

Android Development

Views and Layouts

part 1

SoftUni Team
Teodor Kostadinov



SoftUni
Foundation



BY NC SA



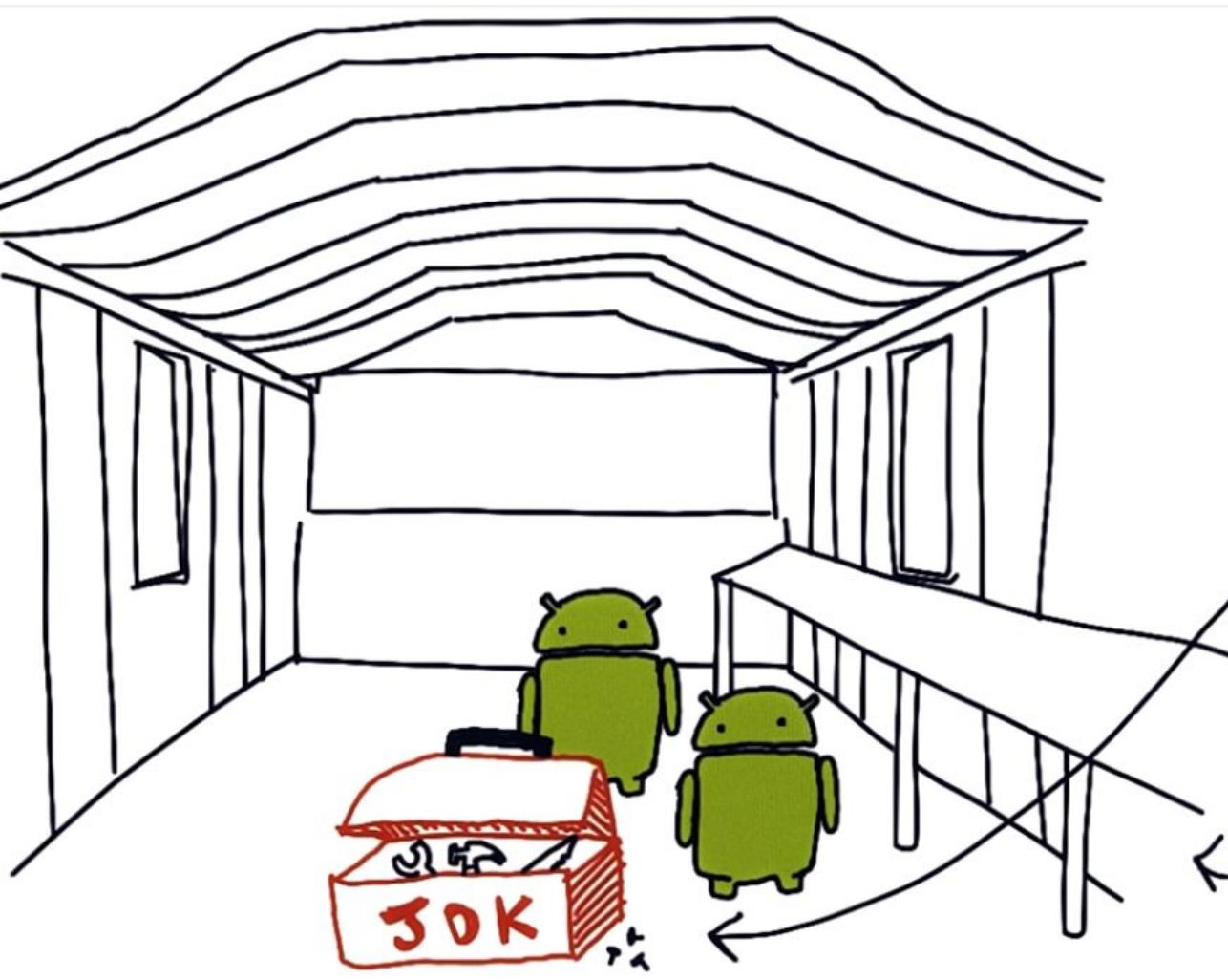
Software University
<http://softuni.bg>



sli.do
#Android

Some notes before we start

- We'll be using a lot of new words - don't be afraid to ask if you forget some
- Here is a glossary for all terms we'll cover: [Terms Glossary](#)
- You have to install [Java JDK](#) if you don't have it
- You have to download and install [Android Studio](#), this is the program we'll be using
- Start the Android Studio and wait until it downloads everything it needs
- You'll need an Android phone and a USB cable also



Installing Android Studio

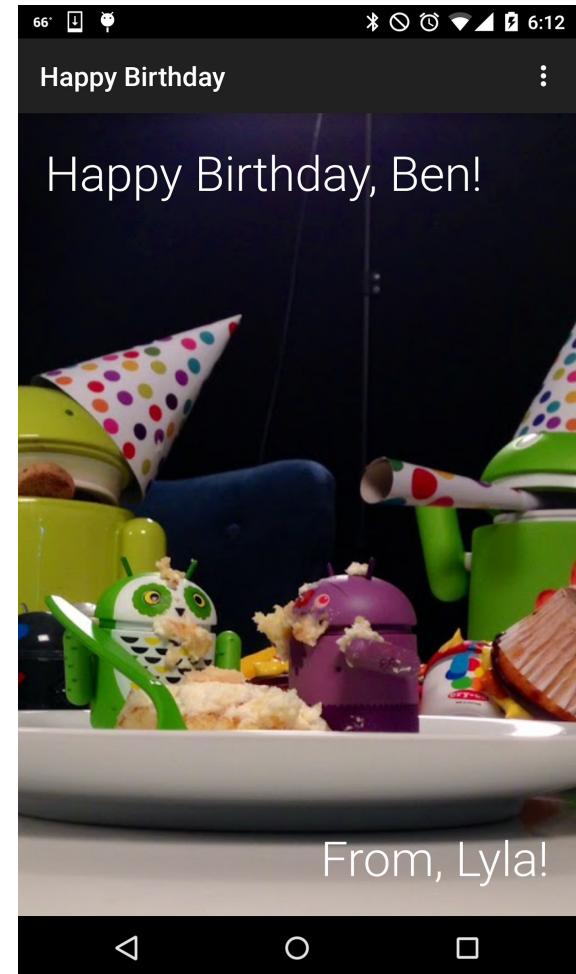
1. Install the
Java Development Kit
2. Install
Android Studio

Installing Your First App



What you'll learn

- Recognize how an app is made
- Create an app's outlook
- Code simple logic in your app



Common obstacles

- Lack of confidence that you can be a developer
- Code seems intimidating
- Tools seem complex
- Too much jargon
- Concepts are too abstract

sharing time

What did you learn last time?

views &
layouts

VIEWS



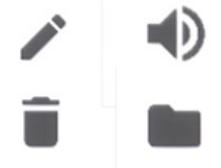
Text View

Morning Pizza

I'll be in your neighborhood doing errands
this weekend if you want to meet up for
brunch. Lemme know.

Show notifications

ImageView



Button



BUTTON

BUTTON

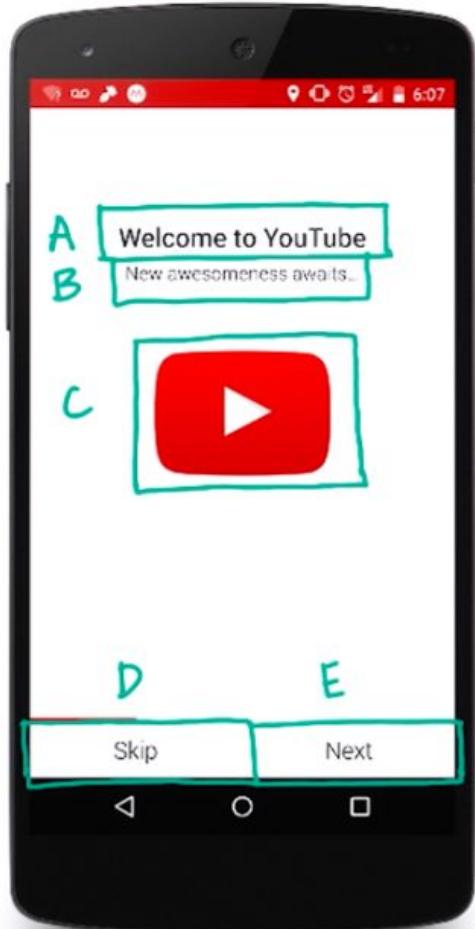




VIEWS

Identify the type of View

A	ImageView	ImageView	Button
B	TextView	ImageView	Button
C	TextView	ImageView	Button
D	TextView	ImageView	Button
E	TextView	ImageView	Button



VIEWS

Identify the type of View

A	TextView	ImageView	Button
B	TextView	ImageView	Button
C	TextView	ImageView	Button
D	TextView	ImageView	Button
E	TextView	ImageView	Button

xml

Code

- Tell us about a time when you gave specific instructions for something
- Tell us about a time when you learned a new software

**Code is written instructions that Android Studio
understands and does a specific thing**

TEXTVIEW

```
<TextView  
    android:text="Happy Birthday!"  
    android:background="@android:color/darker_gray"  
    android:layout_width="150dp"  
    android:layout_height="75dp" />
```

Write your observations here :



XML

- Programming language
- Specific syntax
- Describes how a View looks

XML SYNTAX

```
<TextView  
    android:text="Happy Birthday!"  
    android:background="@android:color/darker_gray"  
    android:layout_width="150dp"  
    android:layout_height="75dp">  
</TextView>
```

XML

- Uses < and > to show where a view begins and ends
- The name is called a TAG
- Views have an opening TAG and a closing TAG
- Simplest TAG looks like this:

<TAG

attribute="Value" >

</TAG>

<TAG

attribute="Value"

/

XML SYNTAX

```
<TextView  
    android:text="Happy Birthday!"  
    android:background="@android:color/darker_gray"  
    android:layout_width="150dp"  
    android:layout_height="75dp" />
```

Attributes:

- width
- height
- background
- text



Attributes:

- black hair
- brown eyes

Line
numbers ↴

XML SYNTAX

```
1 <TextView  
2     android:text="Happy Birthday!"  
3     android:textColor="@android:color/white"  
4     android:background="@android:color/black"  
5     android:layout_width="200dp"  
6     android:layout_height="300dp" />
```

1. What's the name of the XML element?
2. List all attribute names (not attribute values) separated by commas
3. On what line number is the tag closed?

XML SYNTAX

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:text="Happy Birthday"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:text="You're the best!"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

You are
writing!
XML!

CHANGE THE TEXTVIEW

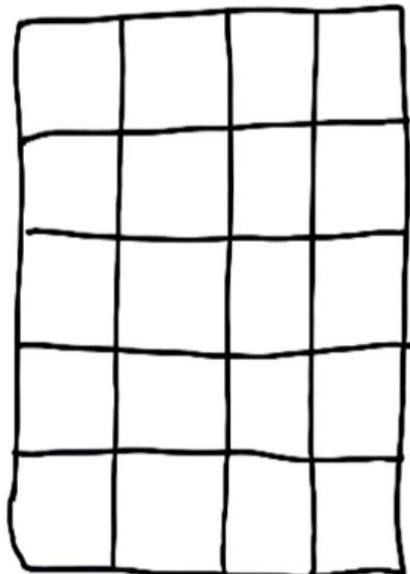
- Change the text in the TextView
- Try changing the size of the TextView.
- Try to create an error. Then undo it.
 

link:

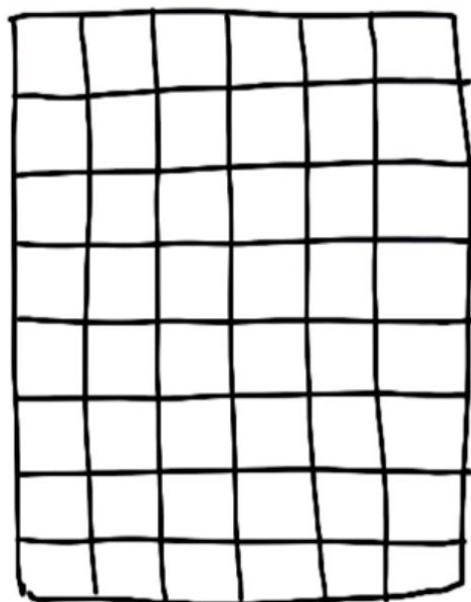
<http://labs.udacity.com/android-visualizer/#/android/text-view>

size: dp

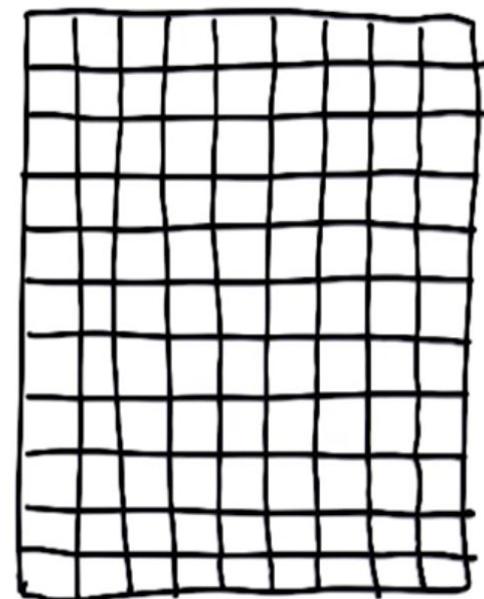
DENSITY-INDEPENDENT PIXELS



Medium Resolution Device



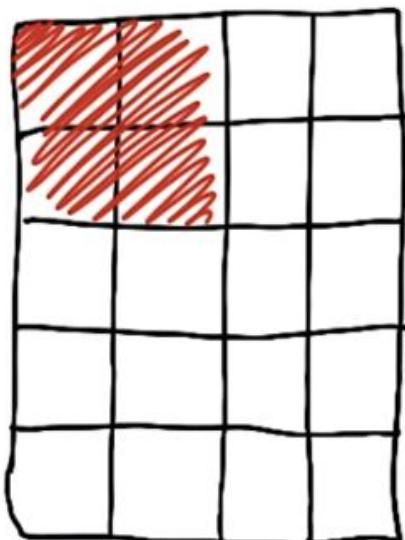
High Resolution Device



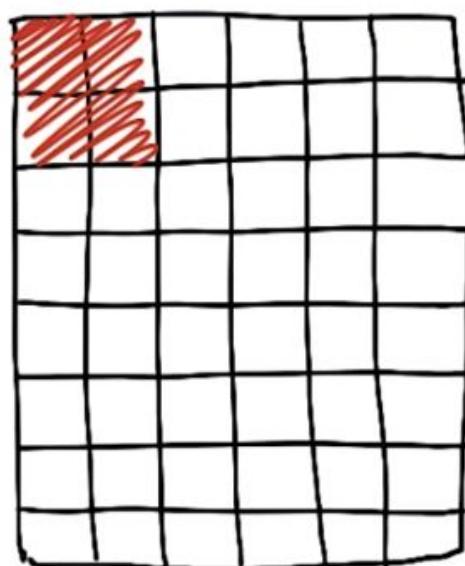
Extra-High Resolution Device

DENSITY-INDEPENDENT PIXELS

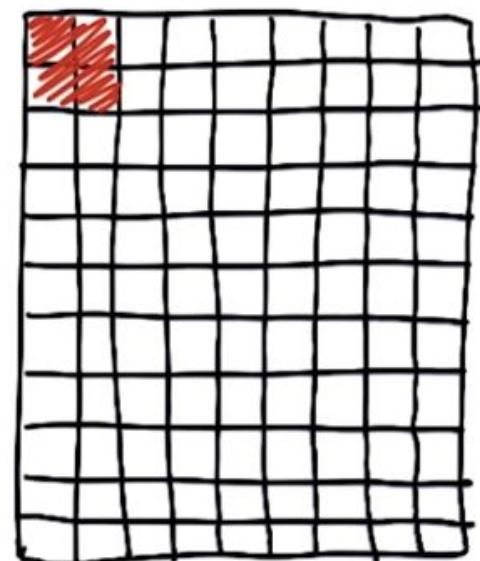
2 pixels by 2 pixels



Medium Resolution Device



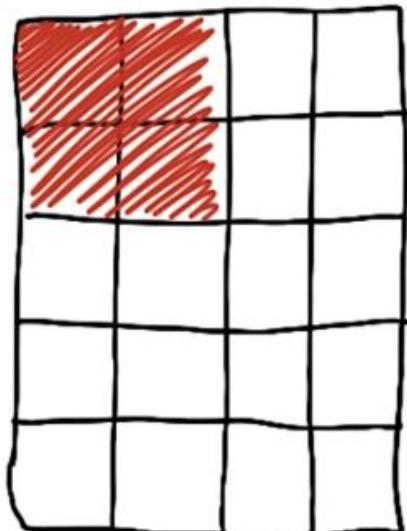
High Resolution Device



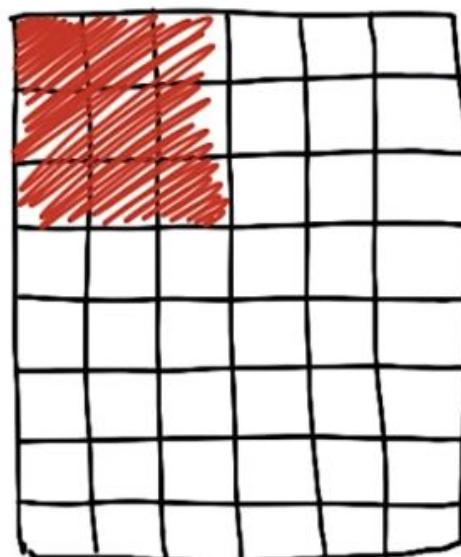
Extra-High Resolution Device

DENSITY-INDEPENDENT PIXELS

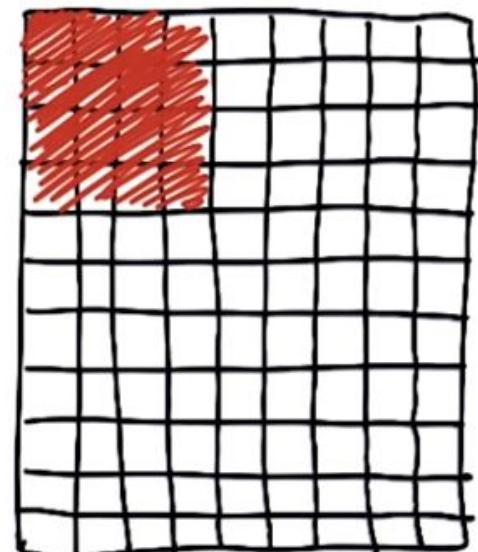
2 dp by 2 dp



Medium Resolution Device



High Resolution Device



Extra-High Resolution Device

errors

GETTING PAST ERRORS

1. Read ⚡ the error message
2. Compare to working code
3. Undo ↺
4. Ask for help ☺

Fix the errors yourself

- Go to <http://labs.udacity.com/android-visualizer/#/android/xml-syntax-errors>
- Try to find all 4 errors
- What were they?

size: wrap_content

Wrap content

- Makes the size of the View just right, so the content of the View is not cut off
- Can be used both for width and height

WRAP_CONTENT

Change the width and height of the TextView to wrap-content

Change the text to be more than 1 line of text on the device

Link:

<http://labs.udacity.com/android-visualizer/#/android/wrap-content>

Quick! Wrap this mug!



more attributes

Text Size

- android:textSize
- Can use dp values, but prefer using sp, instead
- Sp is same as dp, but the user can change it in Settings
- Sp is only for text size
- Material Design Text Tips:
<http://www.google.com/design/spec/style/typography.html#typography-styles>

TEXT SIZE

Link:

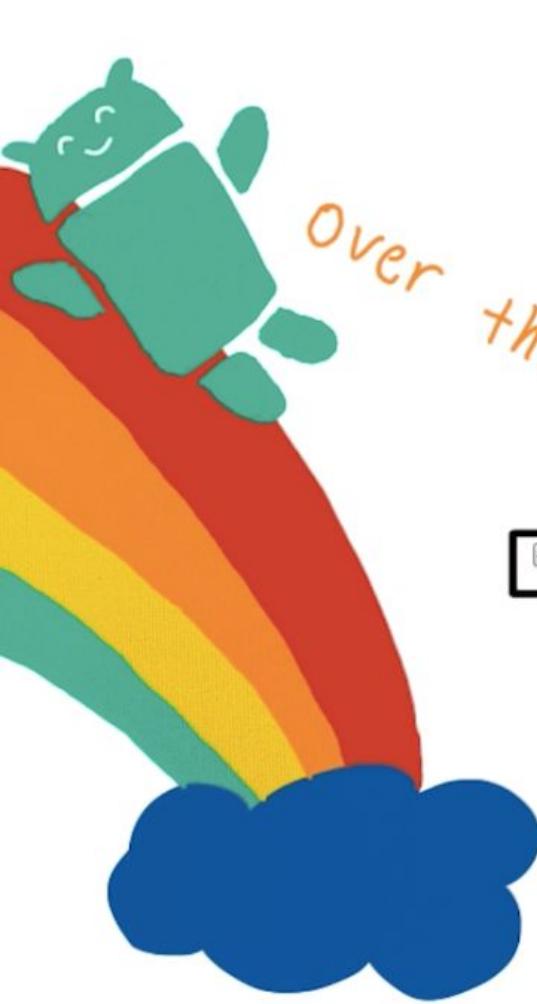
<http://labs.udacity.com/android-visualizer/#/android/text-size>

<https://plus.google.com/+AndroidDevelopers/posts/gQuBtuk6iG>

- Change the TextView font size using sizes from Material Design Spec
- Try learning about text appearance from  #AndroidDev #Protip

Color

- android:background
- android:textColor
- Uses either Android system colors or RBG hex values
 - #FF0000 - red
 - #00FF00 - blue
 - #0000FF - green
- Material Colors:
<https://material.io/guidelines/style/color.html#color-color-palette>
- Hex colors information:
<http://www.htmlgoodies.com/tutorials/colors/article.php/3478951>



TEXT COLOR

Over the rainbow and into a XML visualizer

- Change the color of the background and text to 2 different colors from the Material Design spec

Image View

ImageView

- android: src
 - Used to choose the image to show
- android: scaleType
 - Tells the View how to crop and center the image

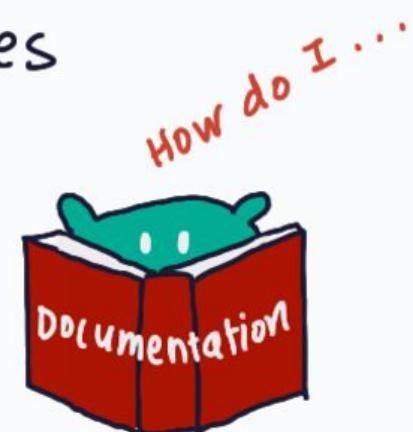
IMAGE VIEW

- Change the height and width to be a fixed dp value (i.e. 100dp)
- Change scaleType between center and centerCrop
- Change to a different image (see Available Images link)

Link: <http://labs.udacity.com/android-visualizer/#/android/simple-imageview>

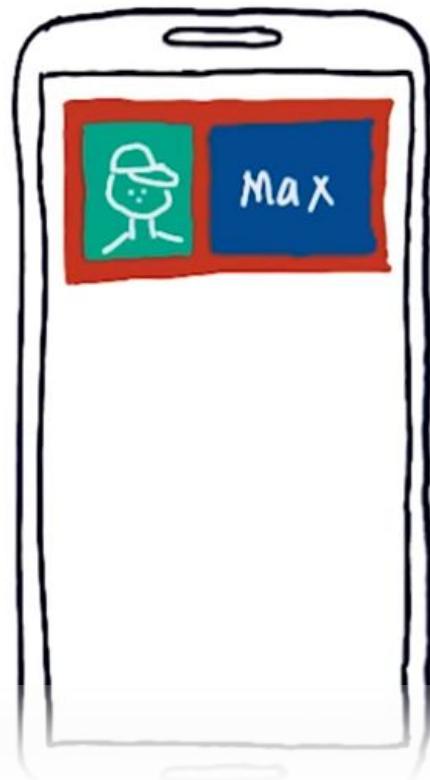
LOOKING AT DOCUMENTATION

- Google search for "textview android"
- Use Find feature (CtI+F or Cmd+F) in web browser to search the page, for the XML attribute that makes TextView text bold or italic
- Use that XML attribute!



view groups

VIEW GROUPS



ViewGroup

- Contain other Views
- Used for grouping of views and telling them how to arrange on the screen
- Views inside a ViewGroup are called children to that ViewGroup, the ViewGroup is their parent
- A View is a sibling to another View, if both are inside the same ViewGroup

MEET THE VIEWS

Parent Views



Ray
RelativeLayout
ViewGroup



Laura
LinearLayout
ViewGroup

Children Views



Tommy
TextView



Ivy
ImageView

LinearLayout

PARENTAL SUPERVISION



Laura

LinearLayout



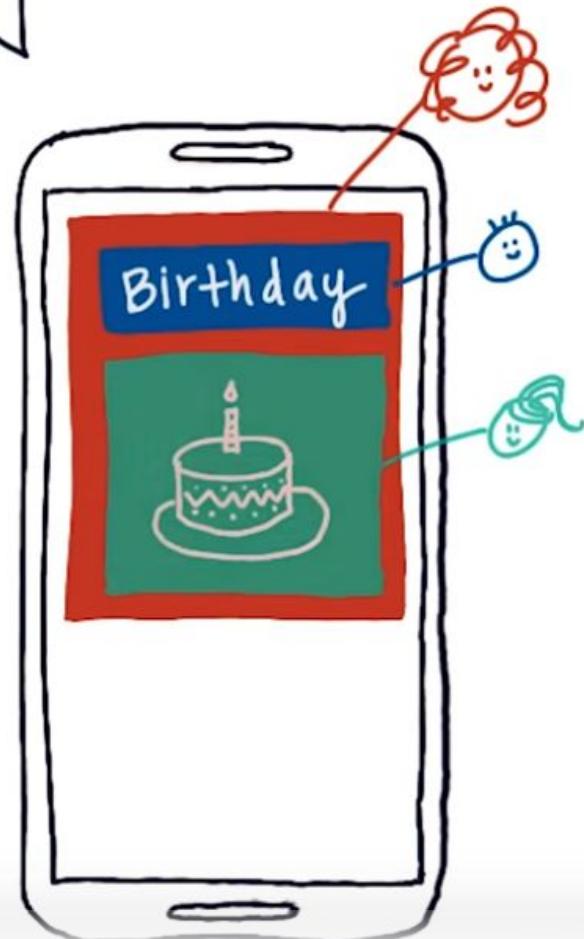
Tommy

TextView



Ivy

ImageView



VIEW GROUPS

1. How many views are there? 4

2. The mountain image is a _____ of the red ViewGroup.

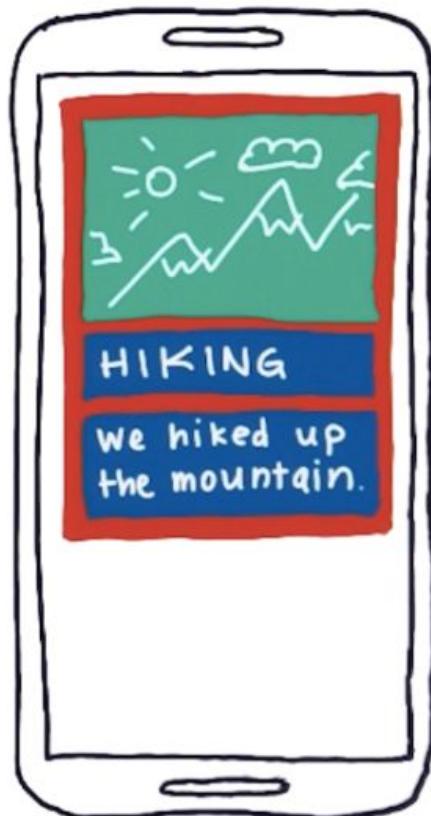
- child parent

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



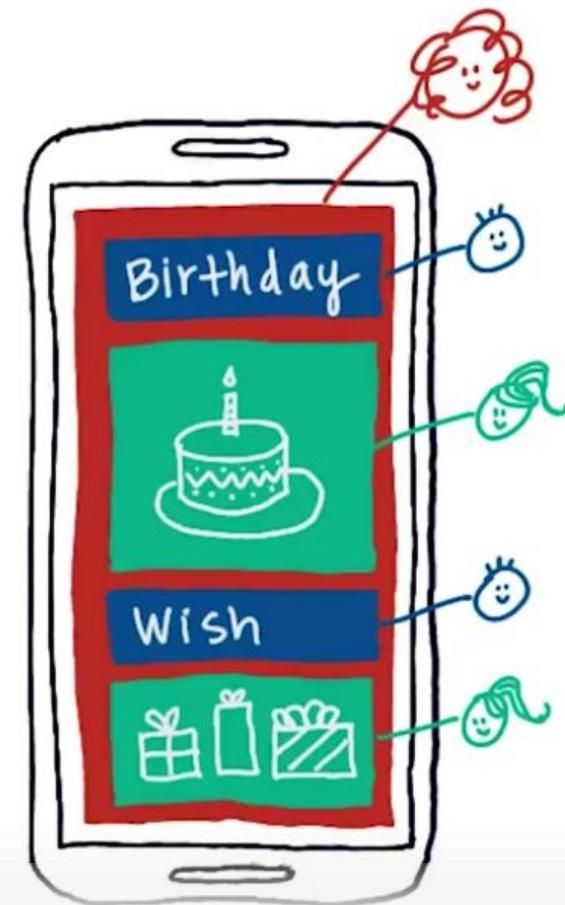
Tommy

TextView



Ivy

ImageView



LINEAR LAYOUT



Laura

LinearLayout



Tommy

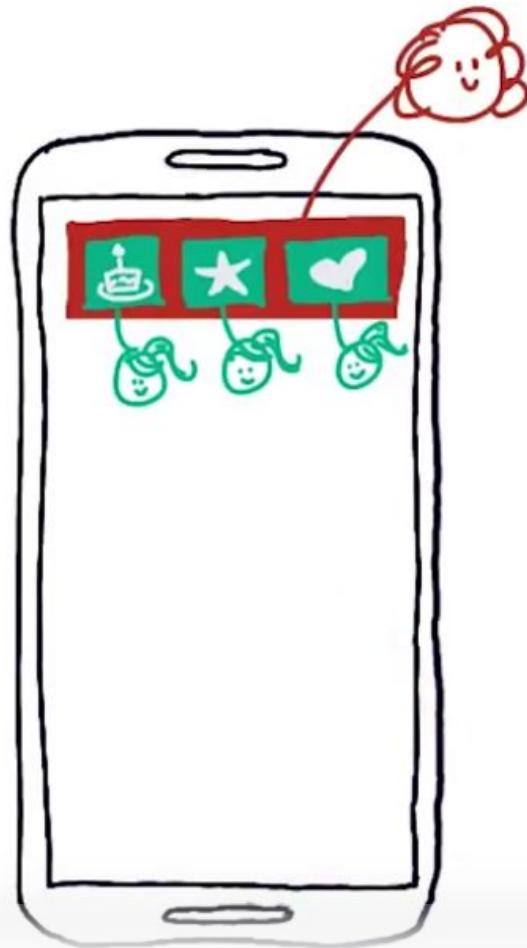
TextView



Ivy

ImageView

Horizontal row



RELATIVE LAYOUT

Relative to parent



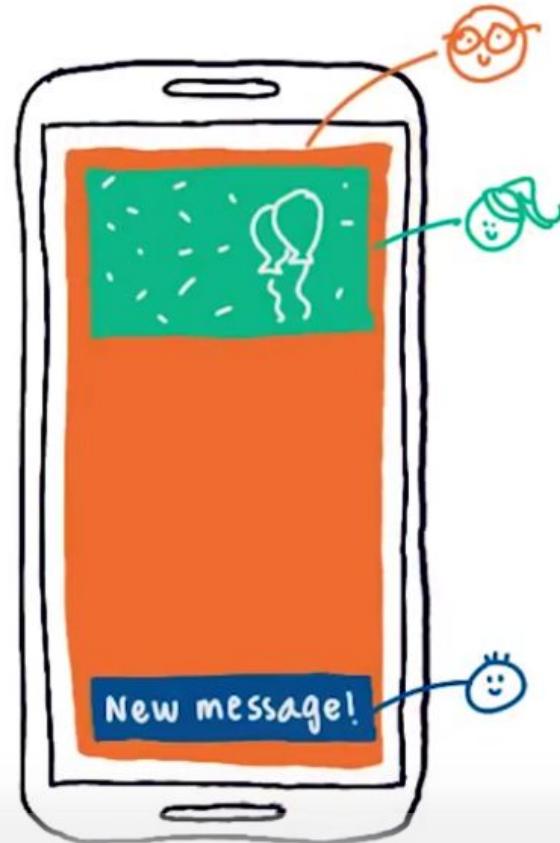
Ray
RelativeLayout



Tommy
TextView

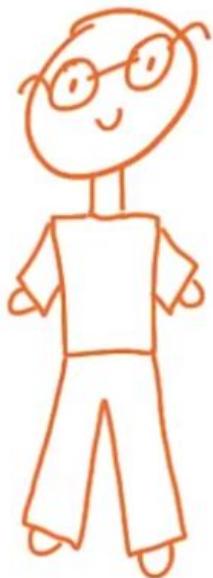


Ivy
ImageView



RELATIVE LAYOUT

Relative to other children



Ray

Relative Layout

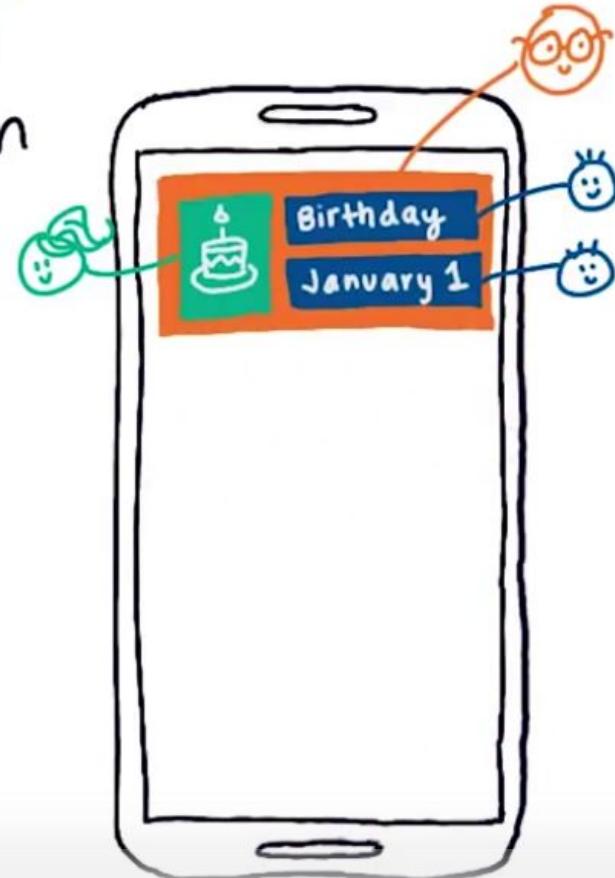


Tommy

TextView



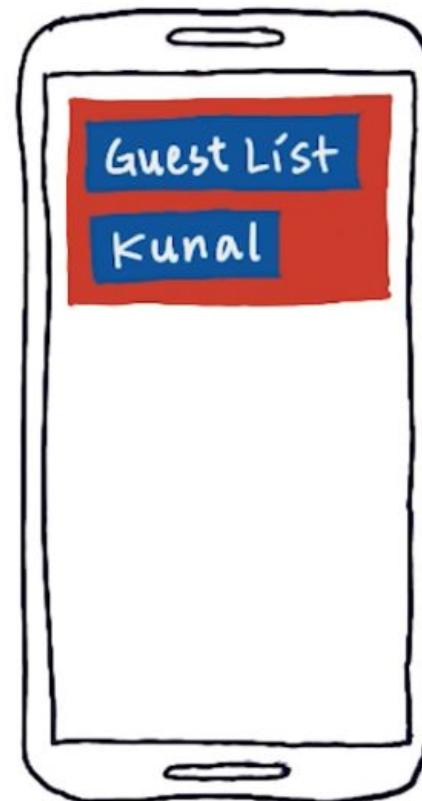
Ivy
ImageView



LINEAR LAYOUT

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

Vertical or Horizontal



size: match_parent

DIFFERENT WIDTHS ON CHILD VIEWS

200 dp



wrap-content



match-parent



DIFFERENT HEIGHTS ON CHILD VIEWS

200 dp



wrap-content



match-parent

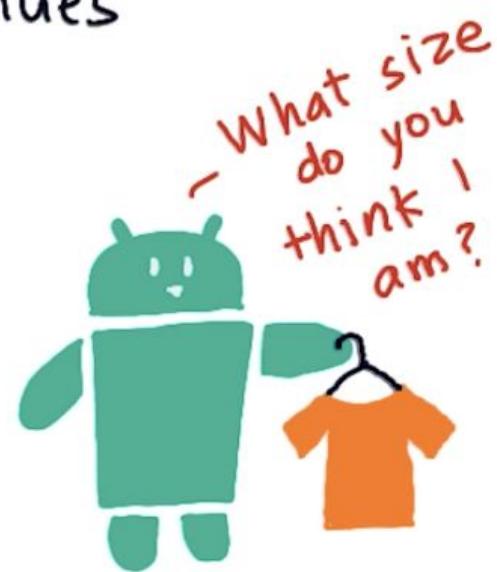


WIDTH and HEIGHT

- Try different width and height values for each TextView

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

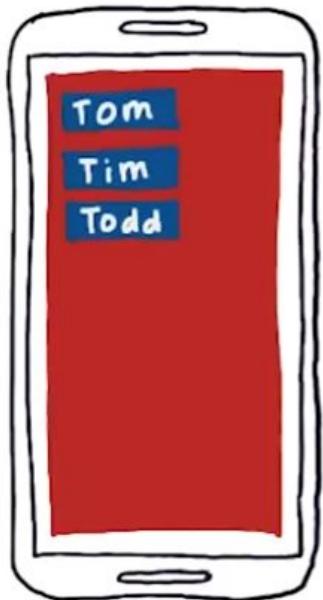
- * Remember that width and height
don't have to match *



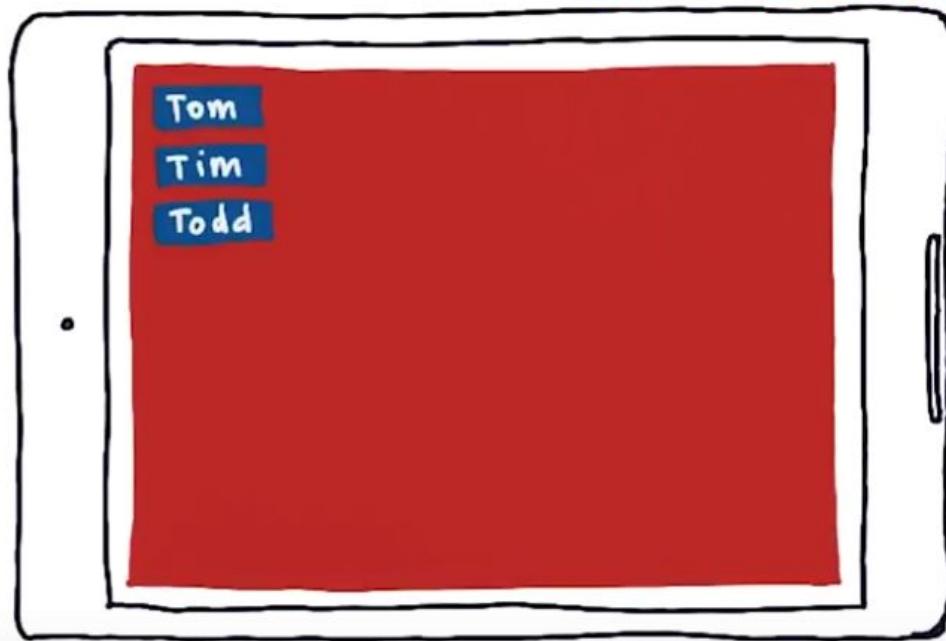
size: weight

TAKING ADVANTAGE OF SCREEN REAL ESTATE

Portrait Mode



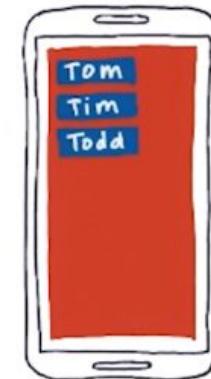
Tablet Landscape



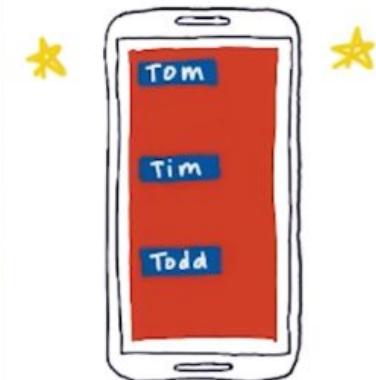
LINEAR LAYOUT WEIGHT

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:



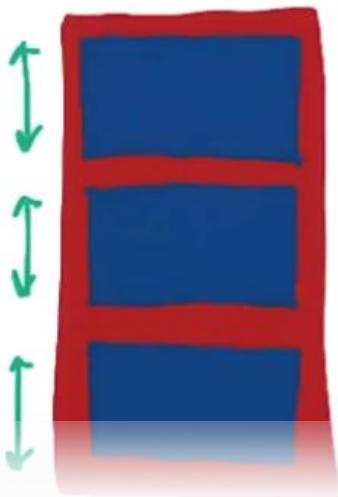
Link:

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

EditText

ImageView

Height

wrap-content

wrap-content

wrap-content

width

wrap-content

0dp

wrap-content

weight

0

1

0

MAPS APP



Horizontal linear layout



TextView

Height

wrap-content

TextView

wrap-content

TextView

wrap-content



CALL



SAVE



WEBSITE

0dp

0dp

0dp

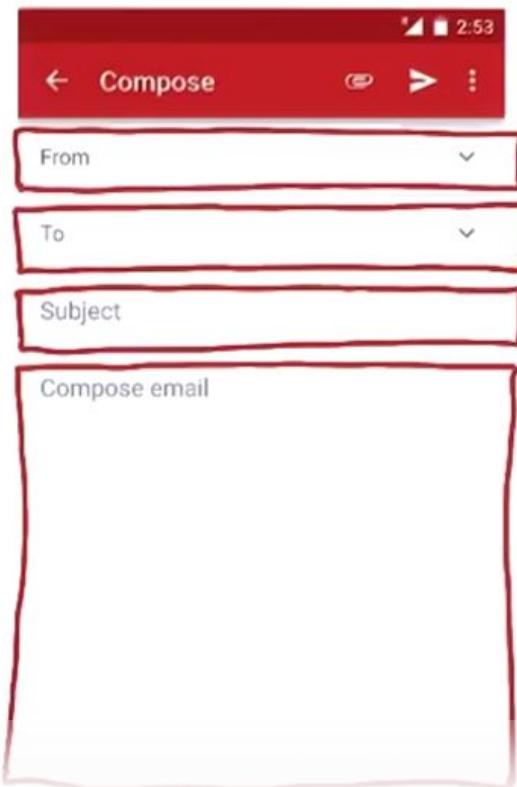
1

1

1

weight

EMAIL APP



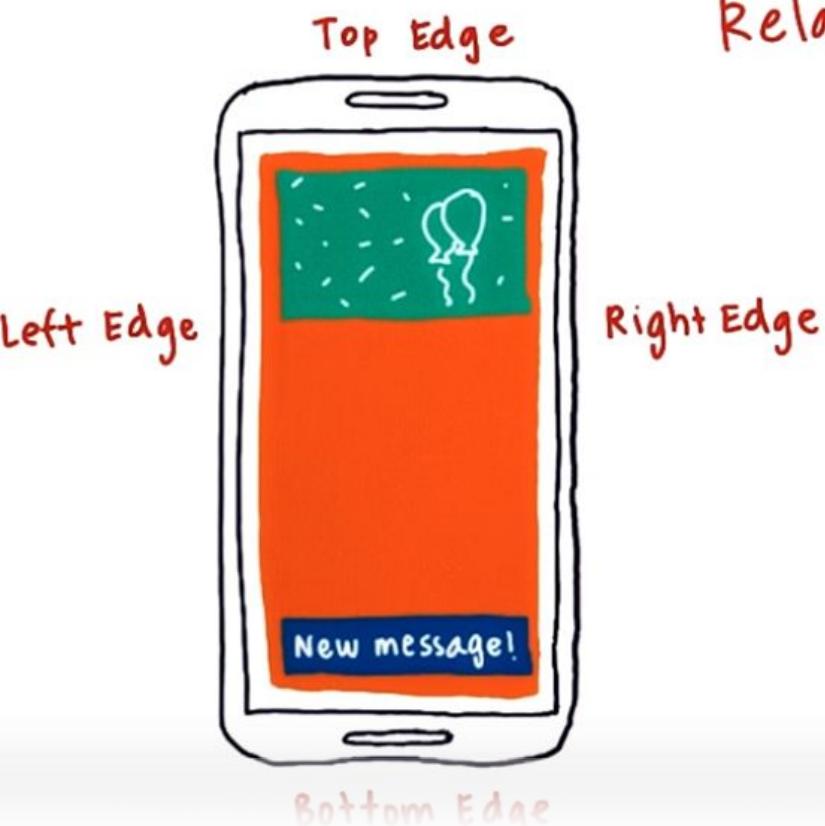
Vertical Linear Layout

	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

RelativeLayout

RELATIVE LAYOUT

Relative to Parent



Child view attributes:

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

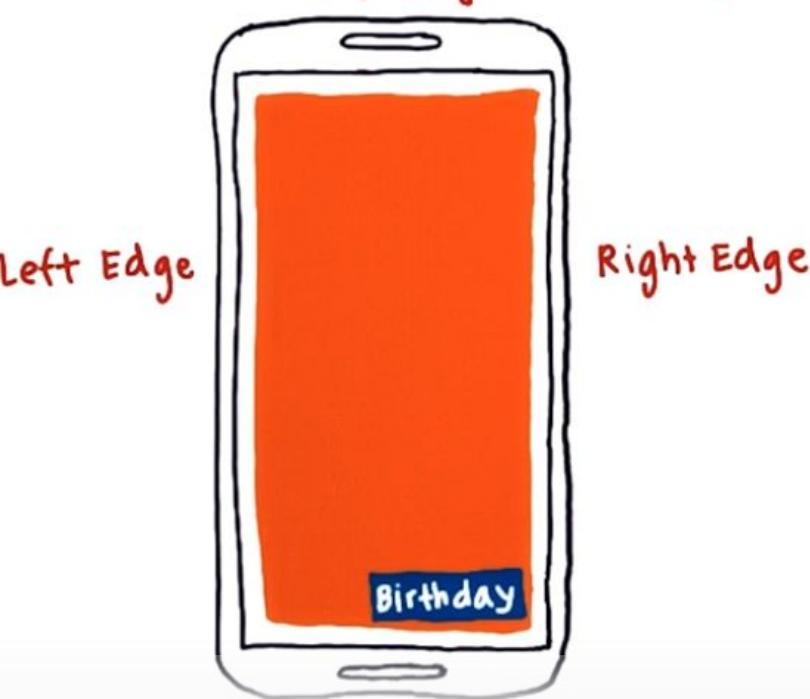
`android:layout_alignParentBottom`

`android:layout_alignParentLeft`

`android:layout_alignParentRight`

RELATIVE LAYOUT

Relative to Parent



Child view attributes:

`android:layout_align Parent Top`

`android:layout_align Parent Bottom = "true"`

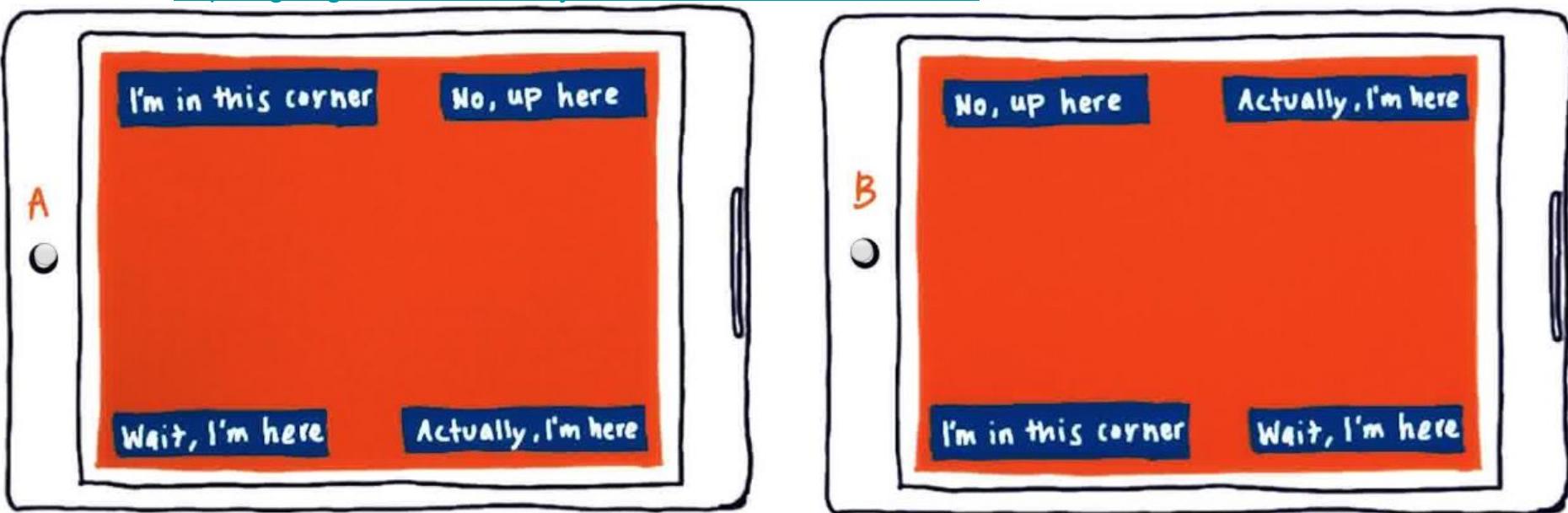
`android:layout_align Parent Left`

`android:layout_align Parent Right = "true"`

RELATIVE LAYOUT

Relative to parent

Link: <https://gist.github.com/anonymous/e7eb5ceff9dedfc376eb>



Which layout does the below XML create on the device?

Kunal Ben Kagure

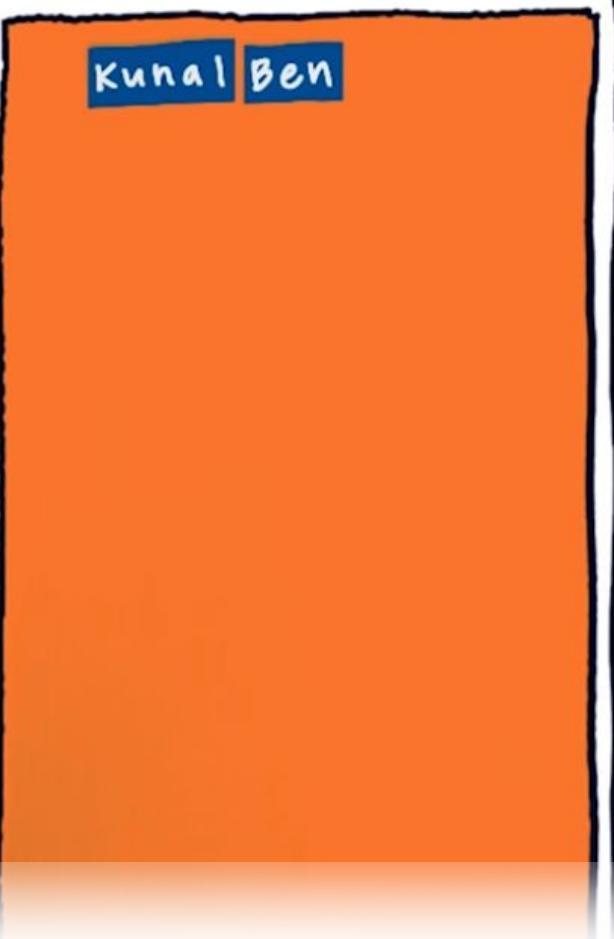
Natalie
Lyla Me

Amy
Omoju
Jennie

RELATIVE LAYOUT

Positioning children relative
to other views

Lyla	Ben	Jennie
Natalie	Kunal	Omoju
Me	Kagure	Amy



Kunal Ben

→ Assigning view ID names

On Ben TextView

android:id="@+id/ben_text_view"

→ Positioning children relative
to other views

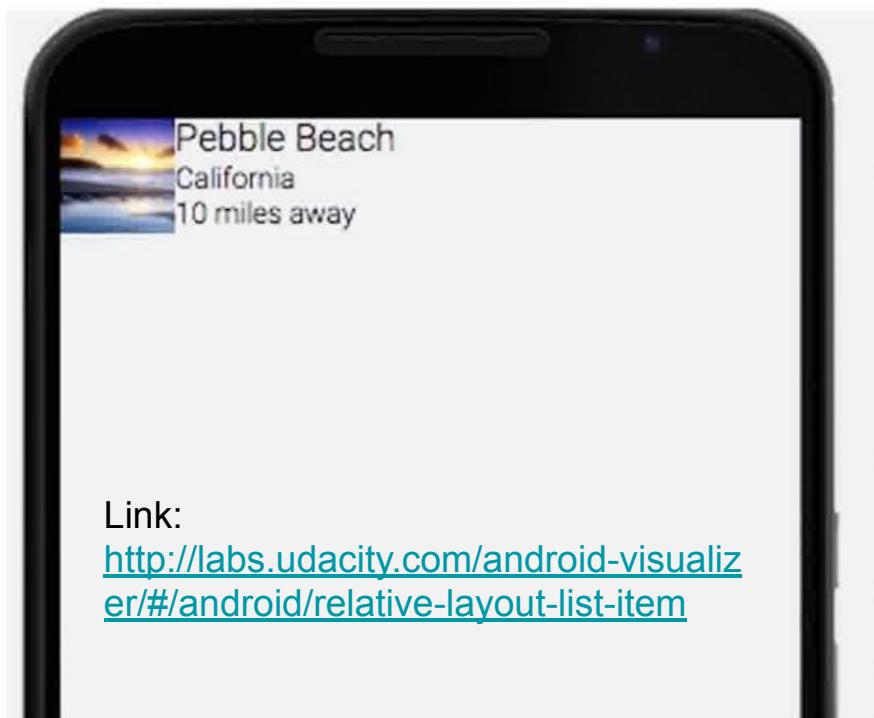
on Kunal TextView:

android:layout_toLeftOf =

RELATIVE LAYOUT for a list item

- Modify the provided XML to build this

Hint: switch from
Linear Layout to
Relative Layout

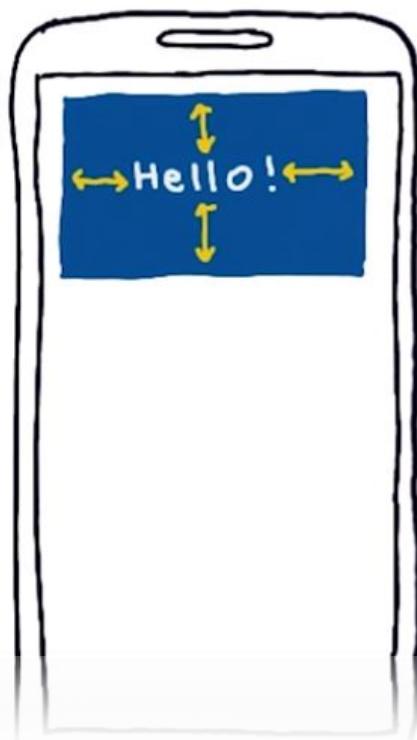


Link:

<http://labs.udacity.com/android-visualizer/#/android/relative-layout-list-item>

margin vs padding

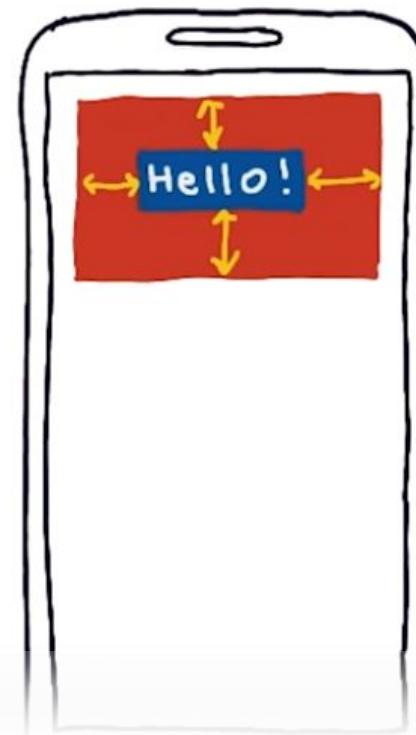
PADDING



8 dp
padding
on
TextView

Hello!

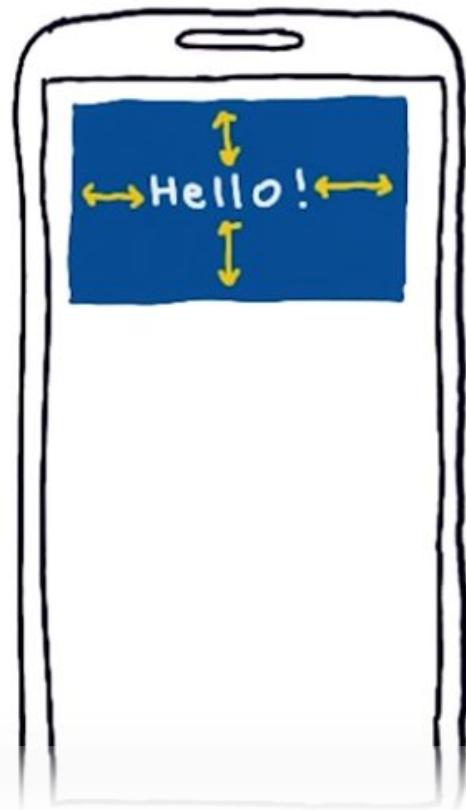
MARGIN



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"

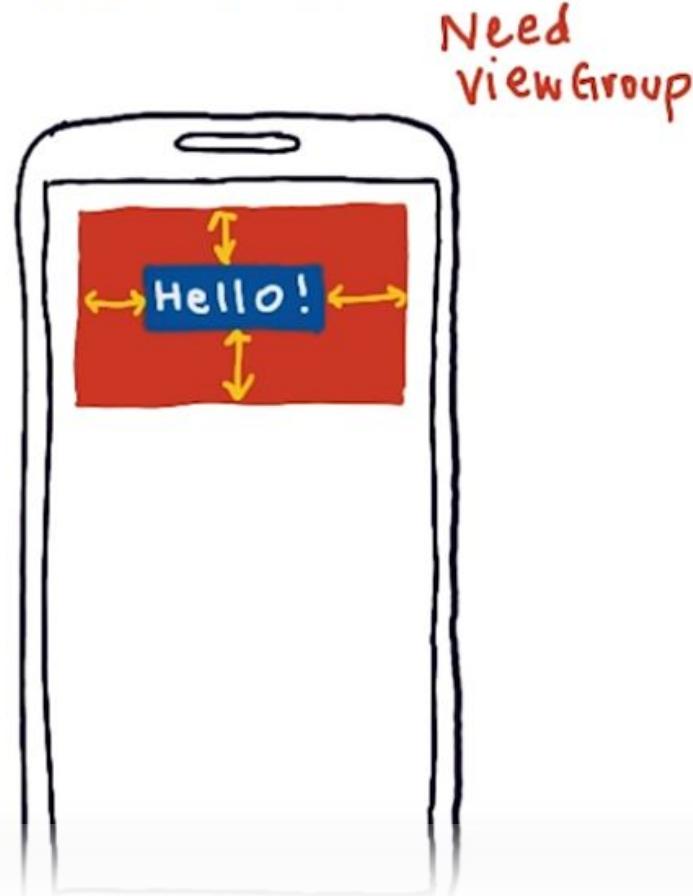
OR

android:layout_marginLeft="8dp"

android:layout_marginRight="8dp"

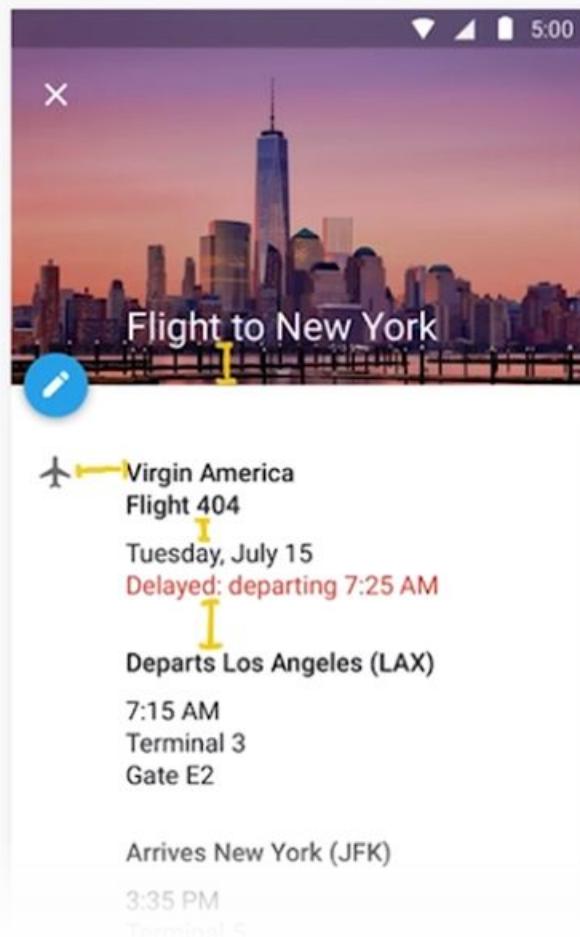
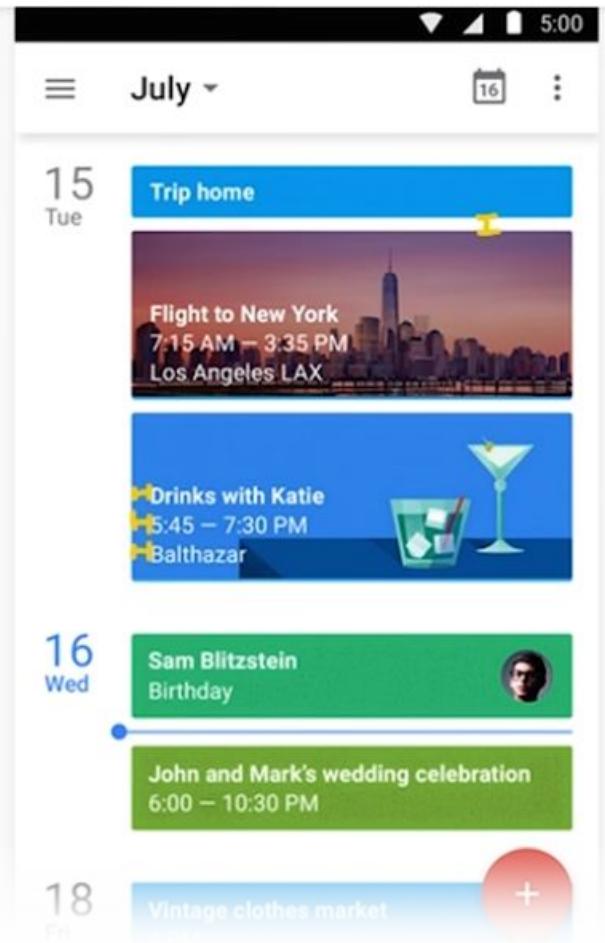
android:layout_marginTop="8dp"

android:layout_marginBottom="8dp"



Spacing

- Margin adds space from the outside of a View
- Padding adds space between the View's borders and its content inside
- More about spacing: <https://material.io/guidelines/layout/metrics-keylines.html>



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.

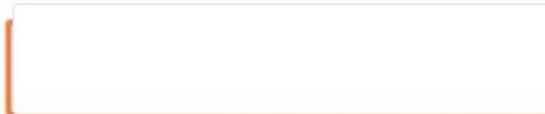


```
1 <LinearLayout  
2     android:layout_width="match_parent"  
3     android:layout_height="match_parent"  
4     android:orientation="vertical">  
5  
6     <TextView  
7         android:layout_width="match_parent"  
8         android:layout_height="wrap_content"  
9         android:text="To Ben"  
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?



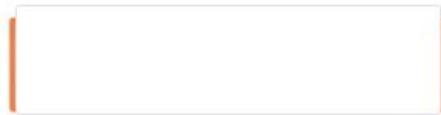
(give the line number)

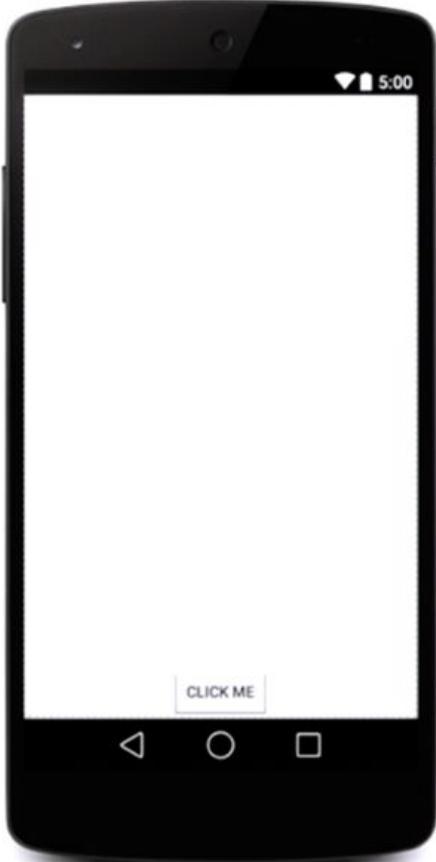
Coffee Break

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?





Coffee Break Challenge

```
<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.

```
<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>
```



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

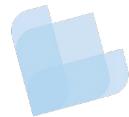


Coffee Break Challenge

You Are
DONE!



SoftUni Diamond Partners



XSsoftware



SBTech
we know sports



telenor



SoftwareGroup
doing it right

NETPEAK



SmartIT



Postbank

Решения за твоето утре

SUPER HOSTING
.BG

INDEAVR
Serving the high achievers

INFRASTISTICS®

LIEBHERR

æternity

codexio

SoftUni Organizational Partners



Trainings @ Software University (SoftUni)



- Software University – High-Quality Education and Employment Opportunities
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg



- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

