

[Open in app](#)[Get started](#)

Published in Mobile App Development Publication

This is your **last** free member-only story this month. [Sign up for Medium and get an extra one](#)



Elye

[Follow](#)Aug 3, 2019 · 6 min read · ✨ · 🎧 [Listen](#)

Save



## Expounding Android Canvas' DrawText

Photo by [Jason Leung](#) on [Unsplash](#)

In Android, when we want to have some Text, we would just use `TextView`. However, in case we want to have a custom view and have better control of the text, we could use the `Canvas` ' `drawText` ' api to draw the text.

Check out the below simple App I made to show some fun with `drawText` (you could get the code in the below link)




[Open in app](#)
[Get started](#)


In this blog, I'll be explaining some `drawText` features provided.

## The function arguments

It consist of 4 arguments.

```
canvas.drawText(text, coordinateX, coordinateY, paint)
```

- The `text` is just the text in `string`. There are other similar function which takes in



[Open in app](#)[Get started](#)

- The `paint` to decide how one would draw the text.

## The coordinate

If we were to draw the text using coordinate 0, 0 as below

```
canvas.drawText(text, 0, 0, paint)
```

We'll not see the text shown at all. This is because the coordinate is set as below

(coordinateX, coordinateY)



## Y coordinate

This is set at the base line of the text (not the most bottom, as you notice it is not at the bottom tip of the `y` letter, but below all other letters that doesn't have a 'tail').

## X coordinate

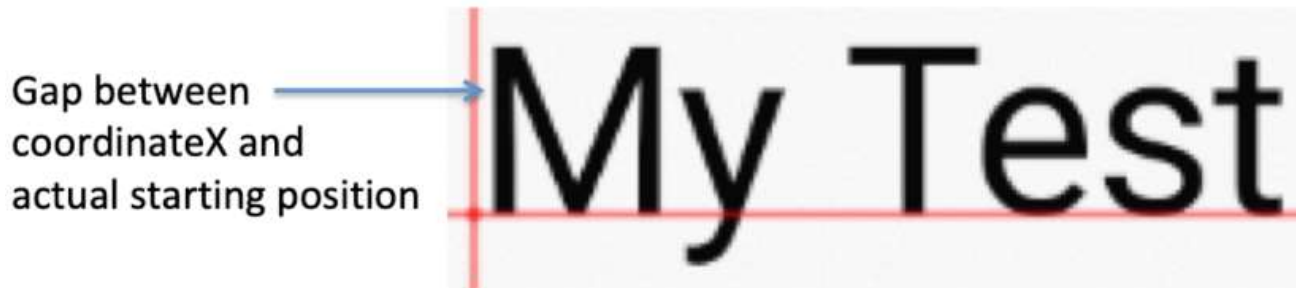
The X coordinate is by default set on the left side of the text. This is because by default the text is left aligned. One could set to be Center Align or Right Align through the paint as shown below.

```
paint.textAlign = Paint.Align.CENTER  
paint.textAlign = Paint.Align.RIGHT  
paint.textAlign = Paint.Align.LEFT // Default
```



[Open in app](#)[Get started](#)

The other point of X coordinate is, the text doesn't start per the X coordinate exactly. But leave a little gap first. That gap is basically the gap between each letter.



### Measuring the height and width of the text.

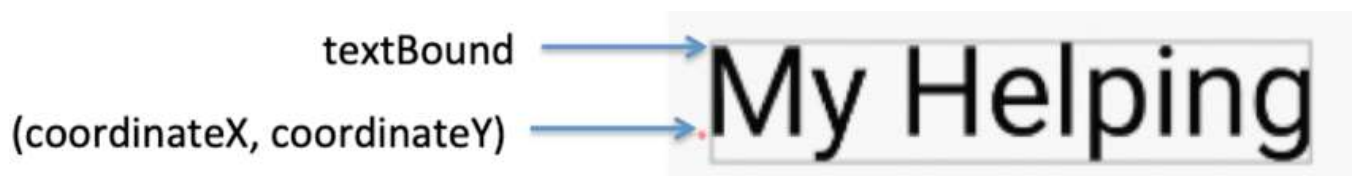
Given that the Y coordinate is on the baseline of the text and not the top, it is important for one to know the height of the text in order to set the right coordinate.

Similarly as there's a gap between the coordinate X and the actual Text start drawing area. We need to know the exact measure, to know where to set the X coordinate.

One way to do so is to get the text bound. This could be done by using the `getTextBounds` function of `paint` object. The `textBounds` is a `Rect` object that need to be allocated outside the function, and sent in to be filled.

```
val textBound = Rect()
paint.getTextBounds(TEXT, 0, TEXT.length, textBound)
```

With the above it will produce measure the TextBound per the exact size of the text, and based in the `coordinateX` and `coordinateY` as (0, 0)





Open in app

Get started



From the above, we know that, we'll need at least 59px of height to make the entire text visible. As for the X axis, there's an extra 6px width there. We could consider using these value to decide the coordinate.

*Note: Other than `getTextBounds`, there's another function `measureLength`. They are different event in the length itself. More elaboration in the link below*

### Android Paint: `.measureText()` vs `.getTextBounds()`

I'm measuring text using `Paint.getTextBounds()`, since I'm interested in getting both the height and width of the text...

[stackoverflow.com](https://stackoverflow.com)

## The paint

After positioning the text, we decide how we want the text to look there. This are done through the `paint` parameter. There are many things could be done through `paint`, but I'll be illustrating few interesting one that particularly apply to `drawText`

### textScaleX

```
// Set the paint's horizontal scale factor for text. The default
// value is 1.0. Values > 1.0 will stretch the text wider.
// Values < 1.0 will stretch the text narrower.
```

```
paintText.textScaleX = <float>
```



[Open in app](#)[Get started](#)

# Drawing Text!

## letterSpacing

```
// Set the paint's letter-spacing for text. The default value  
// is 0. The value is in 'EM' units. Typical values for slight  
// expansion will be around 0.05. Negative values tighten text.
```

```
paintText.letterSpacing = <float>
```

# Drawing Text!

*Note: this differ from `textScaleX` as the letter doesn't grow fatter horizontally, only the space in between the text changed*

## textSkewX

```
// Set the paint's horizontal skew factor for text. The default  
// value is 0.  
// For approximating oblique text, use values around -0.25.
```

```
paintText.textSize = <float>
```

# Drawing Text!




[Open in app](#)
[Get started](#)

There's some settings that's either TRUE or FALSE, and can be combined. Listed below are the few of them.

**Drawing Text!**

default

**Drawing Text!**

`paintText.isFakeBoldText = true`

~~Drawing Text!~~

`paintText.isStrikeThruText = true`

Drawing Text!

`paintText.isUnderlineText = true`

### style

This is like other drawing that define either just stroke outline or fill everything, or a combine of both.

```
paintText.style = <Paint.Style>
```

**Drawing Text!**

**Fill**

**Drawing Text!**

**FILL\_AND\_STROKE**

Drawing Text!

**STROKE**




[Open in app](#)
[Get started](#)

`Typeface.DEFAULT_BOLD`, `Typeface.SAN_SERIF`, `Typeface.SERIF` or `Typeface.MONOSPACE`

```
paintText.typeface = <Typeface>
```

One could also set a custom one like the Waltz Disney font below, using

`Typeface.createFromAsset` function.

```
Typeface.createFromAsset(assets, "<YOURFONT.ttf stored in assets>")
```

Drawing Text!

**San Serif / Default**

**Drawing Text!**

**Default Bold**

Drawing Text!

**Serif**

Drawing Text!

**Monospace**

DRAWING TEXT!

**Custom (Waltz Disney Typeface)**

Other than the above, there are more text attribute. The Chinese Blog below (use Chrome translate) shows you more

**HenCoder Android 开发进阶：自定义 View 1-3 drawText() 文字的绘制**

这期是 HenCoder 自定义绘制的第三期：文字的绘制。之前的内容在这





[Open in app](#)[Get started](#)

If you like to get the code of the example app shown above, it is in

### **elye/demo\_android\_drawtext**

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or...

github.com

If you like to learn about other Android Canvas Draw, check out

### **Learn ALL Android canvas draw functions**

If you would like to create custom view from grown up in Android, knowing what Canvas' draw functions available out...

medium.com

I hope this post is helpful to you. You could check out my other interesting topics [here](#).

Follow me on [medium](#), [Twitter](#), [Facebook](#) or [Reddit](#) for little tips and learning on Android, Kotlin etc related topics. ~Elye~

---

## **Get the latest update whenever Elye publishes**

Make learning Android, iOS, and software development more fun and easier





Open in app

Get started



Subscribe

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

