

Line width formatting standard

[closed]

Asked 16 years, 1 month ago Modified 11 years, 9 months ago

Viewed 63k times



36



As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, [visit the help center](#) for guidance.

Closed 12 years ago.

Do you follow a standard for wrapping long lines in source code? What line length do you find most comfortable to read?

Sometimes I find people who program on wide-screen monitors and like to use its full width for displaying source code. I prefer shorter lines, around 80-100 characters, but I have a hard time trying to convince colleagues with the ever-increasing popularity of wide-screen devices.

Edit:

Similar questions:

- [Is there a valid reason for enforcing a maximum width of 80 characters in a code file, this day and age?](#)

coding-style

formatting

Share

edited May 23, 2017 at 10:31

Improve this question

Follow

community wiki

5 revs, 3 users 54%

[Rômulo Ceccon](#)

Very similar question to stackoverflow.com/questions/110928
– [Andrew Edgecombe](#) Nov 10, 2008 at 1:25

Thanks, Andrew. A quick search on SO and the suggested questions that appear when you're typing yours turned out nothing. – [Rômulo Ceccon](#) Nov 10, 2008 at 10:49

9 Answers

Sorted by:

Highest score (default)



64

Don't compromise readability for dogmatic rules on the exact number of characters in a row. Horizontal scrolling is undesirable but an 81-character line is easier to read than an indentation-confusingly line-wrapped version.



80 characters is likely to be inadequate for programming styles with large indentations and/or verbose variable





names. Keep the amount of logical complexity down to a per-line maximum, not the number of characters.



Share Improve this answer

answered [Nov 10, 2008 at 0:40](#)

Follow

community wiki
[bobince](#)

-
- 2 If you go over 80 characters you've got another much bigger problem --- you are being overly verbose. A line width limit forces you to think better names for your variables and refactor away unnecessary scopes. – [Hyena](#) Jun 19, 2020 at 9:35
-



I stick to the 80 lines rule (And try to convince everyone to do the same). Some reasons:

23






1. You can open 2 (or more) editors at once.
2. Same thing with compare tools. - most (all?) of them display the two(some three (some more ?)) files side by side.
3. Sometimes you need to work from remote, on a different workstation, or a laptop, and suddenly, your nicely formatted 120 char's to line code looks horrible.



Share Improve this answer

answered [Nov 9, 2008 at 15:53](#)

Follow

-
- 47 8 years later, with dual monitors and 1440p monitors being less than \$300, all you reasons listed here are mitigated. So, I choose to break up a line at such a place which doesn't make the code unreadable. I hope you still aren't trying to get others to stick to 80 characters. – [Monir](#) Jan 25, 2016 at 16:57 
-
- 9 80 chars (not *lines*) was computer terminal resolution 50 years ago, it makes no sense to follow this standard today. – [Oleg Mikheev](#) Mar 22, 2016 at 21:27
-
- 7 Python conventions still suggest 80 character attempt to keep the code clean and readable in a small window. Just because you have a 3000sq/ft house does not mean you should fill it with junk. If you need it to support your 5 person family - then by all means, fill it up. – [cgseller](#) Dec 15, 2016 at 18:04
-
- 4 Whether you use a FHD screen or a USQFHD screen, I bet you *will* change the font size so about the same amount of text fits on all screens (except when you have a 2.21:1 cinematic editing screen or 1:1 MRI screen). (Side note: Python PEP8 actually specifies 79 instead of 80) – [Mark Jeronimus](#) Feb 7, 2017 at 10:32 
-
- 1 When everyone is using 4k screens will people write even longer lines? Because you have a wider monitor does not mean you should write longer lines. Some people prefer monitors in portrait mode, too. As a quick test, see if you can read this, or if you have to make your browser window smaller: pbm.com/~lindah/brief.html – [Spacen Jasset](#) Sep 21, 2018 at 15:14 
-



7



I prefer longer lines for one simple reason: I can fit more code into my window. There is a *huge* difference between having to scroll vertically to read a function and being able to fit it in a single screen. If everything is line-wrapped so that the function scrolls off the bottom while the right half of my screen is empty, I consider that to be a huge waste. Note that opening two editor windows doesn't help here either.

Share Improve this answer

answered [Nov 9, 2008 at 17:18](#)

Follow

community wiki
[staktrace](#)

-
- 1 This is great if you or only a select few of your colleagues (whom you know also like to read code like you) are going to read the code. But if you are coding for a larger group of users, then it can be inconsiderate. – [shahensha](#) Dec 5, 2020 at 4:44
-



5



You should not have to scroll horizontally to read the code. But larger screens does not mean longer lines! There is also a limit to how much there should go on in a single line.

So I say: Keep it at 70-80 chars just as always. Larger screens just means that the IDE as more room.

Share Improve this answer

Follow

answered Nov 9, 2008 at 15:43

community wiki
[myplacedk](#)



5



It also depends on other conventions that you're using. At one job, we were programming in Java and the convention was to use long and descriptive identifiers, which meant that only a couple of them could fit on a line without running into the 80-character limit. I thought that was pretty stupid considering every developer in the company was given a widescreen monitor that could easily fit 200 characters. With hardware consistency like that it makes no sense to enforce a stupidly small line wrap limit.

Share Improve this answer

answered Nov 9, 2008 at 17:14

Follow

community wiki
[staktrace](#)



3



Bigger screen -- Bigger font. I use GVim with `Consolas` 14pt maximized at 1280x800 screen resolution. I try to wrap at about 80-90% screen width.

Share Improve this answer

answered Nov 9, 2008 at 16:37

Follow



3

We use coding standard of 80 characters in line. The original reason for 80 char limitation is not relevant today, but some number should be picked...



Beside the obvious (code organization and readability) usually i found that long lines are result of bad styling and folowing such rule improve code quality and reduce errors. Just compare the following examples :



```
status = do_something();
if (status == error)
{
    do_error_handling();
    return;
}
/* do you regular flow */
status = do_more();
if (status == error)
{
    do_error_handling();
    return;
}
/* do more of you regular flow and keep you line
80 chars*/
```

instead :

```
status = do_something();
if (status == succes)
{
    /* do you regular flow */
    status = do_more();
}
```

```

        if (status == success)
        {
            /* do you regular flow */
            /* nest again and get line behind
visible screen */
        }
        else
        {
            /* do error handling */
        }

    }
    else
    {
        /* do error handling */
    }
}

```

Second example is much less readable hard to maintain and probably will lead to some problem on the way ...

Edit

Replaced `goto` with `do_error_handling()` in the code to avoid not relevant discussion.

As i stated before 80 characters not relevant today it's just a number 100 is good as well.

For anyone that found second example more readable please nest it few more times with real code and try read again :)

Share Improve this answer

edited Nov 9, 2008 at 18:36

Follow

3 revs
Ilya

16 funnily enough, i find the 2nd one much more readable, and expected that to be your example of 'readable' code when i first eyeballed your post. I've never understood this fascination with 80 chars in modern coding practises (i understand the historical significance). I usually keep it to 100chars – [Karan](#) Nov 9, 2008 at 17:20

so you're suggesting that we use GOTO statements? =)
– [Can Berk Güder](#) Nov 9, 2008 at 17:24

1 we will not start goto discussion here :) i will update an example – [Ilya](#) Nov 9, 2008 at 17:31

don't worry, I'm j/k. =) – [Can Berk Güder](#) Nov 9, 2008 at 17:39

2 I also find the second example more readable. My counter to the counter of "nest it a few more times with real code and try to read again" would be that you should try and factor those extra nests into additional small private methods.
– [Bane](#) Oct 28, 2014 at 13:37



1



I program almost exclusively on a laptop, so I agree with the shorter lines. Granted, I'm usually designing screens for PDAs, so I can get away with it. But if code is shared between developers it will end up on somebody's laptop eventually, and scroll bars make me cry.



Share Improve this answer

answered [Nov 9, 2008 at 15:52](#)



Follow



0



I use around 72-75 columns to ensure that I can print the code on letter-format pages without too much trouble. I also use spaces instead of tabs and am careful about the layout.

To notice when I am going off the right margin, I often put in a text line that I can use as a ruler. I set the IDE display window so the ruler just fits the horizontal width and then I make sure I don't go outside of it.

I do this in .txt documents as well as .c, .java, .cpp, batch files, etc. This makes it easier to send snippets in e-mail, post on blogs, put in comments, etc. The ruler is often just beneath a top line that identifies the file and the format of the text:

```
/* example.txt 0.00                                UTF-8
dh:2008-11-09
* ---|---1---|---2---|---3---|---4---|---
-5---|---6---|---7---*
*/
```

Of course, the comment convention of the particular kind of file is used.

Share Improve this answer

answered Nov 9, 2008 at 21:21

Follow

