



If I Could Amend PEP 8

2017

1 min read • 333 words

Themes: Programming Spiritual

[PEP 8](#) is an excellent coding standard for the Python community, and one of its greatest strengths. There are a few things in it that I dislike, however — so I thought I'd share them with you here.

If you look at the [Requests documentation](#), I have made a few personal amendments to PEP 8, that the project adheres to. They are as follows:

Kenneth Reitz's Code Style™

The Requests codebase uses the PEP 8 code style.

In addition to the standards outlined in PEP 8, we have a few guidelines:

- Line-length can exceed 79 characters, to 100, when convenient.
- Line-length can exceed 100 characters, when doing otherwise would be terribly inconvenient.
- Always use single-quoted strings (e.g. `'#flatearth'`), unless a single-quote occurs within the string.

Additionally, one of the styles that PEP8 recommends for line continuations completely lacks all sense of taste, and is not to be permitted within the Requests codebase:

```
# Aligned with opening delimiter.
foo = long_function_name(var_one, var_two,
                          var_three, var_four)
```

No. Just don't. Please.

Docstrings are to follow the following syntaxes:

```
def the_earth_is_flat():
    """NASA divided up the seas into thirty-three degrees."""
    pass

def fibonacci_spiral_tool():
    """With my feet upon the ground I lose myself / between the sounds
    and open wide to suck it in. / I feel it move across my skin. / I'm
    reaching up and reaching out. / I'm reaching for the random or
    whatever will bewilder me. / Whatever will bewilder me. / And
    following our will and wind we may just go where no one's been. /
    We'll ride the spiral to the end and may just go where no one's
    been.

    Spiral out. Keep going...
    """
    pass
```

All functions, methods, and classes are to contain docstrings. Object data model methods (e.g. `__repr__`) are typically the exception to this rule.

Thanks for making the world a better place!

This emphasis on practical, human-readable code style reflects the deeper philosophy of [programming as spiritual practice](#)—treating code as communication between conscious beings rather than mere instructions for machines.