

Vaishnavi. Nayak

FYIT. 59

IT - Tools Assignment

* Introduction

This document gives coding conventions for the Python code comprising the standard library in the main Python distribution. Please see the companion informational PEP describing style guidelines for the C code in the C implementation of Python.

This document and PEP 257 (Docstring Conventions) were adapted from Guido's original Python Style Guide essay, with some additions from Barry's style guide.

This style guide evolves over time as additional conventions are identified and past conventions are rendered obsolete by changes in the language itself.

Many projects have their own coding style guidelines. In the event of any conflicts, such project-specific guides take precedence for the project.

Vaishnavi Nayak

FYIT-59

A Foolish Consistency is the Hobgoblin of Little Minds.

One of Guido's key insights is that code is read much more often than it is written. The guidelines provided here are intended to improve the readability of code and make it consistent across the wide spectrum of Python code. As PEP 20 says, "Readability counts."

A style guide is about consistency. Consistency with this style guide is important. Consistency within a project is more important. Consistency within one module or function is the most important.

However, know when to be inconsistent. Sometimes style guide recommendations just aren't applicable. When in doubt, use your best judgement.

In particular: do not break backward compatibility just to comply with PEP!

Some other good reasons to ignore a particular guideline:

1. When applying the guideline would make the code less readable, even for someone who is used to reading code that follows this PEP.
2. To be consistent with surrounding code that also breaks it -- although this is also an opportunity to clean up someone else's mess.
3. Because the code in question predates the introduction of the guideline and there is no other reason to be modifying that code.
4. When the code needs to remain compatible with older versions of Python that don't support the feature recommended by the style guide.

* Code lay-out.

* Indentation

Use 4 spaces per indentation level.

Continuation lines should align wrapped elements either vertically using Python's implicit line joining inside parentheses, brackets and braces, or using a hanging indent. When using a hanging indent the following should be considered; these

Vaishnavi Nayak
FYIT-59

Should be no arguments on the first line and further indentation should be used to clearly distinguish itself as a continuation line:

Correct :

Aligned with opening delimiter.
foo = long_function_name(var_one,
var_two, var_three, var_four.)
Add 4 spaces (an extra level of
indentation) to distinguish arguments
from 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)

Wrong :

Arguments on first line forbidden
when not using vertical alignment
foo = long_function_name(var_one,
var_two, var_three, var_four)

Further indentation required as
indentation is not distinguishable

Vaishnavi Nayak
FYIT-54

```
def long_function_name(  
    var_one, var_two, var_three,  
    var_four):  
    print(var_one)
```

The 4-space rule is optional for continuation lines.

Optional:

Hanging indents *may* be indented to other than 4 spaces.

```
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, it's worth noting that the combination of a two character keyword, plus a single space, plus an opening parenthesis creates a natural 4-space indent for the subsequent lines of multiple conditionals. This can produce a visual conflict with indented suite of code nested inside 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.