

Л2. VIEW ГРУПИ. ANDROID BASICS

У попередній лекції розглядався тільки один `TextView` або один `ImageView` на екрані. Якщо необхідно додати ще один `View` елемент нижче, то можна його скопіювати та вставити нижче. Однак виникає помилка, яка пов'язана з тим, що у `xml`-документу більше однієї корневої `View`.

<https://labs.udacity.com/android-visualizer/>

XML Layout Code

UNDO REDO Code has not been saved RESET CODE

```
1 <TextView
2   android:text="I'm a lonely TextView!"
3   android:layout_width="wrap_content"
4   android:layout_height="wrap_content"
5   android:textSize="36sp" />
6
7 <TextView
8   android:text="I'm a lonely TextView!"
9   android:layout_width="wrap_content"
10  android:layout_height="wrap_content"
11  android:textSize="36sp" />
```

Your XML document has more than one root view. There can only be one root view, and it should enclose all of your other views.

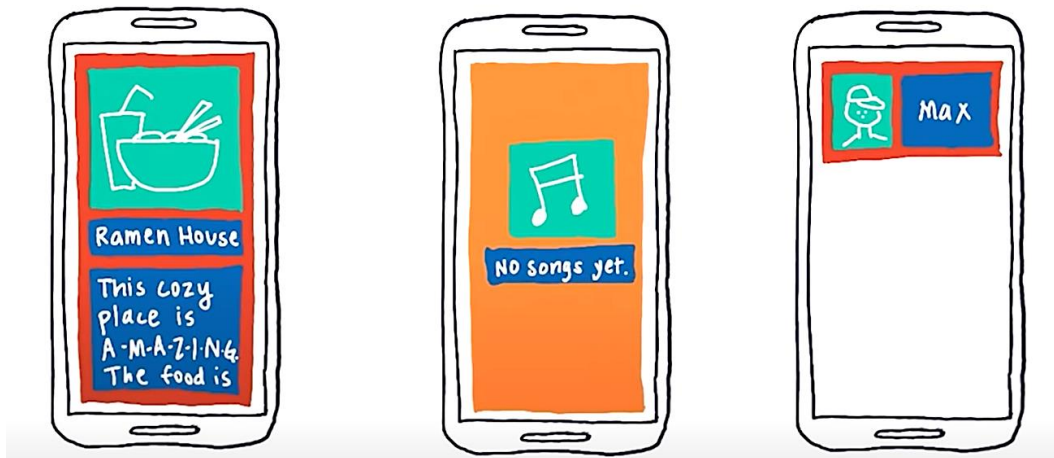
The computer wasn't able to understand your XML. (If any of your code is colored red, the error might be just before it.)

I'm a lonely TextView!

Л2. VIEW ГРУПИ. ANDROID BASICS

Коренева **View** може бути тільки одна, а всі інші повинні входити в неї. Якщо хочемо щоб обидві `TextView` залишилися на екрані, тоді потрібно помістити їх в групу `View`. В цьому випадку група `View` буде тією єдиною кореневої `View` для макета.

VIEW GROUPS



View Groups самі є `View`. На екрані вони мають прямокутну форму. Наприклад червоні та помаранчеві прямокутники. Це все `View Groups`. У них є ширина висота і колір фону, а також інші атрибути. `View Groups` можна представити як контейнер для `View`. Якщо `View` містить інші `View`, вона називається **батьком** цих `View`.

Л2. VIEW ГРУПИ. ANDROID BASICS

Parent Views



Ray
Relative Layout
ViewGroup



Laura
Linear Layout
ViewGroup

MEET THE VIEWS

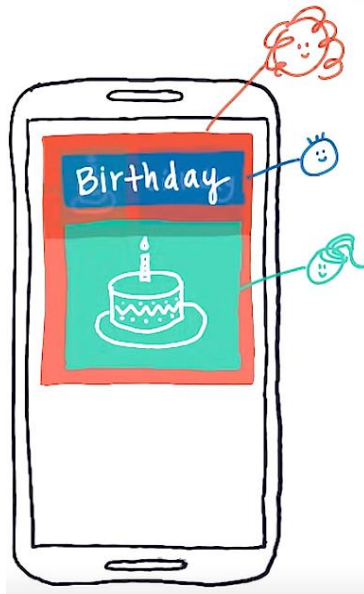
Children Views



Tommy
TextView



Ivy
ImageView



Дитячі View розміщуються відносно їх батьків



Батьки контролюють позицію своєї дитини, наприклад, одна дитяча TextView може бути у верхній частині екрану, а інша дитяча ImageView під нею або якщо батьки захотять, то може розташувати дітей по-іншому. Дитина ImageView може бути зліва, а тоді дитина TextView праворуч.

Л2. VIEW ГРУПИ. ANDROID BASICS

Дасмо відповіді на запитання

1. How many views are there?

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

child parent

3. The red ViewGroup is the _____ of the "Hiking" TextView.

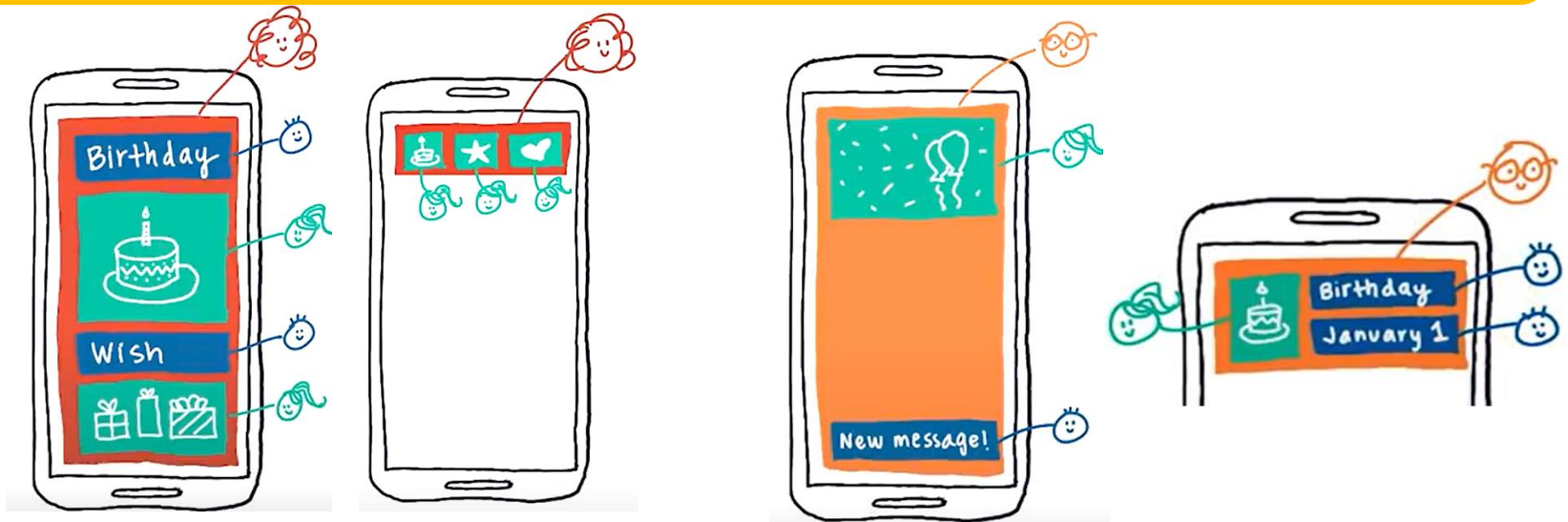
child parent

4. Which views are siblings of each other?



ТИПИ VIEW-ГРУП

У кожній *ViewGroup* є певні правила за якими дочірні елементи розташовуються усередині групи. У ***LinearLayout*** дочірні елементи можуть розташовуватися у **вертикальній колонці**. Немає обмежень по кількості *TextView* та *ImageView*. Їх можемо додавати у будь якій кількості як і інші типи *View*. У *LinearLayout* дочірні елементи також можуть розташовуватися в **горизонтальному ряду**.



При ***RelativeLayout*** застосовуються свої правила розташування дочірніх елементів на екрані, наприклад, можна розташувати дочірній *ImageView* зверху над батьківським, *TextView* внизу щодо батьківського. Ще одна корисна функція *Relative Layout* - можливість розташувати одні дочірні елементи щодо інших. *ImageView* можна помістити зліва від батька, а потім розташувати текст праворуч і ще текст нижче

LinearLayout

XML Layout Code

UNDO

REDO

Code has been saved

RESET CODE

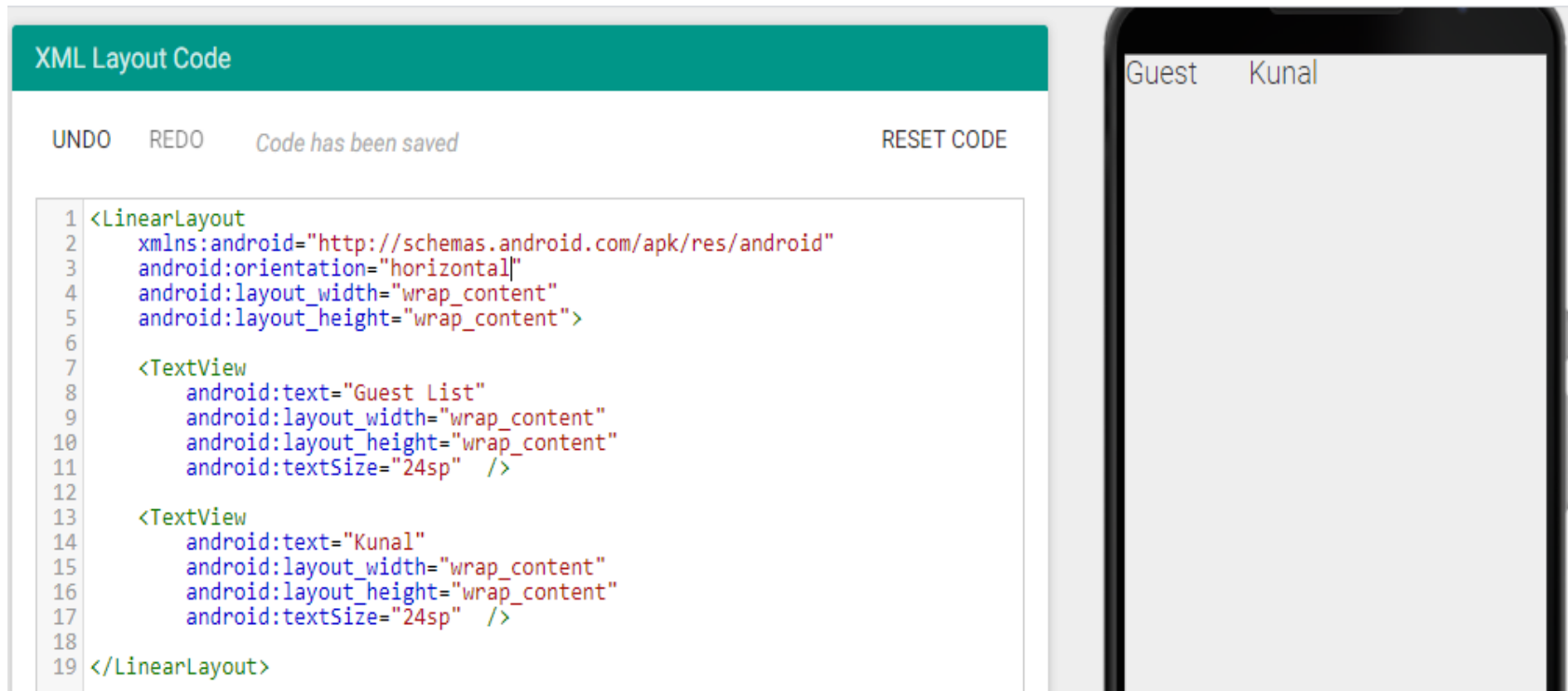
```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="wrap_content"
5   android:layout_height="wrap_content">
6
7   <TextView
8     android:text="Guest List"
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content"
11    android:textSize="24sp" />
12
13   <TextView
14     android:text="Kunal"
15     android:layout_width="wrap_content"
16     android:layout_height="wrap_content"
17     android:textSize="24sp" />
18
19 </LinearLayout>
```



Guest List
Kunal

Давайте розглянемо код уважніше. Він починається з кутової дужки, що відкриває, потім ім'я View групи **LinearLayout**, потім перераховуються атрибути та в кінці кутова дужка, що закриває. Дочірні елементи вставляються між тегами, що відкривають та закривають. Атрибут **android: orientation** визначає чи є розмітка стовпцем або рядком. Значення цього атрибута **horizontal** для рядка та **vertical** для стовпця

LinearLayout



The image shows a screenshot of an Android Studio XML editor. The left pane displays the XML layout code for a LinearLayout. The code defines two TextViews: one with the text "Guest List" and another with the text "Kunal". Both TextViews are set to wrap content and have a text size of 24sp. The right pane shows a preview of the UI, displaying the text "Guest" and "Kunal" side-by-side.

```
XML Layout Code
```

UNDO REDO Code has been saved RESET CODE

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="horizontal"
4   android:layout_width="wrap_content"
5   android:layout_height="wrap_content">
6
7   <TextView
8     android:text="Guest List"
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content"
11    android:textSize="24sp" />
12
13   <TextView
14     android:text="Kunal"
15     android:layout_width="wrap_content"
16     android:layout_height="wrap_content"
17     android:textSize="24sp" />
18
19 </LinearLayout>
```

Guest Kunal

Рядок **`xmlns:android=http://schemas.android.com/apk/res/android`** це оголошення простору імен XML, щоб вказати що всі атрибути відносяться до Андроїд. Тому всі атрибути починаються з `android`

match_Parent

Параметр атрибуту `match_Parent` вказує на відповідність батьківському View. Розмір по ширині та висоті дочірніх елементів буде відповідати батьківському.



Задаємо для кожного дочірнього елементу фіксовану ширину 200 dp. Якщо контент більше **200 dp**, то обрізається його частина. Якщо задаємо для кожного дочірнього елементу ширину **wrap-content**, то ширина контенту кожного елементу різна. Якщо задати ширину **match-parent**, то ширина усіх дочірніх елементів буде однаковою та не залежить від їх контенту.



Аналогічно змінюється і висота дочірніх елементів, якщо встановити значення 200dp, `wrap-content`, `match-parent`.

match_Parent

Розглянемо XML код з атрибутом `layout_widht`, який має значення `wrap_content` та `match_parent` (`1_match_Parent`).

XML Layout Code

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:background="@android:color/darker_gray">
7
8   <TextView
9     android:text="VIP List"
10    android:layout_width="wrap_content"
11    android:layout_height="wrap_content"
12    android:background="#4CAF50"
13    android:textSize="24sp" />
14
15   <TextView
16     android:text="Kunal"
17     android:layout_width="wrap_content"
18     android:layout_height="wrap_content"
19     android:background="#4CAF50"
20     android:textSize="24sp" />
21
22   <TextView
23     android:text="Kagure"
24     android:layout_width="wrap_content"
25     android:layout_height="wrap_content"
26     android:background="#4CAF50"
27     android:textSize="24sp" />
```



VIP List
Kunal
Kagure
Lyla

XML Layout Code

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:background="@android:color/darker_gray">
7
8   <TextView
9     android:text="VIP List"
10    android:layout_width="match_parent"
11    android:layout_height="wrap_content"
12    android:background="#4CAF50"
13    android:textSize="24sp" />
14
15   <TextView
16     android:text="Kunal"
17     android:layout_width="match_parent"
18     android:layout_height="wrap_content"
19     android:background="#4CAF50"
20     android:textSize="24sp" />
21
22   <TextView
23     android:text="Kagure"
24     android:layout_width="match_parent"
25     android:layout_height="wrap_content"
26     android:background="#4CAF50"
27     android:textSize="24sp" />
28
29   <TextView
```



VIP List
Kunal
Kagure
Lyla

Рівномірний розподіл дочірніх View-елементів (android:layout_weight)

Розглянемо XML код

XML Layout Code

UNDO

REDO

Code has been saved

RESET CODE

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="wrap_content"
5   android:layout_height="wrap_content">
6
7   <TextView
8     android:text="Tom"
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content"
11    android:textSize="24sp" />
12
13   <TextView
14     android:text="Tim"
15     android:layout_width="wrap_content"
16     android:layout_height="wrap_content"
17     android:textSize="24sp" />
18
19   <TextView
20     android:text="Todd"
21     android:layout_width="wrap_content"
22     android:layout_height="wrap_content"
23     android:textSize="24sp" />
24
25 </LinearLayout>
```

AVAILABLE IMAGES

REPORT ISSUE

CHEAT SHEET



Рівномірний розподіл дочірніх View-елементів (`android:layout_weight`)

При такому рішенні `TextView` розподіляться більш рівномірно по екрану, однак таке рішення буде працювати, тільки для певного розміру екрану.

XML Layout Code

UNDO

REDO

Code has been saved

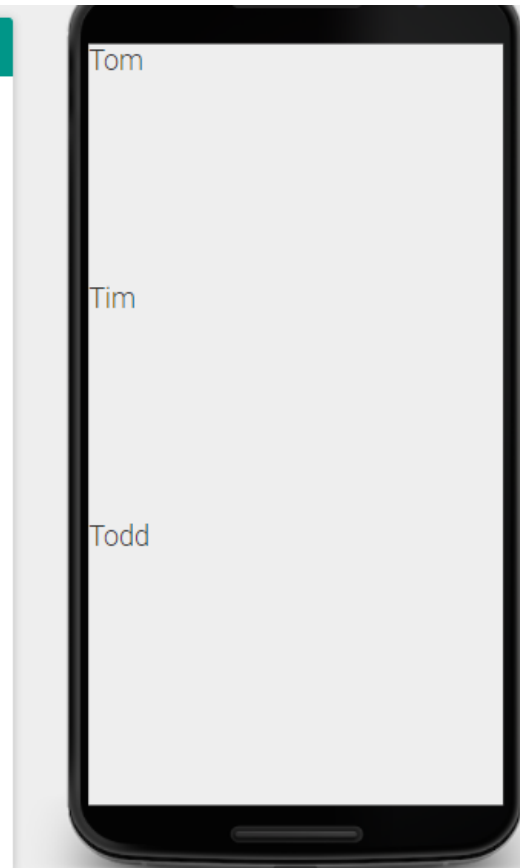
RESET CODE

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="wrap_content"
5   android:layout_height="wrap_content">
6
7   <TextView
8     android:text="Tom"
9     android:layout_width="wrap_content"
10    android:layout_height="200dp"
11    android:textSize="24sp" />
12
13  <TextView
14    android:text="Tim"
15    android:layout_width="wrap_content"
16    android:layout_height="200dp"
17    android:textSize="24sp" />
18
19  <TextView
20    android:text="Todd"
21    android:layout_width="wrap_content"
22    android:layout_height="200dp"
23    android:textSize="24sp" />
24
25 </LinearLayout>
```

AVAILABLE IMAGES

REPORT ISSUE

CHEAT SHEET



Якщо взяти значення атрибуту **`match_parent`**, то перший елемент буде мати висоту батьківського елемента та виштовхує усі інші елементи за межі екрану (всі інші елементи не будуть видні).

Рівномірний розподіл дочірних View-елементів (`android:layout_weight`)

Для того щоб рівномірно розподілити усі елементи по вертикалі, необхідно задати для всіх елементів **`android:layout_height="0dp"`** та **`android:layout_weight="1"`** (вагу елементів).

```
<LinearLayout
```

```
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@android:color/darker_gray"
  android:paddingLeft="16dp"
  android:paddingRight="16dp">
```

```
<TextView
```

```
  android:text="VIP List"
  android:layout_width="0dp"
  android:layout_height="0dp"
  android:background="#000080"
  android:layout_weight="1"
  android:textSize="24sp" />
```

```
<TextView
```

```
  android:text="Kunal"
  android:layout_width="0dp"
  android:layout_height="0dp"
  android:background="#008080"
  android:layout_weight="1"
  android:textSize="24sp" />
```

```
<TextView
```

```
  android:text="Kagure"
  android:layout_width="0dp"
  android:layout_height="0dp"
  android:layout_weight="1"
  android:gravity="top"
  android:background="#00FF7F"
  android:textSize="24sp" />
```

```
<TextView
```

```
  android:text="Lyla"
  android:layout_width="0dp"
  android:layout_height="0dp"
  android:background="#9ACD32"
  android:layout_weight="1"
  android:textSize="24sp" />
```

```
</LinearLayout>
```

Рівномірний розподіл дочірніх View-елементів (`android:layout_weight`)

Для того щоб рівномірно розподілити усі елементи по вертикалі, необхідно задати для всіх елементів **`android:layout_height="0dp"`** та **`android:layout_weight="1"`** (вагу елементів).

```
XML Layout Code

UNDO  REDO  Code has been saved  RESET CODE

1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:background="@android:color/darker_gray"
7   android:paddingLeft="16dp"
8   android:paddingRight="16dp">
9
10  <TextView
11   android:text="VIP List"
12   android:layout_width="0dp"
13   android:layout_height="0dp"
14   android:background="#000080"
15   android:layout_weight="1"
16   android:textSize="24sp" />
17
18  <TextView
19   android:text="Kunal"
20   android:layout_width="0dp"
21   android:layout_height="0dp"
22   android:background="#008080"
23   android:layout_weight="1"
24   android:textSize="24sp" />
25
26  <TextView
27   android:text="Kagure"

```

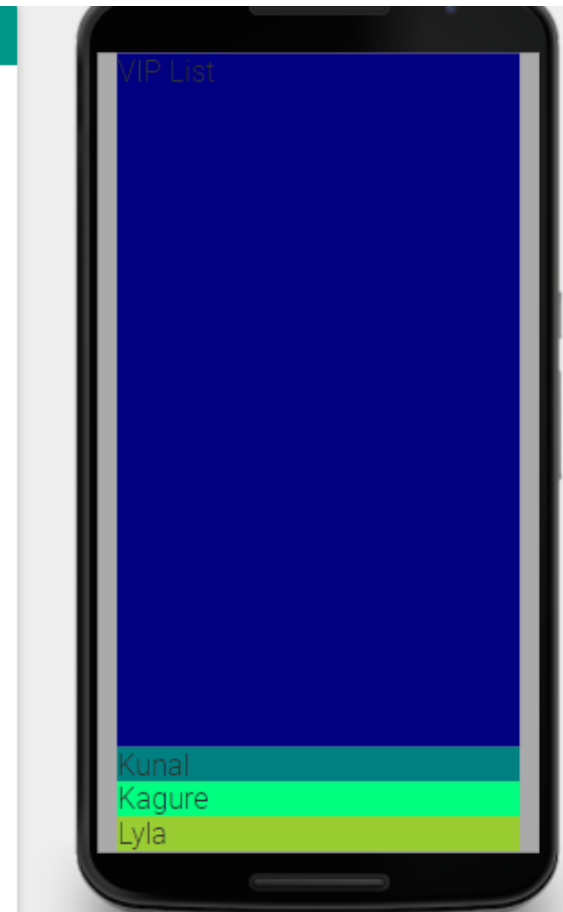
AVAILABLE IMAGES REPORT ISSUE CHEAT SHEET



Рівномірний розподіл дочірніх View-елементів (android:layout_weight)

Чим більше вага елемента, тим більше простір батьковського елемента буде зайнятий. Для Kunal, Kagure, Lyla висота не задана, тому між ними буде виставлена висота, а решта висоти буде відведено елементу з вагою 1.

```
XML Layout Code
UNDO REDO Code has been saved RESET CODE
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:background="@android:color/darker_gray"
7   android:paddingLeft="16dp"
8   android:paddingRight="16dp">
9
10  <TextView
11    android:text="VIP List"
12    android:layout_width="0dp"
13    android:layout_height="0dp"
14    android:background="#000080"
15    android:layout_weight="1"
16    android:textSize="24sp" />
17
18  <TextView
19    android:text="Kunal"
20    android:layout_width="match_parent"
21    android:layout_height="wrap_content"
22    android:background="#008080"
23    android:textSize="24sp" />
24
25  <TextView
26    android:text="Kagure"
27    android:layout_width="match_parent"
28    android:layout_height="wrap_content"
29    android:gravity="top"
30    android:background="#00FF7F"
31    android:textSize="24sp" />
32
33  <TextView
```



Рівномірний розподіл дочірніх View-елементів (android:layout_weight)

Чим більше вага елемента, тим більше простір батьковського елемента буде зайнятий.

XML Layout Code

UNDO

REDO

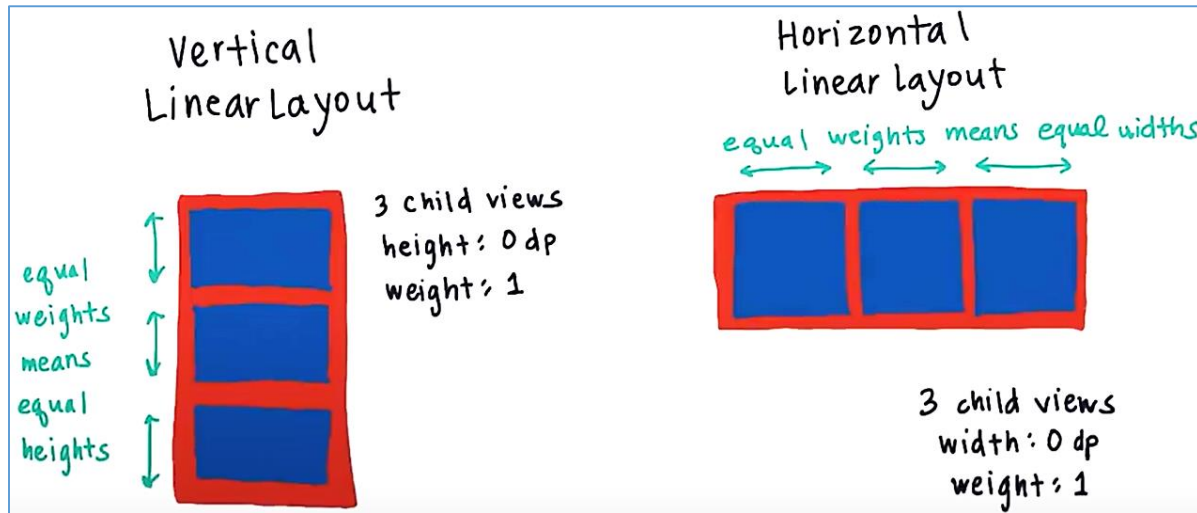
Code has been saved

RESET CODE

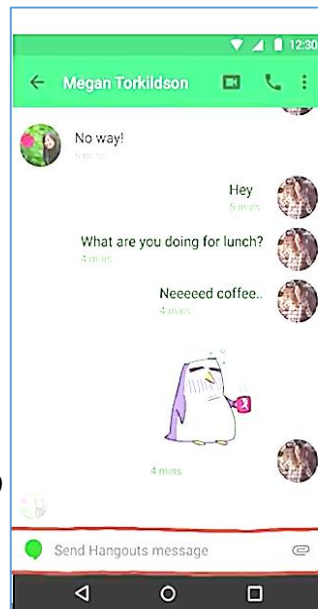
```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:background="@android:color/darker_gray"
7   android:paddingLeft="16dp"
8   android:paddingRight="16dp">
9
10  <TextView
11    android:text="VIP List"
12    android:layout_width="0dp"
13    android:layout_height="0dp"
14    android:background="#000080"
15    android:layout_weight="1"
16    android:textSize="24sp" />
17
18  <TextView
19    android:text="Kunal"
20    android:layout_width="0dp"
21    android:layout_height="0dp"
22    android:background="#008080"
23    android:layout_weight="7"
24    android:textSize="24sp" />
25
26  <TextView
27    android:text="Kagure"
28    android:layout_width="0dp"
29    android:layout_height="0dp"
30    android:layout_weight="2"
31    android:gravity="top"
32    android:background="#00FF7F"
33    android:textSize="24sp" />
```



Рівномірний розподіл дочірніх View-елементів (android:layout_weight)



Розглянемо нижній рядок, який має *Horizontal Linear layout*. Елемент по центру має вагу 1, тому ширина бокових елементів встановлюється по змісту, а він займає усю ширину.



Horizontal Linear layout

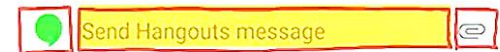
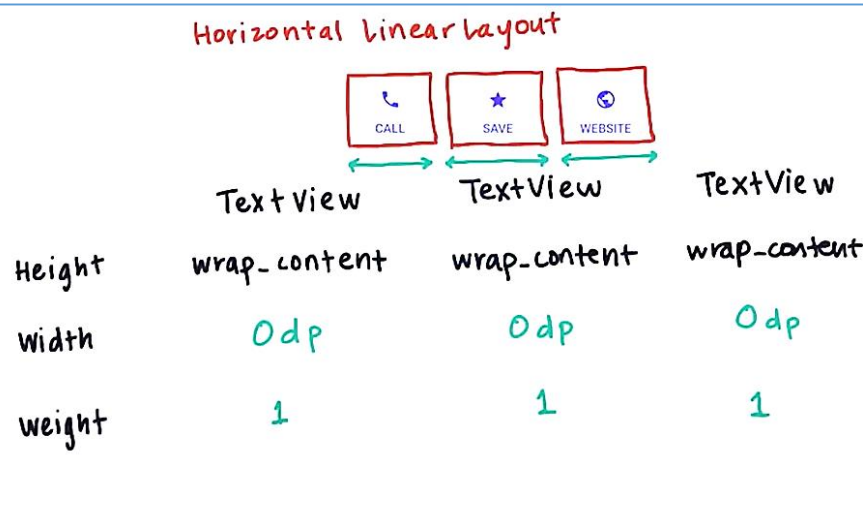
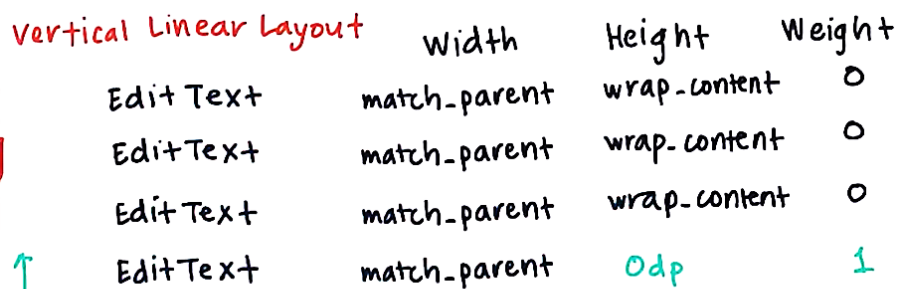
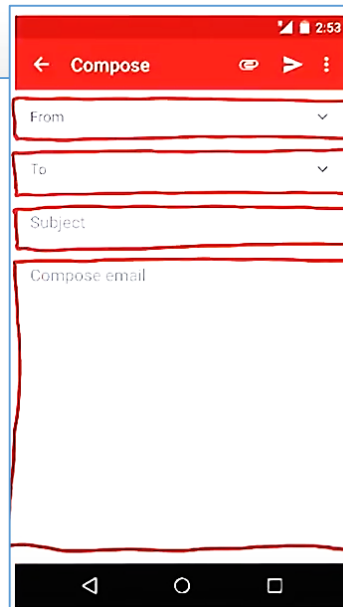


	Image View	EditText	Image View
Height	wrap_content	wrap_content	wrap_content
Width	wrap_content	0dp	wrap_content
weight	0	1	0

Рівномірний розподіл дочірніх View-елементів (android:layout_weight)



Розглянемо групу центральних елементів з Horizontal Linear layout. Важливо щоб усі три елементи рівномірно займали ширину, тому у них усіх вага 1

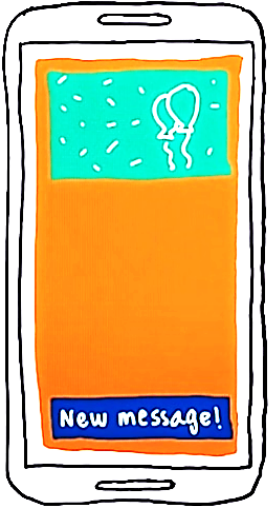


Relative Layout

Дочірні елементи відносно батьківського можна розташувати по правому, лівому, верхньому або нижньому краю.

Top Edge

Left Edge



Right Edge

Bottom Edge

Relative to Parent

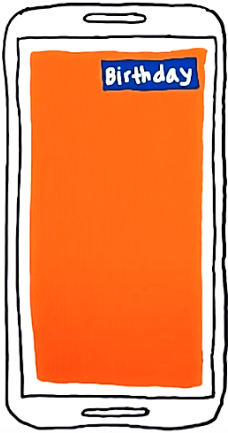
ImageView attributes:

```
android:layout_alignParentTop="true"  
android:layout_alignParentBottom="false"  
android:layout_alignParentLeft="true"  
android:layout_alignParentRight="true"
```

Для вирівнювання елемента по правому верхньому куту необхідно встановити такі атрибути

Top Edge

Left Edge



Right Edge

Bottom Edge


Relative to Parent

Child view attributes:

```
android:layout_alignParentTop="true"  
android:layout_alignParentBottom  
android:layout_alignParentLeft  
android:layout_alignParentRight="true"
```

Relative Layout

Для встановлення тексту в лівому нижньому куту використовуємо такі атрибути. По замовчуванню атрибути встановлені у значенні `false`.



Relative to Parent

Child view attributes:

- `android:layout_alignParentTop`
- `android:layout_alignParentBottom = "true"`
- `android:layout_alignParentLeft = "true"`
- `android:layout_alignParentRight`

Є ще корисні атрибути, які центрують зображення: `layout_centerVertical = "true"` та `layout_centerHorizontal = "true"`. Усі атрибути називаються параметрами групи розмітки.

Relative Layout

XML Layout Code

UNDO REDO Code has been saved RESET CODE
Розглянемо код з *Relative Layout* (4_relativeLayout)

```
1 <RelativeLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent">
5
6   <TextView
7     android:layout_width="wrap_content"
8     android:layout_height="wrap_content"
9     android:layout_alignParentRight="true"
10    android:layout_alignParentTop="true"
11    android:textAppearance="?android:textAppearanceLarge"
12    android:text="Happy" />
13
14   <TextView
15     android:layout_width="wrap_content"
16     android:layout_height="wrap_content"
17     android:layout_alignParentBottom="true"
18     android:layout_alignParentLeft="true"
19     android:textAppearance="?android:textAppearanceLarge"
20     android:text="Birthday" />
21
22   <TextView
23     android:layout_width="wrap_content"
24     android:layout_height="wrap_content"
25     android:textAppearance="?android:textAppearanceLarge"
26     android:text="To You" />
27
28 </RelativeLayout>
```



Є тег, що відкриває, **RelativeLayout** та тег, що закриває в кінці розмітки. Значення ширини та висоти для батьківського елемента задано **match parent**. Значення ширини та висоти кожного дочірнього елемента – **wrap_content** (по змісту). Для всіх дочірніх елементів є атрибути, які вказують відносне розташування (верхній правий кут, нижній лівий, верхній лівий). Якщо не вказано розміщення дочірнього елемента, то за замовчуванням він додається в лівий верхній кут екрану.

Ставлення до інших View-елементів. Ідентифікатор

Припустимо, що один TextView має бути вище іншого або одне зображення має бути ліворуч від іншого. Припустимо, що надписи потрібно розташувати в такому порядку



Робимо через *RelativeLayout*. *Ben* вирівнюється по верхньому краю по центру, *Lyla* - по нижньому лівому краю, *Jennie*- по нижньому правому краю. Вони називається **представленням прив'язки**.

Ставлення до інших View-елементів. Ідентифікатор

Як розташувати Kunal ліворуч від Ben? Існує атрибут **`android:layout_toLeftOf`**, **`android:layout_toRightOf`**. Як задати відносно якого об'єкта? Для цього View елементам задають ID ім'я (ідентифікатор). Наприклад: **`android:id="@+id/ben_text_view"`**



Символ **@(at)** відноситься до вказання на ресурс `android`. Далі вказуємо тип ресурсу. В даному випадку це **id**. Знак **+** вказуємо, що цей `id` оголошуємо в перше. Далі йде слеш та ім'я елемента, яке вибираємо для `ben`. Ім'я не повинно містити пробіли, повинно починатись з букви та не містити незрозумілі символи пунктуації. Можна цифри.

Ставлення до інших View-елементів. Ідентифікатор

Тоді, щоб розташувати Kunal ліворуч від Ben необхідно встановити такий атрибут **`android:layout_toLeftOf="@id/ben_text_view"`**.

Знак + у виразі `@id/ben_text_view` не вказуємо, тому `id` вже визначений. Розглянемо як Omoju розташувати вище Jennie.

Для цього Jennie потрібно задати ідентифікатор

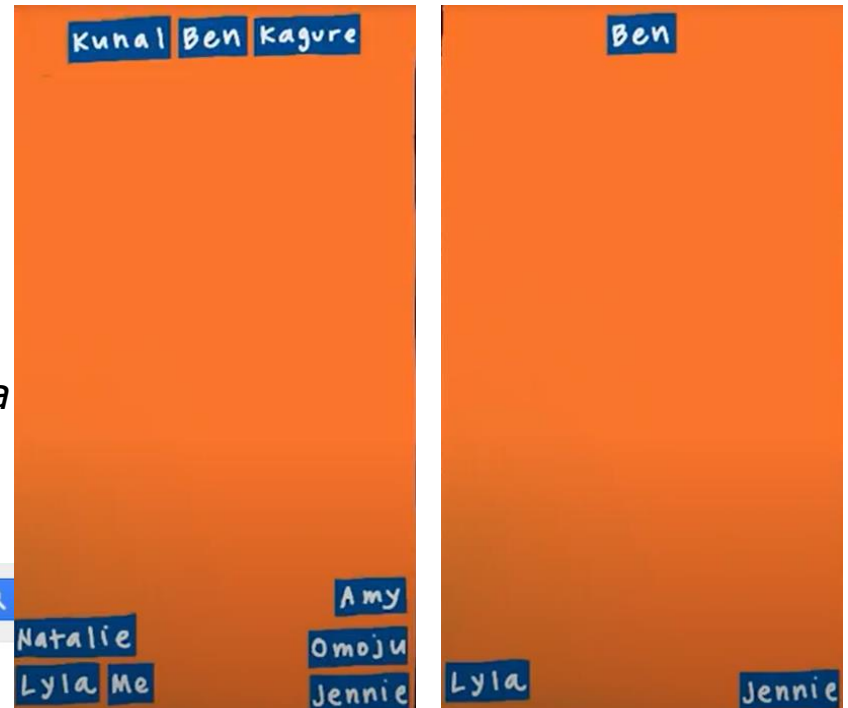
`android:id="@+id/jennie_text_view"`

Для Omoju вказуємо атрибут

`android:layout_above="@id/jennie_text_view"`

Атрибути розташування завжди можна знайти в документації

[RelativeLayout.LayoutParams](#)

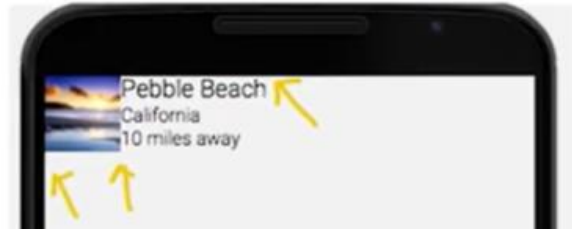


Наприклад, можна знайти атрибут розташувати нижче

`android:layout_below`

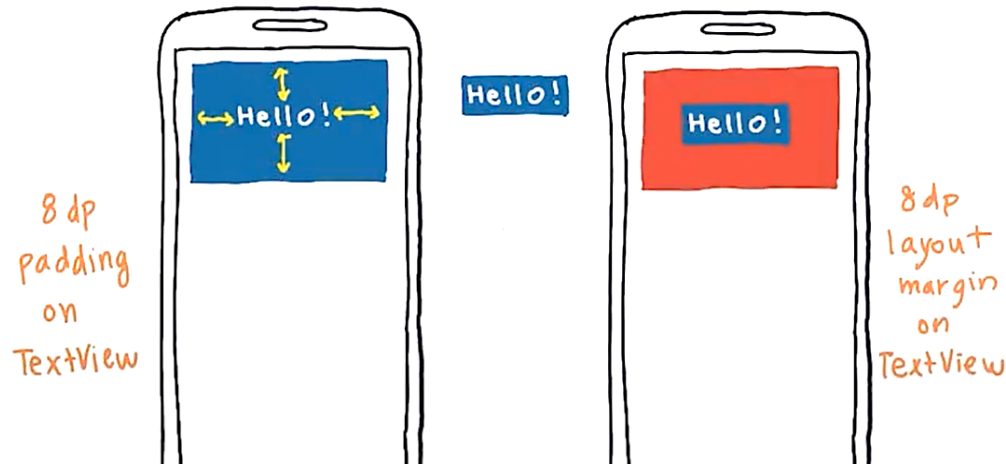
Внутрішній і зовнішній відступ

Для того щоб зображення виглядало краще, необхідно додати відступи



PADDING

MARGIN



Якщо додати **padding 8dp** до елемента `TextView`, то він збільшиться з усіх сторін на 8dp; якщо додати **margin**, то результат буде як на картинці. Розмір тексту скорочується на 8dp. Відступ створюється дочірнім елементом, а поля батьківським елементом.

Внутрішній і зовнішній відступ

Атрибути, що задають відступи

```
android:padding="8dp" або
    android:paddingLeft="8dp"
    android:paddingRight="8dp"
    android:paddingTop="8dp"
    android:paddingBottom="8dp"
```

Атрибути, що задають поля

```
android:layout_margin="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"
```

```
XML Layout Code
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:background="#9E9E9E"
6   android:orientation="vertical" >
7
8   <TextView
9     android:text="Hello!"
10    android:layout_width="match_parent"
11    android:layout_height="wrap_content"
12    android:layout_marginLeft="16dp"
13    android:layout_marginTop="16dp"
14    android:background="#009688"
15    android:textSize="45sp" />
16
17   <TextView
18     android:text="Hello!"
19     android:layout_width="match_parent"
20     android:layout_height="wrap_content"
21     android:layout_marginLeft="16dp"
22     android:layout_marginTop="16dp"
23     android:textSize="45sp" />
24
25   <TextView
26     android:text="Hello!"
27     android:layout_width="match_parent"
28     android:layout_height="wrap_content"
29     android:layout_marginLeft="16dp"
```



```
XML Layout Code
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:background="#9E9E9E"
6   android:orientation="vertical" >
7
8   <TextView
9     android:text="Hello!"
10    android:layout_width="match_parent"
11    android:layout_height="wrap_content"
12    android:paddingLeft="16dp"
13    android:paddingTop="16dp"
14    android:background="#009688"
15    android:textSize="45sp" />
16
17   <TextView
18     android:text="Hello!"
19     android:layout_width="match_parent"
20     android:layout_height="wrap_content"
21     android:paddingLeft="16dp"
22     android:paddingTop="16dp"
23     android:textSize="45sp" />
24
```



Margin поміняли на padding (8_Layout_pading)

Внутрішній і зовнішній відступ

Якщо додати *padding* в батьківському елементі, то появляться виступи з усіх сторін на 16dp

XML Layout Code

UNDO

REDO

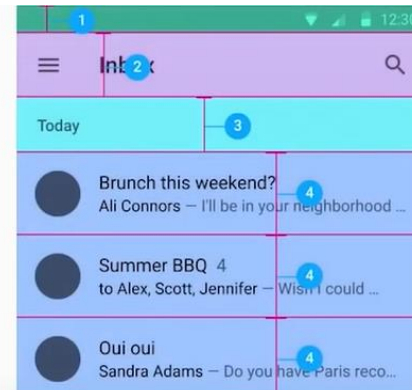
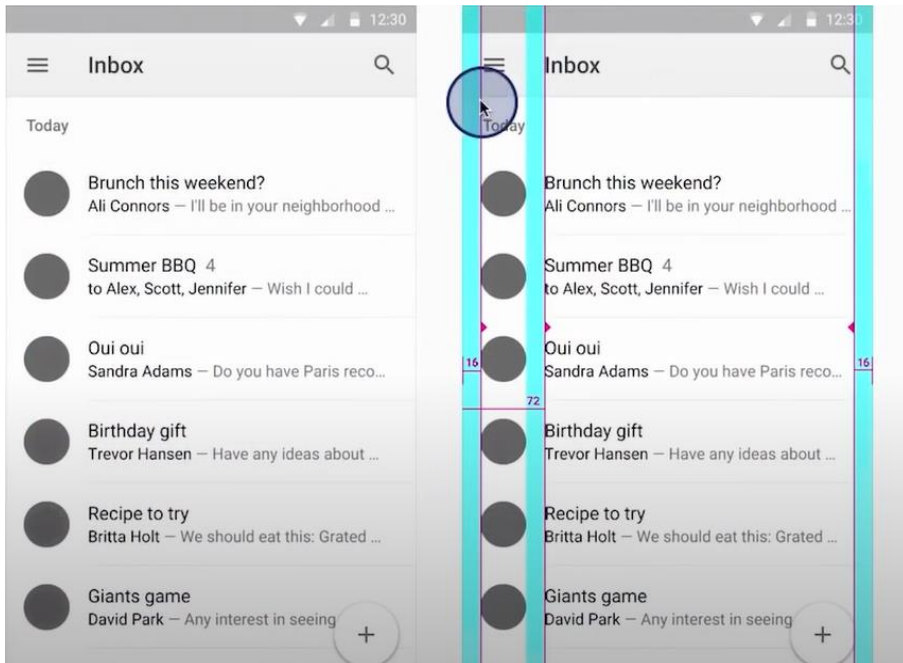
Code has been saved

RESET CODE

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:background="#9E9E9E"
6   android:orientation="vertical"
7   android:padding="16dp">
8
9   <TextView
10    android:text="Hello!"
11    android:layout_width="match_parent"
12    android:layout_height="wrap_content"
13    android:paddingLeft="16dp"
14    android:paddingTop="16dp"
15    android:background="#009688"
16    android:textSize="45sp" />
17  <TextView
18    android:text="Hello!"
19    android:layout_width="match_parent"
20    android:layout_height="wrap_content"
21    android:paddingLeft="16dp"
22    android:paddingTop="16dp"
23    android:background="#009688"
24    android:textSize="45sp" />
25  <TextView
26    android:text="Hello!"
27    android:layout_width="match_parent"
```



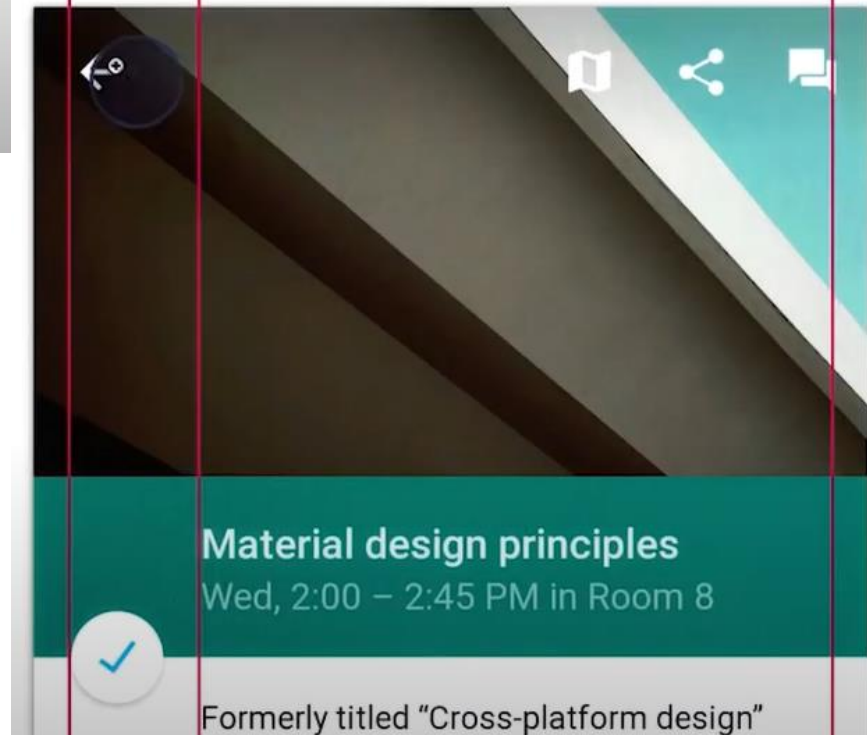
РЕКОМЕНДАЦІЇ MATERIAL DESIGN



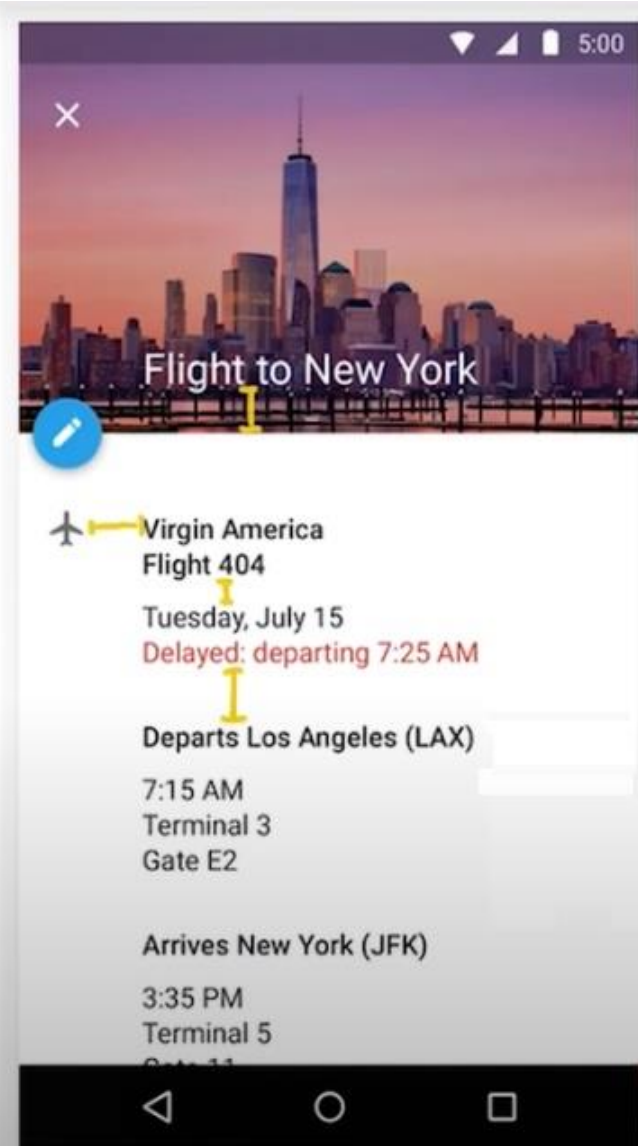
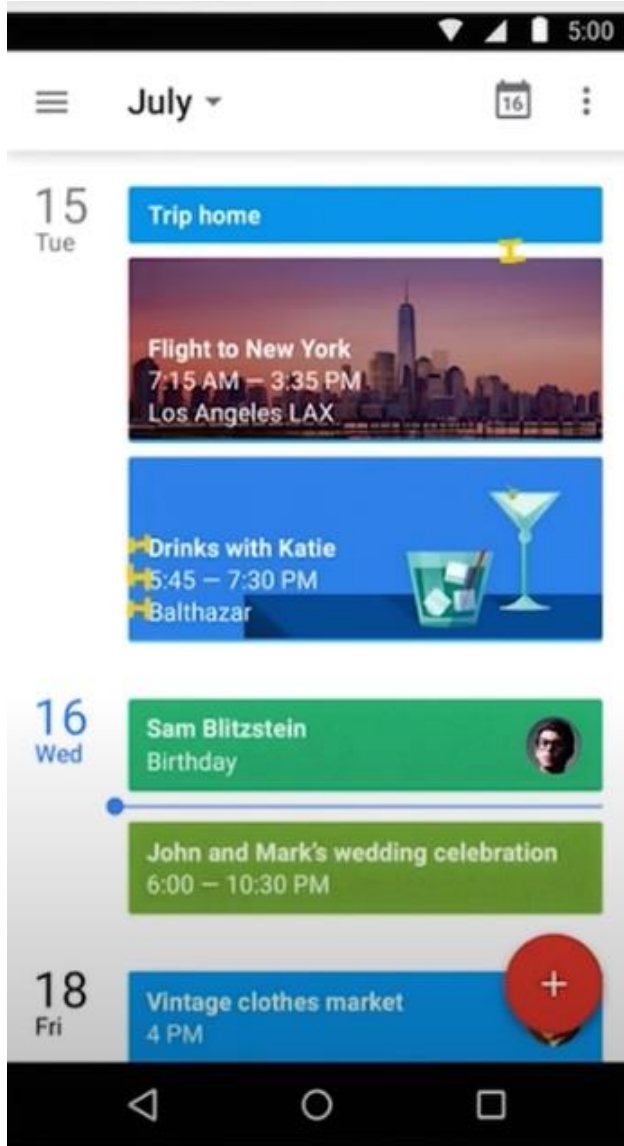
KEYLINE 1
16dp
(24dp tablet)

KEYLINE 2
72dp
(80dp tablet)

KEYLINE 1
16dp
(24dp tablet)



РЕКОМЕНДАЦІІ MATERIAL DESIGN



Корисні посилання

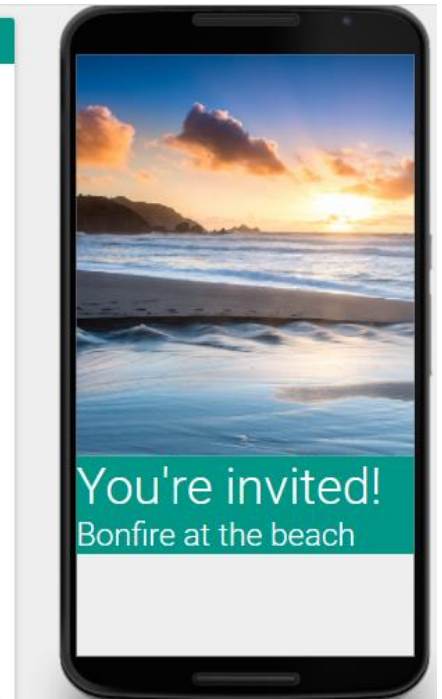
1. XML візуалізатор. URL: <https://labs.udacity.com/android-visualizer>
2. Шпаргалка з найбільш поширених View-елементів. URL: <https://cutt.ly/wEkm3KO>

Домашнє завдання

Змінити код, щоб отримати такий варіант розмітки (використовуємо Vizualizer).
<https://labs.udacity.com/android-visualizer/#/android/linear-layout-weight>



```
XML Layout Code
UNDO REDO Code has been saved RESET CODE
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent">
6
7   <ImageView
8     android:src="@drawable/ocean"
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content"
11    android:scaleType="centerCrop" />
12
13   <TextView
14     android:text="You're invited!"
15     android:layout_width="wrap_content"
16     android:layout_height="wrap_content"
17     android:textColor="@android:color/white"
18     android:textSize="54sp"
19     android:background="#009688" />
20
21   <TextView
22     android:text="Bonfire at the beach"
23     android:layout_width="wrap_content"
24     android:layout_height="wrap_content"
25     android:textColor="@android:color/white"
26     android:textSize="34sp"
27     android:background="#009688" />
28
29 </LinearLayout>
```



Домашнє завдання

Модифікувати XML код, що отримати таку розмітку



```
XML Layout Code
UNDO REDO Code has been saved RESET CODE

1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:paddingLeft="16dp"
6   android:paddingRight="16dp"
7   android:paddingTop="16dp"
8   android:paddingBottom="16dp"
9   android:orientation="vertical">
10
11   <ImageView
12     android:src="@drawable/ocean"
13     android:layout_width="match_parent"
14     android:layout_height="0dp"
15     android:layout_weight="1"
16     android:scaleType="centerCrop"
17     android:layout_marginBottom="8dp" />
18
19   <TextView
20     android:text="You're invited!"
21     android:layout_width="match_parent"
22     android:layout_height="wrap_content"
23     android:textColor="@android:color/white"
24     android:textSize="45sp"
25     android:background="#009688" />
26
27   <TextView
```

