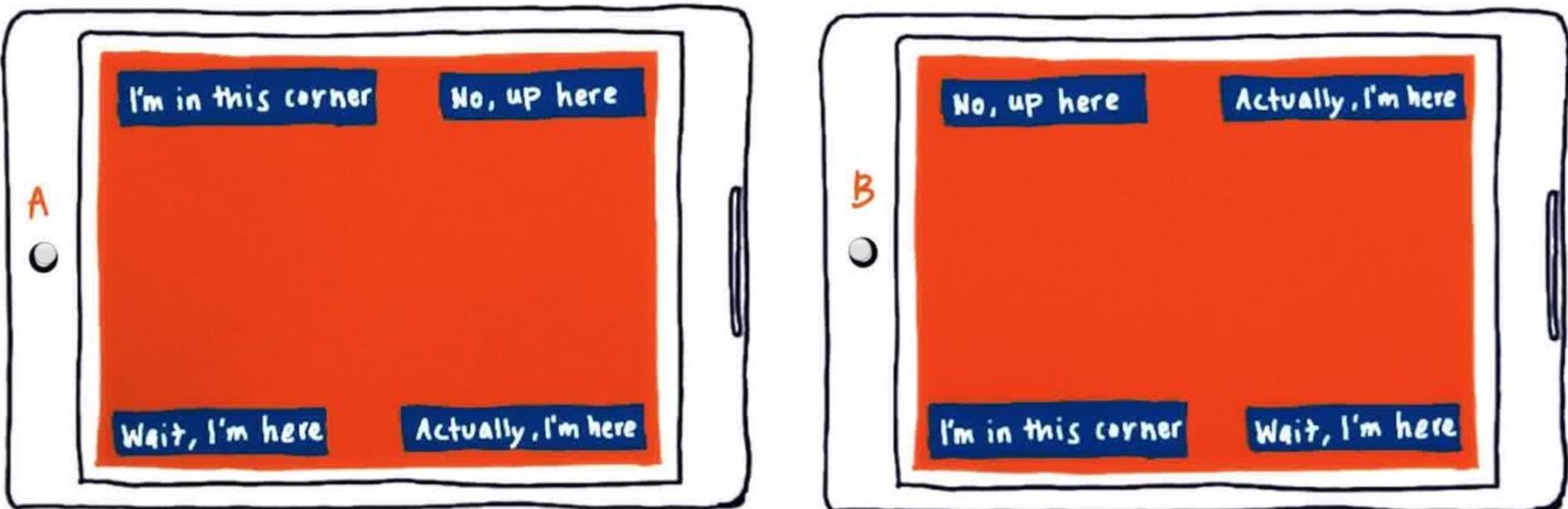
`android:layout_alignParentBottom = "true"`

`android:layout_alignParentLeft`

`android:layout_alignParentRight = "true"`

# RELATIVE LAYOUT

Relative to Parent



Which layout does the below XML create on the device?

Kunal Ben Kagure

Natalie  
Lyla Me

Amy  
Omoju  
Jennie

## RELATIVE LAYOUT

Positioning children relative  
to other views

Lyla

Natalie

Me

Ben

Kunal

Kagure

Jennie

Omoju

Amy

Kunal Ben

→ Assigning view ID names

On Ben TextView

android:id="@+id/ben\_text\_view"

→ Positioning children relative  
to other views

On Kunal TextView:

android:layout\_toLeftOf =

Kunal Ben Kagure

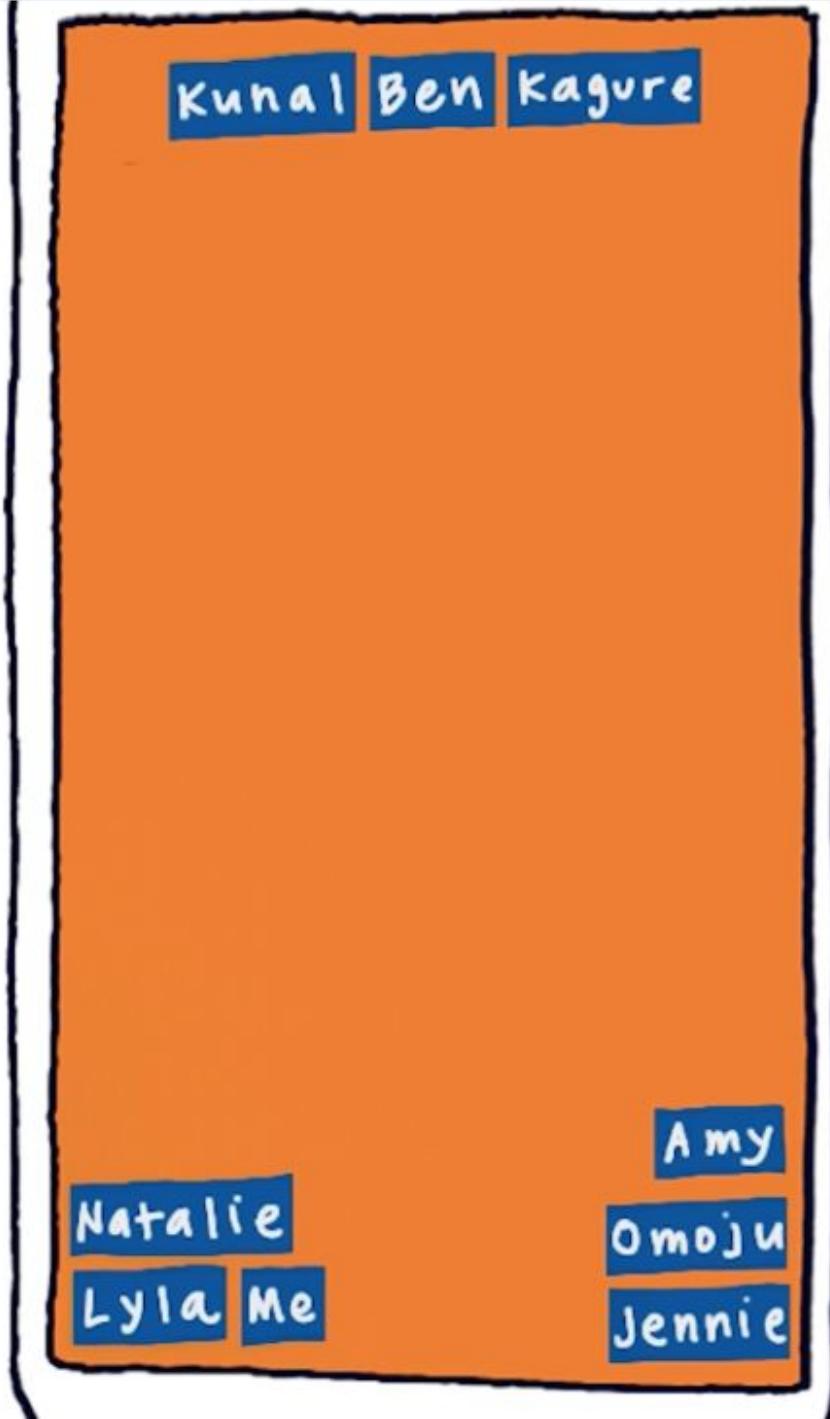
Natalie  
Lyla Me

Amy  
Omoju  
Jennie

# RELATIVE LAYOUT

Positioning children relative  
to other views

- ☐ Modify the provided XML layout to achieve this desired layout



[http://labs.udacity.com/android-visualizer/#/  
android/relative-layout-view-ids](http://labs.udacity.com/android-visualizer/#/android/relative-layout-view-ids)



Вложени ViewGroup-и

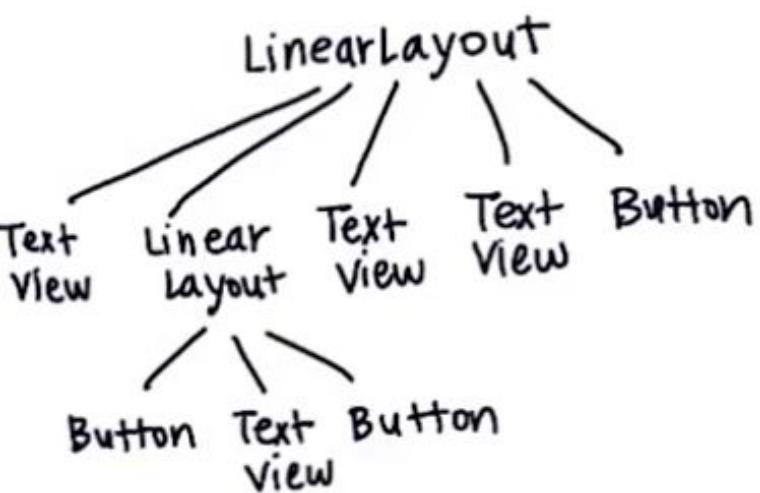
# QUANTITY PICKER

Makin' it  
look better

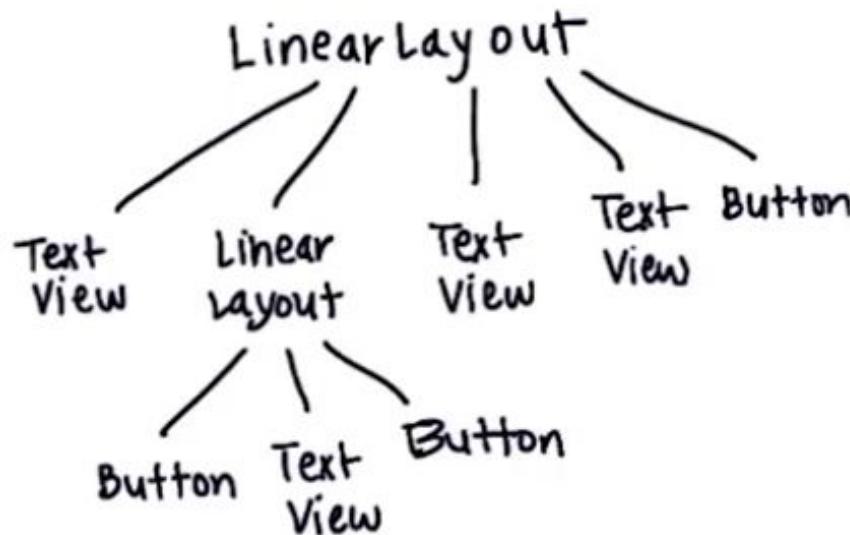


# NESTED VIEWGROUPS

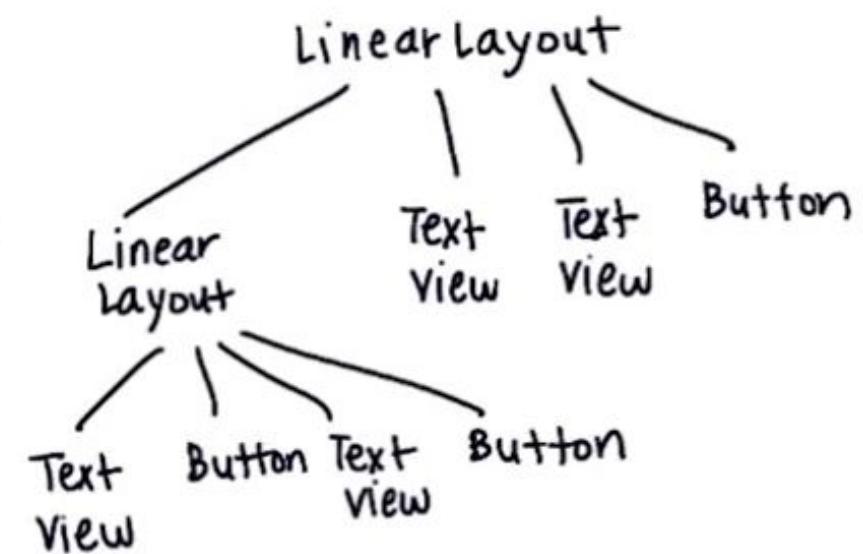
OPTION A



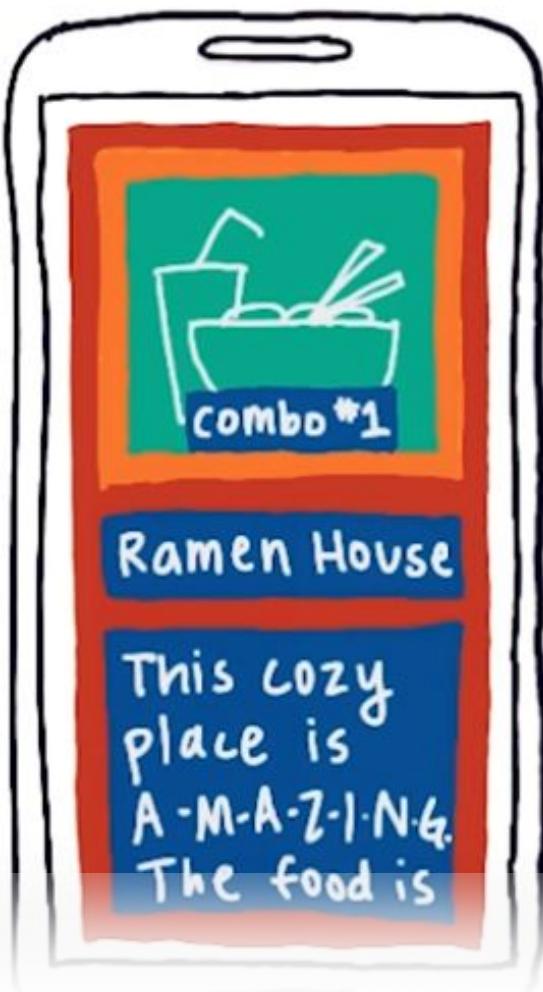
OPTION B



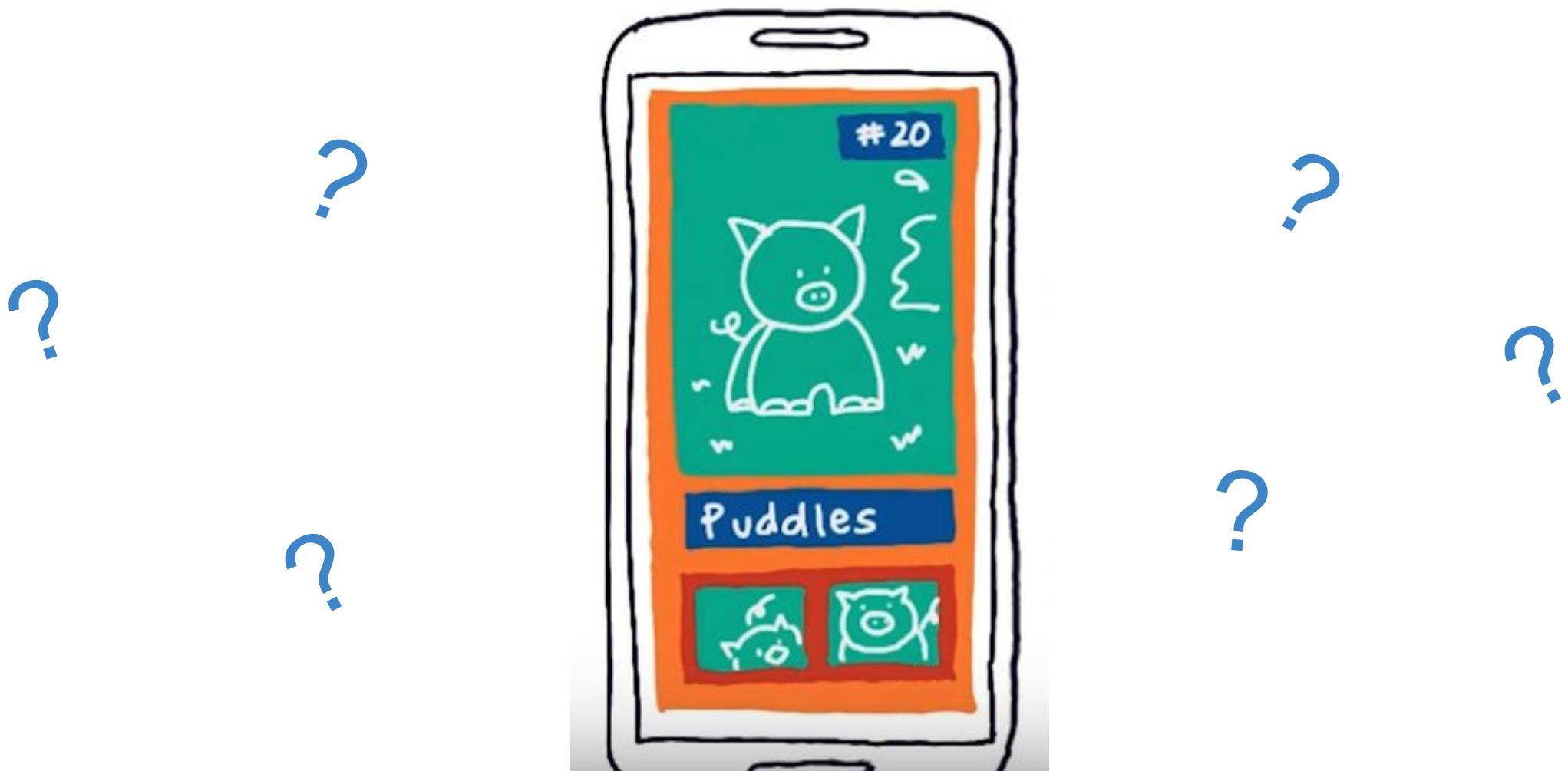
OPTION C



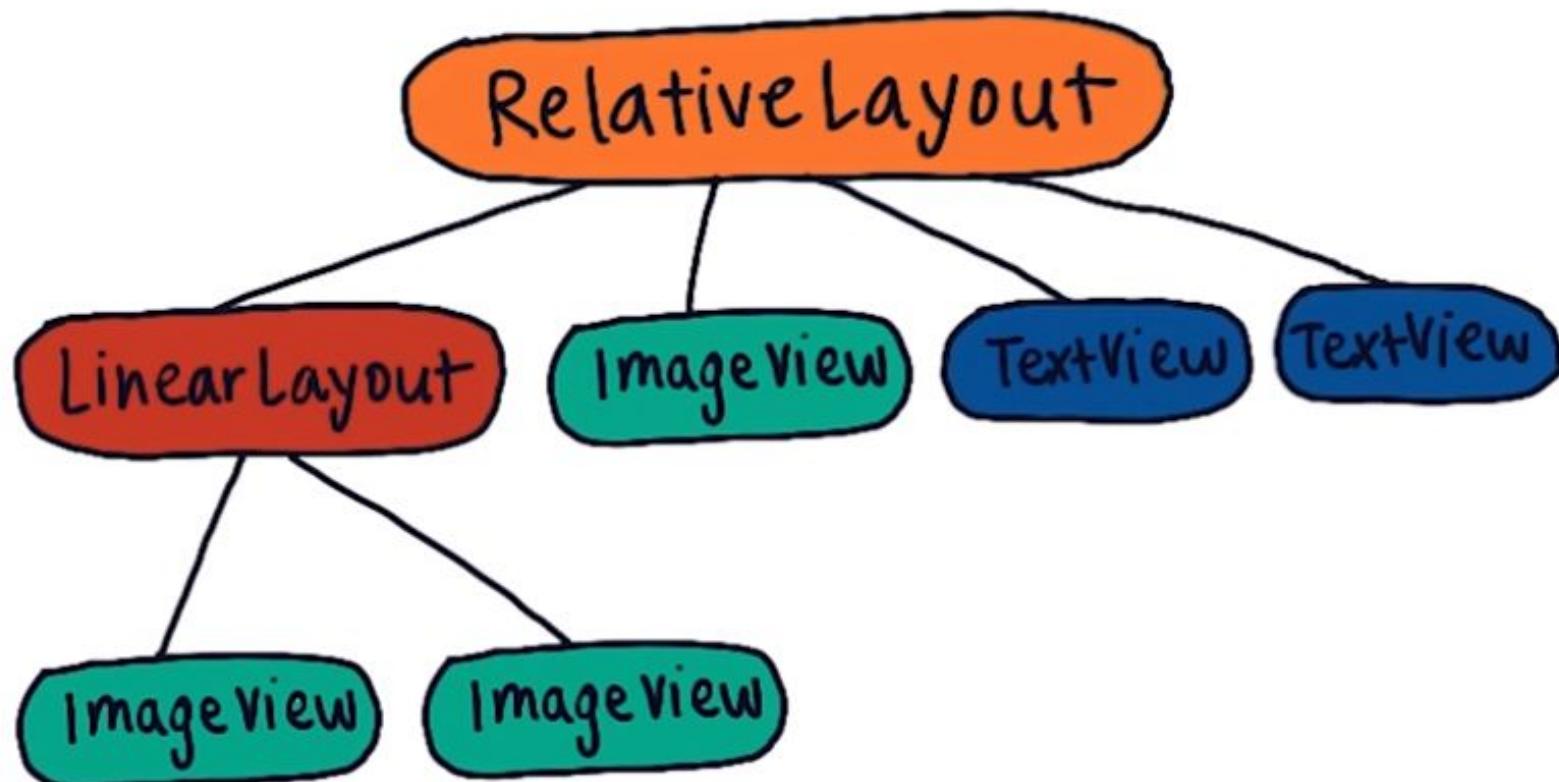
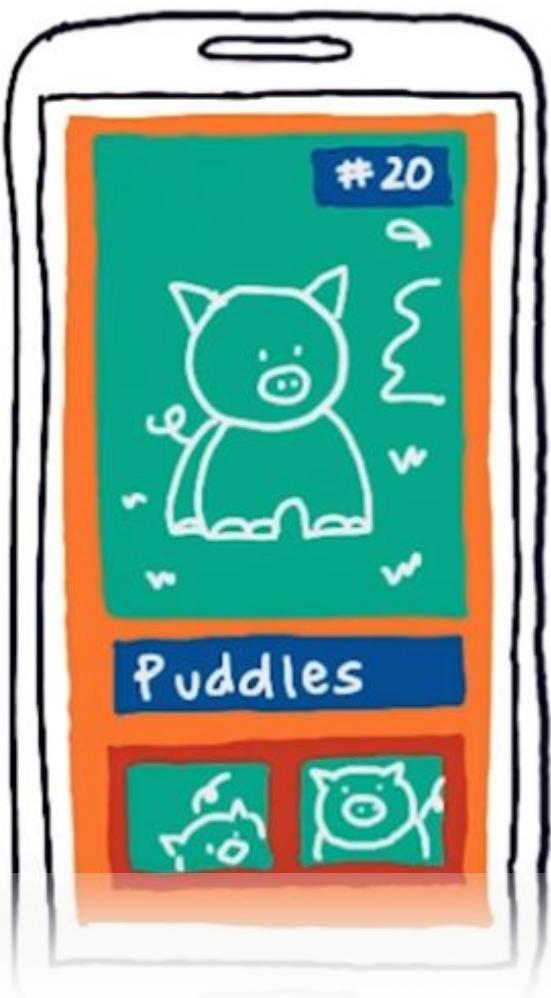
# NESTED VIEW GROUPS



# NESTED VIEWGROUPS



# NESTED VIEW GROUPS



# ConstraintLayout

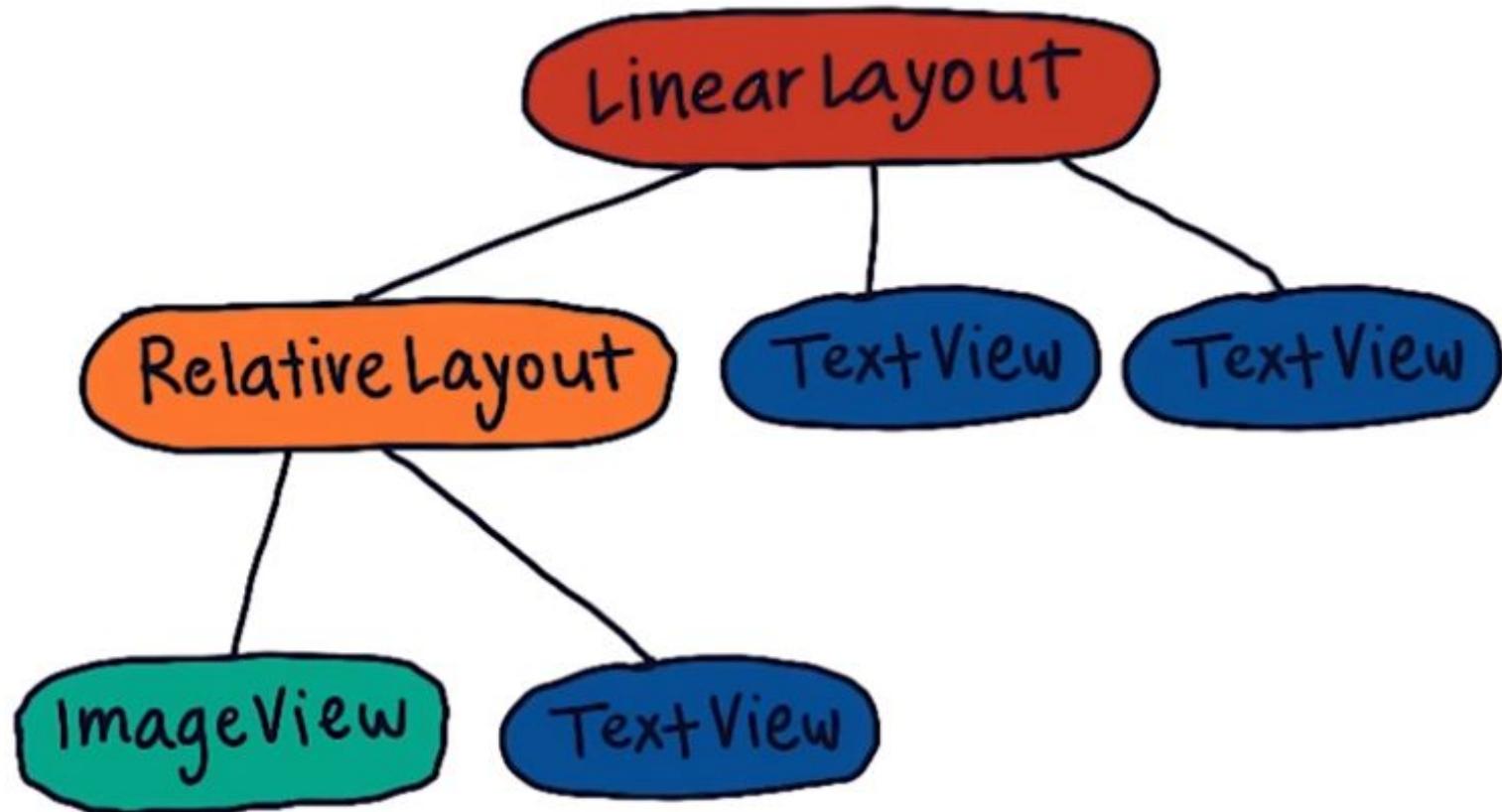
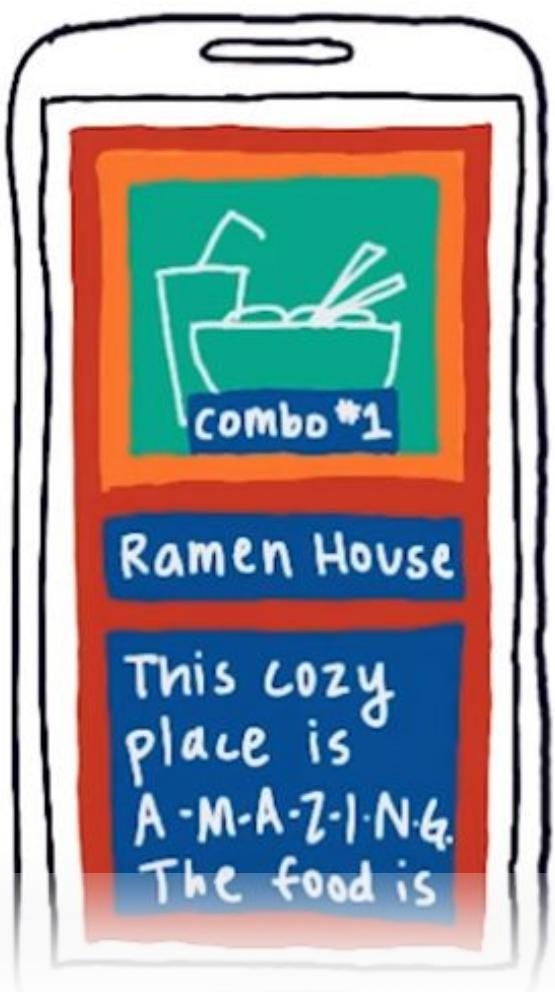
# Какво представлява?

- Благодарение на него можем да създаваме сложни дизайн без да се налага да използваме много вложени едни в други layout-и
- Позволява ни да имаме плоска йерархия
- По-удобно и мощно от RelativeLayout
- Можем да използваме Layout Editor-а на Android Studio

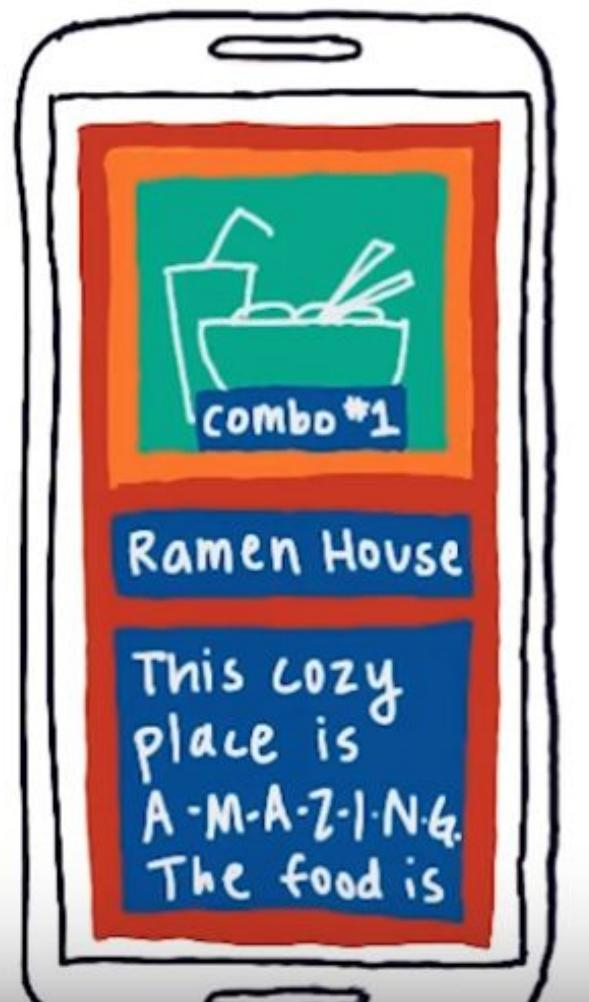
Още информация:

<https://developer.android.com/training/constraint-layout>

# NESTED VIEW GROUPS



~~NESTED~~



VIEW GROUPS

ConstraintLayout

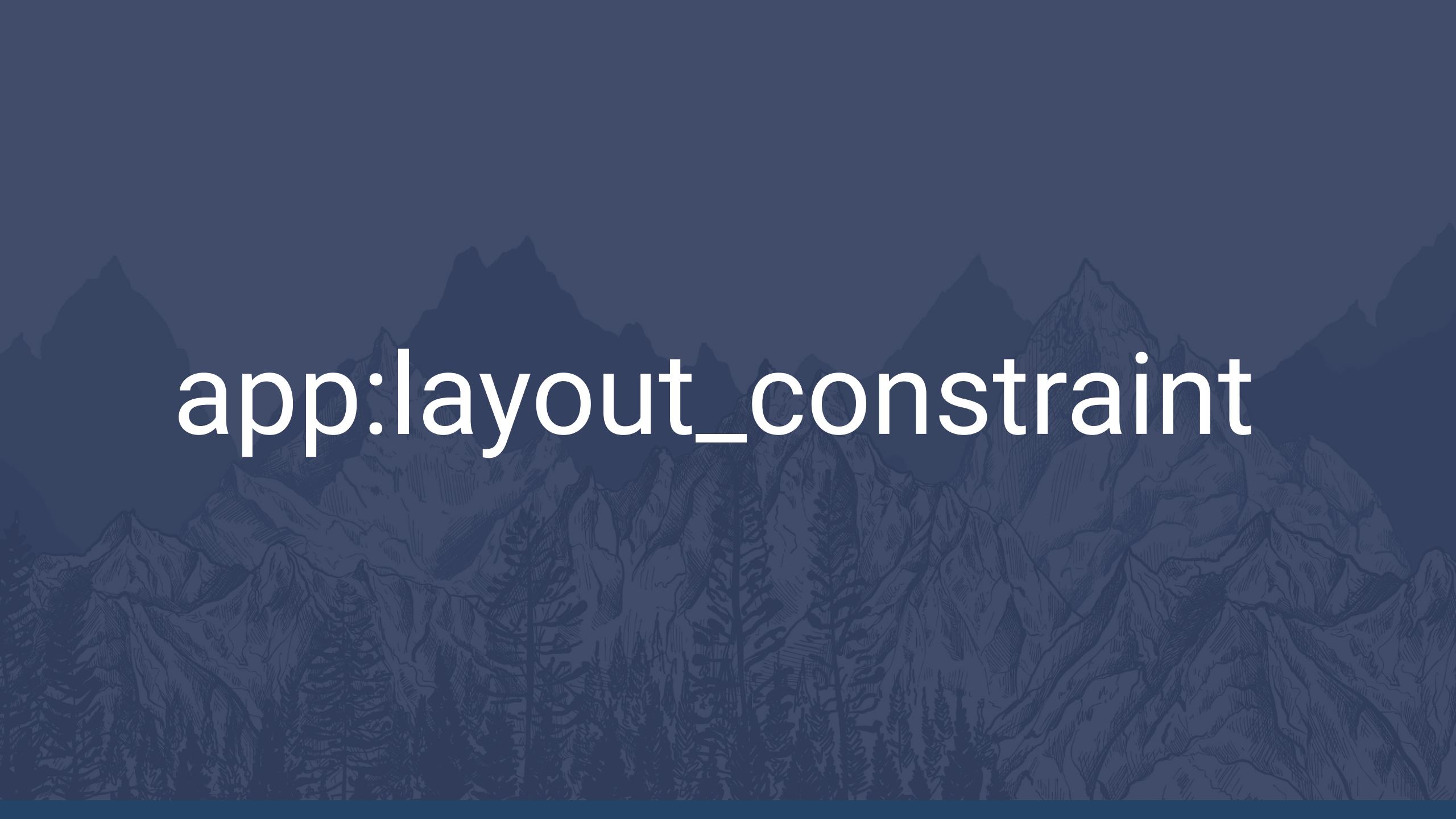
# build.gradle

Добавяме следния ред:

```
dependencies {  
    implementation "androidx.constraintlayout:constraintlayout:2.0.0"  
}
```

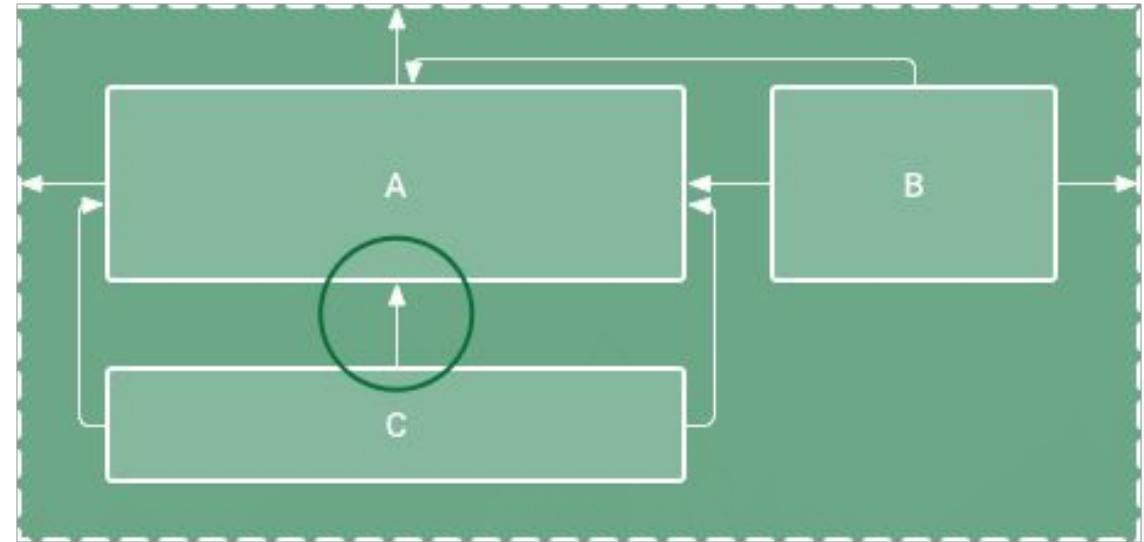
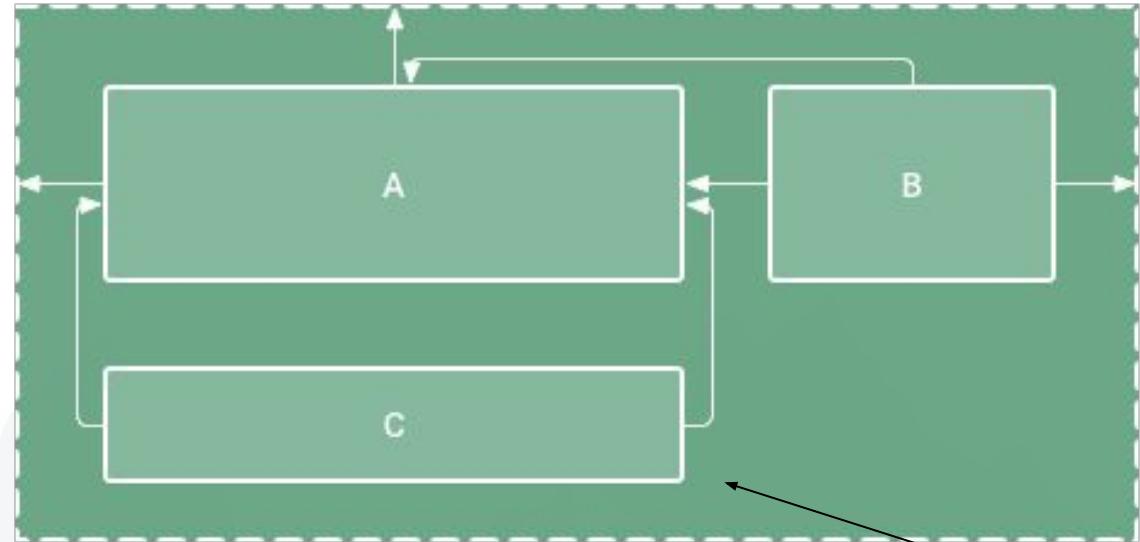
И се уверяваме, че:

```
repositories {  
    google()  
}
```



app:layout\_constraint

- За да може да работи, всяко view трябва да има поне едно хоризонтално и едно вертикално ограничение:
  - `app:layout_constraintStart/constraintEnd`
  - `app:layout_constraintTop/constraintBottom`
- Всяко едно ограничение представлява връзка с друго **view**, **parent layout**-а или **невидим guideline**-и

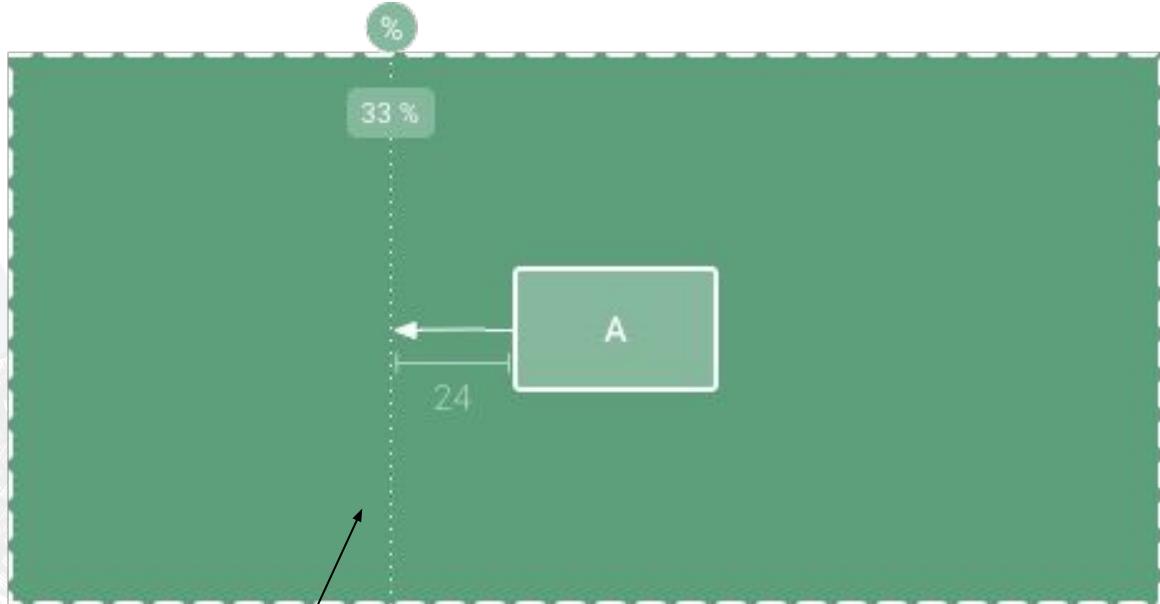


Липсва вертикално  
ограничение

Макар да изглежда привидно на място, при стартиране на приложението, ще бъде изчертано най-отгоре. Задължително трябва да имаме поне по едно ограничение хоризонтално и вертикално.

# Guidelines

Не винаги искаме да подравняваме нашите view-та едно спрямо друго или на база на границите на ViewGroup-ата. За тази цел можем да дефинираме своя собствена невидима по време на изпълнение граница, спрямо която нашите view-та да се организират.



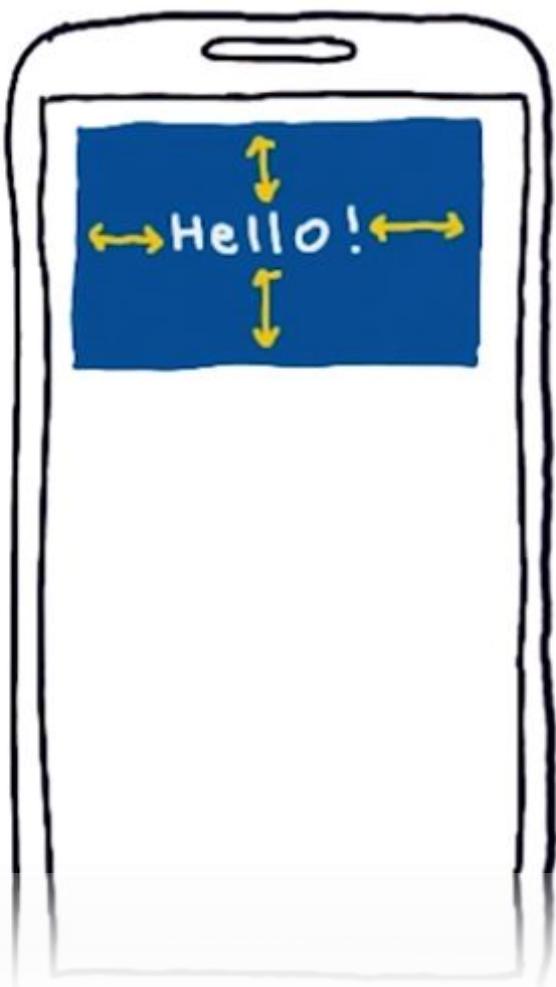
Натискаме Guideline в toolbar-а и после Add Vertical Guideline или Add Horizontal Guideline.





# margin vs padding

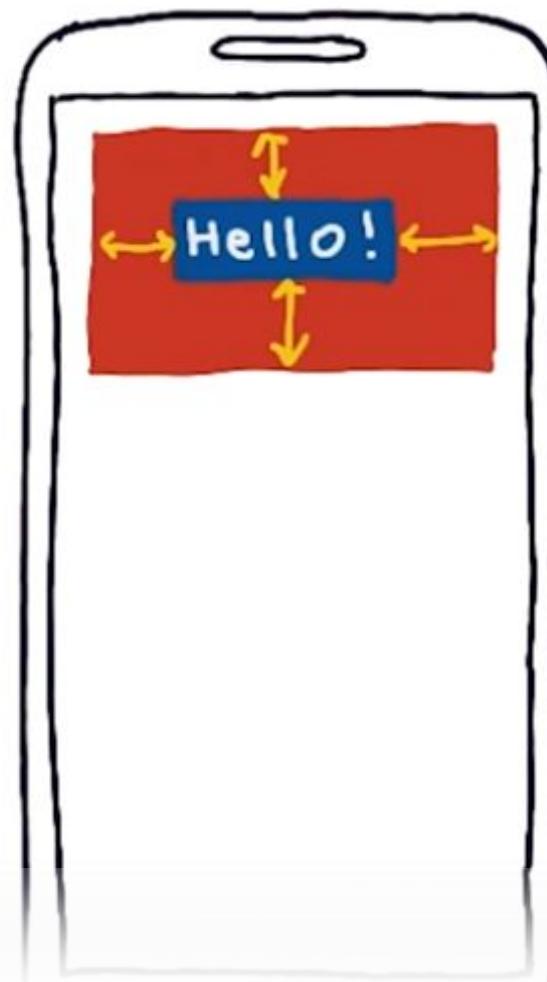
# PADDING



8 dp  
padding  
on  
TextView

Hello!

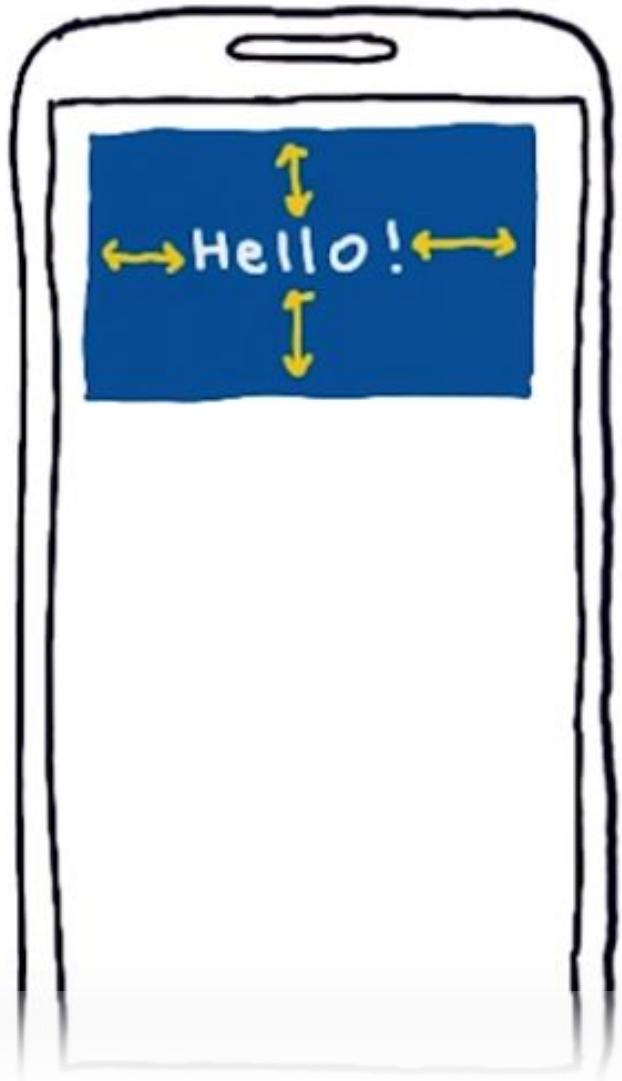
# MARGIN



Need  
ViewGroup

8 dp  
layout  
margin  
on  
TextView

# PADDING



TextView attribute:

`android:padding = "8 dp"`

OR

`android:paddingLeft = "8 dp"`

`android:paddingRight = "8 dp"`

`android:paddingTop = "8 dp"`

`android:paddingBottom = "8 dp"`

# MARGIN

TextView attributes:

android:layout\_margin="8dp"

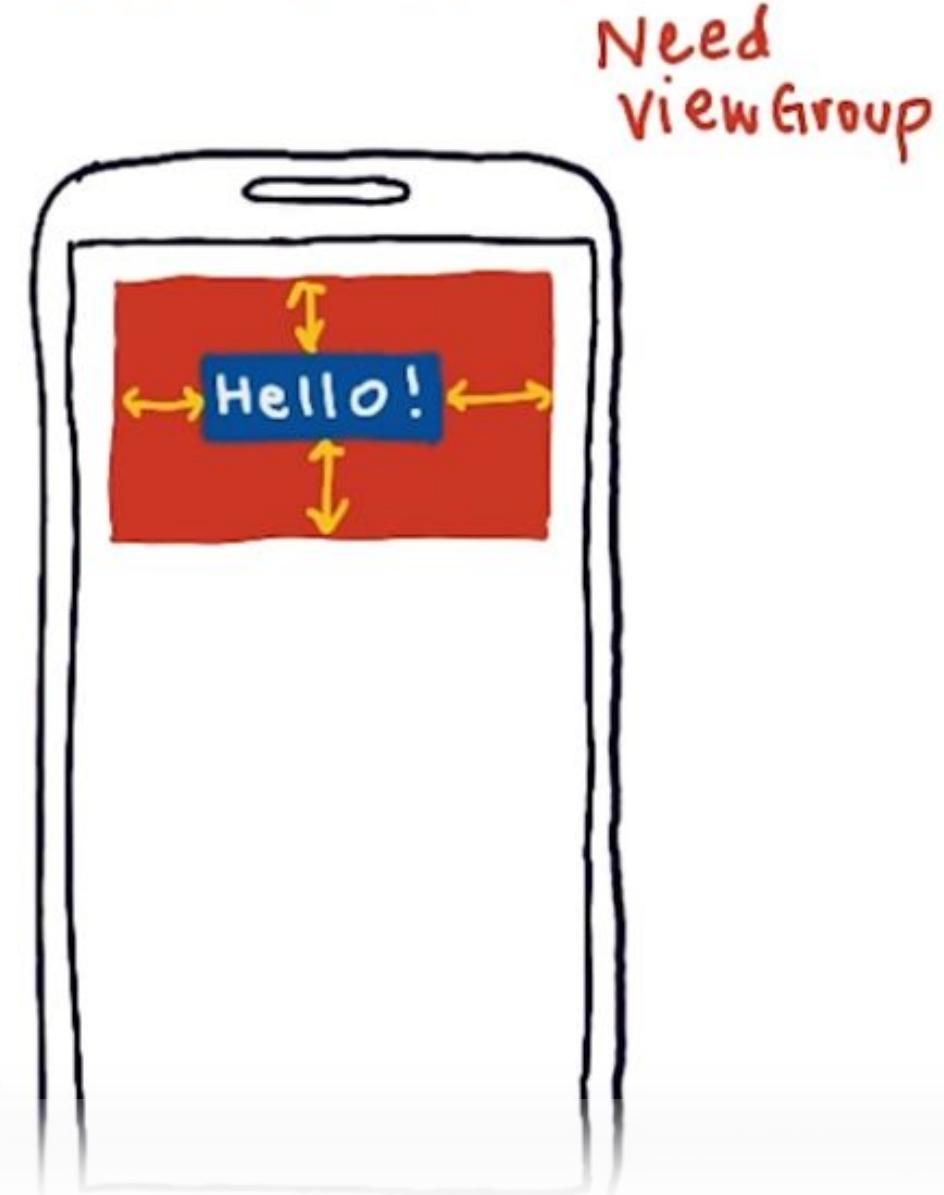
OR

android:layout\_marginLeft="8dp"

android:layout\_marginRight="8dp"

android:layout\_marginTop="8dp"

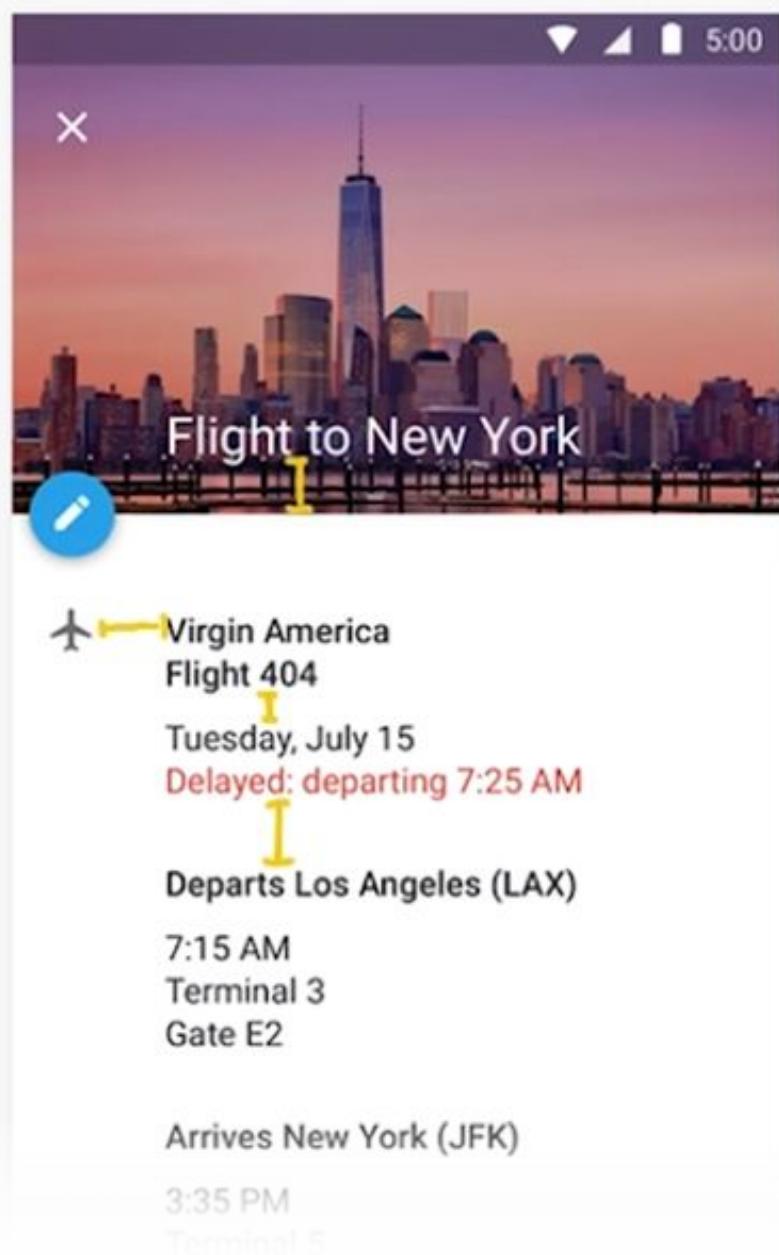
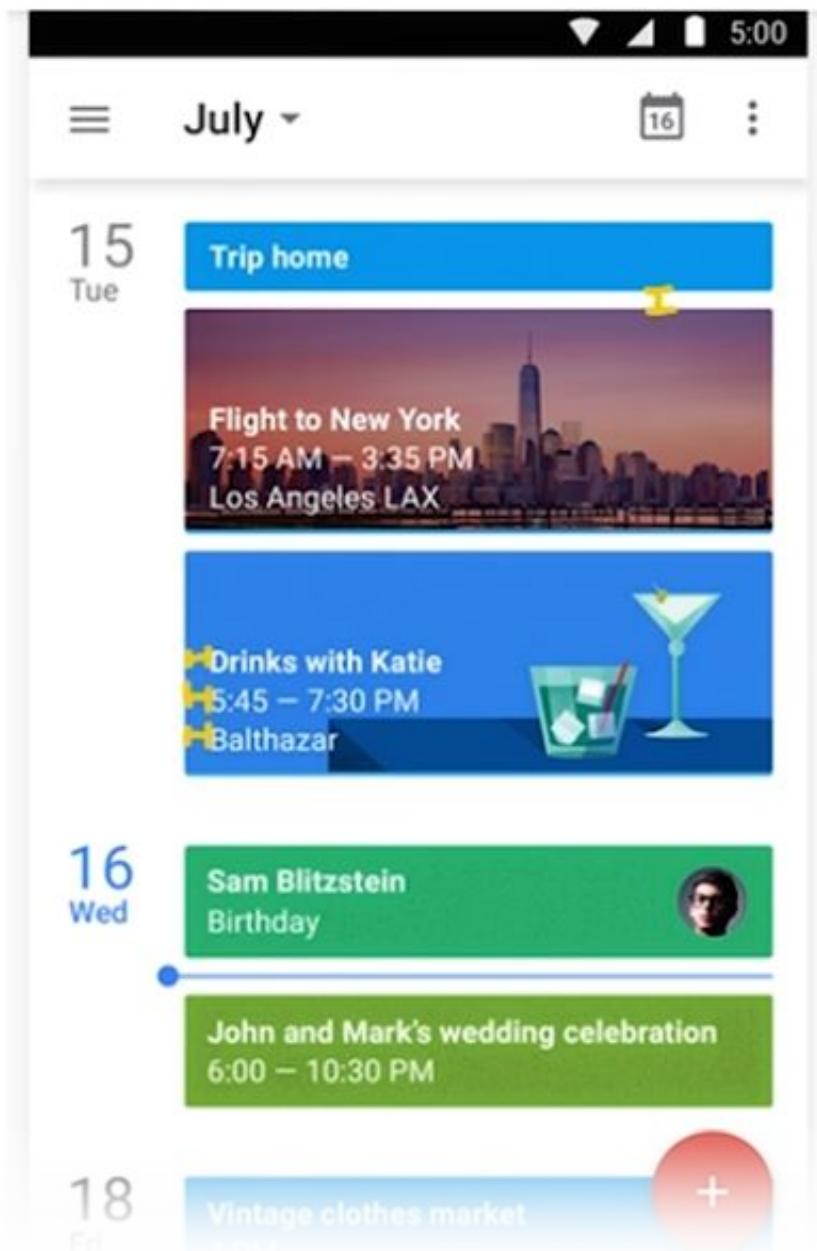
android:layout\_marginBottom="8dp"



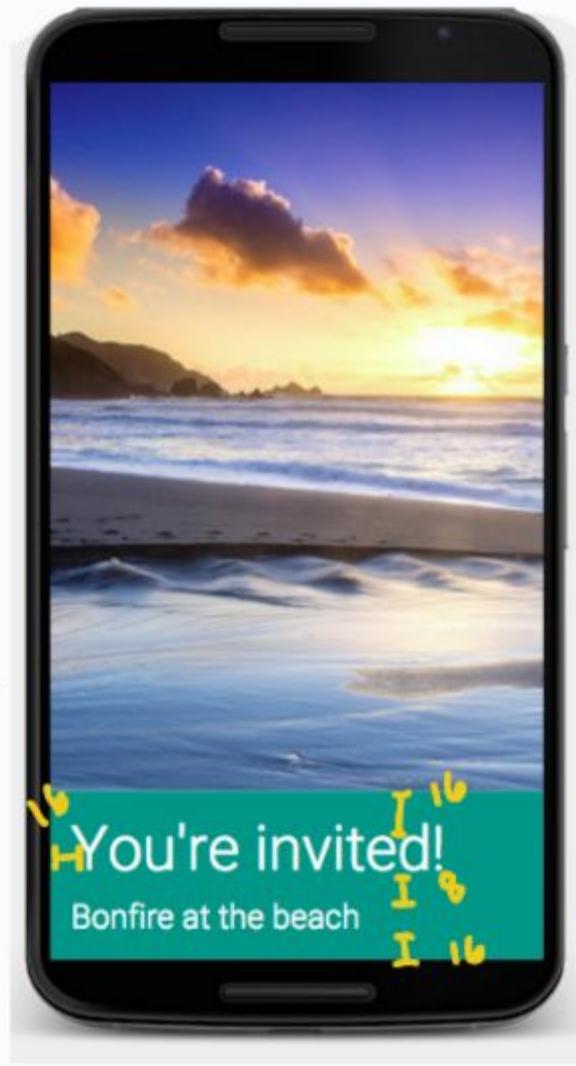
- Ползват се в зависимост от това какво искате да постигнете:
  - margin добавя разстояние извън view-то
  - padding добавя разстояние между съдържанието на view-то и неговите граници

Допълнителна информация:

<https://material.io/guidelines/layout/metrics-keylines.html>



# PADDING & MARGIN



Modify the provided XML to build this desired layout



[http://labs.udacity.com/android-visualizer/#/  
android/padding-and-margin](http://labs.udacity.com/android-visualizer/#/android/padding-and-margin)

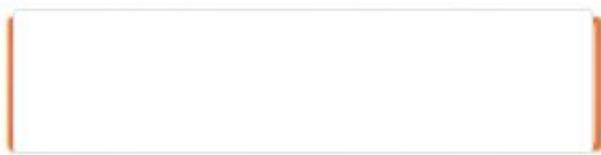
# Coffee Break

Challenge

# Coffee Break

challenge

If I want to show a picture on my phone, what XML element should I use?



# Coffee Break

Challenge

android:layout\_width, android:text, android:src  
are all examples of what?



# Coffee Break

Challenge

Write the sentence "I like practice sets the best" in Camel case.



# Coffee Break

Challenge

Relative Layout and Linear Layout are Known as what?



# Coffee Break

challenge

I always need to [ ] and [ ] my tags.



```
1 <LinearLayout  
2     android:layout_width="match_parent"  
3     android:layout_height="match_parent"  
4     android:orientation="vertical">  
5  
6     <TextView  
7         android:layout_width="match_parent"  
8         android:layout_height="wrap_content"  
9         android:text="To Ben"  
10        android:textSize="24sp" />  
11  
12    <TextView  
13        android:layout_width="match_parent"  
14        android:layout_height="wrap_content"  
15        android:text="Happy Birthday"  
16        android:textSize="34sp" />  
17  
18    <ImageView  
19        android:layout_width="match_parent"  
20        android:layout_height="wrap_content"  
21        android:layout_weight=1  
22        android:src="@drawable/ocean" />  
23  
24    <TextView  
25        android:layout_width="match_parent"  
26        android:layout_height="wrap_content"  
27        android:text="From, Lyla"  
28        android:textSize="24sp" />  
29 </LinearLayout>
```

here are the line numbers



Which line has an XML error on it?

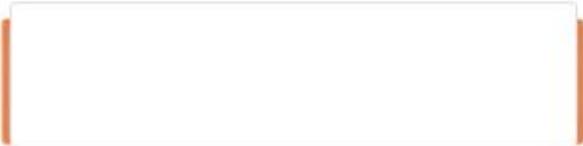
(give the line number)

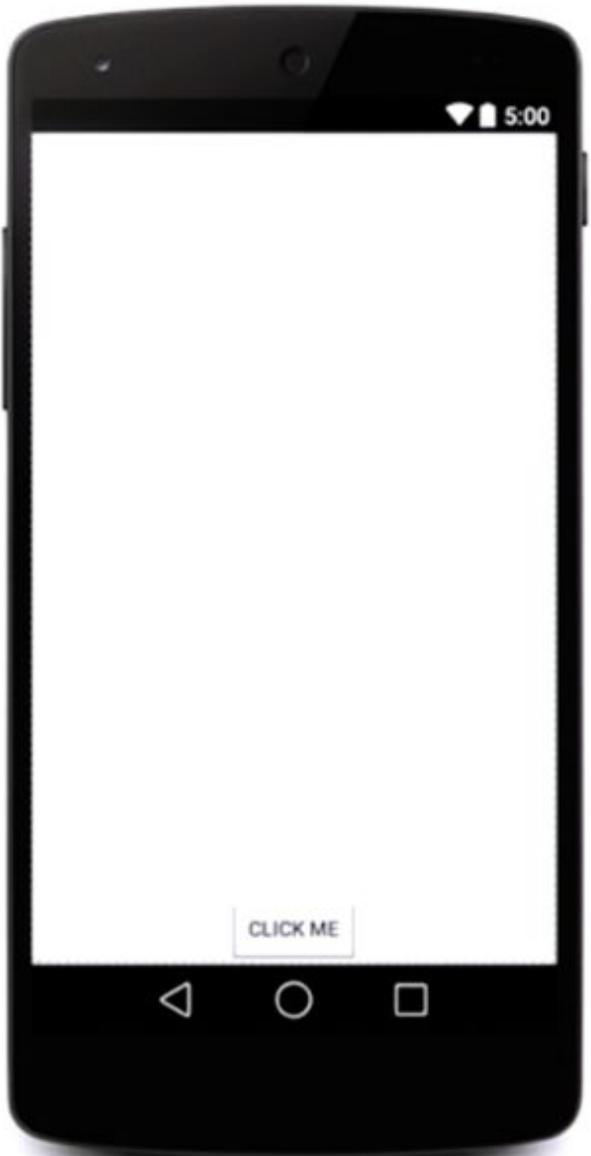
# Coffee Break

challenge

```
<RelativeLayout  
    android:layout_width="350dp"  
    android:layout_height="match_parent">  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="Cup of Coffee" />  
</RelativeLayout>
```

What is the width of the TextView?





```
<RelativeLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">  
    <Button  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Click me"  
        android:layout_weight="0" />  
</RelativeLayout>
```

Complete the code to position the button.

# Coffee Break

challenge

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">  
  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="200dp"  
        android:text="Menu"  
        android:layout_weight="0"/>  
  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="0dp"  
        android:text="Espresso"  
        android:layout_weight="2"/>  
  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="0dp"  
        android:text="Chai Tea"  
        android:layout_weight="1"/>  
  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="0dp"  
        android:text="Cafe Latte"  
        android:layout_weight="1"/>  
  
</LinearLayout>
```

# Coffee Break

challenge

Assume you are using the XML to the left and you have a screen that is 600dp tall. What is the height of the "Chai Tea" TextView?

# Coffee Break Challenge

You Are  
**DONE!**





# HOMEWORK

# #1

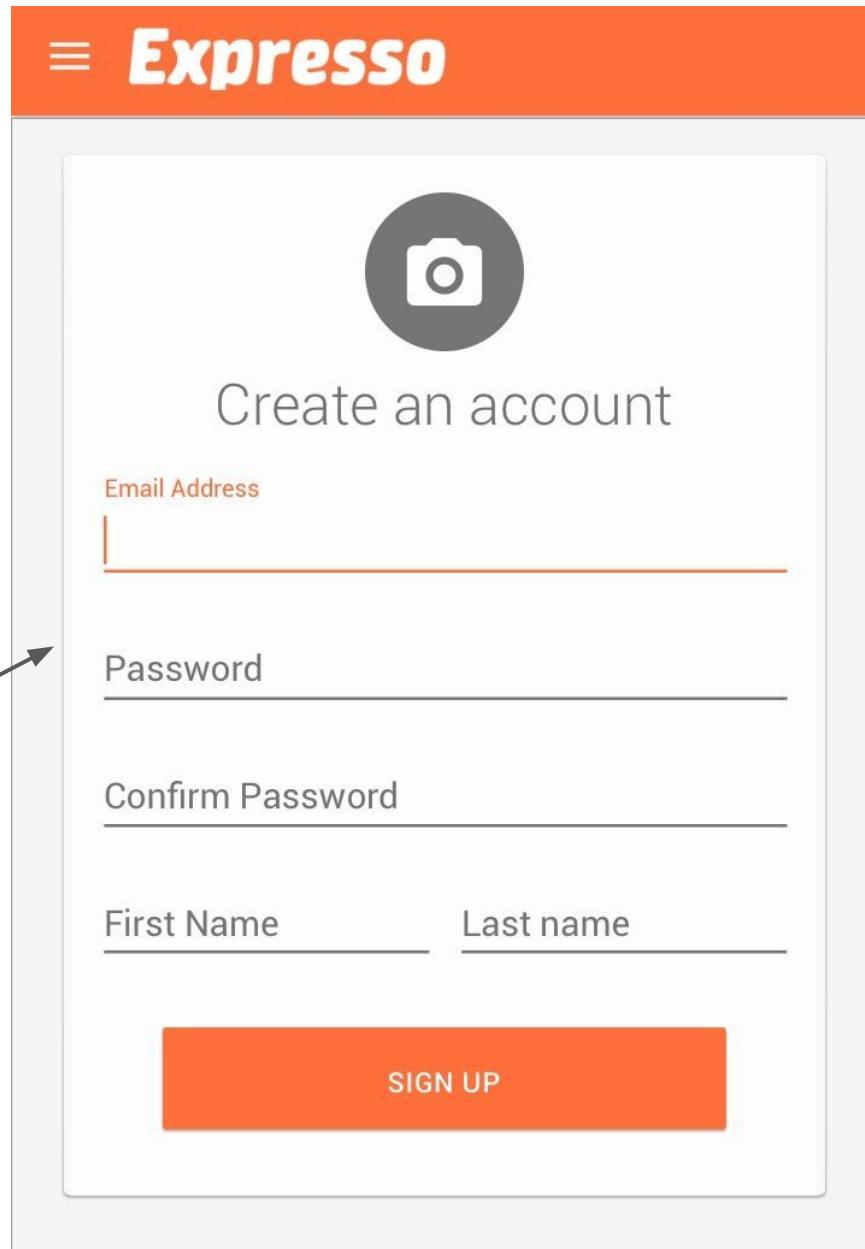
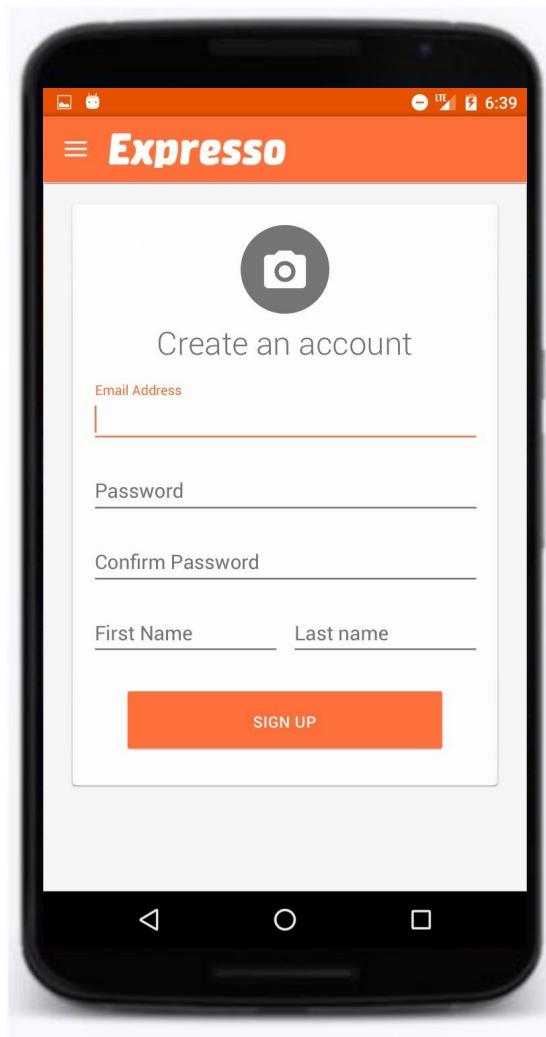
## LINEAR LAYOUT WEIGHT

Build this layout !



1. Determine width of each view first
2. Determine height of each view
3. Make sure the image  takes up any remaining height in parent Linear layout

# #2



Използвайте  
RelativeLayout.

Старайте се да бъдете  
възможно най-точни при  
пресъздаването.

Текстът на паролата да се  
скрива при въвеждане.

Добавете и опция за  
възраст, като при нея да  
се отваря само меню с  
цифри.