

Any Formatter You
Like, As Long As It's
Black



Mika Naylor
@autophagian



Why Use Code Styles?

The background of the slide features a sunburst or radial pattern. It consists of numerous thin, dark gray lines radiating outwards from a central point, creating a starburst effect. The lines are evenly spaced and cover the entire area of the slide.

PEP 8

Style Guide For Python Code



The PSF
The Python Software Foundation
is the organization behind
Python. Become a member of the
PSF and help advance the soft-
ware and our mission.

PEP 8 -- Style Guide for Python Code

PEP:	8
Title:	Style Guide for Python Code
Author:	Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>
Status:	Active
Type:	Process
Created:	05-Jul-2001
Post-History:	05-Jul-2001, 01-Aug-2013

Contents

- [Introduction](#)
- [A Foolish Consistency is the Hobgoblin of Little Minds](#)
- [Code Lay-out](#)
 - [Indentation](#)
 - [Tabs or Spaces?](#)
 - [Maximum Line Length](#)
 - [Should a Line Break Before or After a Binary Operator?](#)
 - [Blank Lines](#)
 - [Source File Encoding](#)
 - [Imports](#)
 - [Module Level Dunder Names](#)
- [String Quotes](#)
- [Whitespace in Expressions and Statements](#)
 - [Pet Peeves](#)
 - [Other Recommendations](#)
- [When to Use Trailing Commas](#)
- [Comments](#)
 - [Block Comments](#)
 - [Inline Comments](#)
 - [Documentation Strings](#)
- [Naming Conventions](#)
 - [Overriding Principle](#)
 - [Descriptive: Naming Styles](#)
 - [Prescriptive: Naming Conventions](#)

<https://www.python.org/dev/peps/pep-0008/>

*“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.”*



No.

“Any formatter you like, as long as it’s Black.”

Horizontal Whitespace

Vertical Whitespace

Line Length

Consistency

Horizontal Whitespace

Vertical Whitespace

Line Length

Consistency

Horizontal Whitespace

Vertical Whitespace

Line Length

Consistency

```
def a_nice_function(param_a: str, param_b: dict, path: os.PathLike,  
verbose: bool = False):  
    with open(path, "w") as file:  
        pass
```

```
def a_nice_function(  
    param_a: str,  
    param_b: dict,  
    path: os.PathLike,  
    verbose: bool = False,  
):  
    with open(path, "w") as file:  
        pass
```

After Blackening

```
if (  
    this_long_variable_is_true  
    and this_other_variable == 10  
    or this_other_variable > 100  
    and this_function_call() < 100  
):  
    pass
```

After Blackening

Horizontal Whitespace

Vertical Whitespace

Line Length

Consistency

$$79 < ?? < 120$$

$$79 < \mathbf{88} < 120$$

Horizontal Whitespace

Vertical Whitespace

Line Length

Consistency

```
my_list = [  
    var_1,  
    var_2,  
    var_3,  
    var_4,  
    var_5,  
    var_6,  
    var_7,  
]
```

After Blackening

```
my_list = [  
    var_1,  
    var_2,  
    var_3,  
    var_4,  
    var_5,  
    var_6,  
    var_7,  
    var_8,  
]
```

After Blackening

```
my_list = [  
    var_1,  
    var_2,  
    var_3,  
    var_4,  
    var_5,  
    var_6,  
    var_7,  
    var_7,  
    var_8  
]
```

Before Blackening

```
my_var = f"Hello world! I am {user}"
```

After Blackening

```
my_var = f"Hello world! I'm {user}"
```

After Blackening

Horizontal Whitespace

Vertical Whitespace

Line Length

Consistency


```
pip install black
```

```
black my_file.py
```

```
black my_files/
```

```
black --check my_file.py
```

```
black --diff my_file.py
```



8,940

<https://github.com/ambv/black>

“I’ve used Black extensively on several projects, and much like f-strings, the last Pink Floyd album, and broccoli, have found I really like something I didn’t think I would.”



Twitter

@autophagian

Github

autophagy

Email

mail@autophagy.io