

Python PEP8 Style Guide

Table of contents

Variable and function names	1
Spacing	1
Breaking long lines	2
Comments	2

A **PEP** (Python Enhancement Proposal) is a design document describing conventions for how to style code. This course follows Google's PEP, a variant of **PEP8**.

Variable and function names

Names must start with a letter (not a digit) and can otherwise only contain letters, digits, and underscores (_). Names should be lowercase, with words separated by underscores for readability:

Yes	No
<code>descriptive_variable_name</code>	<code>DescriptiveVariableName</code>

Spacing

Put single spaces around binary operators; don't pad the inside of brackets:

Yes	No
<code>i = i + 1</code>	<code>i=i+1</code>
<code>hypot2 = x*x + y*y</code>	<code>hypot2 = x * x + y * y</code>
<code>c = (a+b) * (a-b)</code>	<code>c = (a + b) * (a - b)</code>

Breaking long lines

All lines should be strictly less than 80 characters wide. Wrap a long expression in a redundant enclosing bracket so it can be split across multiple lines (otherwise pressing enter would evaluate the expression early):

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

Comments

Anything following a `#` is ignored by Python. Comments (and docstrings, see [python-functions](#)¹) should explain *why* something not obvious was done, not simply restate what the code already shows:

```
x = x + 1    # Not helpful: Increment x
x = x + 1    # Helpful: Compensate for border
```

¹[courses/csse1001/python-functions.pdf](#)