



Python Indent

Kevin Rose | 6,214,660 installs | (48) | Free

Correct Python indentation

Installation

Launch VS Code Quick Open (Ctrl+P), paste the following command, and press enter.

```
ext install KevinRose.vsc-python-indent
```

[Copy](#)[More Info](#)

Overview

Version History

Rating & Review

Python Indent

Correct Python indentation in Visual Studio Code. See the extension on the [VSCoDe Marketplace](#) and its source code on [GitHub](#).

Theme used in gif: *Community Theme Palenight* from [Community Material Theme](#) v1.4.4.

Azure Pipelines **succeeded**

How it works

Every time you press the Enter key in a Python context, this extension will parse your Python file up to the location of your cursor, and determine exactly how much the next line (or two in the case of hanging indents) should be indented and how much nearby lines should be un-indented.

There are three main cases when determining the correct indentation, described below.

Between bracket pairs

In cases when you have your cursor between an open bracket (one of [({) and its closing bracket pair (the corresponding one of })]), this extension will keep subsequent lines indented just to the right of where it was opened:

```
data = {'a': 0,  
      | # <- pressing enter should put your cursor at th
```

Categories

[Programming Languages](#)[Keymaps](#)[Formatters](#)

Tags

[dedent](#)[indent](#)[indentation](#)[jupyter](#)[keybindings](#)[python](#)[whitespace](#)

Works with

Universal

Resources

[Issues](#)[Repository](#)[Homepage](#)[License](#)[Changelog](#)[Download Extension](#)

Project Details

[kbrose/vsc-python-indent](#)

[No Pull Requests](#)

[Last Commit: a month ago](#)

[9 Open Issues](#)

More Info

Version 1.18.0

Released on 21.03.2019,
09:30:29

```
| # <- this is where default VS Code puts your cursor
```

Last updated 04.10.2022, 07:26:13

Publisher Kevin Rose

Unique Identifier KevinRose.vsc-python-indent

Report [Report Abuse](#)



Even heavily nested brackets are handled:

```
data = {'a': 0,
        'b': [[1, 2],
               | # <- match the more recently opened [ inst
               | # <- default behavior of VS Code
```

```
data = {'a': 0,
        'b': [[1, 2],
               [3, 4]],
        | # <- since the lists are all closed, go back to
        | # <- default behavior of VS Code
```

```
data = {'a': 0,
        'b': [[1, 2],
               [3, 4]],
        'c': 5}
| # <- go back to indentation level before any brackets we
| # <- default behavior of VS Code
```

In the full example below, default VS Code required nine extra key presses (three tabs, two spaces, and four backspaces) to match the *automatic* indentation of this extension.

```
data = {'a': 0,
        'b': [[1, 2],
               [3, 4]],
        'c': 5}
done(data)
```

Hanging indents

When you have opened a bracket, but not yet inserted any content, pressing Enter will create a hanging indent, matching the base behavior of VS Code.

```
result = my_func(
    | # <- your cursor should end up here
) # <- the closing bracket should end up here
```

You can use the setting `useTabOnHangingIndent` to make it so that when you are done typing you can simply press Tab to be taken to the closing bracket.

If there is content to the right of your cursor when you press Enter, then this extension falls back on just indenting by your set tab size.

```
# The "|" is your cursor's location
result = my_func(|x, y, z)
# and when you press Enter...
result = my_func(
    |x, y, z)
```

If you never want to have the closing bracket end up on its own line (i.e. you always want to just indent by the set tab size), use the `keepHangingBracketOnLine` configuration setting. *Warning:* This may cause confusing indentation with function definitions as the argument list and the function code may end up at the same indentation level.

It's not often used, but a backslash to continue a line will also result in the next line being indented.

```
my_long_calculation = 1234 + \
    5678
```

Keywords

Some keywords in Python imply certain indentation behaviors. For example, if there is a `return` statement, then we know the next line can be un-indented (or *dedented*) since no statements can follow a `return` in the same code block. Other keywords that follow the same pattern are `pass`, `break`, `continue`, and `raise`.

Similarly, if there is an `else:` on the current line, then the current line needs to be dedented, and the next line needs to be indented *relative to* the new position of the `else:`. Other keywords that follow the same pattern are `elif ...:`, `except ...:`, and `finally:`. Some examples are shown below.

```
if True:
    pass
else:|
# and when you press Enter...
if True:
    pass
else:
    |
```

But if you have manually changed the indentation, the extension should not change it for you:

```
if True:
    if True:
        pass
    else:|
# and when you press Enter, do NOT dedent!
if True:
    if True:
        pass
    else:
        |
```

```
# Or even more nested
if True:
    if True:
        if True:
            pass
        else:|
# and when you press Enter, still do NOT dedent
if True:
    if True:
        if True:
            pass
        else:
            |
```

Extending comments

If (and only if) you press Enter while your cursor is in the middle of a comment, then the next line will automatically be made into a comment.

```
# As always, the "|" indicates your cursor
def f():
    # This function is |gonna be REAL good!

def f():
    # This function is
    # |gonna be REAL good
```

Trimming whitespace lines

You can trim whitespace from lines that contain *only* whitespace by using the `trimLinesWithOnlyWhitespace` configuration setting (the default is to not trim whitespace in this way). This setting brings the behavior closer to native VSCode behavior.

```
# In the below code, the character "." represents a space
def f():
    ....|

# The default of false preserves whitespace
def f():
    ....
    ....|

# Setting trimLinesWithOnlyWhitespace = true will trim the
def f():
    ....|
```



Why is it needed?

There are many related issues on GitHub ([\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#)) asking for improved Python indentation in VS Code. It seems like the maintainers of the Python extension at microsoft are not prioritizing indentation, since there has been no progress in the years since it was first asked for.

Caveats

Known caveats are listed below.

- Using tabs (\t) for your indentation will not work.
- If your Python code is not correctly formatted, you may not get correct indentation.
- The extension works by registering the Enter key as a keyboard shortcut. The conditions when the shortcut is triggered have been heavily restricted, but there may still be times this extension is unexpectedly overriding Enter behavior. Specifically, vim related plugins seem to require special attention. See the [when](#) clause in [package.json](#).

If you experience any problems, please submit an [issue](#), or better yet a [pull request](#).

Release Notes

See the [change log](#).

Developing

See the [developer docs](#) for pointers on how to develop this extension.

[Contact us](#) [Jobs](#) [Privacy](#) [Terms of use](#) [Trademarks](#)

© 2023 Microsoft