

Tabs or spaces?

Indenting for code readability

Coding conventions¹

- Erik - Naming conventions
- Now - Indentation
- Poll
 - Tabs or spaces?
 - Tab size?

1. Convention: *A practice or procedure widely observed in a group, especially to facilitate social interaction; a custom.*

- Wiktionary

6	Language-specific conventions
6.1	ActionScript
6.2	Ada
6.3	APL
6.4	C and C++
6.5	C#
6.6	Go
6.7	Java
6.8	JavaScript
6.9	Lisp
6.10	.NET
6.11	Objective-C
6.12	Pascal, Modula-2 and Oberon
6.13	Perl
6.14	PHP
6.15	Python and Ruby
6.16	R
6.17	Raku
6.18	Rust
6.19	Swift

SAS...?

Poll

Tabs or spaces?

1. Tabs
2. Spaces
3. I write everything on a single line – unindented

https://forms.office.com/Pages/ResponsePage.aspx?id=XE3BsSU2s0WkMJVSNzoML-PZMtVZiNRCmITT_6ZQO0VUREtZWUtIQlk3R1IXQIU2TFNGM1hNUTg1VC4u

Poll results

Tabs or spaces?

1. Tabs **7**
2. Spaces **6**
3. I write everything on a single line – unindented **0**

https://forms.office.com/Pages/ResponsePage.aspx?id=XE3BsSU2s0WkMJVSNzoML-PZMtVZiNRCmITT_6ZQO0VUREtZWUtIQlk3R1IXQIU2TFNGM1hNUTg1VC4u

Poll

Tab size?

- 2
- 4
- 6
- 8
- Other

https://forms.office.com/Pages/ResponsePage.aspx?id=XE3BsSU2s0WkMJVSNzoML-PZMtVZiNRCmITT_6ZQO0VUOFIwRExVNFRTWINXMFE0NVQ2T0dGNVBGQi4u

Poll results

Tab size?

- 2 **3**
- 4 **5**
- 6 **1**
- 8
- Other **2** (*that use a size of 3*)

https://forms.office.com/Pages/ResponsePage.aspx?id=XE3BsSU2s0WkMJVSNzoML-PZMtVZiNRCmITT_6ZQO0VUOFIwRExVNFRTWINXMFE0NVQ2T0dGNVBGQi4u

My experience – Coding conventions

- 2019: New team, mostly new code
- Agreed to use **PEP8**
- “The boy scout rule”

PEP8 - Coding conventions

- PEP "Python Enhancement Proposal"
- PEP8 "Style Guide for Python Code"



6	Language-specific conventions
6.1	ActionScript
6.2	Ada
6.3	APL
6.4	C and C++
6.5	C#
6.6	Go
6.7	Java
6.8	JavaScript
6.9	Lisp
6.10	.NET
6.11	Objective-C
6.12	Pascal, Modula-2 and Oberon
6.13	Perl
6.14	PHP
6.15	Python and Ruby
6.16	R
6.17	Raku
6.18	Rust
6.19	Swift

PEP8

A Foolish Consistency is the Hobgoblin of Little Minds

One of Guido's key insights is that code is read much more often than it is written. The guidelines provided here are intended to improve the readability of code and make it consistent across the wide spectrum of Python code. As [PEP 20](#) says, "Readability counts".

- *Code is read much more often than it is written*
- *Readability counts*
- *Foolish consistency*

PEP8 contents

Indentation →

Naming conventions →

- [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](#)
 - [Names to Avoid](#)
 - [ASCII Compatibility](#)
 - [Package and Module Names](#)
 - [Class Names](#)
 - [Type Variable Names](#)
 - [Exception Names](#)
 - [Global Variable Names](#)
 - [Function and Variable Names](#)
 - [Function and Method Arguments](#)
 - [Method Names and Instance Variables](#)
 - [Constants](#)
 - [Designing for Inheritance](#)
 - [Public and Internal Interfaces](#)
- [Programming Recommendations](#)
 - [Function Annotations](#)
 - [Variable Annotations](#)
- [References](#)
- [Copyright](#)

PEP8 Indentation

“Correct”

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

- *Aligned*

```
# Hanging indents should add a level.  
foo = long_function_name(  
    var_one, var_two,  
    var_three, var_four)
```

- *Hanging*

```
# Add 4 spaces (an extra level of indentation) to distinguish arguments from the rest.  
def long_function_name(  
    var_one, var_two, var_three,  
    var_four):  
    print(var_one)
```

“Wrong”

```
# Arguments on first line forbidden when not using vertical alignment.  
foo = long_function_name(var_one, var_two,  
                          var_three, var_four)
```

- *Vilda västern*

```
# Further indentation required as indentation is not distinguishable.  
def long_function_name(  
    var_one, var_two, var_three,  
    var_four):  
    print(var_one)
```

Poll

How consistent are you with your indentation?

1. I always use X (aligned, hanging or other)
2. I use different ones, depending on the case
3. I don't have a preference
4. I haven't thought about it until now
5. I use tabs on even lines and spaces on odd lines

https://forms.office.com/Pages/ResponsePage.aspx?id=XE3BsSU2s0WkMJVSNzoML-PZMtVZiNRCmITT_6ZQO0VUOEoyVFdBTDhTUTRCM0xGQjQzRVQwVIJRTS4u

Poll results

How consistent are you with your indentation?

1. I always use X (aligned, hanging or other) **5**
2. I use different ones, depending on the case **8**
3. I don't have a preference **2**
4. I haven't thought about it until now **0**
5. I use tabs on even lines and spaces on odd lines **0**

https://forms.office.com/Pages/ResponsePage.aspx?id=XE3BsSU2s0WkMJVSNzoML-PZMtVZiNRCmITT_6ZQO0VUOEoyVFdBTDhTUTRCM0xGQjQzRVQwVIJRTS4u

The end