

# Are there any pitfalls with using whitespace in Python?

Asked 15 years, 9 months ago   Modified 14 years, 5 months ago

Viewed 5k times



3



At the moment I have never had a problem with whitespace in Python (although I've only used it in two projects and I was the only programmer). What are some potential pitfalls with whitespace and indentation in Python for someone learning the language?



python

whitespace



Share

Improve this question

Follow

edited Jun 29, 2010 at 19:34



Lance Roberts

22.8k ● 32 ● 114 ● 132

asked Mar 12, 2009 at 3:50



user34537

This question might be more useful if it gives a specific problem (with code) with the corresponding error.

– RedBlueThing Mar 12, 2009 at 4:01

@acidzombie24 - I made some fairly substantial edits to your question, but I think it's still asking what you want without

sounding confrontational or flamebait-y. But rollback if I went too far. – [Dana](#) Mar 12, 2009 at 4:23

It's deceptive to say "whitespace for scope in Python" simply because variable scope in Python isn't defined in terms of whitespace. – [Ignacio Vazquez-Abrams](#) Mar 12, 2009 at 4:49

@ignacio surely local scope is defined by white space indentation in Python? – [tarn](#) Mar 12, 2009 at 5:48

- 1 Whitespace does not "define" scope per se, but delineates basic blocks, doesn't it? – [Craig S](#) Mar 12, 2009 at 14:11

## 17 Answers

Sorted by:

Highest score (default)



11

It can be confusing in some editors where one line is indented with spaces and the next is indented with a tab. This is confusing as the indentation looks the same but causes an error.



Also when your copying code, if your editor doesn't have a function to indent entire blocks, it could be annoying fixing all the indentation.

But with a good editor and a bit of practice, this shouldn't be a problem. I personally really like the way Python uses white space.

Share Improve this answer

answered Mar 12, 2009 at 3:59


Follow



[tarn](#)

2,192 ● 2 ● 13 ● 17

- 
- 1 The "looks same but is wrong" problem only happen when the editor and Python disagree on the size of a tab. When they agree, it works fine. Most Python people tend to prefer spaces-only, though. – user25148 Mar 12, 2009 at 6:10
- 

From the Python docs, section 2.1.8 (Indentation): ["tabs are replaced \(from left to right\) by one to eight spaces such that the total number of characters up to and including the replacement is a multiple of eight \(this is intended to be the same rule as used by Unix\)."](#) So, if everyone's editor follows this rule, we're fine and the point is moot. Unfortunately, that's rarely the case. Hence, PEP 8 suggests spaces only.  
– Mike DeSimone Jun 29, 2010 at 19:40 

---



10



yeah there are some pitfalls, but most of the time, in practice, they turn out to be enemy [windmills of the Quixotic style](#), i.e. imaginary, and nothing to worry about in reality.



I would estimate that the pitfalls one is *most likely to encounter* are (including mitigating steps identified):



### 1. **working with others a.k.a. collaboration**



a. if you have others which for whatever reason refuse to adhere to [PEP 8](#), then it could become a pain to maintain code. I've never seen this in practice once I point out to them the almost universal convention for python is **indent level == four spaces**

b. get anyone/everyone you work with to accept the convention and have them figure out how to have

their editor automatically do it (or better yet, if you use the same editor, show them how to configure it) such that copy-and-paste and stuff *just works*.

2. **having to invest in a "decent" editor other than your current preferred one, if your current preferred editor is not python friendly** -- not really a pitfall, more an investment requirement to avoid the other pitfalls mentioned associated with copy-and-paste, re-factoring, etc. stop using Notepad and you'll thank yourself in the morning.

a. your efficiency in editing the code will be much higher under an editor which understands `python`

b. most *modern* code editors handle python decently. I myself prefer GNU Emacs, and recent versions come with excellent `python-mode` support out-of-the-box. There are plenty of [other editors to explore](#), including many free alternatives and [IDEs](#).

c. python itself comes out of the box with a "smart" python editor, `idle`. Check it out if you are not familiar, as it is probably already available with your python install, and may even support `python` better than your current editor. [PyCrust](#) is another option for a python editor implemented in python, and comes as part of wxPython.

3. **some code generation or templating environments that incorporate python (think HTML generation or python CGI/WSGI apps) can have quirks**

- a. most of them, if they touch python, have taken steps to minimize the nature of python as an issue, but it still pops up once in a while.
- b. if you encounter this, familiarize yourself with the steps that the framework authors have already taken to minimize the impact, and read their suggestions (*and yes they will have some if it has ever been encountered in their project*), and it will be simple to avoid the pitfalls related to python on this.

Share Improve this answer

edited Mar 12, 2009 at 13:15

Follow

answered Mar 12, 2009 at 4:56



popcnt

4,645 ● 1 ● 18 ● 14

---

+1: 1. PEP 8. 2. All modern editors handle Python indent-outdent. Point 3 is so obscure as to be a non-concern -- I like the irony of picking something really obscure. – [S.Lott](#) Mar 12, 2009 at 11:41

---

@S. Lott: thanks for the feedback. yeah, I felt like I was stretching with #3. mostly because the other two "pitfalls" are also mentioned above, but IMHO do not usually amount to anything in the "real world"... – [popcnt](#) Mar 12, 2009 at 13:08

---



2

That actually kept me away from Python for a while. Coming from a strong C background, I felt like I was driving without a seat belt.



It was aggravating when I was trying to fill up a snippet library in my editor with boilerplate, frequently used classes. I learn best by example, so I was grabbing as many interesting snippets as I could with the aim of writing a useful program while learning.

After I got in the habit of re-formatting everything that I borrowed, it wasn't so bad. But it still felt really awkward. I had to get used to a dynamically typed language PLUS indentation controlling my code.

It was quite a leap for me :)

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

answered Mar 12, 2009 at 4:11

[Follow](#)



user50049

---

I always thought that in comparing Py and C, C would be the "without the seatbelt, steering wheel and brakes" option :)

– [Rook](#) Mar 12, 2009 at 17:37

---

Depends on whether you're getting things to compile or whether you're getting them to run right. C is a very easy to start chainsaw of a language. – [David Thornley](#) Mar 12, 2009 at 18:10

---

@David, I was mostly talking about controlling the scope of the code. It just felt REALLY weird to do it via indentation alone. – user50049 Mar 13, 2009 at 10:02

---

As opposed to braces alone? I haven't run across a C compiler that points out when the braces and indentation don't sync. – [Mike DeSimone](#) Jun 29, 2010 at 19:45

---

@Mike, note the braces, dealing with a compiler that cares not about indentation :) – user50049 Jun 29, 2010 at 23:58

---



When I look at C and Java code, it's always nicely indented.

2

Always. Nicely. Indented.



Clearly, C and Java folks spend a lot of time getting their whitespace right.

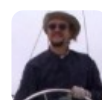


So do Python programmers.

Share Improve this answer

answered Mar 12, 2009 at 11:02

Follow



[S.Lott](#)

391k ● 82 ● 517 ● 788

---

1 the experienced python programmers do not spend a lot of time doing this, armed with a good editor. :) – [popcnt](#) Mar 12, 2009 at 13:09

---

@popcnt: Not a lot of time; about the same amount of time as Java and C programmers. – [S.Lott](#) Mar 12, 2009 at 15:48

---

good point, but to nitpick, this doesn't really answer the question, rather it's a response to the critics! – [hasen](#) Mar 12, 2009 at 17:37

---

The question is merely a criticism of whitespace. Which I always find silly because so many programmers spend so much time getting it right, even in languages where it doesn't matter. Given the care lavished on whitespace, there are no pitfalls. – [S.Lott](#) Mar 12, 2009 at 18:38

---

I'm a big fan of not having the "where to put the braces" argument. – [Mike DeSimone](#) Jun 29, 2010 at 19:43

---



1



Whitespace block delimiters force a certain amount of code formatting, which seems to irritate some programmers. Some in our shop seem to be of the attitude that they are too busy, or can't be bothered to pay attention to formatting standards, and a language that forces it rubs them raw. Sometimes the same folks gripe when others do not follow the same patterns of putting curly braces on a new line ;)

I find that Python code from the web is more commonly "readable", since this minor formatting requirement is in place. IMO, this requirement is a very useful feature.



IIRC, does not Haskell, OCaml (#light), and F# also use whitespace in the same fashion? For some reason, I have not seen any complaints about these languages.

Share Improve this answer

answered Mar 12, 2009 at 4:18

Follow



kermatt

1,662 ● 2 ● 17 ● 36

---

I was thinking that maybe the surface syntax of a language should be an IDE thing rather than a language thing - maybe you can use simple rewrite rules in Emacs or whatever to make your Lisp code look like Python or Smalltalk code or APL or Haskell code. Language tutorials can use the same rewrite rules so you can pick which one to use. Though its easier to make Lisp "view" to look like any other language, than to make C look like Python its just AST -> string in Lisp, vs C -> parse -> stringify -> Python. – [aoeu256](#) Sep 11, 2019 at 14:16

---



1



Long ago, in an environment far, far away, there were languages (such as RPG) that depended on the column structure of punch cards. This was a tedious and annoying system, and led to many errors, and newer languages such as BASIC, pascal, and so forth were designed without this dependency.



A generation of programmers were trained on these languages and told repeatedly that the freedom to put anything anywhere was a wonderful feature of the newer languages, and they should be grateful. The freedom was used, abused, and calibrated (cf the [IOCC](#)) for many years.

Now the pendulum has begun to swing back, but many people still remember that forced layout is bad in some way vague, and resist it.

IMHO, the thing to do is to work with languages on their own terms, and not get hung up on tastes-great-less-filling battles.

Share Improve this answer

answered Mar 12, 2009 at 4:18

Follow



MarkusQ

21.9k ● 4 ● 57 ● 68



1



When python programmers don't follow the common convention of "Use 4 spaces per indentation level" defined in [PEP 8](#). *(If your a python programmer and haven't read it please do so)*

Then you run into copy paste issues.



Share Improve this answer

answered Mar 12, 2009 at 4:33

Follow



monkut

43.7k ● 26 ● 128 ● 160



1



Some people say that they don't like python indentation, because it can cause errors, which would be immensely hard to detect in case if tabs and spaces are mixed. For example:

```
1 if needFrobnicating:
2     frobnicate()
```





3 update()

Depending on the tab width, line 3 may appear to be in the same block as line 2, or in the enclosing block. This won't cause runtime or compile error, but the program would do unexpected thing.

Though I program in python for 10 years and never seen an error caused by mixing tabs and spaces

Share Improve this answer

edited Mar 12, 2009 at 4:36

Follow

answered Mar 12, 2009 at 4:20



Eugene Morozov

15.8k ● 3 ● 27 ● 33

---

1 +1: never seen a problem except n00bs on day 1 or day 2 of learning Python. Then they configure their editor to use spaces and never make a mistake again. – [S.Lott](#) Mar 12, 2009 at 11:01

---

but I use tabs, and do spaces -> tabs on my editor : ( – [aoeu256](#) Sep 11, 2019 at 14:11

---



1



Pick a good editor. You'd want features such as:

1. Automatic indentation that mimics the last indented line
2. Automatic indentation that you can control (tabs vs. spaces)



3. Show whitespace characters



4. Detection and mimicking of whitespace convention when loading a file

For example, Vim lets me highlight tabs with these settings:

```
set list
set listchars=tab:|_
highlight SpecialKey ctermbg=Red guibg=Red
highlight SpecialKey ctermfg=White guifg=White
```

Which can be turned off at any time using:

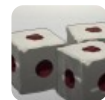
```
set nolist
```

IMO, I dislike editor settings that convert tabs to spaces or vice versa, because you end up with a mix of tabs and spaces, which can be nasty.

Share Improve this answer

answered Mar 12, 2009 at 4:37

Follow



**Tung Nguyen**

13k ● 4 ● 20 ● 10

---

on the contrary, I like editors that convert tabs into spaces! I hate tabs, imho all editors should treat the tab character as if it doesn't exist, and just replace it with spaces all the time.

– [hasen](#) Mar 12, 2009 at 7:32

---



I used to think that the white space issues was just a question of getting used to it.

1



Someone pointed out some serious flaws with Python indentation and I think they are quite valid and some subconscious understanding of these is what makes experienced programmers nervous about the whole thing:-



- Cut and paste just doesn't work anymore! You cannot cut boiler plate code from one app and drop it into another app.
- Your editor becomes powerless to help you. With C/Java etc. there are two things going on the "official" curly brackets indentation, and, the "unofficial" white space indentation. Most editors are able to reformat the white space indentation to match the curly brackets nesting -- which gives you a strong visual clue that something is wrong if the indentation is not what you expected. With Python's "space is syntax" paradigm your editor cannot help you.
- The sheer pain of introducing another condition into already complex logic. Adding another if then else into an existing condition involves lots of silly error prone inserting of spaces on many lines line by hand.
- Refactoring is a nightmare. Moving blocks of code around your classes is so painful it's easier to put up with a "wrong" class structure than refactor it into a better one.

Share Improve this answer

answered Mar 12, 2009 at 4:48

Follow



James Anderson

27.4k ● 7 ● 54 ● 80

---

there are many editors that have excellent support for python and can definitely help you in the situations you describe, including reformatting/re-indenting, and refactoring. I use python-mode in Emacs to do these very things. – [popcnt](#) Mar 12, 2009 at 5:00

---

Selecting a block and using Tab/Shift-Tab to set it to the appropriate indent level is trivially easy in any non-awful editor, you shouldn't be manually inserting spaces on every line. Stop using Notepad! – [bobince](#) Mar 12, 2009 at 5:30

---

(And for sanity's sake, please fix the indenting of cut-and-pasted code even when the language doesn't require it!)  
– [bobince](#) Mar 12, 2009 at 5:31

---

1 You need a better editor. I've never had *any* of those problems. – [kquinn](#) Mar 12, 2009 at 5:59

---

You need a better editor. I have most of those problems in C++, since I refuse to leave the indentation not matching the program logic. They don't give me significant problems.  
– [David Thornley](#) Mar 12, 2009 at 18:14

---



1



If you use Eclipse as your IDE, you should take a look at PyDev; it handles indentation and spacing automatically. You can copy-paste from mixed-spacing sources, and it will convert them for you. Since I started learning the language, I've never once had to think about spacing.

And it's really a non-issue; sane programmers indent anyway. With Python, you just do what you've always



done, minus having to type and match braces.

Share Improve this answer

answered Mar 12, 2009 at 13:05

Follow



DNS

38.2k ● 20 ● 96 ● 133



## Pitfalls

1



- It can be annoying posting code snippets on web sites that ignore your indentation.
- Its hard to see how multi-line anonymous functions (lambdas) can fit in with the syntax of the language.
- It makes it hard to embed Python in HTML files to make templates in the way that PHP or C# can be embedded in PHP or ASP.NET pages. But that's not necessarily the best way to design templates anyway.
- If your editor does not have sensible commands for block indent and outdent it will be tedious to realign code.

## Advantages

- Forces even lazy programmers to produce legible code. I've seen examples of brace-language code that I had to spend hours reformatting to be able to read it...
- Python programmers do not need to spend hours discussing whether braces should go at the ends of

lines K&R style or on lines on their own in the Microsoft style.

- Frees the brace characters for use for dictionary and set expressions.
- Is automatically pretty legible

Share Improve this answer

answered Mar 12, 2009 at 17:33

Follow



[pdc](#)

2,394 ● 21 ● 28



0



The problem is that in Python, if you use spaces to indent basic blocks in one area of a file, and tabs to indent in another, you get a run-time error. This is quite different from semicolons in C.

This isn't really a programming question, though, is it?



Share Improve this answer

answered Mar 12, 2009 at 3:59

Follow



[Craig S](#)

952 ● 4 ● 7



---

it is, i want to learn of issues of a new language or API before i start and get myself into trouble. – user34537 Mar 12, 2009 at 4:08

---

Tabs and spaces are mixable, as long as the editor and Python agree on the size of tabs. – user25148 Mar 12, 2009 at 6:11

---

@acidzombie24 if spaces and tabs in Python are your biggest concern of pitfalls, you're going to have fun with the language. – [Craig S](#) Mar 12, 2009 at 14:08

---



0

The only trouble I've ever had is minor annoyances when I'm using code I wrote before I settled on whether I liked tabs or spaces, or cutting and posting code from a website.



I think most decent editors these days have a convert tabs-to-spaces and back option. Textmate certainly does.



Beyond that, the indentation has never caused me any trouble.

Share Improve this answer

answered Mar 12, 2009 at 4:27

Follow



[Dana](#)

32.9k ● 17 ● 64 ● 73



0

If your using emacs, set a hard tab length of 8 and a soft tab length of 4. This way you will be alerted to any extraneous tab characters. You should always uses 4 spaces instead of tabs.



Share Improve this answer

answered Mar 12, 2009 at 15:40

Follow



kcbanner



0

One drawback I experienced as a beginner with python was forgetting to set softtabs in my editors gave me lots of trouble.



But after a year of serious use of the language I'm not able to write poorly indented code anymore in any other language.



Share Improve this answer

answered Mar 12, 2009 at 15:51

Follow



Vasil

38k ● 27 ● 93 ● 116



0

No, I would say that is one thing to which I can find no downfalls. Yes, it is no doubt irritating to some, but that is just because they have a different habit about their style of formatting. Learn it early, and it's gonna stick. Just look how many discussions we have over a style matter in languages like C, Cpp, Java and such. You don't see those (useless, no doubt) discussions about languages like Python, F77, and some others mentioned here which have a "fixed" formatting style.



The only thing which is variable is spaces vs. tabs (can be changed with a little trouble with any good editor), and

amount of spaces tab accounts for (can be changed with no trouble with any good editor). Voila ! Style discussion complete.

Now I can do something useful :)

Share Improve this answer

answered Mar 12, 2009 at 17:55

Follow



Rook

62.4k ● 53 ● 168 ● 247

---