



I'm not robot



[Continue](#)

Formatting c code in visual studio

Although this tutorial has content that we believe is of great benefit to our community, we have not yet tested or edited it to ensure you have a bug-free learning experience. It's on our list, and we're working on it! You can help us by reporting a problem button at the bottom of the tutorial to use. Formatting code consistently is a pain, especially when working on a team. The beauty of contemporary web development is that the tools have turned so much better! In this article we will look at setting up Prettier to automatically format your code in Visual Studio Code. Sample code For demo purposes, here's the sample code we'll format. If you're picky about code formatting, you'll immediately pick up some obvious missteps. mix of single vs double quotes the first property of the person object must be on its own line the console declaration inside of the function must be intruded you may or may not like the optional parentheses around the parameter of the arrow function Installing the Nicier Extension To work with prettier in Visual Studio Code, you need to install the extension. Search for Prettier - Code Formatter. You can see the extension below. If you install it for the first time, you'll see an install button instead of the uninstall button you see on mine. The Format Document Command with the Nicier extension installed, we can now use it to format our code. We'll work more on this later, but to get started, we can set up the Format Document Command. To open the command palette, you can use Command **⇧**+ Shift+P on Mac or Control **⇧**+ Shift+P on Windows. In the command palette search format, then select **Format Document**. You can then be asked to choose which formatter to use. To do this, click the Configure button. Then select Nicier **-** Code Formatter. And the VOILA! Your code is nice and formatted. Notice all the fancy improvements! spacing line wrap consistent quotes Prettier also works with CSS files! The great thing is that it works on CSS files too! From this... For this! Format automatically on Save so far, we have had to manually run a command to format our code. Instead, you can select a setting in VS Code to have your files automatically formatted when you save. It has some great advantages. You never need to format your code manually again! ensure code is formatted without thinking of it code being checked in it is not formatted to change this setting, use **⇧**+ Command + , on Mac or Control + , on Windows to open the settings menu. Then search for Editor: Format on Save and make sure it's checked out. With this setting in place, you can proceed to your business writing sloppily formatted code as we all know it will all be taken care of for you automatically! More beautiful configuration in Code Settings More Beautiful does a lot of things for you by default, but you can also customize the settings. Here are some of the most common settings. Single Single - select between single and double quotes Semi - select or semi-colons at the end of lines include Tab Width - how many spaces you want a tab to consist of Open the Settings menu as above. Then look for Prettier. This will bring up all the settings you can just change in your editor there. For example, what if I change the tab width to 10. Then save my file. Pretty easy right?! It's probably not the tab width size you want to keep, but it's all up to you! Creating a nicer configuration file The downside of using the spoils in settings menu in VS Code is that it does not ensure consistency across developers on your team. If you change settings in your VS code, someone else might have a very different set of settings in theirs. Establish consistent formatting across your team by creating a configuration file! To resolve this, you can create a Nicier configuration file. It should be titled .prettierrc. (ext) with one of the following extensions. yml, yaml, or json js toml include package.json file (alternate option) I usually prefer JSON configuration files where you can define key **key**: value pairs for your settings. VS Code will even provide some intellisense for you if you type. Here is an example of a simple configuration file. For more details about the configuration files, check out the Nicier Docs. After you create one of these and check it in with your project, you can ensure that each team member follows the same formatting rules. Conclusion Do not waste your time manually formatting your code manually. It takes time that can be better spent writing more code. Take use of the amazing modern tools out there and set up Prettier! Since the day Microsoft released Visual Studio Code, I've installed it on Windows instead of notepad++. Because VS Code does not have a built-in code format or beautifier by default, I was eager to see a more powerful VS code with extensions. Now there it is. We can find many useful extensions on Visual Studio Marketplace. Recently, I wrote C/C++ code on Ubuntu and found the extension Clang-Format for beautifying C/C++ code. Let's take a look at how to make clang format work with Visual Studio Code on Windows and Linux. Getting started with Clang-Format To install an extension, we can press Ctrl+Shift+P and type in install extension. When all extensions are listed, search for format, and you'll see the Clang Format: After installing the extension, you need to reboot VSCode. To format code, you can call Command Palette again with Ctrl+Shift+P, and then enter format: The shortcut Ctrl+Shift+I is for Linux. If you want to use it on Windows, you must use Alt+Shift+F. If you don't have Clang-Format installed on your system, you'll see the prompt: The 'clang-format' command is not available. Please check your and make sure it is installed. How to install Clang Format on Windows Download Clang for Windows (32-bit) or Clang for (64-bit). Install the package and add the path of %LLVM%\bin to your system environment. The shortcut Alt+Shift+F now works in Visual Studio Code for Windows. How to install Clang Format on Ubuntu 14.04 There are two ways to install clang format on Ubuntu 14.04: the stand-alone clang format-3.4 or Clang for x86_64 Ubuntu 14.04. The package size of stand-alone is much smaller than the full LLVM. If you choose to install clang format-3.4, the VS Code extension cannot work immediately. It will still ask you that no clang format is found. Why? The installed clang format tool is named clang-format-3.4: \$trueis clang-format-3.4 clang-format-3: /usr/bin/clang-format-3.4/usr/bin/X11/clang-format-3.4 To make it work, you just need to create a symlink: sudo ln -s/usr/bin/clang-format-3.4/usr/bin/clang-format Alternatively, if you download and extract the LLVM package, you'll find clang format under clang+llvm/bin: Likewise, create a symbolic link for clang format: sudo ln -s /home/xiao/clang+llvm-3.7.0-x86_64-linux-gnu-ubuntu-14.04/bin/clang-format/usr/ Bin/clang format Now, you can format code with Ctrl+Shift+I in Visual Studio When I did the formatting for the C language (in Linux) using Ctrl+Shift+I, the opening arms that were on the same line of the function name when after the next line. Like this: void view() {}becamevoid display(){} don't like to happen and how can I change this rule? Note – I don't know if the formatting is provided by the extensions. The only C extension I've installed is the C/C++ extension provided by Microsoft (URL - .Zam has tried many IDE for developing C++ project, but none of these are smart for coding style, e.g. Sublime. Dev C++ . Qt Creator. But fortunately, I found the VSCode and the extension called C/C++ for Visual Studio Code, although that's not good enough, but I'm satisfying for it. How to customize your own coding style rule, install your VSCode and Extension C/C++ for Visual Studio Code firstChoose your favor argumentArgument Link: Clang Format Style OptionsI would like to recommend an argument I feel usefulBasedOnStyle: Choose your side, I chose Google! IndentWidth: the space number before code each lineColumnLimit: the line numbers, 0 means unlimitedSpacesBeforeTrailingCommentsDerivePointerAlignment: If true, analyze the formatted file for the most common alignment of &*. Pointer and reference alignment styles are going to be updated according to the preferences found in the file. PointerAlignment is then used only as a relapse. Cpp11BracedListStyle:This is all about {}PointerAlignment: Left, Right, MiddleAccessModifierOffset:The extra indent or outer access modifiers, Change SettingSTEP 1 Go Preferences -> SettingsSTEP 2-1 Search C_Cpp.clang_format_fallbackStyleSTEP 2-2 Change the file:setting.json directwindows %APPDATA%\Code\User\settings.jsonmacOS %APPDATA%\Code\User\settings.jsonmacOS Support/Code/User/Settings.jsonLinux \$HOME/.config/Code/User/settings.jsonSTEP3 The Google style is really good, but still change to individuality your gunsmodify C_Cpp.clang_format_fallbackStyle and the original value wasVisual StudioThe resultSettingsetting.jsonReferenceVSCode SettingClang Format Style OptionsIf you like this one or feel useful, please applause or follow to Zam, I really appreciate you doing what