Style Guide – Python for Scientists

Use descriptive names, even if it increases line length slightly. count is more descriptive than c.

In general, avoid using single character variable names, since they are often difficult to follow and read. Never use the characters ‘l’ (lowercase letter el), ‘O’ (uppercase letter oh), ‘I’ (uppercase letter eye), ‘1’ (number one), or ‘0’ (number zero) as single character variable names. Avoid using ‘L’ (uppercase letter el) when possible. In some fonts, these characters are indistinguishable from the numerals one and zero.

# Capitalization Conventions

* Variables and objects: camelCase (count, numRuns)
* Functions: snake case (sum, sum\_of, get\_result)
* Classes: CapCase (GameScore, Runs)
* Constants: ALLCAPS (PI, FIELDLENGTH)

# Whitespace

Use whitespace wisely. Remember, whitespace takes the form of both horizontal whitespace (spaces and indentation) and vertical whitespace (blank lines). Both too much and too little whitespace make your source code difficult to read.

Leave one space around initializations and boolean operators.

runs = 1 # Good

if (runs >= 10): # Good

runs=3 # Bad

Observe how the equal sign in line 1 is surrounded by spaces. This is an example of space around initialization. Also observe how the greater than/equal to sign in line 2 is surrounded by spaces without a space between components of the boolean operator. This ensures that the syntax is correct for the entire boolean operator (the >= is one 1 unit, not a separate > and =) while still providing adequate whitespace. This is an example of space around a boolean operator. Also leave space before and after comment demarcations, as shown in lines 1-3. The comment demarcation in Python is a #, and there is a space before and after.

Leave an extra space between function arguments. Do not leave an extra space before or after function parentheses.

atlRuns = GetRuns('ATL') # Good

ariWins = GetWins('ARI', 'away') # Good

bosRuns = GetRuns( 'BOS' ) # Bad , too much space around args

chiWins = GetWins( 'CHI', 'home' ) # Bad , too much space around args

dalWins = GetWins('DAL','away') # Bad , no space between args

Indentation

In connection with whitespace, make sure you follow indentation conventions for your language. Python enforces indentation, so make sure you use consistent indentation.

Indent using one tab, which should indent two spaces. Indent anything nested, including function contents, logic statement bodies, loops, and nested objects (mainly arrays, lists, and dictionaries).

Do not put a space before a colon in a conditional or logic statement.

if (numRuns > 3): # Good

print("More than three runs!") # Good , two spaces of indentation

else : # Bad

print("Not more than three runs.") # Bad, inconsistent indentation (3 spaces)

Soft-wrap lines in your editor, not by manually splitting a line into multiple lines. Not everyone’s editor window size and font size is the same as yours.