

Making your code easy to read: good Python coding practices*

*many of these ideas are also applicable to other languages

ASTR400B
Ekta Patel

Python Syntax

- If you are not already familiar with them, familiarize yourself with the syntax associated with your version of Python (2.7 or 3.6). Some things you will use often include:
 - print statements
 - differentiating between floats, integers
 - commenting
 - indents
 - quotes for strings
 - white spaces
 - brackets, parentheses, square brackets

https://www.tutorialspoint.com/python/python_basic_syntax.htm

The following suggestions are strongly encouraged.

Comment!

- All of your code should have comments. Why?
 1. You may need to look back at your code after some time and comments will help jog your memory.
 2. Others may need to read your code to collaborate with you, grade your assignments, or to build some extension of what you've written.

Organize your code logically

- Section your code so that related lines can be found in the same place, separated by line break from other parts of your code.
- A good general order to stick to for this class is:
 - A short comment on the goal of the code.
 - Imports
 - Functions
 - Read data
 - Do something with your data
 - Close and save your results.
- ***Good, final versions of your code should follow your thought process, but not necessarily the order in which you wrote it.***

**The following suggestions are recommended but
not required.**

Naming

- Do not rename variables in a single script!
 - example: `this_number = 2` --> `this_number = 2*2`
 - instead: `this_number = 2` --> `new_number = 2*2`
- lowercase: modules, functions, packages
- constants: all caps
- class names and exceptions: capital words (CapWords)

White spaces

- Immediately inside parentheses, brackets or braces.

Yes: `spam(ham[1], {eggs: 2})`

No: `spam(ham[1], { eggs: 2 })`

- Between a trailing comma and a following close parenthesis.

Yes: `foo = (0,)`

No: `bar = (0,)`

- Immediately before a comma, semicolon, or colon:

Yes: `if x == 4: print x, y; x, y = y, x`

No: `if x == 4 : print x , y ; x , y = y , x`

- Immediately before the open parenthesis that starts the argument list of a function call:

Yes: `spam(1)`

No: `spam (1)`

- Immediately before the open parenthesis that starts an indexing or slicing:

Yes: `dct['key'] = lst[index]`

No: `dct ['key'] = lst [index]`

- More than one space around an assignment (or other) operator to align it with another.

Yes:

`x = 1`

`y = 2`

`long_variable = 3`

No:

`x = 1`

`y = 2`

`long_variable = 3`

Line Lengths

- 79 characters python code
- 72 characters for long comments
- these are helpful conventions for having multiple files open on your screen
- requires knowledge of how to appropriately break your lines without hitting syntax errors

References

- <http://python.net/~goodger/projects/pycon/2007/idiomatic/handout.html>
- <http://docs.python-guide.org/en/latest/writing/style/>
- python style guide: <http://legacy.python.org/dev/peps/pep-0008/#code-lay-out>