GUI function documentation

The area on which your program draws is called a canvas. The canvas has a coordinate system just like you're used to. The origin is in the center of the canvas. If you move up the canvas, this is the positive y direction; move to the right on the canvas, that is the positive x direction.

The functions below all belong to the canvas object – (actually, when a function belongs to an object, we call them **methods**) So we need to use the syntax: **objectName.methodName()** similar to when we call functions from the math module.

clear() – erases all the drawings from the canvas

config(***) -- configures the canvas. We'll use this to change the background color.

The argument syntax is something new. config() has many parameters, but instead of requiring the call to list an argument for every single one, config allows the call to specify which parameter is being initialized and by what argument. (These are known as keyword arguments)

The one parameter that we'll use is the **bg** parameter (for background). If we want to set the background of the canvas to blue, the syntax is: **canvas.config(bg = 'blue')**

Each of the methods on the next page draws a shape on the canvas. They have similar parameters. Some are required, some are optional:

- One or more **points**. A point is an x, y pair. The required syntax is to put the coordinates within square brackets. For example [3, 4] (This Python syntax, of putting data within square brackets, separated by commas, is known as a list.) If a function requires more than one point, then those points must be a list. For example [[0,0] , [10,17], [2,3]]

- An optional **fill** parameter, specifying if the shape should be filled in with a color. The color options are:

  "white" "black" "red" "green"
  "blue" "cyan" "yellow" "magenta"

  The syntax for **fill** is the same as **bg** above. If you don't specify this parameter, then the shape is not filled in, but is only an outline. Even when a shape is colored in, there is still an outline in black

- An optional **width** parameter, specifying how thick to make the border of the object (or, when drawing a line, how thick to make the line)

- An optional **outline** parameter (like fill) where you can specify the color of the outline.

- All of the numeric values (coordinates, linear measurements like radius or height) must be integers

circle(center, radius, *)** - draws a circle
center: a point specifying tne center of the circle
radius: a length

rectangle(points, *)** – draws a rectangle
The first parameter is a list of two points. The first point is the lower left corner of the rectangle. The second point is the upper right corner.

line(points, *)** – draws lines connecting the points
The first parameter is a list of points. A line segement is drawn between each point (but no connecting the last point to the first point)

polygon(points, *)** – draws a polygon with the given points as vertices
The first parameter again is a list of points (like line). The significant difference between this function and the line function is that the last point is connected to the first point to complete the shape.

Example method calls:

```
canvas.circle([10, 10], 25,  fill = 'red')
canvas.rectangle( [ [20, 30], [70, 32] ],   outline = 'magenta')
canvas.line([ [1, 1], [20, 20], [40, 1] ], width = 5)
```