

Why does Python pep-8 strongly recommend spaces over tabs for indentation? [closed]

Asked 16 years, 3 months ago Modified 1 year, 1 month ago

Viewed 118k times



208



Closed. This question is [opinion-based](#). It is not currently accepting answers.



Want to improve this question? Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 3 years ago.

[Improve this question](#)

I see on Stack Overflow and [PEP 8](#) that the recommendation is to use spaces only for indentation in Python programs. I can understand the need for consistent indentation and I have felt that pain.

Is there an underlying reason for spaces to be preferred? I would have thought that tabs were far easier to work with.

[python](#)[indentation](#)[pep8](#)[Share](#)[Improve this question](#)[Follow](#)

edited Sep 15, 2021 at 16:06



TylerH

21.2k ● 76 ● 79 ● 110

asked Sep 23, 2008 at 13:20



quamrana

39.3k ● 13 ● 55 ● 76

-
- 11 i think the subjective tag is (for once) wrong. the question asks for the reasoning or reason behind the decision in the canoncial text about the subject! – user3850 Oct 10, 2008 at 13:49
-
- 209 1 level of indentation is... 1. It is completely illogical to have to agree on using N spaces when you could all use a single tab. Which, by the way, is exactly meant to do that. Indent. Once. 1 level of indentation = 1 single character, i.e. 1 single tab. And they are more handy because each coder may chose freely how to visualize it. Using spaces is dumb, I've never ever seen a single argument for it which is non stupid. – o0'. Jan 19, 2012 at 14:53
-
- 61 @BlueBomber and why don't you force people to have a font size and a color scheme you like, while you are at it? Still stupid. – o0'. Feb 22, 2013 at 22:20
-
- 33 @BlueBomber no, no, it is EXACTLY on the same level of absurdness. – o0'. Feb 22, 2013 at 22:57
-
- 33 @BlueBomber What exactly is the difference? You're reducing a degree of freedom in another developer's configuration of their environment without achieving any noticeable benefit. If you want to be a dictator and force

everyone to look at the code with an indent corresponding to 2 or 4 or 29 spaces you can still do this with tabs. Just ask your underlings to set their IDE to display tabs as corresponding to your preferred number of spaces. If you don't have the authority to do this, maybe you should let them decide for themselves how wide a unit of indentation is comfortable to their eyes. – [Asad Saeeduddin](#) Sep 23, 2014 at 23:44

18 Answers

Sorted by:

Highest score (default)



253

Well well, seems like everybody is strongly biased towards spaces. I use tabs exclusively. I know very well why.



Tabs are actually a cool invention, that came **after** spaces. It allows you to indent without pushing space millions of times or using a fake tab (that produces spaces).



I really don't get why everybody is discriminating against the use of tabs. It is very much like old people discriminating against younger people for choosing a newer more efficient technology and complaining that pulse dialing works on **every phone**, not just on these fancy new ones. "Tone dialing doesn't work on every phone, that's why it is wrong".

Your editor cannot handle tabs properly? Well, get a **modern** editor. Might be darn time, we are now in the 21st century and the time when an editor was a high tech complicated piece of software is long past. We have now

tons and tons of editors to choose from, all of them that support tabs just fine. Also, you can define how much a tab should be, a thing that you cannot do with spaces. Cannot see tabs? What is that for an argument? Well, you cannot see spaces neither!

May I be so bold to suggest to get a better editor? One of these high tech ones, that were released some 10 years ago already, that **display invisible characters**?
(sarcasm off)

Using spaces causes a lot more deleting and formatting work. That is why (and all other people that know this and agree with me) use tabs for Python.

Mixing tabs and spaces is a no-no and no argument about that. That is a mess and can never work.

Share Improve this answer

Follow

edited Nov 2, 2023 at 7:20



Josh Correia

4,318 ● 3 ● 40 ● 61

answered Feb 19, 2011 at 0:56



nigratruo

2,637 ● 1 ● 15 ● 2

62 To add to that, take a look at your keyboard, the TAB key's symbol clearly portrays indentation - it's the key's indented purpose not SPACE's. PEP8 recommends to use spaces is a mistake IMHO but it is just a recommendation -

en.wikipedia.org/wiki/Tab_character#Tab_characters

– Daniel Sokolowski May 23, 2013 at 14:21

57 Agreed with this post entirely. Using spaces is for fools who enjoy getting tripped up on those 1 space off indentation errors. If your indentation was off by 1 tab, I guarantee you'd notice it. – [Sepero](#) Jun 6, 2013 at 10:37

26 @Zingham *Nobody can use tabs exclusively: tabs and spaces are always used in combination and this leads to inconsistency.* I and thousands of others are using them quite consistently every day. Tabs for indentation, spaces for alignment. Exactly which part of this concept do you find so terribly hard to grasp, and why are you convinced that it is impossible to apply it consistently? – [antred](#) Jan 22, 2016 at 14:11

1 The actual problem with tabs is that you cannot get exact indendation to-the-character recommended by the same PEP8 under the comment "# Aligned with opening delimiter." That only is the reason to prefer spaces: to get the indentation correct! – [user541905](#) Feb 6, 2017 at 10:17

16 The question is "**Why does Python pep-8 strongly recommend spaces over tabs for indentation?**". This answer does not ever mention anything about the PEP8. ||| Instead of trying to answer the question... this answer comes across to me as one huge condescending **primarily opinion** piece. There are 276 words used to say "tabs are better than spaces and here's why...". – [Trevor Boyd Smith](#) Sep 25, 2018 at 18:49



The answer was given right there in the PEP [ed: this passage has been edited out in [2013](#)]. I quote:

153



The **most popular** way of indenting Python is with spaces only.



What other underlying reason do you need?



To put it less bluntly: Consider also the scope of the PEP as stated in the very first paragraph:



This document gives coding conventions for the Python code comprising the standard library in the main Python distribution.

The intention is to make *all code that goes in the official python distribution* consistently formatted (I hope we can agree that this is universally a Good Thing™).

Since the decision between spaces and tabs for an individual programmer is a) really a matter of taste and b) easily dealt with by technical means (editors, conversion scripts, etc.), there is a clear way to end all discussion: choose one.

Guido was the one to choose. He didn't even have to give a reason, but he still did by referring to empirical data.

For all other purposes you can either take this PEP as a recommendation, or you can ignore it -- your choice, or your team's, or your team leaders.

But if I may give you one advice: don't mix'em ;-) [ed: Mixing tabs and spaces is no longer an option.]

Share Improve this answer

edited Jul 18, 2017 at 10:58

Follow

answered Sep 23, 2008 at 15:19



user3850

-
- 14 Agreed. Consistency is more important than tabs vs. X spaces vs. Y spaces. – [Mike Clark](#) Dec 10, 2010 at 6:20
-
- 23 makes me wonder... why does the standard library have so many mixedCase method names? – [Kyle Wild](#) Feb 23, 2011 at 19:22
-
- 7 @dorkitude: a) nobody is perfect. b) historical reasons. – user3850 May 3, 2012 at 14:55
-
- 12 So then why did so many programmers choose to use spaces before PEP-8? That's what I really want to know. The advantages of tabs seem obvious to me, but not spaces. – [einnocent](#) Mar 1, 2014 at 0:40
-
- 24 "My recommendation is to always use tabs on the Mac -- then it will look good on the Mac and at least parse correctly everywhere. For the same reason I recommend always using tabs on Unix as well (thus indenting by 8 positions there), but the majority of Python users seem to be against me."—Guido van Rossum, 1994 – [noumenal](#) Aug 12, 2016 at 12:53
-



44



~~I personally don't agree with spaces over tabs. To me, tabs are a document layout character/mechanism while spaces are for content or delineation between commands in the case of code.~~

I have to agree with Jim's comments that tabs aren't really the issue, it is people and how they want to mix



tabs and spaces.

That said, I've forced myself to use spaces for the sake of convention. I value consistency over personal preference.

Edit 2022: Use spaces. Follow the language conventions and those set in the particular project you're working on. Use a linter to ensure those conventions are maintained. Format on save. Lint on commit. This will reduce "bikeshedding" on a team. As I mentioned so many years ago, **consistency over personal preference!**

Bikeshedding, also known as Parkinson's law of triviality, describes our tendency to devote a disproportionate amount of our time to menial and trivial matters while leaving important matters unattended.

Share Improve this answer

edited Jun 1, 2022 at 15:18


Follow

answered May 16, 2009 at 17:41



Soviut

91.4k ● 53 ● 205 ● 279

5 I tried to force myself to use spaces too, but editor (at least Eclipse + PyDev) wise tabs wins especially if you enable show invisible characters. And I can easily set tabs to be 4, 8, 6 spaces visually. So in my code at least I value personal preference, and stick to spaces if that is the established convention in existing code base. – [Daniel Sokolowski](#) May 16, 2013 at 18:29 

3 That's fine as long as you're not coding in a team. Once you're in a team, you agree upon a single convention and stick to it. – [Soviut](#) May 16, 2013 at 20:22

1 @Soviut That seems mostly like wishful thinking to me. In all the teams I've ever been in, the official party line was "use spaces". The reality was that virtually every file was a total mess of both tabs and spaces, and even in files that consistently used only spaces or only tabs, the indentation was still all over the place. – [antred](#) Jun 16, 2015 at 21:40

Yeah but that's what I meant. Some consensus is eventually reached and enforced. Like you said, it's usually "just use spaces". – [Soviut](#) Jun 18, 2015 at 4:52

1 this is the main reason why I prefer tabs. It just makes sense to have separate characters for layout and word delimiting – [woojoo666](#) Jun 28, 2016 at 4:33



32



The main problems with indentation occur when you mix tabs and spaces. Obviously this doesn't tell you which you should choose, but it is a good reason to to recommend one, even if you pick it by flipping a coin.

However, IMHO there are a few minor reasons to favour spaces over tabs:



- Different tools. Sometimes code gets displayed outside of a programmer's editor. Eg. posted to a newsgroup or forum. Spaces generally do better than tabs here - everywhere spaces would get mangled, tabs do as well, but not vice-versa.
- Programmers see the source differently. This is deeply subjective - its either the main benefit of tabs, or a reason to avoid them depending on which side you're on. On the plus side, developers can view the source with their preferred indentation, so a developer preferring 2-space indent can work with an 8-space developer on the same source and still see it as they like. The downside is that there are repercussions to this - some people like 8-space because it gives very visible feedback that they're too deeply nested - they may see code checked in by the 2-indenter constantly wrapping in their editor. Having every developer see the code the same way leads to more consistency wrt line lengths, and other matters too.
- Continued line indentation. Sometimes you want to indent a line to indicate it is carried from the previous one. eg.

```
def foo():  
    x = some_function_with_lots_of_args(foo, bar,  
                                         xzyzy, bla
```

If using tabs, theres no way to align this for people using different tabstops in their editor without mixing

spaces and tabs. This effectively kills the above benefit.

Obviously though, this is a deeply religious issue, which programming is plagued with. The most important issue is that we should choose one - even if that's not the one you favour. Sometimes I think that the biggest advantage of significant indentation is that at least we're spared brace placement flamewars.

Also worth reading is [this](#) article by Jamie Zawinski on the issue.

Share Improve this answer

answered Sep 27, 2008 at 17:32

Follow



Brian


119k ● 29 ● 110 ● 114

4 The alignment is trivial though. I simply use the brackets like a block and indent each arg. Also, in your example you very well can use spaces since you're inside an argument list and you can stack as many spaces in there as you want. – [Soviut](#) May 16, 2009 at 17:44

5 @Soviut: If you indent with spaces, the alignment is messed up as soon as it's viewed with a different tab size. The only way to preserve it is to use tabs to the indent level, and then spaces for the rest - ie mix spaces and tabs, which leads to its own problems. – [Brian](#) May 16, 2009 at 20:19

1 Yeah, which is why i tend to just use the python convention of block indenting on my arguments anyways. Sure, they may not line up with the open brace, but it's still clear which line or command they belong to. JQuery syntax operates on a similar principle. – [Soviut](#) May 17, 2009 at 4:20

5 @Brian: I don't see how that leads to any problems. It's exactly the right way, tabs for indentation, spaces for alignment. It's not at all the same as mixing spaces and tabs for **indentation**. – [antred](#) Feb 4, 2015 at 17:28

5 @CoreDumpError Uh no, it certainly doesn't. I know that because Python 3 never complains about any of my scripts, and I use tabs for indentation / spaces to align all the damned time. Also, PEP8 cannot "forbid" anything, as it's merely a recommendation (and a harebrained one, in my opinion). – [antred](#) Apr 27, 2016 at 23:41 



32

The reason for spaces is that tabs are optional. Spaces are the actual lowest-common denominator in punctuation.



Every decent text editor has a "replace tabs with spaces" and many people use this. But not always.



While some text editors might replace a run of spaces with a tab, this is really rare.

Bottom Line. You can't go wrong with spaces. You *might* go wrong with tabs. So don't use tabs and reduce the risk of mistakes.

Share Improve this answer

[edited Jun 3, 2010 at 14:49](#)

Follow

answered Sep 23, 2008 at 14:45



[S.Lott](#)

391k ● 82 ● 517 ● 788

28 I would never vow to do something the wrong way just because a lot of others (people, text editors, etc.) also happen to be it the wrong way. In 2015, a text editor that doesn't handle tabs well belongs in the garbage can.
– [antred](#) Nov 8, 2015 at 15:13

12 "You can't go wrong with spaces. You might go wrong with tabs". I've found that to be 100% topsy-turvy incorrect. In my experience: "You can't go wrong with Tabs. You might go wrong with spaces" ... especially when sharing code.
– [cmroanirgo](#) Apr 11, 2017 at 6:59

13 So someone has finally said it : use spaces because it's the lowest common denominator. The same rule has gotten us to hold on to MBR, BIOS, and paper forms at government offices. Except that these actually have conceptual problems, while tabs vs spaces is a 100% stupid user problem.
– [Milind R](#) Jun 18, 2017 at 9:38

10 This seems to me a Argumentum ad populum: fallacious argument that concludes that a proposition is true because many or most people believe it. Because every editor can replace tabs with spaces then spaces are the right choice is a fallacy!! – [Djunzu](#) Oct 31, 2017 at 15:02



31



The problem with tabs is that they are invisible, and people can never agree on the width of tabs. When you mix tabs and spaces, and you set tabstops at something other than Python (which uses tabstops every 8 spaces) you will be seeing the code in a different layout than Python sees it. And because the layout determines blocks, you will be seeing different logic. It leads to subtle bugs.

If you insist on defying PEP 8 and using tabs -- or worse, mixing tabs and spaces -- at least always run python with the '-tt' argument, which makes *inconsistent* indentation (sometimes a tab, sometimes a space for the same indentation level) an error. Also, if possible, set your editor to display tabs differently. But really, the best approach is not to use tabs, period.

Share Improve this answer

answered Sep 23, 2008 at 13:24

Follow



Thomas Wouters

133k ● 23 ● 152 ● 123

61 It's true that tabs are invisible and people can't agree on the width of tabs. But the same is also true for spaces. When you mix tabs and spaces, things go wrong. But why are you blaming that situation on tabs and not spaces? – Jim Sep 23, 2008 at 14:01

51 No, the same is not true for spaces. People can agree on the width of spaces. – Rafał Dowgird Sep 23, 2008 at 14:49

35 A single space may always be the same width, but indentation with spaces is not always the same width. I fail to see how agreeing to use tabs n spaces wide is any different to agreeing to indent with n spaces. – Jim Sep 23, 2008 at 14:57

32 Yes, I know the problems mixing the two can cause. What I don't understand is why some people blame that on tabs. The problem is mixing them, not tabs in particular. You could solve the problem by replacing tabs with spaces, but you could also solve the problem by replacing spaces with tabs. – Jim Sep 23, 2008 at 18:09

85 And no, if I use 8-wide tabs and you use 6-wide tabs, and we share code, it doesn't get messed up. It's all just a single tab



Note that the use of tabs confuses another aspect of PEP 8:

19



Limit all lines to a maximum of 79 characters.



Let's say, hypothetically, that you use a tab width of 2 and I use a tab width of 8. You write all your code so your longest lines reach 79 characters, then I start to work on your file. Now I've got hard-to-read code because (as the PEP states):

The default wrapping in most tools disrupts the visual structure of the code

If we all use 4 spaces, it's ALWAYS the same. Anyone whose editor can support an 80 character width can comfortably read the code. *Note: The 80 character limit is a holy war in and of itself, so let's not start that here.*

Any non-sucky editor should have an option to use spaces as if they were tabs (both inserting and deleting), so that really shouldn't be a valid argument.

[Share](#) [Improve this answer](#)

answered Aug 3, 2013 at 15:49

[Follow](#)



[Thane Brimhall](#)

9,535 ● 7 ● 37 ● 50



11



The answer to the question is: PEP-8 wants to make a recommendation and has decided that since spaces are more popular it will strongly recommend spaces over tabs.



Notes on PEP-8

PEP-8 says *'Use 4 spaces per indentation level.'*
Its clear that this is the standard recommendation.

'For really old code that you don't want to mess up, you can continue to use 8-space tabs.'

Its clear that there are SOME circumstances when tabs can be used.

'Never mix tabs and spaces.'

This is a clear prohibition of mixing - I think we all agree on this. Python can detect this and often chokes. Using the -tt argument makes this an explicit error.

'The most popular way of indenting Python is with spaces only. The second-most popular way is with tabs only.'

This clearly states that both are used. Just to be ultra-clear: You should still never mix spaces and tabs in same file.

'For new projects, spaces-only are strongly recommended over tabs.'

This is a clear recommendation, and a strong one, but not a prohibition of tabs.

I can't find a good answer to my own question in PEP-8. I use tabs, which I have used historically in other languages. Python accepts source with exclusive use of tabs. That's good enough for me.

I thought I would have a go at working with spaces. In my editor, I configured a file type to use spaces exclusively and so it inserts 4 spaces if I press tab. If I press tab too many times, I have to delete the spaces! **Arrgh!** Four times as many deletes as tabs! My editor can't tell that I'm using 4 spaces for indents (although AN editor might be able to do this) and obviously insists on deleting the spaces one at a time.

Couldn't Python be told to consider tabs to be n spaces when its reading indentations? If we could agree on 4 spaces per indentation and 4 spaces per tab and allow Python to accept this, then there would be no problems. We should find win-win solutions to problems.

Share Improve this answer

edited Sep 27, 2008 at 17:53

Follow

answered Sep 27, 2008 at 16:56




quamrana

39.3k ● 13 ● 55 ● 76

-
- 1 What editor are you using? Most I've used have an option to dedent on backspace (emacs behaves this way for instance), regardless of the implementation of the indenting.
– [Brian](#) Sep 27, 2008 at 17:36
-

You're right - I don't see an option to dedent on backspace, but you can probably deal with it using shift-tab, or reduce indent (ctrl-shift-i by default) instead. – [Brian](#) Sep 27, 2008 at 20:15

I'm just trying PyScripter which seems loads better at using spaces when you press tab and removing them in 4's when you press backspace. – [quamrana](#) Aug 23, 2009 at 19:11

34 "I have to delete the spaces! Arrgh! Four times as many deletes as tabs!" - This is the single reason I use tabs for everything, and why I think people who use spaces are insane. :) I've never had problems, except when I paste something in from the web that uses spaces. Then a simple find-replace fixes that. – [Aphex](#) Jan 13, 2011 at 15:43 



4



I've always used tabs in my code. That said, I've recently found a reason to use spaces: When developing on my Nokia N900 internet tablet, I now had a keyboard without a tab key. This forced me to either copy and paste tabs or re-write my code with spaces. I've run into the same problem with other phones. Granted, this is not a standard use of Python, but something to keep in mind.

Share Improve this answer

answered Jan 5, 2012 at 18:31

Follow



[Skyler](#)

939 ● 1 ● 10 ● 28

play.google.com/store/apps/... – [Smart Manoj](#) Oct 9, 2020 at 13:32



2

Since python relies on indentation in order to recognize program structure, a clear way to identify indentation is required. This is the reason to pick either spaces or tabs.



However, python also has a strong philosophy of only having one way to do things, therefore there should be an official recommendation for one way to do indentation.



Both spaces and tabs pose unique challenges for an editor to handle as indentation. The handling of tabs themselves is not uniform across editors or even user settings. Since spaces are not configurable, they pose the more logical choice as they guarantee that the outcome will look everywhere the same.

[Share](#) [Improve this answer](#)

answered Sep 23, 2008 at 13:35

[Follow](#)



[Florian Bösch](#)

27.7k ● 12 ● 51 ● 53

-
- 11 And since every editor also may chose its color scheme, do you think they should mandate also which color scheme to use? – [o0'](#) Jan 19, 2012 at 14:55
-
- 11 Yes but doesn't this inconsistency actually make more sense? Because it is simply a matter of visual preference. If I prefer a larger "looking" indent in my editor I can set my tabs to be 8 spaces, if I prefer less I can set it to 2. That way the code, without actually changing the formatting, better suits the individual who is observing it. – [denmat](#) Apr 22, 2013 at 13:32
-
- 10 I agree with denmat- If I visually prefer 2 spaces, and Guido visually prefers 4 spaces, then the logical choice is to use tab indentation. – [Sepero](#) Jun 6, 2013 at 10:21
-

[JWZ says it best:](#)

2

When [people are] reading code, and when they're done writing new code, they care about how many screen columns by which the code tends to indent when a new scope (or sexpr, or whatever) opens...

...My opinion is that the best way to solve the technical issues is to mandate that the ASCII #9 TAB character never appear in disk files:
program your editor to expand TABs to an appropriate number of spaces before writing the lines to disk...

...This assumes that you never use tabs in places where they are actually significant, like in string or character constants, but I never do that: when it matters that it is a tab, I always use `'\t'` instead.

Share Improve this answer

edited Jun 20, 2020 at 9:12

Follow



Community Bot

1 • 1

answered Sep 27, 2008 at 17:50



Mark Cidade

99.8k • 33 • 229 • 237

10 I'd do the contrary: tabs have a semantic meaning for indentation, so it is more sensed to have tabs stored and spaces displayed. The user could chose a formatting style and the editor would expand tabs accordingly. – [AkiRoss](#) Sep 27, 2013 at 10:35

1 You still have the problem of mixed tabs and spaces, as well as one author using 1 column per tab and indenting 4+ times, which would look crazy in a text editor set to display each tab character as 4 columns wide. Tabs for indentation make the most sense in a variable-width text editor, like a word processor using proportionally-spaced fonts. Not so much with a fixed-width text editor. – [Mark Cidade](#) Sep 27, 2013 at 16:04 ✎

2 No, I was meaning that a text editor should be able to parse the language grammar and *understand* when a tabulation is occurring, so that tabs could be used as only formatting mean and there would be no need of using spaces for indentation. "tab" is not required to have a fixed width, and I find generally shameful that with today's techniques (e.g.

machine learning), formatting is still a problem for programmers. Everything should be automated and it should be automated and transparent. – [AkiRoss](#) Sep 28, 2013 at 11:32

I don't see how it would be able to understand that.
– [Mark Cidade](#) Sep 29, 2013 at 0:20



0



The most significant advantage I can tell of spaces over tabs is that a lot of programmers and projects use a set number of columns for the source code, and if someone commits a change with their tabstop set to 2 spaces and the project uses 4 spaces as the tabstop the long lines are going to be too long for other people's editor window. I agree that tabs are easier to work with but I think spaces are easier for collaboration, which is important on a large open source project like Python.

Share Improve this answer

answered Sep 23, 2008 at 13:30

Follow



[sirwart](#)

2,474 ● 2 ● 15 ● 7

3 this is wrong: this happens only if you mix tabs and spaces, and you would equally solve it by forcing everyone to use tabs instead of spaces. – [o0'](#) Jan 19, 2012 at 14:54



0

You can have your cake and eat it to. Set your editor to expand tabs into spaces automatically.

(That would be `:set expandtab` in Vim.)



Share Improve this answer

answered Sep 23, 2008 at 13:52

Follow



Rod Daunoravicius

2,518 ● 1 ● 16 ● 10



0



Besides all the other reasons already named (consistency, never mixing spaces and tabs etc) I believe there are a few more reasons for the 4 spaces convention to note. These only apply to Python (and maybe other languages where indentation has meaning). Tabs may be nicer in other languages, depending on individual preferences.



1. If an editor doesn't show tabs (which happens, depending on the configuration, in quite a few), another author might assume that your code uses 4 spaces, b/c almost all of the Python code being publicly available does; if that same editor happens to have a tab width of 4, nasty things may happen - at least, that poor person will lose time over an indentation issue that would have been very easy to avoid by sticking to the convention. So for me, the number one reason is to avoid bugs with consistency.
2. Reframing the question of which is better, tabs or spaces, one should ask which the advantages of tabs are; I've seen plenty posts praising tabs, but few compelling arguments for them; good editors like emacs, vi(m), kate, ... do proper indentation

depending on the semantics of your code - even without tabs; the same editors can easily be configured to unindent on backspace etc.

3. Some people have very strong preferences when it comes to their freedom in deciding the look/ layout of code; others value consistency over this freedom. Python drastically reduces this freedom by dictating that indentation is used for blocks etc. This may be seen as a bug or a feature, but it sort of comes with choosing Python. Personally, I like this consistency - when starting to code on a new project, at least the layout is close to what I'm used to, so it's fairly easy to read. Almost always.
4. Using spaces for indentation allows "layout tricks" that may facilitate to comprehend code; some examples of these are listed in PEP8; eg.

```
foo = long_function_name(var_one, var_two,
                          var_three, var_four)

# the same for lists
a_long_list = [1,
               2,
               # ...
               79]

# or dictionaries
a_dict = {"a_key": "a_value",
          "another_key": "another_value"}
```

Of course, the above can also be written nicely as

```
foo = long_function_name(
    var_one, var_two,
    var_three, var_four)
```



```
# the same for lists
a_long_list = [
    1,
    2,
    # ...
    79]

# or dictionaries
a_dict = {
    "a_key": "a_value",
    "another_key": "another_value"}
```

However, the latter takes more lines of code and less lines are sometimes argued to be better (b/c you get more on a single screen). But if you like alignment, spaces (preferably assisted by a good editor) give you, in a sense, more freedom in Python than tabs. [Well, I guess some editors allow you to do the same w/ tabs ;) - but with spaces, all of them do...]

5. Coming back to the same argument that everybody else makes - PEP 8 dictates (ok, strongly recommends) spaces. If coming to a project that uses tabs only, of course, you have little choice. But because of the establishment of the PEP 8 conventions, almost all Python programmers are used to this style. This makes it sooooo much easier to find a consensus on a style that is accepted by most programmers... and having individuals agree on style might be very hard otherwise.
6. Tools that help enforcing style are usually aware of PEP 8 without extra effort. That's not a great reason, but it's just nice to have things work ~out of the box.

Share Improve this answer

answered Feb 25, 2013 at 21:11

Follow



Gerald Senarclens de Grancy

7,334 ● 8 ● 44 ● 55



0

My guess is that most the linux text editors make defaults look ridiculously large by default. I can't think of any other good reason to use spaces over tabs.



Share Improve this answer

answered Feb 26, 2019 at 0:27

Follow



brian cronrath

67 ● 6



0

Well, I would say that there is not such "recommendation" in the PEP 8. It is stated as a recommendation since they won't prohibit you to write tabs but since code must be written in the most standardized way, use spaces we must.



That said, if I were the one to write the standard guide, I would recommend tabs since they are a modern and more practical way to indent code.

Finally, I'll stress, I am **not encouraging** anybody to use tabs, instead, I am saying that all of us **should use spaces** as stated in the [style guide](#).

Share Improve this answer

edited Jan 14, 2021 at 19:41

Follow

answered Jan 14, 2021 at 19:33



Maxinfofamily

23 ● 1 ● 1 ● 4



The universal problem with tabs is that they can be represented differently in different environment.

-1

In a given editor, a tab might be 8 spaces or it might be 2. In some editors, you can control this, while in others you can't.



Another issue with tabs is how they are represented in printed output. I believe most printers interpret a tab as 8 spaces.

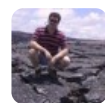


With spaces, there is no doubt. Everything will line up as the author intended.

Share Improve this answer

answered Sep 23, 2008 at 13:25

Follow



Benoit

38.9k ● 24 ● 85 ● 117

15 Another one who has fundamentally misunderstood the tab... get a mechanical typewriter and play with it for a while, really! 1 tab is not equal to 8 spaces! it is equal to *up_to_8_spaces*! otoh: with proportional fonts, tabs are the *only* way to guarantee alignment. – user3850 Oct 10, 2008 at 14:14

6 "In a given editor, a tab might be 8 spaces or it might be 2". If I like 4-spaces and my friend likes 8-spaces or 2-spaces or 3-spaces or, etc. Then we can both agree on tabs because (being dedicated *indentation* characters,) the editor knows

what they are and can display them accordingly. I see the code with 4-space wide indents, you see it with 8-space wide indents, our weird friend uses his 3-spaces, and everything's cool. The circumstances (especially in Python!) where the tab width itself would ever matter are so rare that even space-proponents rarely bring them up.

– [JamesTheAwesomeDude](#) Mar 27, 2015 at 0:31 

using tabs (which can have different spacings depending on the editor) are less of a problem when they are at the beginning of a line. however it is problematic when they are in the middle of a line of code as they align the next character to a multiple of the tab spacing, they do not insert a fixed spacing. the space of a tab when placed after another statement can be anything up to the tab spacing. so with a tab space of 8, a tab could result in 1,2,3,4,5,6,7 or 8 characters spacing to the next character on the line. This can make some code unreadable.

– [EmbeddedSoftwareEngineer](#) Oct 14, 2021 at 21:09



On the discussion between [Jim and Thomas Wouters](#) in the comments.

-1



The issue was... since the width of tabs and spaces both can vary -- and since programmers can't agree on either width -- why is it that tabs bear the blame.



I agree with Jim on that -- tabs are NOT evil in and of themselves. But there is a problem...

With spaces I can control how **"MY OWN CODE"** looks in EVERY editor in the world. If I use 4 spaces -- then no matter what editor you open my code in, it will have the same distance from the left margin. With tabs I am at the

mercy of the tab-width setting for the editor -- even for MY OWN CODE. And I don't like that.

So while it is true that even spaces can't guarantee consistency -- they at least afford you more control over the look of your OWN code everywhere -- something that tabs can't.

I think it's NOT the consistency in the programmers writing the code -- but the consistency in editors showing that code -- that spaces make easier to achieve (and impose).

Share Improve this answer

Follow

edited May 23, 2017 at 11:54



Community Bot

1 • 1

answered Aug 6, 2011 at 6:53



treecoder

45k • 23 • 70 • 95

11 You're "at the mercy of the tab-width setting for the editor"? If your editor doesn't let you set the tab-width you want, you may be using notepad.exe – [user137369](#) Apr 14, 2013 at 12:59

5 @zigg That's absolutely irrelevant to the argument, since he (she?) is talking specifically about **his/her own code** (that information is even bolded, in italics, and all caps). Nowhere to the discussion is sharing code relevant. – [user137369](#) Aug 4, 2013 at 19:26 ✎

2 Editors aren't the only tools to view code with. There are also diffs, tracebacks, Github and other web pages, and so on

that will all choose some tab width out of your control (probably 8). – [RemcoGerlich](#) Nov 23, 2015 at 22:22

- 4 I see your point. You indeed control how everyone sees your code (regarding indentation). Your next step is to control the font type and coloring that everyone will use to see your code. After that, you are prepared to dominate the world itself and not only code editors!! – [Djunzu](#) Oct 31, 2017 at 15:15
-