Personal Clean Code cheat sheet by Dennis Fast

(based on PEP 8 -- Style Guide for Python Code https://www.python.org/dev/peps/pep-0008/)

Code Lay-out

- Use 4 spaces per indentation level
- Limit all lines to a maximum of 79 characters
- Line breaks before a binary operator
- Imports should usually be on separate lines

Whitespace in Expressions and Statements

- Avoid extraneous whitespace in the following situations:
  - Immediately inside parentheses, brackets or braces
  - Between a trailing comma and a following close parenthesis
  - Immediately before a comma, semicolon, or colon
  - Immediately before the open parenthesis that starts the argument list of a function call
  - Immediately before the open parenthesis that starts indexing or slicing
  - More than one space around an assignment (or other) operator to align it with another

Other Recommendations

- Always surround these binary operators with a single space on either side: assignment (=), augmented assignment (+=, -= etc.), comparisons (==, <, >, !=, <>, <=, >=, in, not in, is, is not), Booleans (and, or, not).
- If operators with different priorities are used, consider adding whitespace around the operators with the lowest priority(ies). Use your own judgment; however, never use more than one space, and always have the same amount of whitespace on both sides of a binary operator
- Function annotations should use the normal rules for colons and always have spaces around the -> arrow if present. (See Function Annotations below for more about function annotations.):
- Don't use spaces around the = sign when used to indicate a keyword argument, or when used to indicate a default value for an unannotated function parameter

Naming Conventions

- Python packages should have short, all-lowercase names, although the use of underscores is discouraged.
- Class names should normally use the CapWords convention.
- Function names should be lowercase, with words separated by underscores as necessary to improve readability
- Variable names follow the same convention as function names.