Pretty Pictures!

Jumping into Art

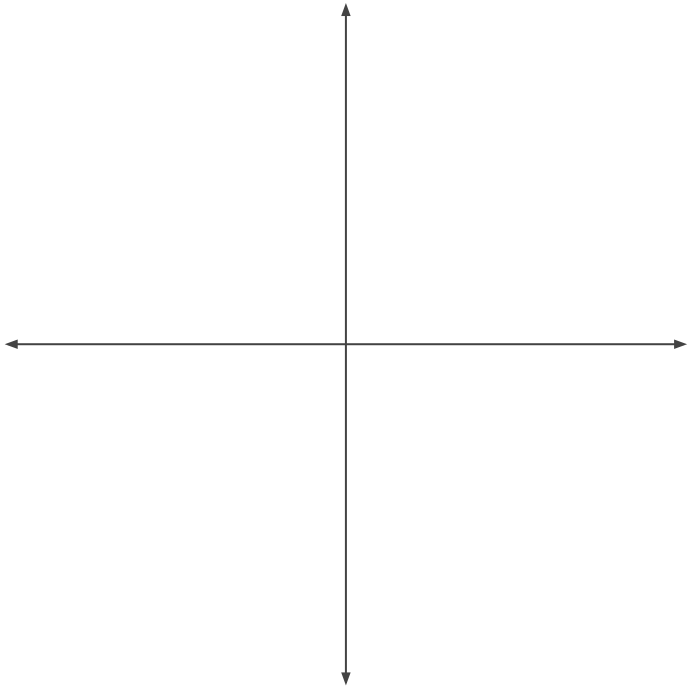
Today we're gonna move into a realm of even greater interactivity - displaying **art** on the screen!

This will give our programs more interactivity than we could get from simply displaying text.

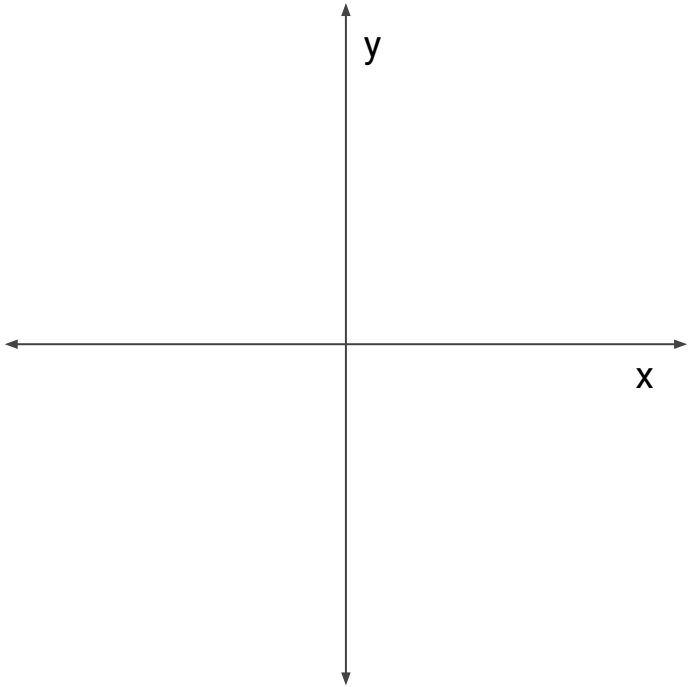
Where we gon' draw?

On the Canvas!

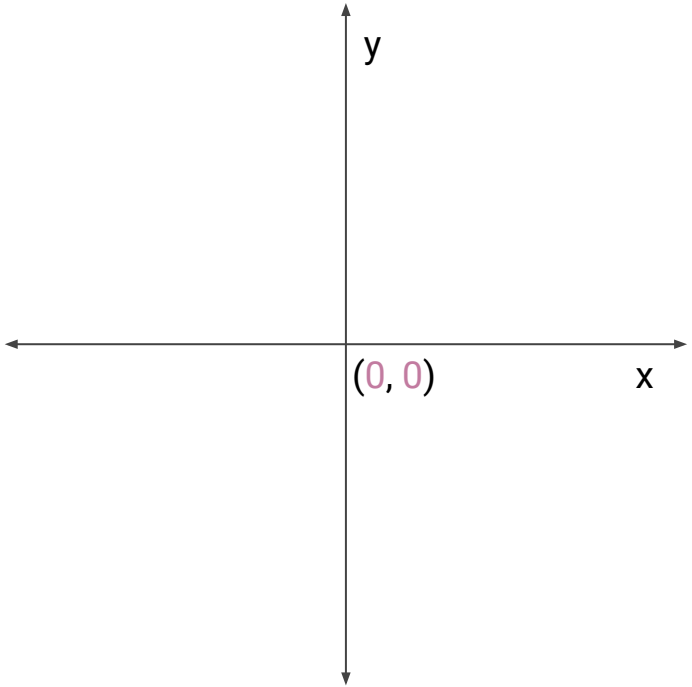
Coordinate Plane (Math)



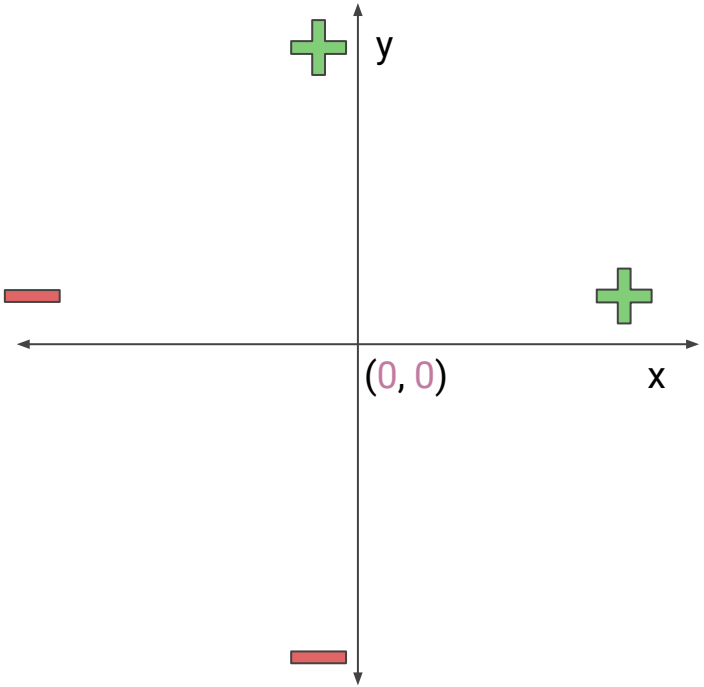
Coordinate Plane (Math)



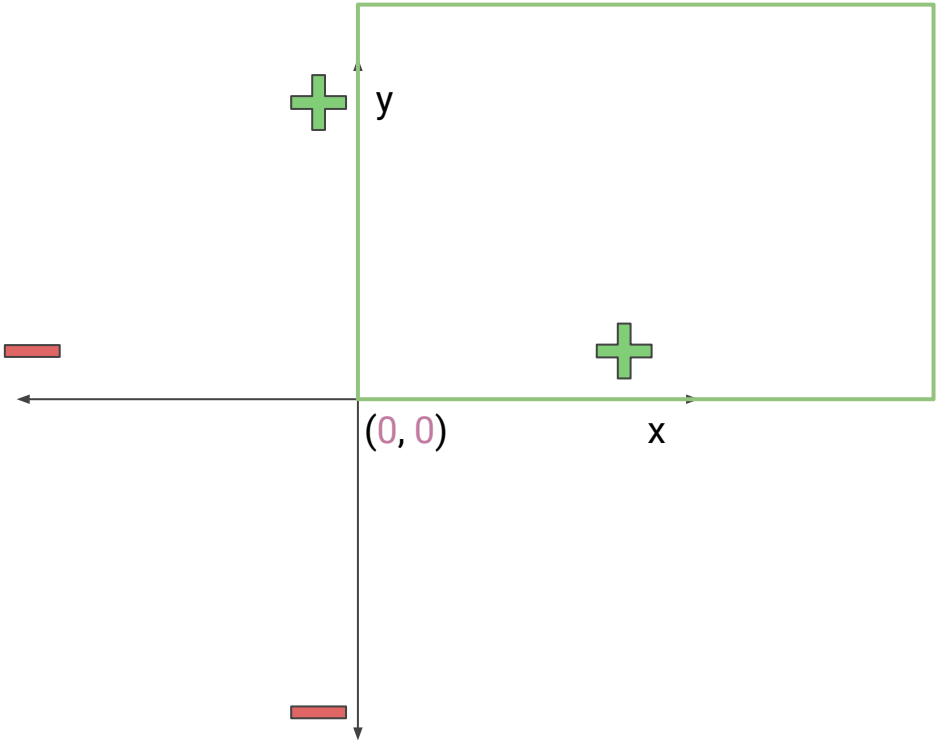
Coordinate Plane (Math)



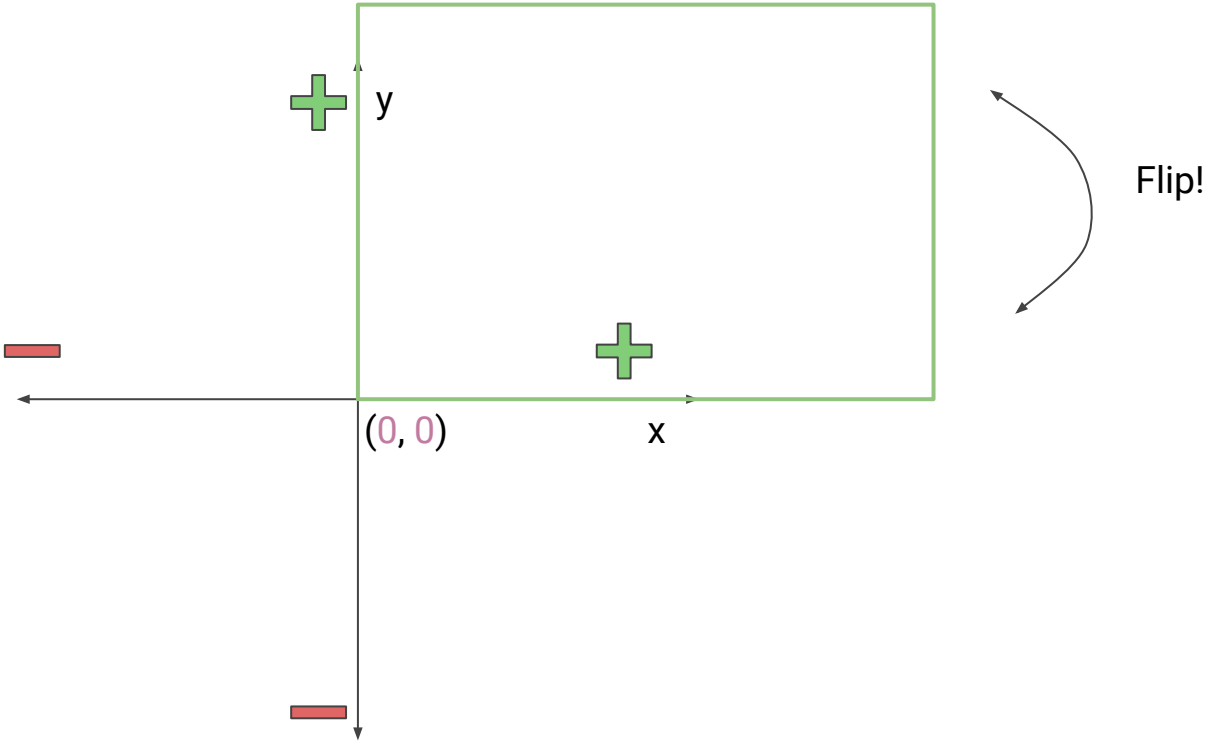
Coordinate Plane (Math)



Coordinate Plane (Graphics)



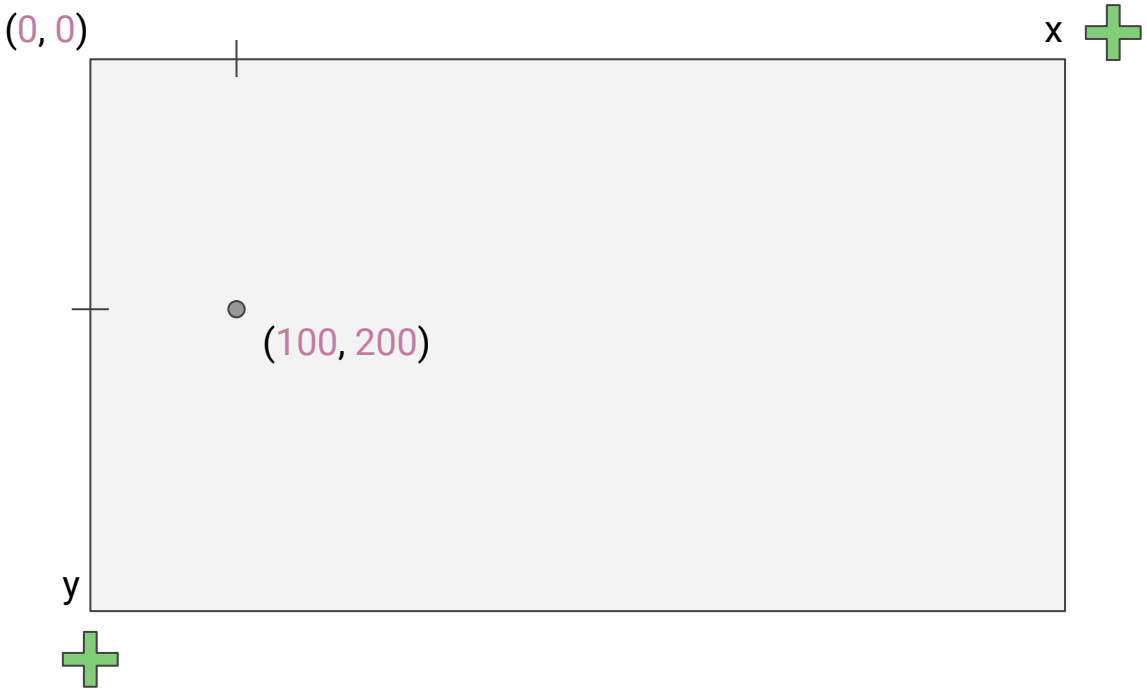
Coordinate Plane (Graphics)



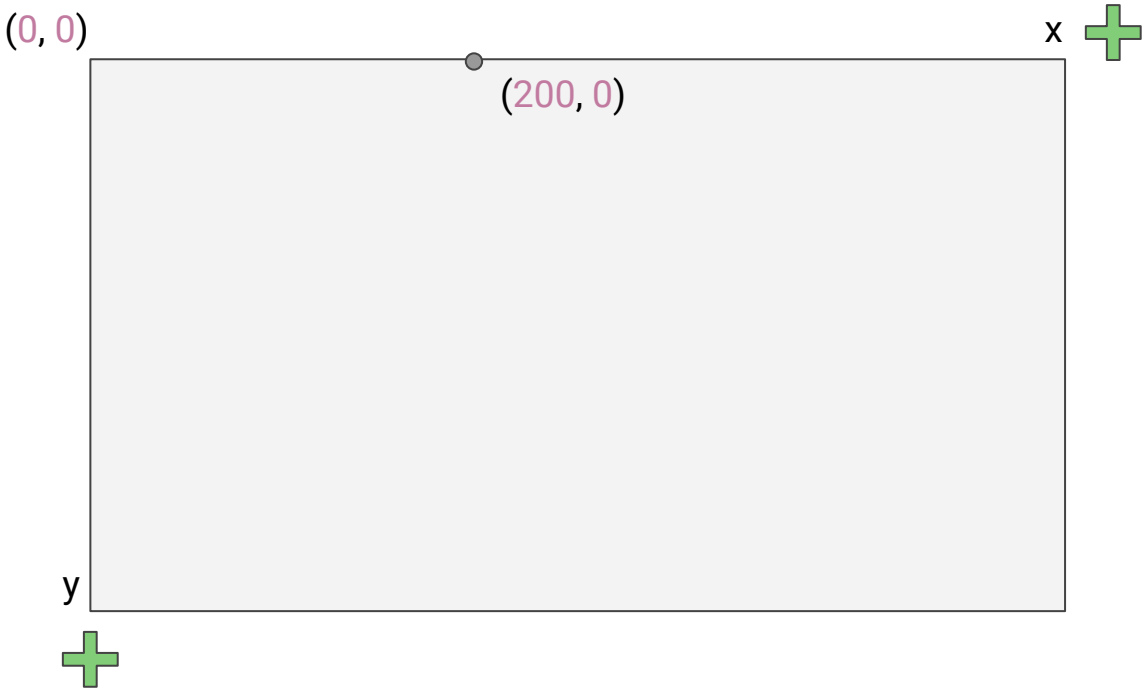
Flip!

Coordinate Plane (Graphics)

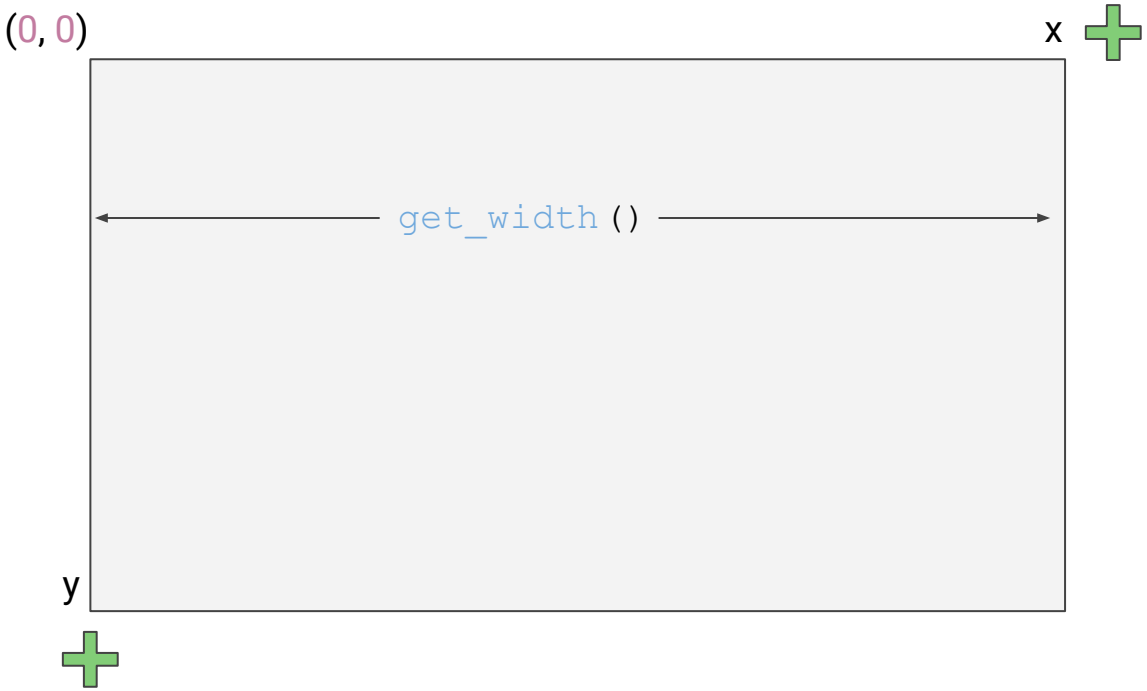
Coordinate Plane (Graphics)



Coordinate Plane (Graphics)

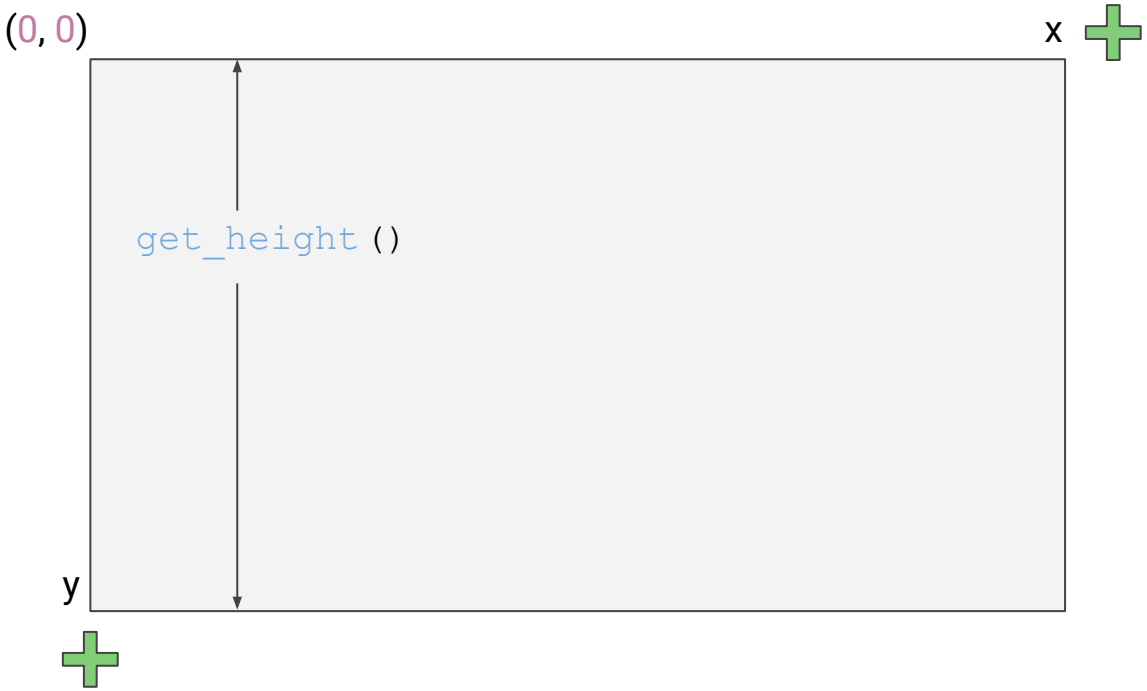


Coordinate Plane (Graphics)



The `get_width()` function can be used to tell us how many pixels **wide** our canvas is.

Coordinate Plane (Graphics)



The `get_height()` function can be used to tell us how many pixels **tall** our canvas is.

Displaying Text on Canvas

In order to display text on our canvas, we need to create a variable to store that text.

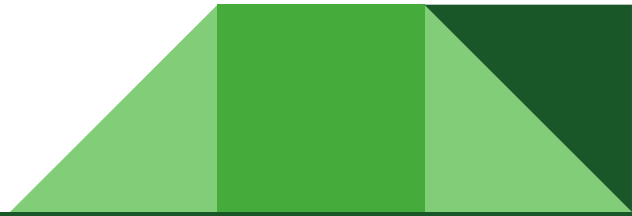
```
my_text = Text("Hello World!")
```

Displaying Text on Canvas

In order to display text on our canvas, we need to create a variable to store that text.

```
my_text = Text("Hello World!")
```

`my_text` is the variable we're using to store the Text value.



Displaying Text on Canvas

In order to display text on our canvas, we need to create a variable to store that text.

```
my_text = Text("Hello World!")
```

`my_text` is the variable we're using to store the Text value.

`Text()` will create a value that's ready to be displayed on the canvas.

Displaying Text on Canvas

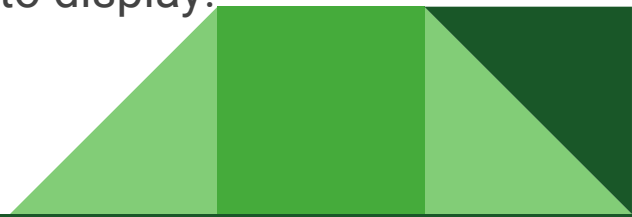
In order to display text on our canvas, we need to create a variable to store that text.

```
my_text = Text("Hello World!")
```

`my_text` is the variable we're using to store the Text value.

`Text()` will create a value that's ready to be displayed on the canvas.

The value we give to `Text()` is the string value we want to display!



Displaying Text on Canvas

Once we have a variable, we need to choose where on our canvas the value will be placed. Then (finally!) we can add it to our canvas!

```
my_text = Text("Hello World!")

my_text.set_position(100, 100)

add(my_text)
```

Displaying Text on Canvas

Once we have a variable, we need to choose where on our canvas the value will be placed. Then (finally!) we can add it to our canvas!

```
my_text = Text("Hello World!")

my_text.set_position(100, 100)

add(my_text)
```

This is the (x, y) coordinate of the **bottom left** corner of the text!

Displaying Text on Canvas

Once we have a variable, we need to choose where on our canvas the value will be placed. Then (finally!) we can add it to our canvas!

```
my_text = Text("Hello World!")

my_text.set_position(100, 100)

my_text.set_color(Color.red)

my_text.set_font("50pt Arial")

add(my_text)
```

There's even more options! We can select a color, as well as change the font and size of the letters!

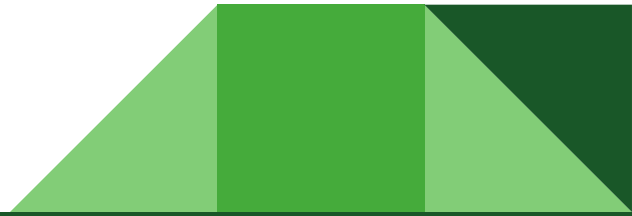
Displaying Text on Canvas


```
my_text = Text("Hello World!")
my_text.set_position(100, 100)
my_text.set_color(Color.red)
my_text.set_font("50pt Arial")
add(my_text)
```

Drawing Circles!

A **circle** can be drawn in a similar fashion - we need to create a variable, and **initialize** it with a Circle value, like so:

```
my_circle = Circle(25)

my_circle.set_position(100, 100)

my_circle.set_color(Color.blue)

add(my_circle)
```



Drawing Circles!

A **circle** can be drawn in a similar fashion - we need to create a variable, and **initialize** it with a Circle value, like so:

```
my_circle = Circle(25)

my_circle.set_position(100, 100)

my_circle.set_color(Color.blue)

add(my_circle)
```

This number determines the **radius** (distance from center to edge) of the circle.

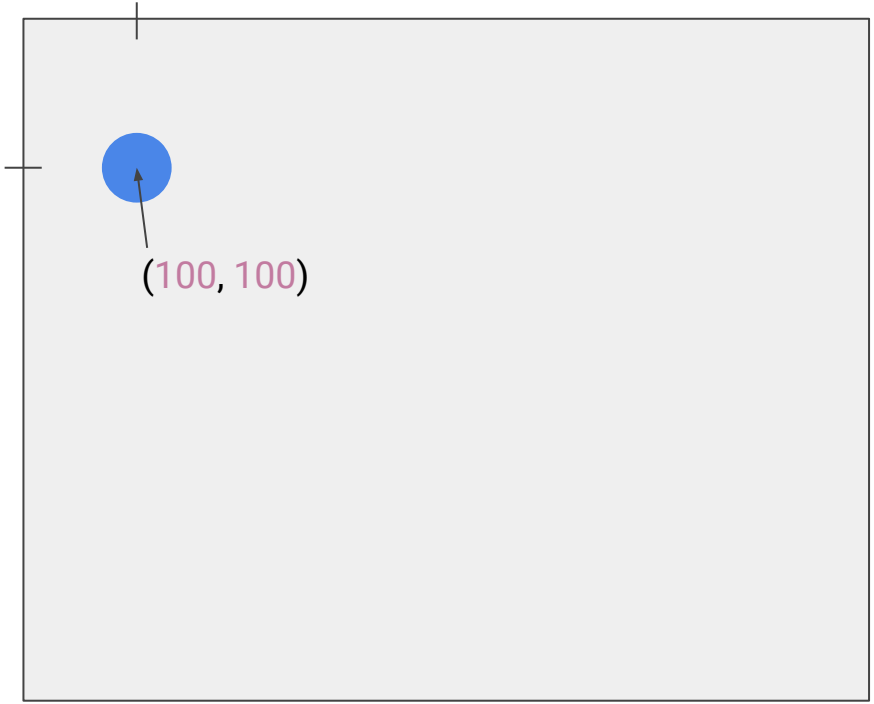
Drawing Circles!

A **circle** can be drawn in a similar fashion - we need to create a variable, and **initialize** it with a Circle value, like so:

```
my_circle = Circle(25)

my_circle.set_position(100, 100)

my_circle.set_color(Color.blue)

add(my_circle)
```

This is the (x, y) coordinate of the **center** of the circle!

Drawing Circles!

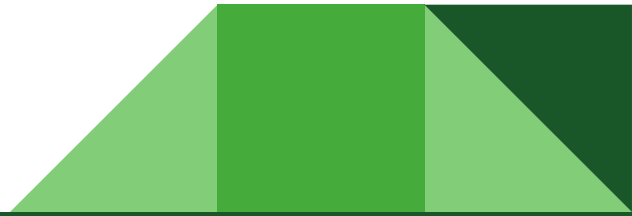



```
my_circle = Circle(25)
my_circle.set_position(100, 100)
my_circle.set_color(Color.blue)
add(my_circle)
```

Drawing Rectangles!

A **rectangle** can be drawn in almost the same way as a circle! We need to create a variable, and **initialize** it with a Rectangle value, like so:

```
my_rect = Rectangle(50, 100)

my_rect.set_position(100, 100)

my_rect.set_color(Color.green)

add(my_rect)
```



Drawing Rectangles!

A **rectangle** can be drawn in almost the same way as a circle! We need to create a variable, and **initialize** it with a Rectangle value, like so:

```
my_rect = Rectangle(50, 100)

my_rect.set_position(100, 100)

my_rect.set_color(Color.green)

add(my_rect)
```

These numbers determines the **width** and **height** of the rectangle.

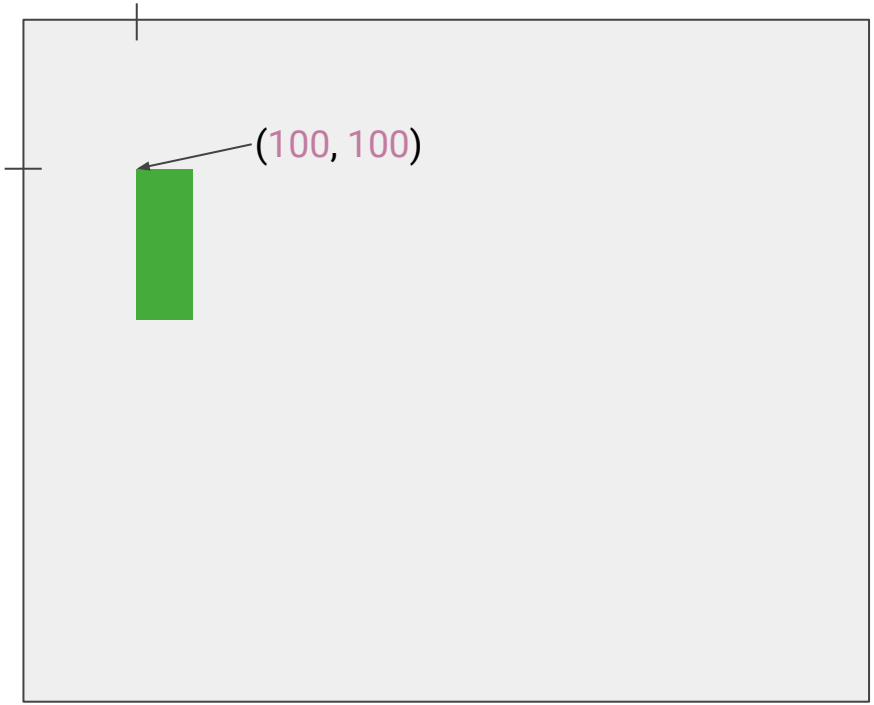
Drawing Rectangles!

A **rectangle** can be drawn in almost the same way as a circle! We need to create a variable, and **initialize** it with a Rectangle value, like so:

```
my_rect = Rectangle(50, 100)

my_rect.set_position(100, 100)

my_rect.set_color(Color.green)

add(my_rect)
```

This is the (x, y) coordinate of the **top left** corner of the rectangle!

Drawing Rectangles!




```
my_rect = Rectangle(50, 100)
my_rect.set_position(100, 100)
my_rect.set_color(Color.green)
add(my_rect)
```

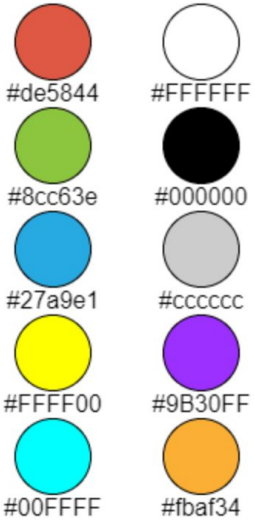
Available Colors

`Color.red`		`Color.white`	
`Color.green`		`Color.black`	
`Color.blue`		`Color.gray`	
`Color.yellow`		`Color.purple`	
`Color.cyan`		`Color.orange`	

But what if I want… more colors?

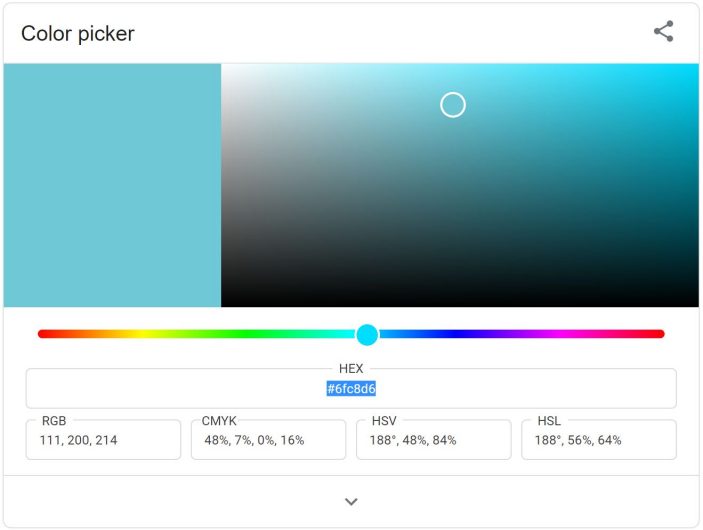
You're in luck! The values stored in those variables are actually **string**s! They store the **hex code** for the color they correspond to. If you want to add your own colors, you can simply put your desired color's **hex code** as a string in the `set_color()`!

```
my_rect = Rectangle(50, 100)
my_rect.set_position(100, 100)
my_rect.set_color("#6fc8d6")
```



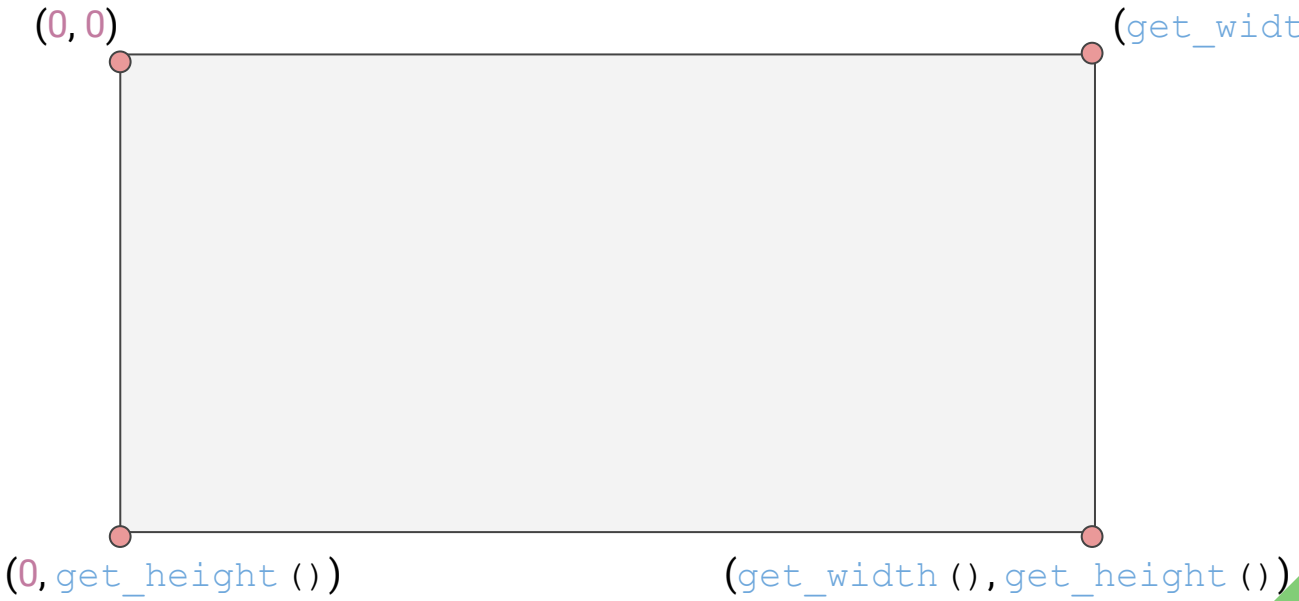
Where to find Hex Codes

The quickest and easiest way to find the hex code for the color of your choice is to simply Google "color picker". Google will present you with an interface like the one below, and you can pick whatever color you want!



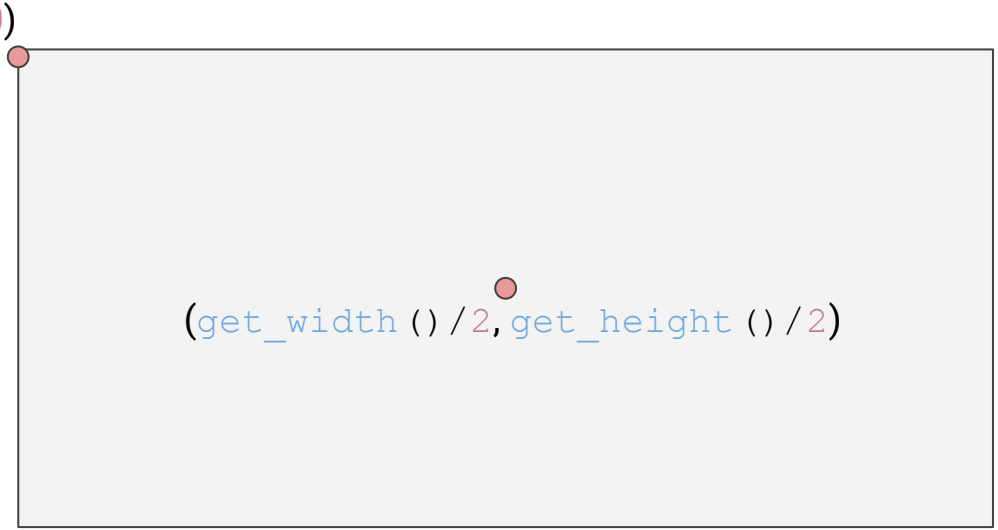
Relative Positioning

Using the `get_width()` and `get_height()` functions, we can determine how large our canvas is!



Relative Positioning

Using that information, I can divide each number by 2 and find the center of my canvas!

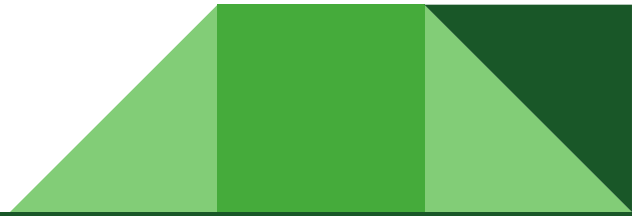


Object Properties

I can also find the width and height of the shapes I've drawn, using those same 2 functions!

```
my_text = Text("Howdy!")
text_width = my_text.get_width()
text_height = my_text.get_height()
```

This can be super helpful for centering an object in the middle of the canvas!

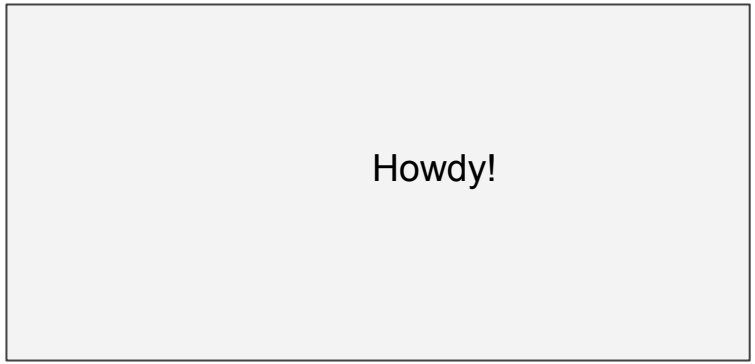


Centering some Text

```
center_x = get_width()/2
center_y = get_height()/2
my_text = Text("Howdy!")
text_wd = my_text.get_width()
text_ht = my_text.get_height()
my_text.set_position(center_x, center_y)
```

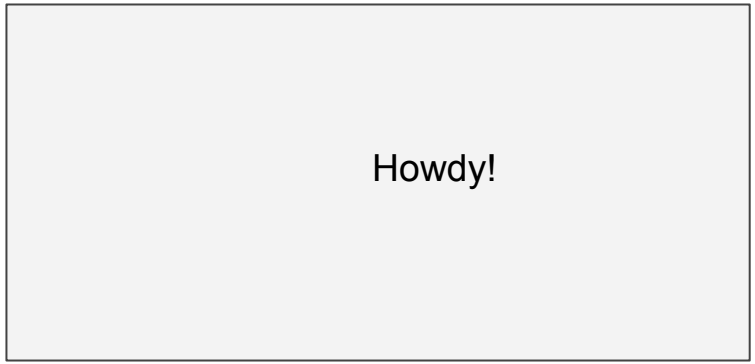
Centering some Text

```
center_x = get_width()/2
center_y = get_height()/2
my_text = Text("Howdy!")
text_wd = my_text.get_width()
text_ht = my_text.get_height()
my_text.set_position(center_x, center_y)
```



Centering some Text

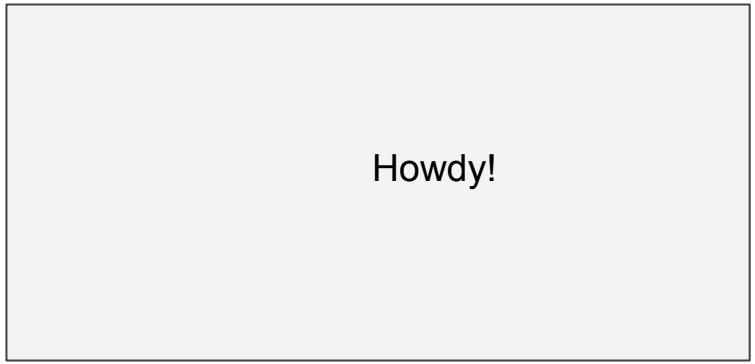
```
center_x = get_width()/2
center_y = get_height()/2
my_text = Text("Howdy!")
text_wd = my_text.get_width()
text_ht = my_text.get_height()
my_text.set_position(center_x, center_y)
```



This text isn't centered on our canvas because text uses its **bottom left** corner to determine its position!

Centering some Text

```
center_x = get_width()/2
center_y = get_height()/2
my_text = Text("Howdy!")
text_wd = my_text.get_width()
text_ht = my_text.get_height()
my_text.set_position(center_x, center_y)
```



This text isn't centered on our canvas because text uses its **bottom left** corner to determine its position!

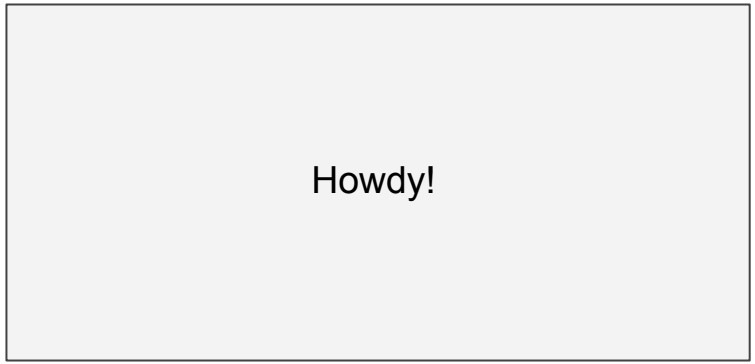
What we can do is adjust the text's coordinates by half of its width and height, thus centering it on the canvas!

Centering some Text

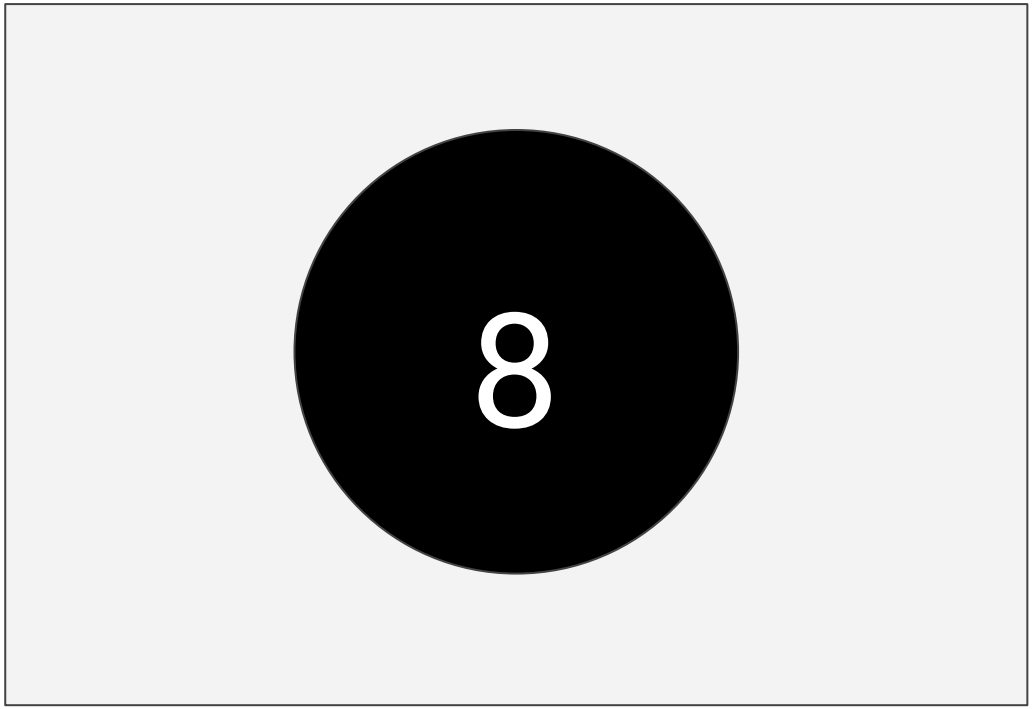
```
center_x = get_width()/2
center_y = get_height()/2
my_text = Text("Howdy!")
text_wd = my_text.get_width()
text_ht = my_text.get_height()
my_text.set_position(center_x - text_wd / 2, center_y + text_ht / 2)
```

This text isn't centered on our canvas because text uses its **bottom left** corner to determine its position!

What we can do is adjust the text's coordinates by half of its width and height, thus centering it on the canvas!



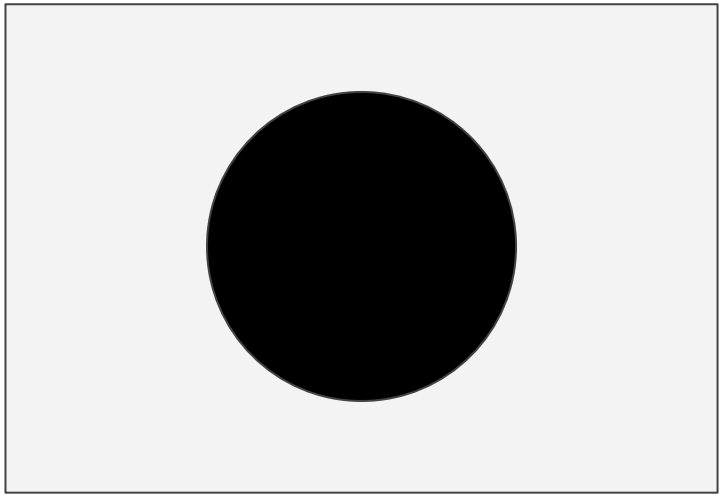
Example: 8 Ball



Step 1: The Circle

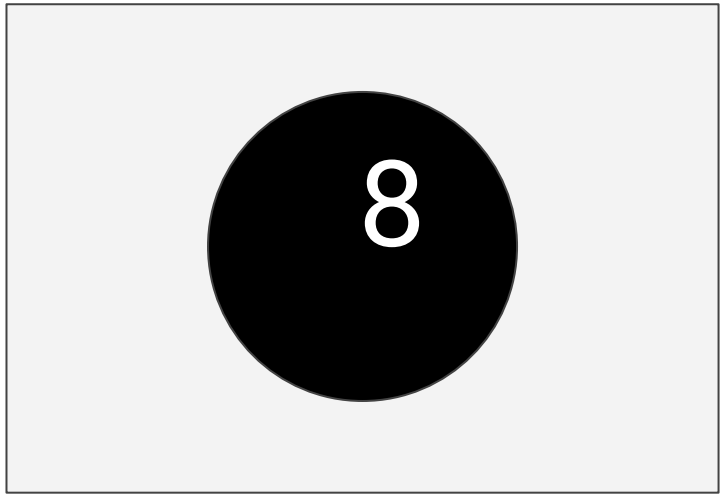
```
center_x = get_width()/2
center_y = get_height()/2
ball = Circle(100)
ball.set_position(center_x, center_y)
add(ball)
```

The default fill color is **black**!



Step 2: The Number

```
center_x = get_width()/2
center_y = get_height()/2
ball = Circle(100)
ball.set_position(center_x, center_y)
add(ball)

number = Text("8")
number.set_color(Color.white)
number.set_font("50pt Arial")
number.set_position(center_x, center_y)
add(number)
```



Fixing Step 2: The Number

```
center_x = get_width()/2
center_y = get_height()/2
ball = Circle(100)
ball.set_position(center_x, center_y)
add(ball)

number = Text("8")
number.set_color(Color.white)
number.set_font("50pt Arial")
num_wd = number.get_width()
num_ht = number.get_height()
num_x = center_x - num_wd / 2
num_y = center_y + num_ht / 2
number.set_position(num_x, num_y)
add(number)
```