**From: NVDA 2023.3.3 User Guide**

[**https://www.nvaccess.org/files/nvda/documentation/userGuide.html**](https://www.nvaccess.org/files/nvda/documentation/userGuide.html)

**Line indentation reporting**

This option allows you to configure how indentation at the beginning of lines is reported. The Report line indentation with combo box has four options.

* Off: NVDA will not treat indentation specially.
* Speech: If speech is selected, when the amount of indentation changes, NVDA will say something like "twelve space" or "four tab."
* Tones: If Tones is selected, when the amount of indentation changes, tones indicate the amount of change in indent. The tone will increase in pitch every space, and for a tab, it will increase in pitch the equivalent of 4 spaces.
* Both Speech and Tones: This option reads indentation using both of the above methods.

If you tick the "Ignore blank lines for line indentation reporting" checkbox, then indentation changes won't be reported for blank lines. This may be useful when reading a document where blank lines are used to separate indented bloks of text, such as in programming source code.

An email:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Jacob Kruger jacob.kruger.work@gmail.com**[**via**](https://support.google.com/mail/answer/1311182?hl=en)**mitalumprod.onmicrosoft.com** | | Feb 16, 2024, 3:17 AM (1 day ago) |  |  |
| |  | | --- | | to pythonvis | | | |

Thunderbird portable is not cooperating this morning - hence starting  
new topic.  
  
  
Anyway, I work with NVDA, and, in terms of python code editors, primary  
two are edSharp and VS code, but, either way, I stick to 4 spaces for  
indentation - in edSharp you can choose if you want a tab character to  
be replaced with four spaces or not, and, it has it's own option to  
announce indentation changes, or to query indentation levels, or else it  
lets you use { and } characters to handle code blocks while typing  
python code, with keystrokes to switch between braces and indentation  
before running/saving code, but, I just stick to indentation, and, if  
you are working with a .py file, then it handles code block navigation,  
etc. well enough.  
  
  
I have also set NVDA to use tones to announce indentation changes.  
  
  
In terms of VS code, I use the NVDA add-on called indentNav, which then  
lets you jump around between code blocks/indentation levels, but, also  
implemented my own scratchpad NVDA add-on to let me query indentation  
levels, working with space characters.  
  
  
Anyway, the primary reason I stick to working with 4 spaces is that  
that's the standard you are going to come across the most, so it makes  
it easier in terms of reviewing code samples, copying and pasting from  
tutorial material, etc. etc. - don't want/need to remember who's code am  
working with at different times.  
  
  
Stay wlel  
  
--  
  
Jacob Kruger  
+2782 413 4791  
"Resistance is futile!...Acceptance is versatile..."