**Drawing Text on a Graphical Screen**

The Pygame module contains a class called the **Font** class. In the Font class are some helpful methods that allow us to draw text as if it was a picture. Here's how:

1. Initialize the font class

pygame.font.init()

Put this line at the top of your code, under your "import" statements. This line tells the font class to set up the parameters so we can begin making fonts.

1. Create a font object

font = pygame.font.Font(None, 600)

The font object keeps track of one font. In our case, the "None" pointer means that we will use the default font type. This is the easiest way to get a basic font up and running. If you want another font, you'll have to do some research yourself. This example creates a font with a point size of 600 (very large). You can change it as you like.

1. Use the font object to create a picture of text

The render() method renders (creates) some text in a certain colour. We use (RGB) code to determine the colour - for example, green would be (0, 255,0).

This code will create a picture of the word “Hello!” in green. It goes in your main section, after you have created a screen:

someText = font.render("Hello!", False, (0,255,0)) # set up some green text

The second argument in the render() method is False. This argument determines whether anti-aliasing is on or off. You can read up on anti-aliasing to understand it, but for now, we turn this option off.

1. Draw the picture of the text on the screen

Now we can draw it using the blit() method:

screen.blit(someText, (20,20)) # draw text at coordinates

pygame.display.flip() # put on screen