

FrameLayout

Последнее обновление:

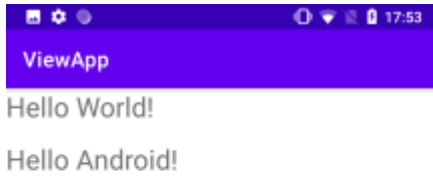
Контейнер `FrameLayout` предназначен для вывода на экран одного помещенного в него визуального элемента. Если же мы поместим несколько элементов, то они будут накладываться друг на друга. Тем не менее также можно располагать в `FrameLayout` несколько элементов.

Допустим, вложим в `FrameLayout` два элемента `TextView`:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Hello World!"
11        android:textSize="26sp"/>
12    <TextView
13        android:layout_width="wrap_content"
14        android:layout_height="wrap_content"
15        android:text="Hello Android!"
16        android:textSize="26sp"
```

```
17         android:layout_marginTop="50dp"/>
18
19     </FrameLayout>
```

Здесь оба элемента позиционируются в одно и то же место - в левый верхний угол контейнера `FrameLayout`, и чтобы избежать наложения, в данном случае у второго `TextView` устанавливается отступ сверху в 50 единиц.



Нередко `FrameLayout` применяется для создания производных контейнеров, например, `ScrollView`, который обеспечивает прокрутку.

Элементы управления, которые помещаются в `FrameLayout`, могут установить свое позиционирование с помощью атрибута **`android:layout_gravity`**:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <FrameLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent">
6
7      <TextView
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:text="Hello World!"
11         android:textSize="26sp"
12         android:layout_gravity="center_horizontal" />
13     <TextView
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
```

```

16         android:text="Welcome to Java World"
17         android:textSize="26sp"
18         android:layout_gravity="center"/>
19     <TextView
20         android:layout_width="wrap_content"
21         android:layout_height="wrap_content"
22         android:text="Hello Android!"
23         android:textSize="26sp"
24         android:layout_gravity="bottom|center_horizontal"/>
25
26 </FrameLayout>

```

При указании значения мы можем комбинировать ряд значений, разделяя их вертикальной чертой: `bottom|center_horizontal`



Welcome to Java World



Программное создание `FrameLayout` в коде `MainActivity`:

```

1  package com.example.viewapp;
2
3  import androidx.appcompat.app.AppCompatActivity;
4
5  import android.os.Bundle;
6  import android.view.Gravity;
7  import android.widget.FrameLayout;
8  import android.widget.TextView;
9
10 public class MainActivity extends AppCompatActivity {
11
12     @Override

```

```
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15
16         FrameLayout frameLayout = new FrameLayout(this);
17         TextView textView = new TextView(this);
18         textView.setText("Hello World!");
19
20         FrameLayout.LayoutParams layoutParams = new FrameLayout.LayoutParams
21             (FrameLayout.LayoutParams.WRAP_CONTENT, FrameLayout.LayoutParams.WRAP_CONTENT);
22         layoutParams.gravity = Gravity.CENTER_HORIZONTAL | Gravity.TOP;
23
24         textView.setLayoutParams(layoutParams);
25         textView.setTextSize(26);
26         frameLayout.addView(textView);
27         setContentView(frameLayout);
28     }
29 }
```