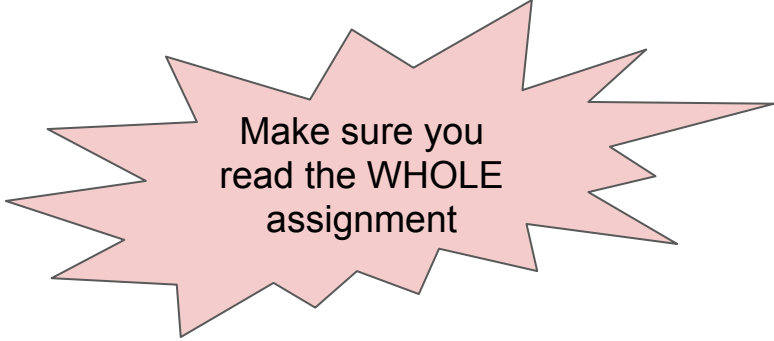

Python Objects and SVG

on the web

The Pledge

Every file you build in python **must** begin with a **comment** containing the academic honesty pledge as shown below. It should be near the top.

Add another, separate comment to the file containing your name.

```
# I honor Parkland's core values by affirming that I have
# followed all academic integrity guidelines for this work.

# your name
```

Overview

- Build three classes and a main, each in a separate python source file.
  - A **`Point`** class to keep track of coordinates.
  - A **`Color`** class to keep track of RGB colors.
  - A **`Rectangle`** class to keep track of a Point, a Color, a Width and a Height associated with a rectangle. (This is a solid, filled rectangle)
  - A main.py CGI enabled script the will create a **list** of 1000 **random** rectangles and generate the SVG code that will display the rectangles on a web page.
- Turn in a link to your working CGI program.

Requirements

- The objects need to be in the files (no caps!)
  - point.py
  - color.py
  - rectangle.py
- You must attach all the .py files in the submission including main.py
- You must place a link to the main CGI script in the message area of the submission.

`class Point`

- Contains 2 coordinates
  - `_across`
  - `_down`
- Contains setters and getters for those coordinates
- The constructor will initialize them to both 0.
- There is code in the file that tests all the methods in the class.

`class Color`

- Contains 3 color values (from 0 to 255)
  - _red
  - _green
  - _blue
- Contains setters and getters for those properties.
- The constructor will initialize them **to 0,0,0**
- Contains a method called **`SVG()`** which outputs the string **`rgb(0,0,0)`** which can be used inside a <circle .. /> tag. See https://www.w3.org/TR/SVGColor12/#Color_syntax and use the "Integer functional"
- There is code in the file that tests all the methods in the class.

`class Rectangle`

- Contains a `_upperleft` (which is a **point** object)
- Contains a `_height` (which is an integer)
- Contains a `_width` (which is an integer)
- Contains a `_fill` (which is a **color** object)
- Contains a method called SVG() with **returns** a string that takes the properties and create an SVG rectangle string as described at https://www.w3schools.com/graphics/svg_rect.asp (This is a solid filled rectangle)
- Contains setters and getters for all the properties.
- The constructor will take 4 parameters. `thepoint, width, height` and `fill` in that order.
- There is code in the file that tests all the methods in the class.

incomplete ideas for the main (main.py)

There will be a separate file that will generate a list of shapes and display them. Here's the necessary HTML & HTTP. Notice all the shapes are in 1 svg tag.

```
# code that creates 1000 or objects and puts them in rectangle_list
# that code will generate Point and Color objects to build the rectangles

print ('<html><head></head><body>')
print ('<svg height="1000" width="1000">')
for rectangle in rectangle_list:
    print (rectangle.SVG()) # or something similar
print ('</svg>')
print ('</body></html>')
```

Importing Files

https://en.m.wikibooks.org/wiki/A_Beginner%27s_Python_Tutorial/Importing_Modules

```
from somefile import ClassName
```

• Only the item ClassName in somefile.py gets Imported.
• After importing ClassName, you can just use it without a module prefix. It's brought into the current namespace.
• Take care! Overwrites the definition of this name if already defined in the current namespace!

SVG

We're going to do other things with the SVG tag in HTML. Here's are some links

- https://www.w3schools.com/graphics/svg_intro.asp
- https://en.wikipedia.org/wiki/Scalable_Vector_Graphics