Ansari Sara .

IT Tools & Pratices

DATE

⟹ PEP - 8 coding Practises in python :

PEP stands for python enhancement proposal. this document gives coding conventions for the python code comprising the standard library in the main python distribution.

code lay-out :

Indentation :

use 4 spaces per indentation level. continuation line should align wrapped element either vertically using pythons implicit line joining inside parentheses, bracket and braces, or using a hanging element, when using a hanging indent the following should be considered, there should be no argument on the first line and further identation should be used to clearly distinguish itself as a continualion line

```
    foo = long_function_name(var_one, var_two
                             var_three, var_four).
# More indentation included to distinguish
this form the rest def long_function_name(
                        var_one, var_two, var_three,
                        var_four):
                print(var_one)
# Hanging indents should add a level.
foo = long_function_name(
        var_one, var_two,
        var_three, var_four).
```

When the conditional part of an if-statement is long enough to require that it be written across multiple lines, its worth noting that the combination of a two character keyword, plus a single space, plus an opening parethesis creates a natural 4 space indent for the subsequent lines of the multiline conditional. This can produce a visual conflict with the idented suite of code nested inside the if-statement, which would also naturally be indented to 4 spaces. This PEP takes no explicit position on how to further visually distinguish such conditional lines from the nested suite inside the if-statement; Acceptable options in this situation include, but or not limited to:

```
# No extra identation
if (this_is_one_thing and
    that_is_another_thing):
    do_something()
# Add a comment, which will provide some
  distinction in
# editors supporting syntax highlighting
if (this_is_one_thing and
    that_is_another_thing):
    # since both conditions are true, we can frabnicate
    do_something()
# Add some extra indentation on the conditional
# continuation line
if (this_one_thing
        and that_is_another_thing):
    do_something()
```

The closing brace/bracket/parenthesis on multi-line constructs may either line up under the first non-whitespace character of the last line of list, as in :-

```
my list = [
    1, 2, 3
    4, 5, 6
    ]
result = some-function-that-takes-arguments(
    'a', 'b', 'c',
    'd', 'e', 'f',
    )
```

Tabs or Spaces :-
Spaces are the preferred indentation method. Tabs should be used solely to remain consistent with tabs. Python 3 disallow mixing the uses of tabs and spaces for identation.

Maximum line length:-
limit all lines to a maximum of 79 characters for flowing long blocks of text with fewer string should be limited to 72 characters.
Limiting the required editor window width makes it possible to have several files open side-by side, & works well when using code review tools that present the two version in adjacent columns.
Backslashes may still be appropriate at times. for example long, multiple with statement with statement cannot use implicit continuation, so backlashes are acceptable

Should a line break before or after a binary operator ?

for decades the recommended style was to break after binary operator. But this can hurt readability in two ways. the operator tend to get scattered across different columns on the screen, and each operator is moved away from its operand and onto the previous line, there, the eye has to do extra work to tell which items are added and which are subtracted.

No: operators sit far away from their operands.

```
income = (gross-wages +
          taxable-interest +
          (dividends - qualified-dividends) -
          ira-deduction -
          student-loan-interest)
```

To solve this readibility problem, mathematicians & their publisher follow the opposite convention. following the tradition from mathematics usually results in more readable code

Yes :

```
income = (gross-wages
          + taxable-interest
          + (dividends - qualified-dividends)
          - ira-deduction
          - student-loan interest.
```

In Python code, it is permissible to break before or after a binary operator, as long as the convention is consistent locally. for new code knuth style is

[illegible]

Blank lines:
Surround top-level function and class definition with two blank lines. Method definition inside a class are surrounded by a single blank line. Extra blank lines may be used to separate groups of related functions. Blank lines may be omitted between a bunch of related one-lines.
Use blank lines in functions, sparingly, to indicate logical sections

Source file Encoding:
Code in the core python distribution should always use.
UTF-8,
For python 3.0 and beyond, the following policy is prescribed for the standard library. All identifiers in the python standard library must use ASCII only identifiers, and should use English words wherever feasible. In addition, string literals as string ~~tit~~ translation of their names. Open source project with a global audience are encouraged to adopt a similar policy.