CS 220 Python Style Guide

{kvedder, aaronweiss, hpoddar, ilevyor, jholzman, jreedie, timothymcnam}@umass.edu,
gordon@cs.umass.edu

January 28, 2017

0: Why?

Style matters. Code is read far more than it's written, and thus writing readable code is critical.

This document aims to outline guidelines that will make your code more readable; it's for your TAs and for your instructor, but it's also for you. Debugging is easier when you can actually read the hodgepodge of code in front of you.

This document ultimately is a very pared down version of the PEP8 style guide. It's short, please read it.

1: Indentation

1. **Use four spaces, not tabs.** Do not mix tabs and spaces; Python 3 does not allow for mixing of the two and will throw errors.
2. **Use hanging indents for long lines:**

   ```
   # Aligned with opening delimiter.
   foo = long_function_name(var_one, var_two,
                            var_three, var_four)
   ```

   Note that the variables line up under one another so that it's clear they are part of the function invocation.
3. **Line limit of 80 characters.** Don't write crazy long lines.

2: Whitespace

1. **Avoid extraneous whitespace.** Don't put spaces between the start of a function parameter list and the first parameter and other places where it doesn't aid readability.
2. **Use whitespace to aid in readability.** Add it after a comma in a list, and places of that nature that aid in clarifying the code's intent.

3: Comments

1. **Use comments sparingly.** Don't make every other line comments, as this results in code that doesn't flow well. Comments should carry their weight, not just restate the line below them.
2. **Avoid inline comments.** Add comments to the line above, not at the end of a line after the code.

4: Naming Convention

1. **Use snake case for variable names.** It's *my_variable_name*, not *myVariableName*.
2. **Use Pascal case for class names.** It's *MyClassName*, not *my_class_name*.
3. **Use all caps for constants with underscores for spaces.** It's *MY_CONSTANT_NAME*, not *my_constant_name*.
4. **Use short, all lowercase names for package and module names.** It's *mypackagename* not *my_package_name*.
5. **Differentiate between public and private variables.** In Java, private variables cannot be accessed outside of the object they are declared inside. Python does not have the concept of Public and Private, but instead follows the convention of beginning private variables with underscores. Thus: *_my_private_name* not *my_private_name*.

   For more information, see the Python docs on private variables.
6. **Avoid single letter variables.** The variable *O* doesn't tell you very much, does it?
7. **Other naming conventions.** The underscore itself does have special significance in some cases in Python. For example, *__init__* is a "magic " attribute that lives in user-controlled namespaces. Do not name you standard variables with leading and trailing double underscores; only use these names as directed by the documentation.