<https://pep8.org/>

"""This is the example module.

This module does stuff.

"""

from \_\_future\_\_ import barry\_as\_FLUFL

\_\_all\_\_ = ['a', 'b', 'c']

\_\_version\_\_ = '0.1'

\_\_author\_\_ = 'Cardinal Biggles'

import os

import sys

If operators with different priorities are used, consider adding whitespace around the operators with the lowest priority(ies). Use your own judgment; however, never use more than one space, and always have the same amount of whitespace on both sides of a binary operator.

Yes:

i = i + 1

submitted += 1

x = x\*2 - 1

hypot2 = x\*x + y\*y

c = (a+b) \* (a-b)

When combining an argument annotation with a default value, use spaces around the **=** sign (but only for those arguments that have both an annotation and a default).

Yes:

def munge(sep: AnyStr = None): ...

def munge(input: AnyStr, sep: AnyStr = None, limit=1000): ...

Yes:

try:

value = collection[key]

except KeyError:

return key\_not\_found(key)

else:

return handle\_value(value)

No:

try:

*# Too broad!*

return handle\_value(collection[key])

except KeyError:

*# Will also catch KeyError raised by handle\_value()*

return key\_not\_found(key)

Yes:

def foo(x):

if x >= 0:

return math.sqrt(x)

else:

return None

def bar(x):

if x < 0:

return None

return math.sqrt(x)

No:

def foo(x):

if x >= 0:

return math.sqrt(x)

def bar(x):

if x < 0:

return

return math.sqrt(x)

Don’t compare boolean values to True or False using **==**:

Yes:

if greeting:

No:

if greeting == True:

Worse:

if greeting is True:

<https://docs.python.org/3/whatsnew/3.9.html>

what’s new:

>>> d = {'spam': 1, 'eggs': 2, 'cheese': 3}

>>> e = {'cheese': 'cheddar', 'aardvark': 'Ethel'}

>>> d | e

{'spam': 1, 'eggs': 2, 'cheese': 'cheddar', 'aardvark': 'Ethel'}

>>> e | d

{'cheese': 3, 'aardvark': 'Ethel', 'spam': 1, 'eggs': 2}

>>> list[str]

list[str]

>>> tuple[int, ...]

tuple[int, ...]

>>> ChainMap[str, list[str]]

collections.ChainMap[str, list[str]]

* Current:
* if test\_func\_name.startswith("test\_"):
* print(test\_func\_name[5:])
* else:
* print(test\_func\_name)
* Improved:

print(test\_func\_name.removeprefix("test\_"))