

Vaibhavi Nayak
FYIT-59

Acceptable options in this situation include, but are not limited to:

No extra indentation.

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 frobulate.
do something()

Add some extra indentation on the
conditional continuation line.

if (this is one thing
and that is another thing):
do something()

The closing brace / bracket / parenthesis on
multiline constructs may either line up under
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 code that is already indented with tabs.
 - Python disallows mixing tabs and spaces for indentation.
- Maximum line length.
 - limit all lines to maximum of 79 characters
 - For flowing long blocks of text with fewer structural restriction, the line length should be limited to 72 characters
 - Limiting the required editor window width makes it possible to have several files open side by side, and works well when using code review tools that present the two versions in adjacent columns.
 - The default wrapping in most tools disrupts the visual structure of the code, making it more difficult to understand. The limits are chosen to avoid wrapping in editor with the window width set to 80, even if

Vaishnavi Nayak
FYIT - 59

the tool places a marker glyph in the final column when wrapping lines.
Some web based tools may not offer dynamic line wrapping at all.

→ Some teams strongly prefer a longer line length. For code maintained exclusively or primarily by a team that can reach agreement on this issue, it is okay to increase the line length limit up to 99 characters, provided that comments & docstrings are still wrapped at 72 characters.

→ The Python Standard Library is conservative and requires limiting lines to 79 characters.

→ The preferred way of wrapping long lines is by using Python's implied line continuation inside parentheses, brackets and braces. Long lines can be broken over multiple lines by wrapping expressions in parentheses. These should be used in preference to using a backslash for line continuation.

→ Backslashes may still be appropriate at times. For example, long, multiple with-statement could not use implicit continuation before Python 3.10, so backslashes were acceptable for that case.

Vaishnavi Nayak
FYIT - 54

with open ('/path/to/some/file/you/
want/to/read') as file_1,
open ('/path/to/some/file/being/
written', 'w') as file_2:
file_2.write(file_1.read())

Another such case is with assert statement

Make sure to indent the continued line
appropriately.

* 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
operators tend to get scattered across
different columns on the screen, and
each operator is moved away from its
operand and onto the previous line.
Here, the eye has to do extra work
to tell which items are added and
which are subtracted:

Wrong:

operators sit far away from their operand
income = (gross - wages +
taxable - interest +

Vaishnavi. Nayak
FYIT-54

(dividends - qualified dividends) -
ira deduction -
student loan - interest)

To solve this readability problem, mathematicians and their publishers follow the opposite convention. Donald Knuth explains the traditional rule in his Computer and Typesetting series: "Although formulas within a paragraph always break after binary operations and relations, displayed formulas always break before binary operations."

Following the tradition from mathematics usually results in more readable code:
Correct:
easy to match operators with operands
 $\text{income} = (\text{gross} - \text{wages}$
+ taxable interest
+ (dividends - qualified dividends)
- ira deduction
- student loan - interest)

In Python code, it is permissible to break or after a binary operator, as long as the convention is consistent locally. For new code Knuth's style is suggested.