View Sizes

DP, SP, Wrap_content, Match_parent

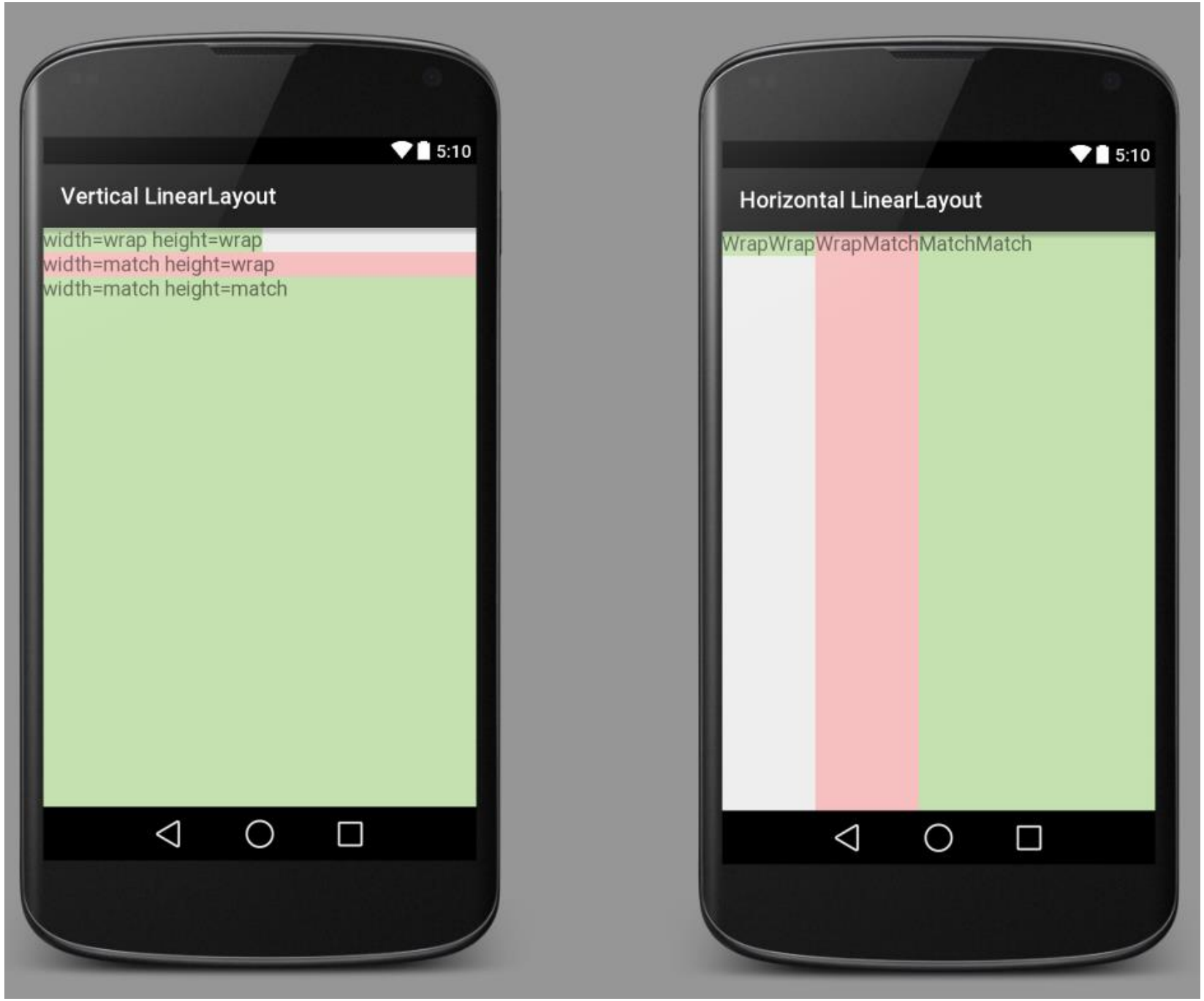
- A widget in Android

View

- TextView = JLabel
- EditText = JTextField
- Button = JButton
- ImageView = JLabel

Widths:

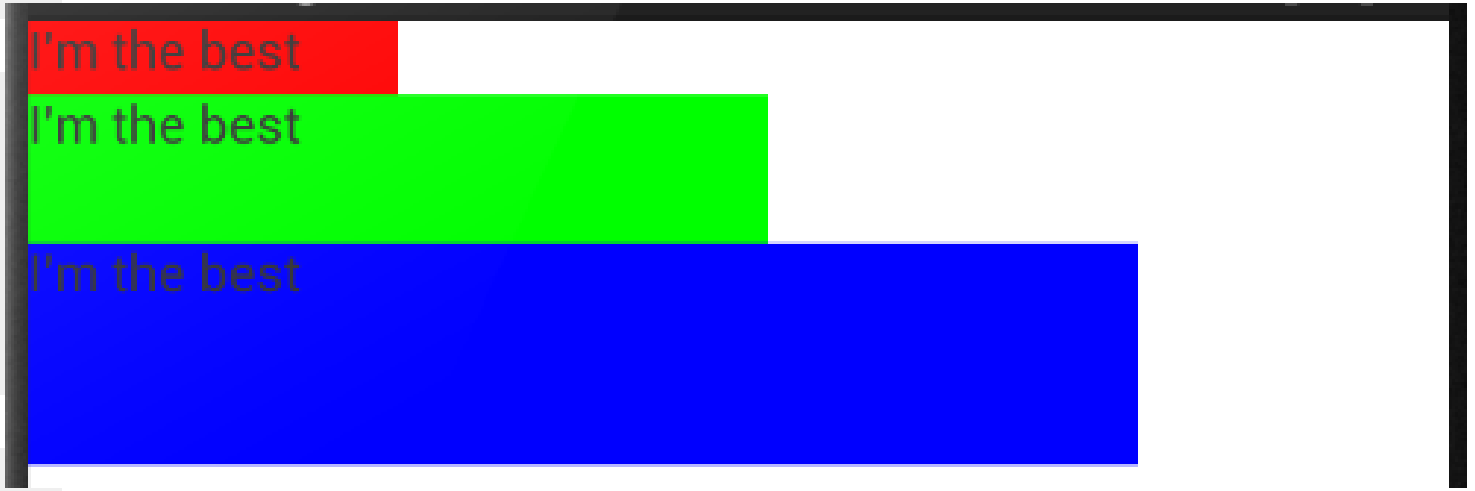
- DP
- Wrap Content
- Match Parent

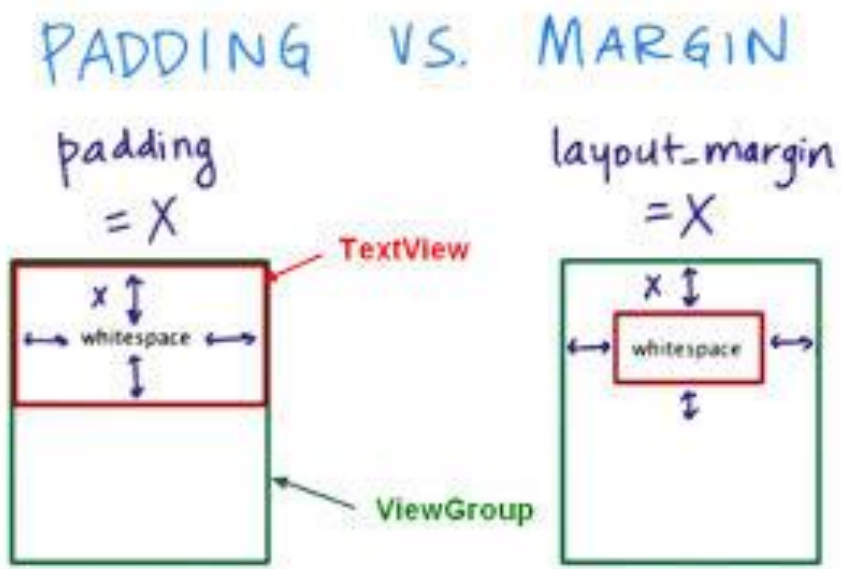


Layout_Width
Layout_Height
Values

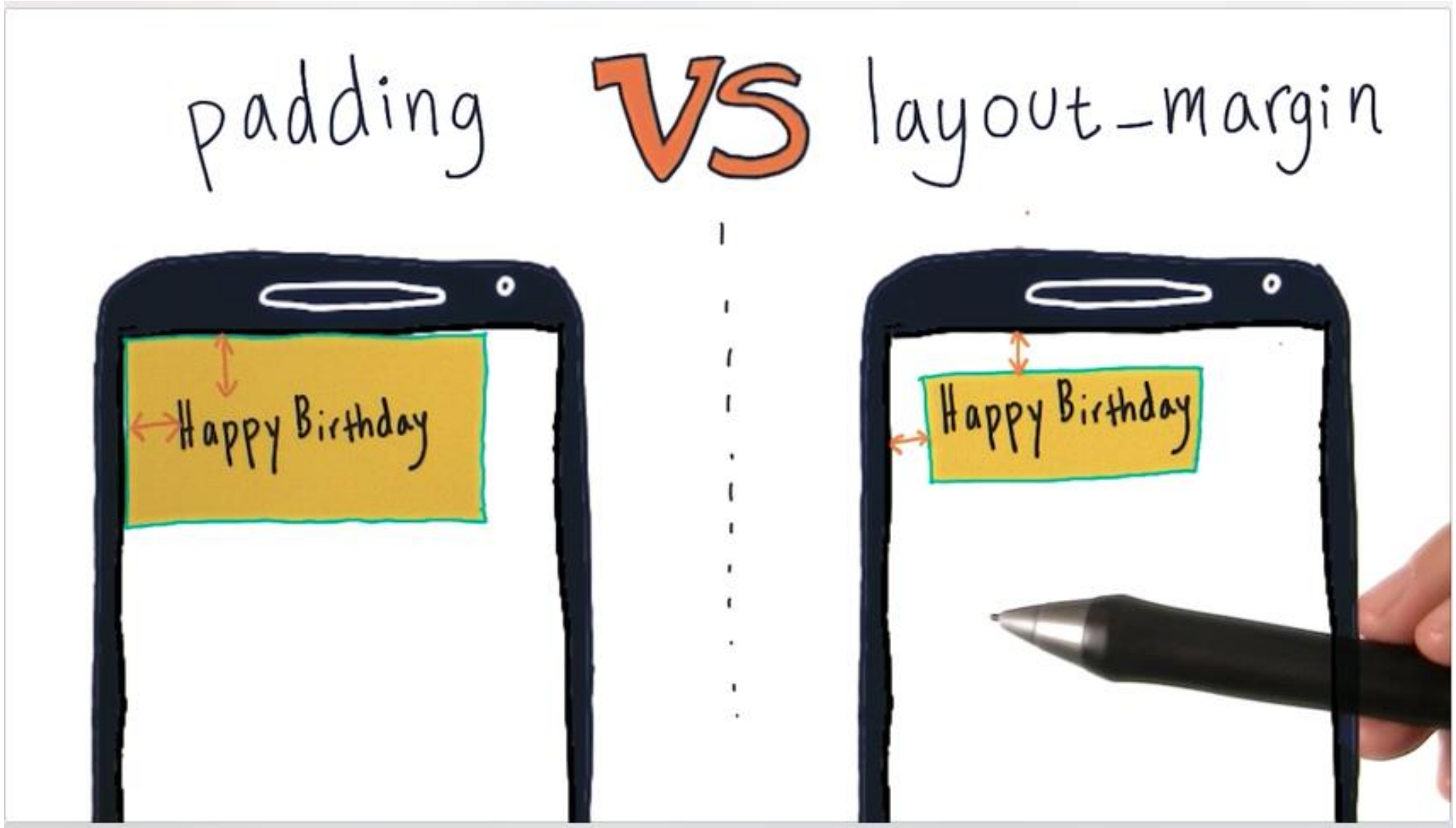
- Match_parent: will fill the parent (or phone) holding it.
- Wrap_content: resize self to hold content

```
<TextView
    android:id="@+id/text1"
    android:layout_width="100dp"
    android:layout_height="20dp"
    android:text="@string/my_best_text"
    android:background="#FF0000"
    />
<TextView
    android:id="@+id/text2"
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:text="@string/my_best_text"
    android:background="#00FF00"
    />
<TextView
    android:id="@+id/text3"
    android:layout_width="300dp"
    android:layout_height="60dp"
    android:text="@string/my_best_text"
    android:background="#0000FF"
    />
```

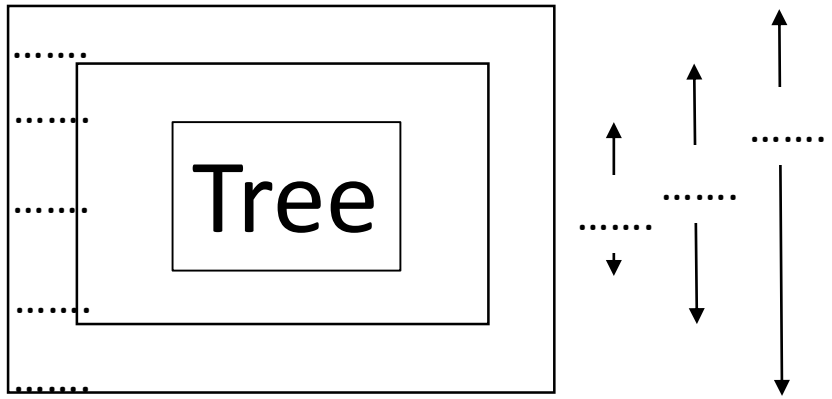



PADDING VS. MARGIN
padding
=X
layout_margin
=X
TextView
x
whitespace
ViewGroup
x
whitespace

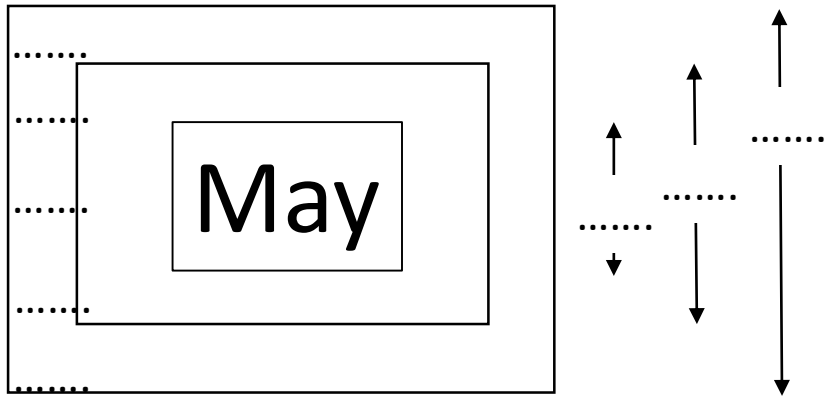

Hello World
Content
Padding
Border
Margin


padding
VS
layout_margin
Happy Birthday
Happy Birthday

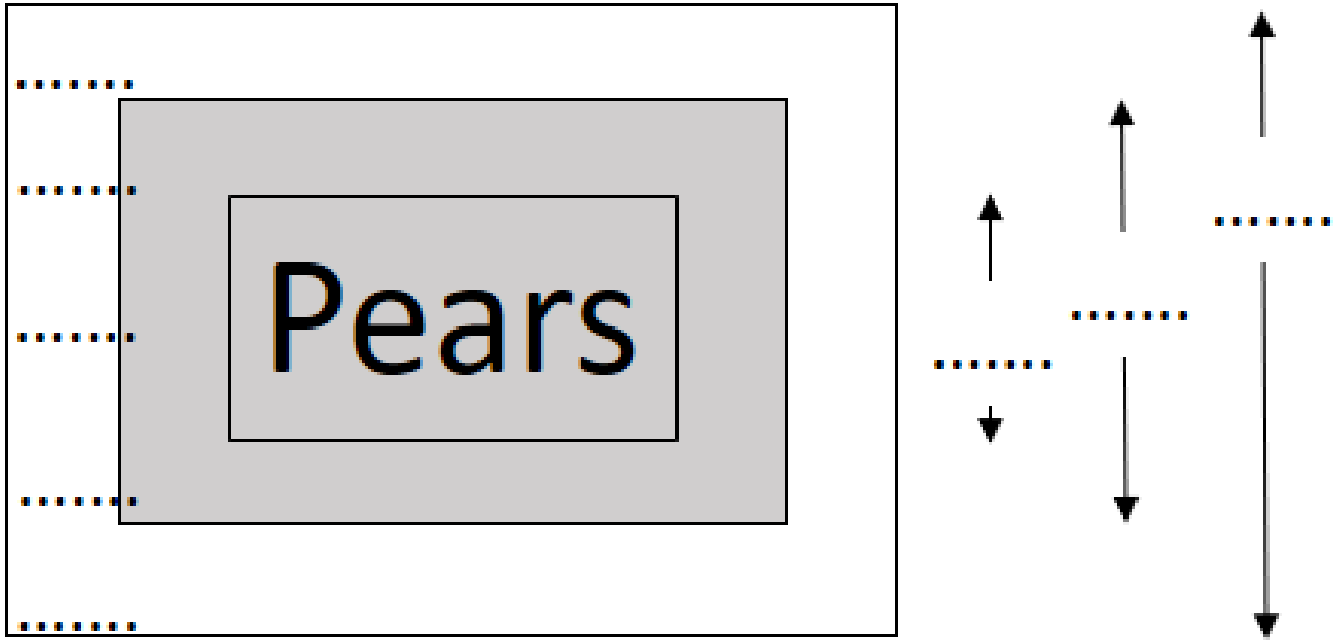
```
<TextView
    android:text="Tree"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="30sp"
    android:padding="20dp"
    android:layout_margin="10dp"/>
```



```
<TextView
    android:text="May"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="50sp"
    android:padding="30dp"
    android:layout_margin="20dp"/>
```

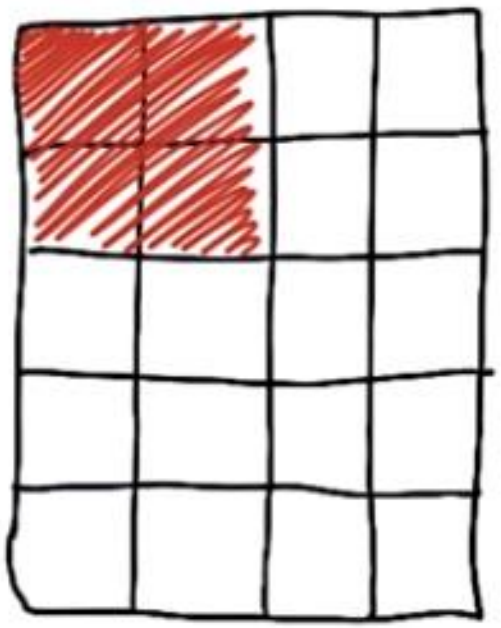


```
<TextView
    android:text="Pears"
    android:layout_width="wrap_content"
    android:layout_height="200dp"
    android:textSize="20sp"
    android:padding="80dp"/>
```

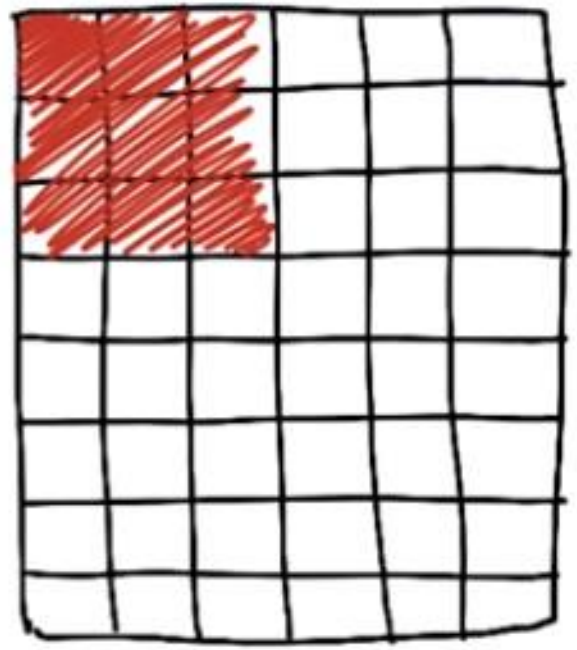


DENSITY-INDEPENDENT PIXELS

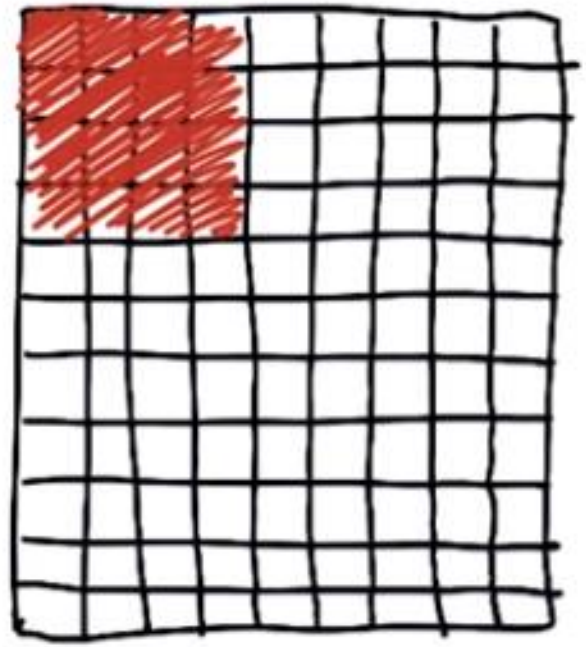
2 dp by 2 dp



Medium Resolution Device



High Resolution Device



Extra-High Resolution Device

dp (Density-Independent Pixel)


```
<TextView
    android:layout_width="160dp"
    android:layout_height="80dp"
    android:background="#00FF00"
    android:text="Hello"/>
```

DESCRIPTION | SAMPLE CODE

The screen of an **Android device** is made of rows and columns of glowing dots called **pixels**. Devices can range in **screen density**, which means how many pixels per inch (or dots per inch) are on the screen. For example, an mdpi (or medium density device) has 160 dots per inch, while an xxhdpi (extra extra high density device) has 480 dots per inch.

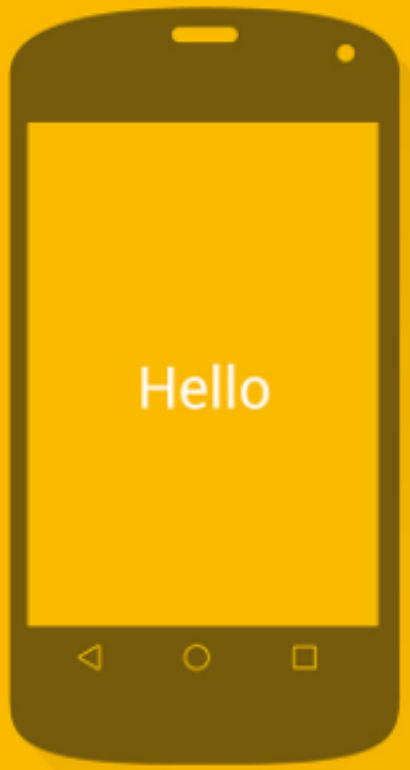
If we specify the size of views in pixel values, then views would appear very small on the higher density devices, where there are many pixels packed into a small area. If a button is too small, then it would be hard for the user to touch it.

To achieve a consistent physical size of Views, across devices of different screen densities, we use a unit of measure called a **density-independent pixel** (**dp** or **dip**, pronounced "dee pee" **or** "dip"). 1 dp is equal to 1 pixel on an mdpi device. 1 dp is equal to 3 pixels on an xxhdpi device, and so on for other devices. Per Material design guidelines, any touch target on the screen should be at least 48dp wide by 48dp tall. That way, a button in an app on one device will be approximately the same physical size as that button in the same app running on a device with a different screen density.

DP

- Density Independent Pixels
- A size unit in Android
- Resizes to handle different phones
- Means the buttons don’t shrink on a better phone resolution.

sp (Scale-Independent Pixel)



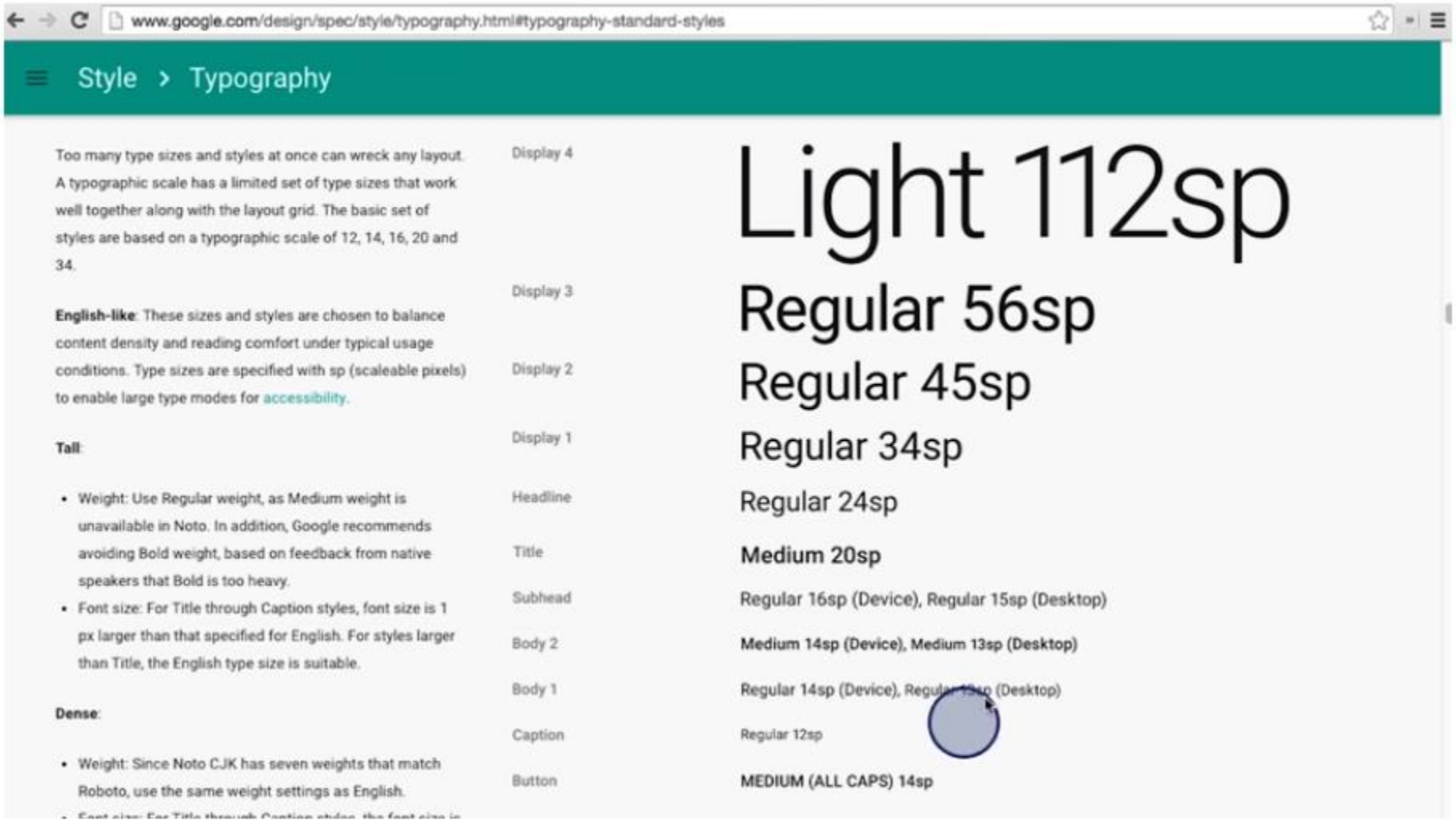
user selects **medium** font setting

DESCRIPTION | SAMPLE CODE

A **scale-independent pixel** (sp) is a unit of length for specifying the size of a font of type. Its length depends on the user's preference for font size, set in the Settings app of the Android device.

To respect the user's preferences, you should specify all font sizes in scale-independent pixels. All other measurements should be given in **device-independent pixels** (dp's).

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="8dp"
    android:textSize="20sp"
    android:text="Hello"/>
```

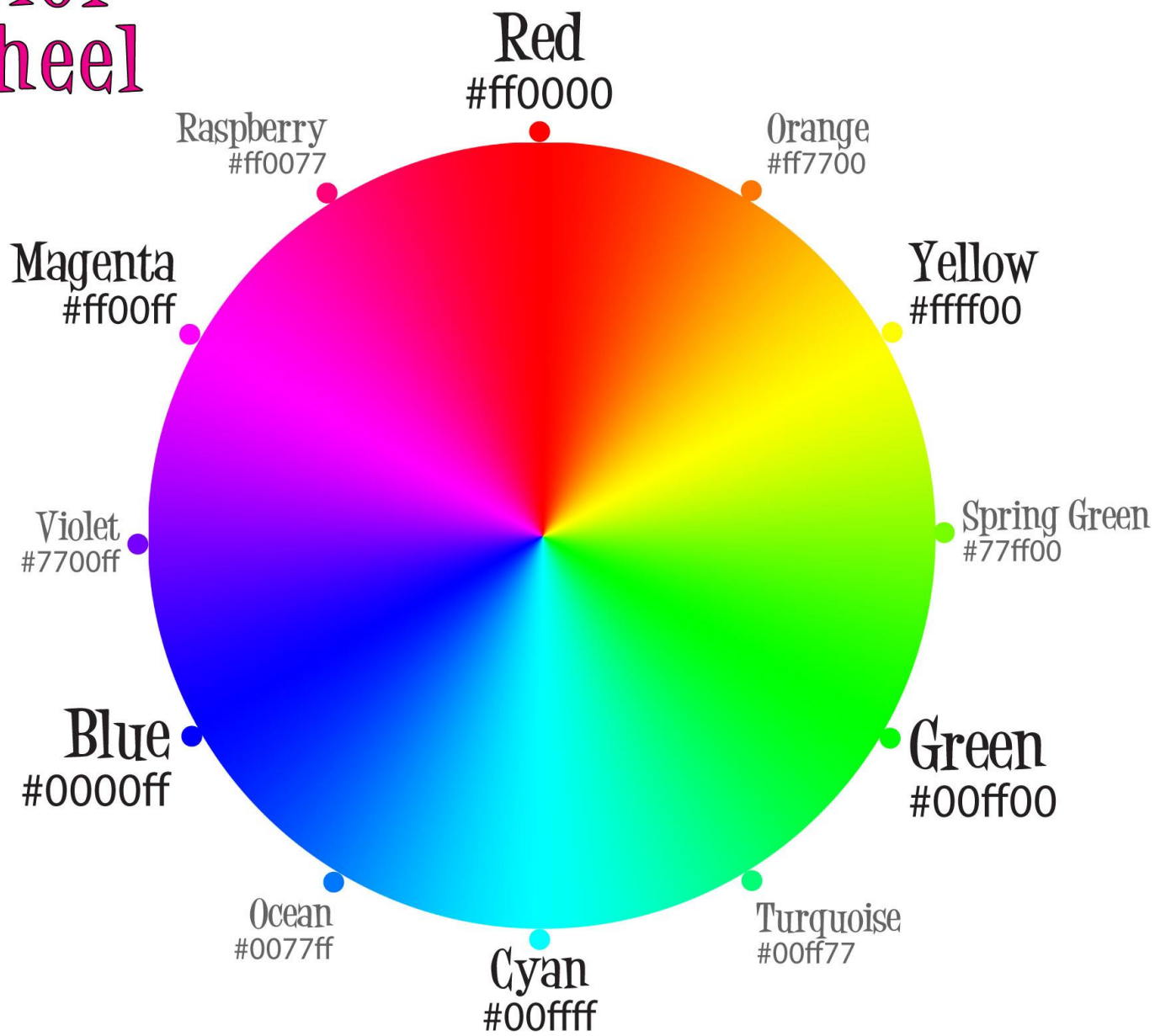



```
android:textAppearance="?android:textAppearanceSmall" />
android:textAppearance="?android:textAppearanceLarge"
```

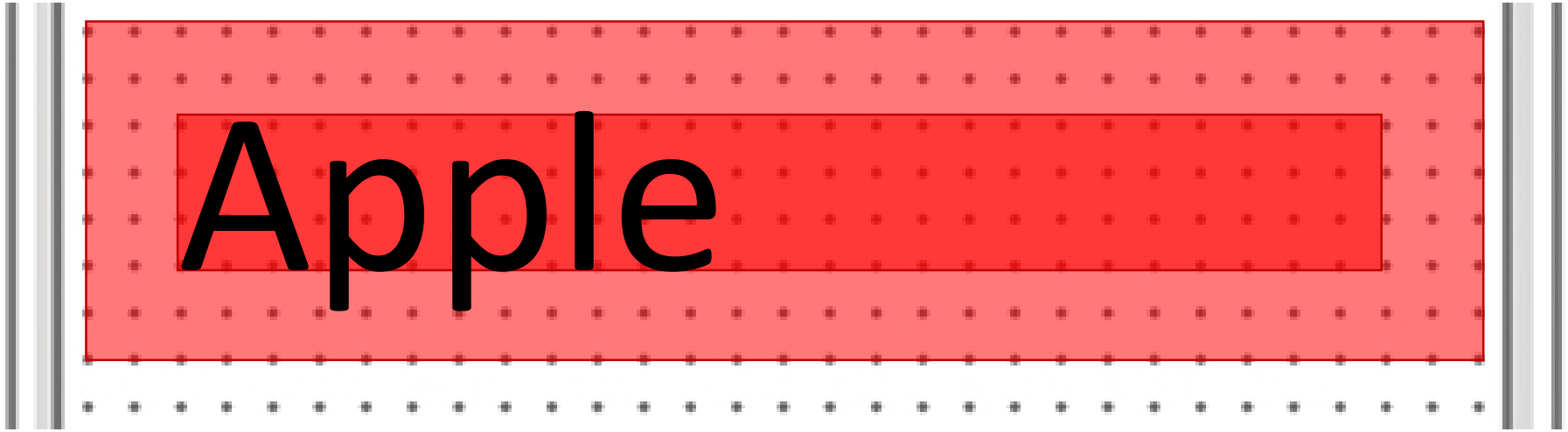
SP

- Scale Independent Pixels
- Works exactly as dp does, and for the same reason.
- It is used for font.

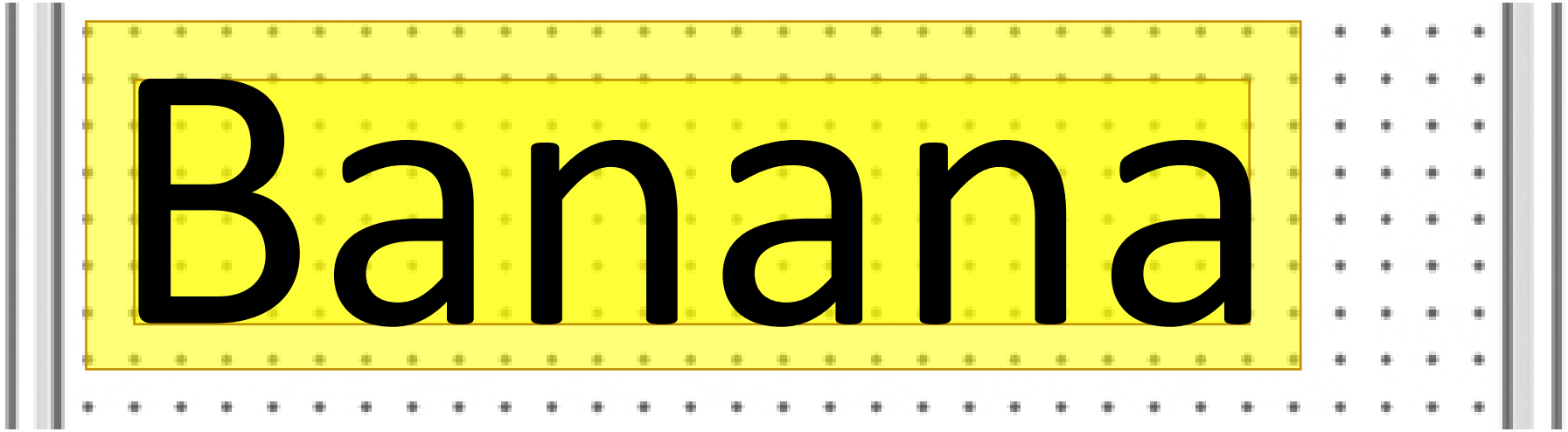
Hexadecimal Color Wheel



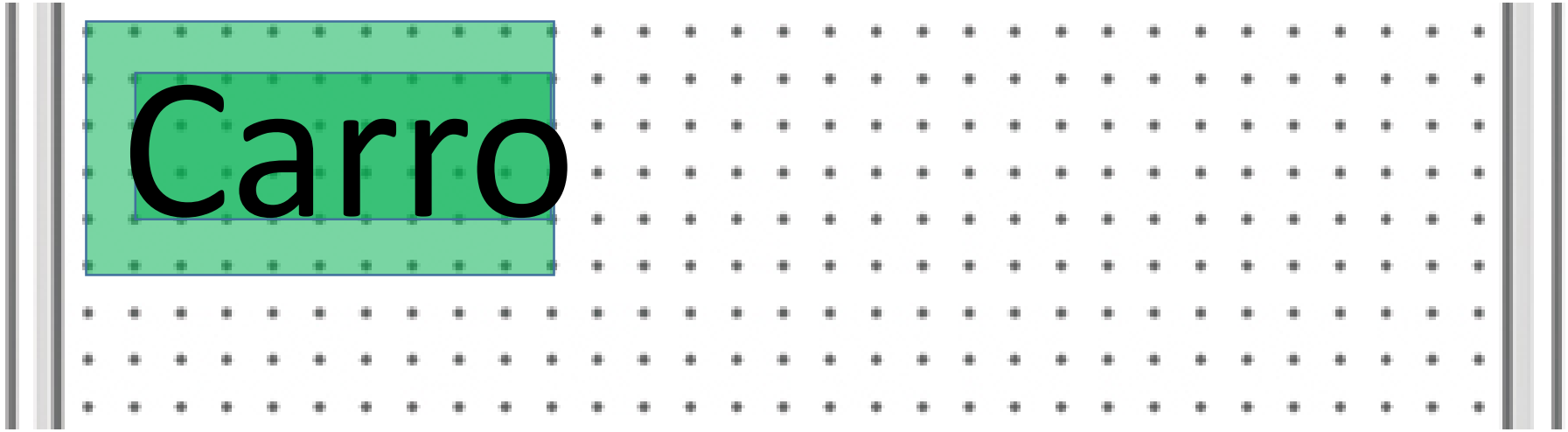
```
<TextView
    android:text="Apple"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FF0000"
    android:textSize="30sp"
    android:padding="20dp"/>
```



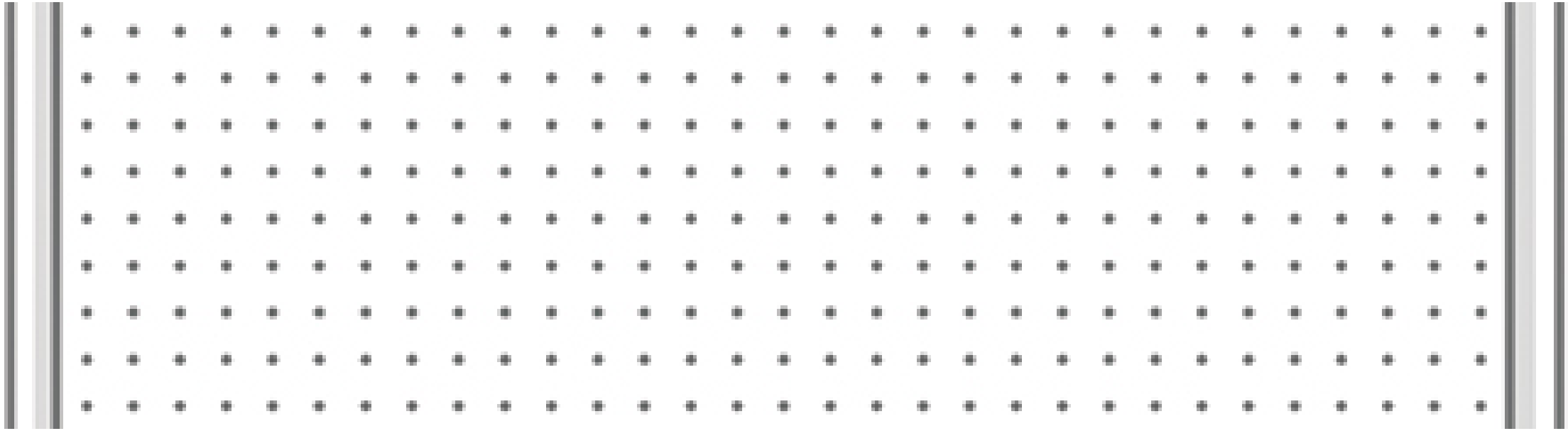
```
<TextView
    android:text="Banana"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#FFFF00"
    android:textSize="50sp"
    android:padding="10dp"/>
```



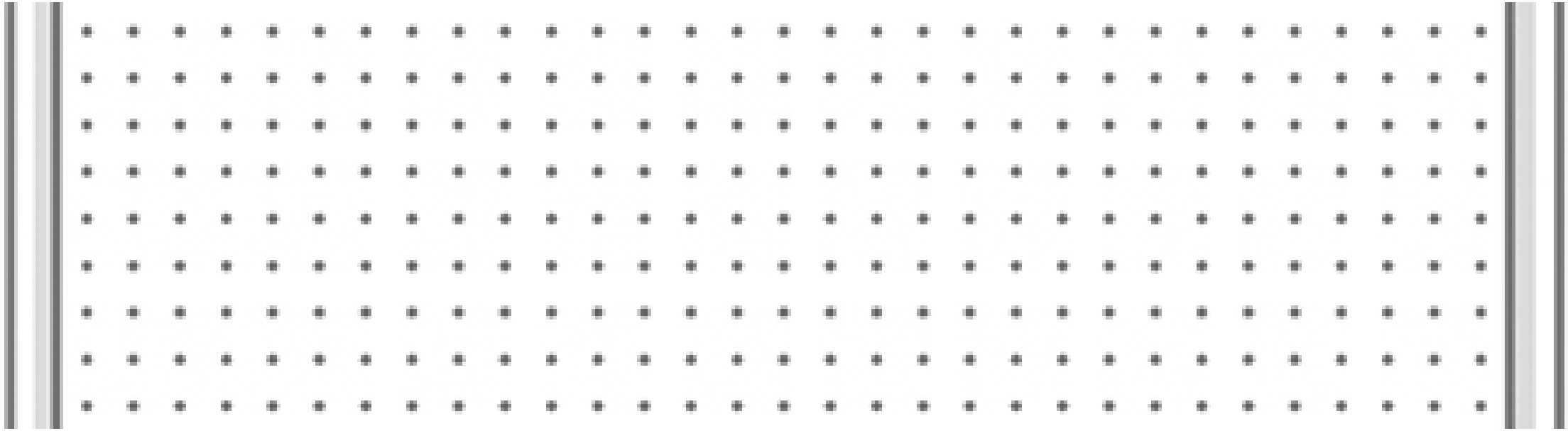
```
<TextView
    android:text="Carrot"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:background="#00FF00"
    android:textSize="30sp"
    android:padding="10dp"/>
```



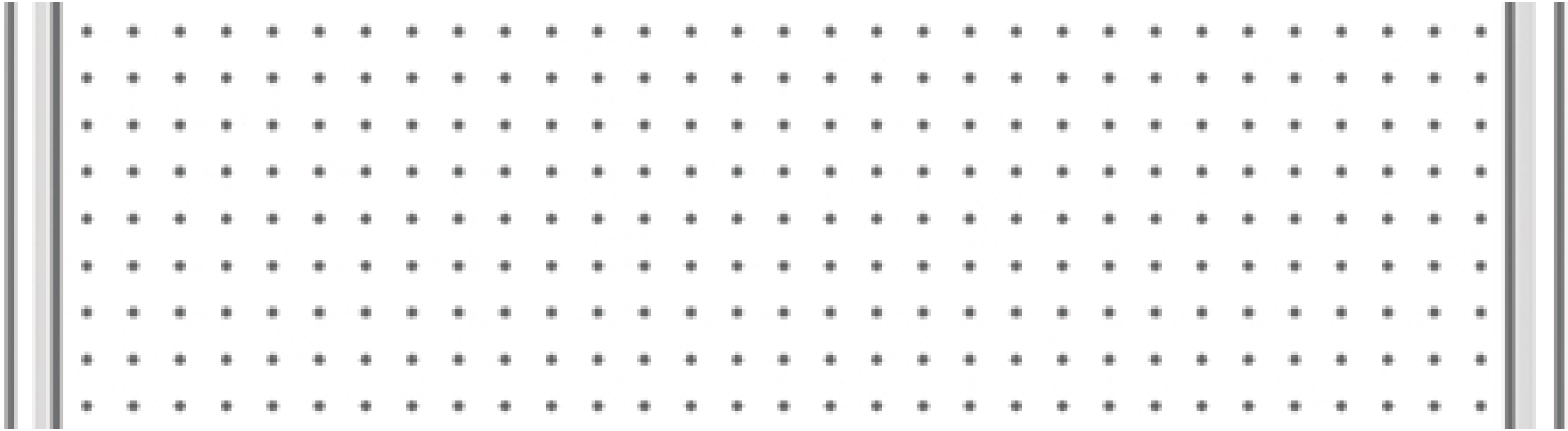
```
<TextView
    android:text="Hamster"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#00FF00"
    android:textSize="40sp"
    android:padding="20dp"/>
```



```
<TextView
    android:text="Hamster"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#00FFFF"
    android:textSize="40sp"
    android:padding="20dp"/>
```



```
<TextView
    android:text="Hamster"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:background="#FFFF00"
    android:textSize="40sp"
    android:padding="20dp"/>
```



```
<TextView
    android:text="Hamster"
    android:layout_width="300dp"
    android:layout_height="200dp"
    android:background="#FF0000"
    android:textSize="40sp"
    android:padding="20dp"/>
```