**Margin vs. Padding Attributes**

There is always a confusion between the **padding** and **margin** attribute when we just start designing User interfaces. Be it web development or Android development, margin and padding is standard parameters to position and style User interface elements.

Both provides extra space/gap inside or outside the container. Then what's the exact difference?

Let's get it cleared. In simple words, margin means to push outside, whereas padding means to push inside.

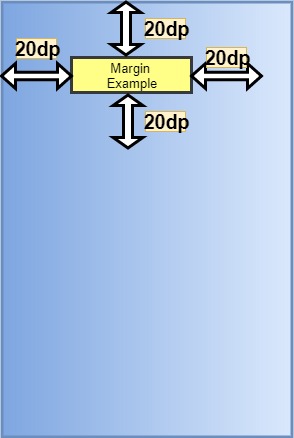
Now, let’s understand what happens when we use these attributes for a **View** in Android.

### Margin and Padding with a View

Let's see how we can use these attributes with a View in android and how it affects the View's position and styling.

#### **Margin → android:layout\_margin**

When we say push outside, the view i.e. the rectangle pushes its surrounding contents from itself by the dimension specified in the margin attribute. Here, the surrounding contents can be other Views. The following diagram will make it more clear.



Here is how we specify this is our layout XML:

android:layout\_margin="20dp"

Hence, the other views will be separated from this view by at least 20dp.

We can also different margin values for all the sides, like:

android:layout\_marginRight="20dp"

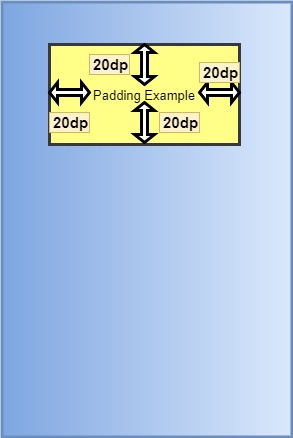
android:layout\_marginLeft="20dp"

android:layout\_marginTop="20dp"

android:layout\_marginBottom="20dp"

#### **Padding → android:padding**

When we say push inside, the view i.e. the rectangle pushes its contents from itself by the dimension specified in the padding attribute towards its center. Padding can be considered as margin but inside the View. The following diagram will make it more clear.



Here is how we specify this is our layout XML:

android:padding="20dp"

Hence, its content (here text) will be pushed inside of the rectangle by 20dp.

We can also different padding values for all the sides, like:

android:paddingRight="20dp"

android:paddingLeft="20dp"

android:paddingTop="20dp"

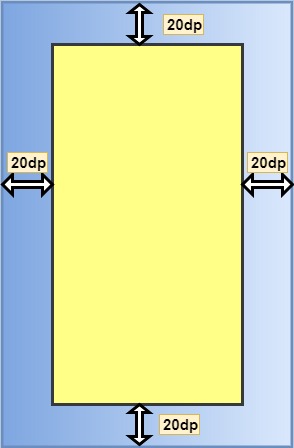
android:paddingBottom="20dp"

### Margin and Padding with Layout

Now, lets understand what happens when we write these attributes for a Layout.

#### **Margin**

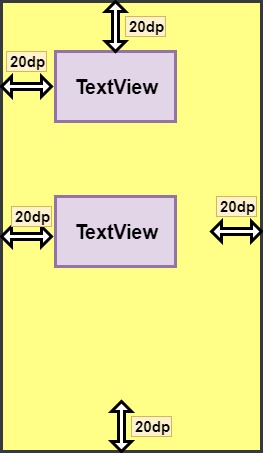
When we say push outside, the root layout i.e. the rectangle pushes its surrounding contents from itself by the dimension specified in the margin attribute. Here, the surrounding content will be the screen of the mobile. Hence, the layout (rectangle) pushes itself from the screen of the mobile. The following diagram will make it more clear.



The syntax to apply margin to a layout is similar to that for a view.

#### **Padding**

When we say push inside, the root layout i.e the rectangle pushes its contents from itself by the dimension specified in the padding attribute. Here, its content will be the different views that it holds like TextView, ImageView etc. The following diagram will make it more clear.



**Note**: In Android, **padding** attribute applied to the parent (root layout) looks the same as **margin** attribute applied to the child (views inside the layout).