

Цепочки элементов в ConstraintLayout

Последнее обновление:



ConstraintLayout позволяет организовать расположение элементов в ряд по горизонтали или по вертикали или то, что в Android называется **chains** или цепочки. Мы можем по цепочке установить позиционирование одного элемента относительно другого и таким образом организовать ряд элементов.

Горизонтальная цепочка элементов

Например, ряд элементов по горизонтали:

```
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:id="@+id/textView1"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:background="#efefef"
```

```
15         android:text="First"
16         android:textSize="30sp"
17         app:layout_constraintRight_toLeftOf="@id/textView2"
18         app:layout_constraintLeft_toLeftOf="parent"
19         app:layout_constraintTop_toTopOf="parent" />
20
21     <TextView
22         android:id="@+id/textView2"
23         android:layout_width="wrap_content"
24         android:layout_height="wrap_content"
25         android:background="#e0e0e0"
26         android:text="Second"
27         android:textSize="30sp"
28         app:layout_constraintLeft_toRightOf="@id/textView1"
29         app:layout_constraintRight_toLeftOf="@id/textView3"
30         app:layout_constraintTop_toTopOf="parent" />
31
32     <TextView
33         android:id="@+id/textView3"
34         android:layout_width="wrap_content"
35         android:layout_height="wrap_content"
36         android:background="#efefef"
37         android:text="Third"
38         android:textSize="30sp"
39         app:layout_constraintLeft_toRightOf="@id/textView2"
40         app:layout_constraintRight_toRightOf="parent"
41         app:layout_constraintTop_toTopOf="parent" />
42
43 </androidx.constraintlayout.widget.ConstraintLayout>
```

В итоге элементы цепочки равномерно будут растянуты по всей ширине контейнера:



Горизонтальная цепочка элементов достигается за счет двух факторов:

- Первый элемент выравнивается относительно левой границы контейнера (`app:layout_constraintLeft_toLeftOf="parent"`), последний элемент выравнивается относительно правой границы контейнера (`app:layout_constraintRight_toRightOf="parent"`).
- Благодаря установке атрибутов `app:layout_constraintLeft_toRightOf` и `app:layout_constraintRight_toLeftOf` располагаем один элемент справа или слева от другого.

Кроме того, `ConstraintLayout` позволяет настроить положение элементов внутри цепочки. Для этого применяется атрибут **`layout_constraintHorizontal_chainStyle`**, который может принимать следующие значения:

- **`spread`**: значение по умолчанию, при котором элементы цепочки равномерно растягиваются по всей длине цепочки, как в примере выше
- **`spread_inside`**: первый и последний элемент цепочки примыкают к границами контейнера
- **`packed`**: элементы цепочки располагаются вплотную друг к другу.

Например, применим значение **`spread_inside`**:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto">
```

```

4  xmlns:tools="http://schemas.android.com/tools"
5  android:layout_width="match_parent"
6  android:layout_height="match_parent"
7  tools:context=".MainActivity">
8
9  <TextView
10     android:id="@+id/textView1"
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:background="#efefef"
14     android:text="First"
15     android:textSize="30sp"
16
17     app:layout_constraintHorizontal_chainStyle="spread_inside"
18
19     app:layout_constraintRight_toLeftOf="@id/textView2"
20     app:layout_constraintLeft_toLeftOf="parent"
21     app:layout_constraintTop_toTopOf="parent" />
22
23  <TextView
24     android:id="@+id/textView2"
25     android:layout_width="wrap_content"
26     android:layout_height="wrap_content"
27     android:background="#e0e0e0"
28     android:text="Second"
29     android:textSize="30sp"
30     app:layout_constraintLeft_toRightOf="@id/textView1"
31     app:layout_constraintRight_toLeftOf="@id/textView3"
32     app:layout_constraintTop_toTopOf="parent" />
33
34  <TextView
35     android:id="@+id/textView3"
36     android:layout_width="wrap_content"
37     android:layout_height="wrap_content"
38     android:background="#efefef"
39     android:text="Third"
40     android:textSize="30sp"
41     app:layout_constraintLeft_toRightOf="@id/textView2"
42     app:layout_constraintRight_toRightOf="parent"
43     app:layout_constraintTop_toTopOf="parent" />
44
45  </androidx.constraintlayout.widget.ConstraintLayout>

```

Причем в данном случае достаточно установить атрибут у первого элемента цепочки:





Значение **packed**:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <TextView
10         android:id="@+id/textView1"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:background="#efefef"
14         android:text="First"
15         android:textSize="30sp"
16
17         app:layout_constraintHorizontal_chainStyle="packed"
18
19         app:layout_constraintRight_toLeftOf="@id/textView2"
20         app:layout_constraintLeft_toLeftOf="parent"
21         app:layout_constraintTop_toTopOf="parent" />
22
23     <TextView
24         android:id="@+id/textView2"
25         android:layout_width="wrap_content"
26         android:layout_height="wrap_content"
27         android:background="#e0e0e0"
```

```

28         android:text="Second"
29         android:textSize="30sp"
30         app:layout_constraintLeft_toRightOf="@id/textView1"
31         app:layout_constraintRight_toLeftOf="@id/textView3"
32         app:layout_constraintTop_toTopOf="parent" />
33
34     <TextView
35         android:id="@+id/textView3"
36         android:layout_width="wrap_content"
37         android:layout_height="wrap_content"
38         android:background="#efefef"
39         android:text="Third"
40         android:textSize="30sp"
41         app:layout_constraintLeft_toRightOf="@id/textView2"
42         app:layout_constraintRight_toRightOf="parent"
43         app:layout_constraintTop_toTopOf="parent" />
44
45 </androidx.constraintlayout.widget.ConstraintLayout>

```



Вес элемента

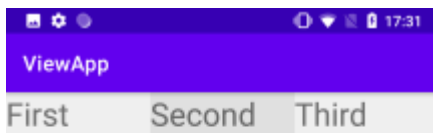
Стоит отметить, что выше у элементов устанавливалась ширина, необходимая для их содержимого. Но мы могли бы установить и нулевую ширину, тогда элементы равномерно бы распределялись по всей цепочки без образования промежутков между ними.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto"

```

```
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context=".MainActivity">
8
9   <TextView
10       android:id="@+id/textView1"
11       android:layout_width="0dp"
12       android:layout_height="wrap_content"
13       android:background="#efefef"
14       android:text="First"
15       android:textSize="30sp"
16       app:layout_constraintRight_toLeftOf="@id/textView2"
17       app:layout_constraintLeft_toLeftOf="parent"
18       app:layout_constraintTop_toTopOf="parent" />
19
20   <TextView
21       android:id="@+id/textView2"
22       android:layout_width="0dp"
23       android:layout_height="wrap_content"
24       android:background="#e0e0e0"
25       android:text="Second"
26       android:textSize="30sp"
27       app:layout_constraintLeft_toRightOf="@id/textView1"
28       app:layout_constraintRight_toLeftOf="@id/textView3"
29       app:layout_constraintTop_toTopOf="parent" />
30
31   <TextView
32       android:id="@+id/textView3"
33       android:layout_width="0dp"
34       android:layout_height="wrap_content"
35       android:background="#efefef"
36       android:text="Third"
37       android:textSize="30sp"
38       app:layout_constraintLeft_toRightOf="@id/textView2"
39       app:layout_constraintRight_toRightOf="parent"
40       app:layout_constraintTop_toTopOf="parent" />
41
42 </androidx.constraintlayout.widget.ConstraintLayout>
```



В этом случае значение атрибута `app:layout_constraintHorizontal_chainStyle` не играет никакой роли, так как все элементы итак растягиваются по всей цепочке.

Однако такое поведение может не устраивать, например, мы хотим, чтобы один элемент был два раза больше другого. И в этом случае мы можем с помощью атрибута **`layout_constraintHorizontal_weight`**. Однако следует учитывать, что при применении весов у элементов, они должны иметь нулевую ширину:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <TextView
10         android:id="@+id/textView1"
11         android:layout_width="0dp"
12         android:layout_height="wrap_content"
13         android:background="#efefef"
14         android:text="First"
15         android:textSize="30sp"
16
17         app:layout_constraintHorizontal_weight="1"
18
19         app:layout_constraintRight_toLeftOf="@id/textView2"
20         app:layout_constraintLeft_toLeftOf="parent"
21         app:layout_constraintTop_toTopOf="parent" />
```

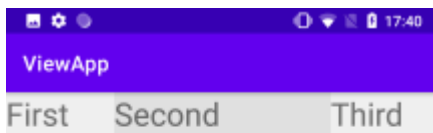


```

22
23 <TextView
24     android:id="@+id/textView2"
25     android:layout_width="0dp"
26     android:layout_height="wrap_content"
27     android:background="#e0e0e0"
28     android:text="Second"
29     android:textSize="30sp"
30
31     app:layout_constraintHorizontal_weight="2"
32
33     app:layout_constraintLeft_toRightOf="@id/textView1"
34     app:layout_constraintRight_toLeftOf="@id/textView3"
35     app:layout_constraintTop_toTopOf="parent" />
36
37 <TextView
38     android:id="@+id/textView3"
39     android:layout_width="0dp"
40     android:layout_height="wrap_content"
41     android:background="#efefef"
42     android:text="Third"
43     android:textSize="30sp"
44
45     app:layout_constraintHorizontal_weight="1"
46
47     app:layout_constraintLeft_toRightOf="@id/textView2"
48     app:layout_constraintRight_toRightOf="parent"
49     app:layout_constraintTop_toTopOf="parent" />
50
51 </androidx.constraintlayout.widget.ConstraintLayout>

```

В качестве значения атрибут `layout_constraintHorizontal_weight` принимает число - вес элемента. Так, в данном случае вес первого элемента - 1, вес второго - 2, а вес третьего - 1. Поэтому вся ширина контейнера будет условно поделена на $1 + 2 + 1 = 4$ частей, из которых по одной части займут первый и третий элемент, а второй займет 2 части, то есть второй элемент будет в два раза больше первого и третьего элемента.



В принципе мы можем оставить элементы и с шириной `"wrap_content"` или конкретным значением, отличным от `"0dp"`, просто в этом случае они не будут участвовать в распределении пространства контейнера и вес у такого элемента роли играть не будет.

Вертикальная цепочка

Для образования вертикальной цепочки также должно соблюдаться два условия:

- Первый элемент выравнивается относительно верхней границы контейнера (`app:layout_constraintTop_toTopOf="parent"`), последний элемент выравнивается относительно нижней границы контейнера (`app:layout_constraintBottom_toBottomOf="parent"`).
- Благодаря установке атрибутов `app:layout_constraintBottom_toTopOf` и `app:layout_constraintBottom_toBottomOf` располагаем один элемент поверх другого.

Чтобы настроить положение элементов внутри цепочки, применяется атрибут **`layout_constraintVertical_chainStyle`**, который может принимать следующие значения:

- **`spread`**: значение по умолчанию, при котором элементы цепочки равномерно растягиваются по всей длине цепочки
- **`spread_inside`**: первый и последний элемент цепочки примыкают к границами контейнера
- **`packed`**: элементы цепочки прилегают вплотную друг к другу.

Например, вертикальная цепочка со значением по умолчанию - **spread**:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <TextView
10         android:id="@+id/textView1"
11         android:layout_width="200dp"
12         android:layout_height="wrap_content"
13         android:background="#efefef"
14         android:text="First"
15         android:textSize="30sp"
16         app:layout_constraintBottom_toTopOf="@id/textView2"
17         app:layout_constraintLeft_toLeftOf="parent"
18         app:layout_constraintRight_toRightOf="parent"
19         app:layout_constraintTop_toTopOf="parent" />
20
21     <TextView
22         android:id="@+id/textView2"
23         android:layout_width="200dp"
24         android:layout_height="wrap_content"
25         android:background="#e0e0e0"
26         android:text="Second"
27         android:textSize="30sp"
28         app:layout_constraintTop_toBottomOf="@id/textView1"
29         app:layout_constraintBottom_toTopOf="@id/textView3"
30         app:layout_constraintLeft_toLeftOf="parent"
31         app:layout_constraintRight_toRightOf="parent" />
32
33     <TextView
34         android:id="@+id/textView3"
35         android:layout_width="200dp"
36         android:layout_height="wrap_content"
37         android:background="#efefef"
38         android:text="Third"
39         android:textSize="30sp"
40
41         app:layout_constraintTop_toBottomOf="@id/textView2"
42         app:layout_constraintLeft_toLeftOf="parent"
43         app:layout_constraintRight_toRightOf="parent"
44         app:layout_constraintBottom_toBottomOf="parent" />
45
46 </androidx.constraintlayout.widget.ConstraintLayout>
```



Также достаточно применить к первому элементу цепочки атрибут `layout_constraintVertical_chainStyle`, чтобы изменить положение элементов:

```
1 <TextView
2     android:id="@+id/textView1"
3     android:layout_width="200dp"
4     android:layout_height="wrap_content"
5     android:background="#efefef"
6     android:text="First"
7     android:textSize="30sp"
8
9     app:layout_constraintVertical_chainStyle="spread_inside"
10
11     app:layout_constraintBottom_toTopOf="@id/textView2"
12     app:layout_constraintLeft_toLeftOf="parent"
13     app:layout_constraintRight_toRightOf="parent"
14     app:layout_constraintTop_toTopOf="parent" />
```

spread_inside

packed



Second

First
Second
Third

Third

И как при горизонтальной ориентации в вертикальной цепочки можно использовать вес элементов с помощью атрибута **layout_constraintVertical_weight**. Для установки веса у элемента в качестве высоты должно быть установлено значение **0dp**

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <TextView
10         android:id="@+id/textView1"
11         android:layout_width="200dp"
12         android:layout_height="0dp"
13         android:background="#efefef"
14         android:text="First"
15         android:textSize="30sp"
16         app:layout_constraintVertical_weight="1"
17         app:layout_constraintBottom_toTopOf="@id/textView2"
18         app:layout_constraintLeft_toLeftOf="parent"
19         app:layout_constraintRight_toRightOf="parent"
20         app:layout_constraintTop_toTopOf="parent" />
21
22     <TextView
23         android:id="@+id/textView2"
```

```
24     android:layout_width="200dp"
25     android:layout_height="0dp"
26     android:background="#e0e0e0"
27     android:text="Second"
28     android:textSize="30sp"
29     app:layout_constraintVertical_weight="3"
30     app:layout_constraintTop_toBottomOf="@id/textView1"
31     app:layout_constraintBottom_toTopOf="@id/textView3"
32     app:layout_constraintLeft_toLeftOf="parent"
33     app:layout_constraintRight_toRightOf="parent" />
34
35     <TextView
36         android:id="@+id/textView3"
37         android:layout_width="200dp"
38         android:layout_height="0dp"
39         android:background="#efefef"
40         android:text="Third"
41         android:textSize="30sp"
42         app:layout_constraintVertical_weight="2"
43         app:layout_constraintTop_toBottomOf="@id/textView2"
44         app:layout_constraintLeft_toLeftOf="parent"
45         app:layout_constraintRight_toRightOf="parent"
46         app:layout_constraintBottom_toBottomOf="parent" />
47
48 </androidx.constraintlayout.widget.ConstraintLayout>
```

Совокупный вес элементов в данном случае $1 + 3 + 2 = 6$. Поэтому вся высота контейнера будет делиться на 6 частей, из которых первый элемент займет 1 часть, второй - 3 части и третий - 2 части в соответствии со своим весом.

