CS4 Python Style Guide

Having good style conventions is important when reading and debugging your code. Please use the guidelines below when formatting your python code. These guidelines are a brief summary of the more in depth **PEP 8 Style Guide for Python Code**.

1 Spacing

- Use four spaces instead of tabs.
- Line length should be no longer than 80 characters
- Separate your function declarations by a new line
- Avoid whitespace:
  - immediately inside parentheses

    Do: `("CS" + "4")`
    Don't: `( "CS" + "4" )`

  - before colons, commas and brackets

    Do: `cs[4]`
    Don't: `cs [4]`

- Put a newline after branching and looping statements

  Do:

```
if class == 4:
    print("It's CS4!")
else:
    print("Not CS4 =(")
```

  Don't:

```
if a > 4: print("It's CS4!")
else: print("Not CS4 =(")
```

2 Naming

- Functions and variables should be written in all lowercase letters with underscore separating words

  Ex: `lower_case, my_function, distance, num_times`

- Constants should be written in all capital letters with underscores separating words

  Ex: `CONSTANT_CASE, HEIGHT, BORDER_WIDTH, MAX_VALUE`

- Class names (and exceptions) should be written with the "CamelCase" convention

  Ex: `CamelCaseClass, MyAwesomeClass, LibraryBook`

- Modules should have short, lowercase names.

  Ex: `module, algos, lower`

3 Commenting

- Write docstrings for every class, function, module and method. The docstring should briefly describe what the function does as well as list inputs and outputs.

```
def multiply(a, b):
    """
    Multiply two numbers together and return the product.
    """
```

```
return a*b
```

- Comments should be used sparingly to explain complex code.

4 Other

- Place your imports at the start of the file, and give each import its own line.
- Don't write compound statements

  Do:

  ```
  function1()
  function2()
  function3()
  ```

  Don't: `function1(); function2(); function3()`