Naming Conventions

Variable Names:

We will be following Camel Casing convention for the variable names.

Example: newVariable.

Constants:

Constants are usually defined on a module level and written in all capital letters with underscores separating words.

Example: CONSTANT, NEW_CONS.

Method Names:

Method names should be in lower case. For multiple words, it should be in lower case with underscores.

Example: def new_method().

Package and Module Names:

Package or module names should be in lower case and short. For multiple words, variable should be in lower case with underscores.

Example: new_module.

Class Name:

We will be following **Pascal Casing** convention for the class name.

Example: class NewClass.

Exception Names:

As exceptions should be classes, the class naming convention applies here. However, the suffix "Error" should be used on the exception names (if the exception actually is an error).

Names to Avoid:

Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names. In some fonts, these characters are indistinguishable from the numerals one and zero. When tempted to use 'l', use 'L' instead.

Code Layout

Indentation:

We will use 4 spaces per indentation level.

Example:

```
if True:
    print("If works")
```

- When you write a big expression, it is best to keep the expression vertically aligned. When you do this, you'll create a "hanging indent".
  - Aligned with opening delimiter.

    Example:

```
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.

  Example:

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

Line Break:

Line Break should be before operator.

Example:

```
income = (gross_wages
          + taxable_interest
          + (dividends - qualified_dividends)
          - ira_deduction
          - student_loan_interest)
```

Blank Lines:

In Python scripts, top-level function and classes are separated by two blank lines. Method definitions inside classes should be separated by one blank line. You can see this clearly in the following example:

```
class SwapTestSuite(unittest.TestCase):
    """
        Swap Operation Test Case
    """
    def setUp(self):
        self.a = 1
        self.b = 2

    def test_swap_operations(self):
        instance = Swap(self.a,self.b)
        value1, value2 =instance.get_swap_values()
        self.assertEqual(self.a, value2)
        self.assertEqual(self.b, value1)


class OddOrEvenTestSuite(unittest.TestCase):
    """
```

Whitespaces in Expressions and Statements

Pet Peeves: Avoid extraneous whitespace in the following situations:

- Immediately inside parentheses, brackets or braces:

  Correct Example: spam(ham[1], {eggs: 2})

- Between a trailing comma and a following close parenthesis:

  Correct Example: foo = (0,)

- Immediately before a comma, semicolon, or colon:

  Correct Example: if x == 4: print x, y; x, y = y, x

- Immediately before the open parenthesis that starts the argument list of a function call:

  Correct Example: spam(1)

- Immediately before the open parenthesis that starts an indexing or slicing:

  Correct Example: dct['key'] = lst[index]

- More than one space around an assignment (or other) operator to align it with another:

  Correct Example:

  x = 1

  y = 2

  long_variable = 3

Using Recommendations:

- Always surround these binary operators with a single space on either side: assignment (=), augmented assignment (+=, -= etc.), comparisons (==, <, >, !=, <>, <=, >=, in, not in, is, is not), Booleans (and, or, not).

  Example:

  i = i + 1

  sum += 1

- If operators with different priorities are used, consider adding whitespace around the operators with the lowest priority(ies).

  Example:

  i = x*x + y*y

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

- To Import

  Example:

  import os

  Import sys

  or

  from config import settings

Comments

Comments start with the # symbol. Anything written after the hashtag does not get executed by the interpreter. Each line of a block comment begins with the hashtag # and a single space. If you need to use more than one paragraph, they should be separated by a line that contains a single #.

Example: # This is a comment