

Внутренние и внешние отступы

Последнее обновление:



Параметры разметки позволяют задать отступы как от внешних границ элемента до границ контейнера, так и внутри самого элемента между его границами и содержимым.

Padding

Для установки внутренних отступов применяется атрибут **android:padding**. Он устанавливает отступы контента от всех четырех сторон контейнера. Можно устанавливать отступы только от одной стороны контейнера, применяя следующие атрибуты: **android:paddingLeft**, **android:paddingRight**, **android:paddingTop** и **android:paddingBottom**.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:padding="50dp"
9     tools:context=".MainActivity">
10
11     <TextView
```

```

12     android:layout_height="wrap_content"
13     android:layout_width="wrap_content"
14     android:text="Hello World!"
15     android:textSize="30sp"
16     android:background="#e0e0e0"
17     app:layout_constraintLeft_toLeftOf="parent"
18     app:layout_constraintTop_toTopOf="parent"
19 />
20
21 </androidx.constraintlayout.widget.ConstraintLayout>

```

У контейнера `ConstraintLayout` установлен только один общий внутренний отступ в 50 единиц. Вложенный элемент `TextView` позиционируется в левом верхнем углу контейнера (благодаря атрибутам `app:layout_constraintLeft_toLeftOf="parent"` и `app:layout_constraintTop_toTopOf="parent"`). Поэтому `TextView` будет отодвигаться от начальной точки (левый верхний угол контейнера `ConstraintLayout`) вниз и влево на 50 единиц. Кроме того, такие же отступы будут действовать справа и снизу, если элемент будет примыкать к нижней или правой границе контейнера.



Установка одного отступа

```

1     android:padding="50dp"

```

Будет аналогична установке четырех отступов

```

1     android:paddingTop="50dp"
2     android:paddingLeft="50dp"
3     android:paddingBottom="50dp"
4     android:paddingRight="50dp"

```

Подобным образом можно установить отступы в других элементах. Например, установим внутри `TextView` сверху и снизу от внутреннего содержимого (то есть текста) отступы в 60 единиц и отступы слева и справа в 40 единиц:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:padding="50dp"
9      tools:context=".MainActivity">
10
11      <TextView
12          android:layout_height="wrap_content"
13          android:layout_width="wrap_content"
14          android:paddingTop="60dp"
15          android:paddingLeft="40dp"
16          android:paddingRight="40dp"
17          android:paddingBottom="60dp"
18          android:text="Hello World!"
19          android:textSize="30sp"
20          android:background="#e0e0e0"
21          app:layout_constraintLeft_toLeftOf="parent"
22          app:layout_constraintTop_toTopOf="parent"
23          />
24
25  </androidx.constraintlayout.widget.ConstraintLayout>
```



Стоит отметить, что вместо атрибутов `android:paddingLeft` и `android:paddingRight` можно применять атрибуты `android:paddingStart` и `android:paddingEnd`, которые разработаны специально адаптации приложения для работы как для языков с левосторонней ориентацией, так и правосторонней ориентацией (арабский, фарси).

Margin

Для установки внешних отступов используется атрибут `layout_margin`. Данный атрибут имеет модификации, которые позволяют задать отступ только от одной стороны: `android:layout_marginBottom`, `android:layout_marginTop`, `android:layout_marginLeft` и `android:layout_marginRight` (отступы соответственно от нижней, верхней, левой и правой границ):

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <TextView
11         android:layout_height="wrap_content"
12         android:layout_width="wrap_content"
13         android:layout_marginTop="50dp"
14         android:layout_marginLeft="60dp"
15         android:text="Hello World!"
16         android:textSize="30sp"
17         android:background="#e0e0e0"
18         app:layout_constraintLeft_toLeftOf="parent"
19         app:layout_constraintTop_toTopOf="parent"
20     />
21
22 </androidx.constraintlayout.widget.ConstraintLayout>
```

Здесь у `TextView` задаются отступы от двух сторон `ConstraintLayout` (слева 60 единиц и сверху 50 единиц):



Программная установка отступов

Для программной установки внутренних отступов у элементы вызывается метод **setPadding(left, top, right, bottom)**, в который передаются четыре значения для каждой из сторон. Также можно по отдельности задать отступы с помощью методов **getPaddingLeft()**, **getPaddingTop()**, **getPaddingRight()** и **getPaddingBottom()**.

Для установки внешних отступов необходимо реализовать объект **LayoutParams** для того контейнера, который применяется. И затем вызвать у этого объекта LayoutParams метод **setMargins(left, top, right, bottom)**:

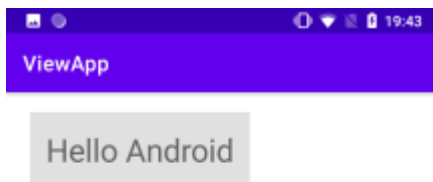
```
1 package com.example.viewapp;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.constraintlayout.widget.ConstraintLayout;
5
6 import android.os.Bundle;
7 import android.widget.TextView;
8
9 public class MainActivity extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14
15         ConstraintLayout constraintLayout = new ConstraintLayout(this);
16         TextView textView = new TextView(this);
17         // установка цвета текстового поля
```

```

18     textView.setBackgroundColor(0xFFE0E0E0);
19     // установка текста текстового поля
20     textView.setText("Hello Android");
21     // установка размера текста
22     textView.setTextSize(30);
23
24     ConstraintLayout.LayoutParams layoutParams = new ConstraintLayout.La
25         (ConstraintLayout.LayoutParams.WRAP_CONTENT , ConstraintLayo
26     // установка внешних отступов
27     layoutParams.setMargins(60, 50, 60, 50);
28     // позиционирование в левом верхнем углу контейнера
29     // эквивалент app:layout_constraintLeft_toLeftOf="parent"
30     layoutParams.leftToLeft = ConstraintLayout.LayoutParams.PARENT_ID;
31     // эквивалент app:layout_constraintTop_toTopOf="parent"
32     layoutParams.topToTop = ConstraintLayout.LayoutParams.PARENT_ID;
33     // устанавливаем размеры
34     textView.setLayoutParams(layoutParams);
35     // установка внутренних отступов
36     textView.setPadding(40, 40, 40, 40);
37     // добавляем TextView в ConstraintLayout
38     constraintLayout.addView(textView);
39
40     setContentView(constraintLayout);
41 }
42 }

```

Поскольку в данном случае элемент `TextView` добавляется в контейнер типа `ConstraintLayout`, то для его позиционирования применяется объект **`ConstraintLayout.LayoutParams`** (соответственно для `LinearLayout` это будет `LinearLayout.LayoutParams`), у которого вызывается метод **`setMargins()`**.



Но если посмотреть на последний скриншот, то можно увидеть, что, несмотря на то, что отступы вроде бы заданы также, что и в предпоследнем примере в файле `layout`, однако в реальности на экране мы увидим отступы со совсем другими значениями. Дело в том, что методы `setPadding()` и `setMargins()` принимают значения в пикселях, тогда как в файле `layout` применялись единицы `dp`. И чтобы использовать `dp` также в коде, необходимо выполнить преобразования:

```
1 package com.example.viewapp;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.constraintlayout.widget.ConstraintLayout;
5
6 import android.os.Bundle;
7 import android.util.TypedValue;
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         ConstraintLayout constraintLayout = new ConstraintLayout(this);
17         TextView textView = new TextView(this);
18         textView.setBackgroundColor(0xFFE0E0E0);
19         textView.setText("Hello Android!");
20         textView.setTextSize(30);
21     }
```

```

22 // получаем отступ в пикселях для 50 dp
23 int margin50inDp = (int) TypedValue.applyDimension(
24     TypedValue.COMPLEX_UNIT_DIP, 50, getResources().getDisplayMetrics()
25 // получаем отступ в пикселях для 60 dp
26 int margin60inDp = (int) TypedValue.applyDimension(
27     TypedValue.COMPLEX_UNIT_DIP, 60, getResources().getDisplayMetrics()
28 // получаем отступ в пикселях для 40 dp
29 int padding40inDp = (int) TypedValue.applyDimension(
30     TypedValue.COMPLEX_UNIT_DIP, 40, getResources().getDisplayMetrics()
31
32 ConstraintLayout.LayoutParams layoutParams = new ConstraintLayout.LayoutParams(
33     (ConstraintLayout.LayoutParams.WRAP_CONTENT, ConstraintLayout.LayoutParams.WRAP_CONTENT)
34 // установка внешних отступов
35 layoutParams.setMargins(margin60inDp, margin50inDp, margin60inDp, margin50inDp);
36 // выравнивание по левому краю ConstraintLayout
37 layoutParams.leftToLeft = ConstraintLayout.LayoutParams.PARENT_ID;
38 // выравнивание по верхней границе ConstraintLayout
39 layoutParams.topToTop = ConstraintLayout.LayoutParams.PARENT_ID;
40 // устанавливаем размеры
41 textView.setLayoutParams(layoutParams);
42 // установка внутренних отступов
43 textView.setPadding(padding40inDp, padding40inDp, padding40inDp, padding40inDp);
44 // добавляем TextView в ConstraintLayout
45 constraintLayout.addView(textView);
46
47 setContentView(constraintLayout);
48 }
49 }

```



Hello Android!

