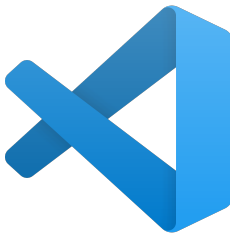


# Customizing (a little) Visual Studio Code

Mathieu Kessler

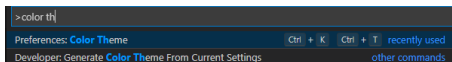
Departamento de Matemática Aplicada y Estadística  
Universidad Politécnica de Cartagena

<https://code.visualstudio.com>



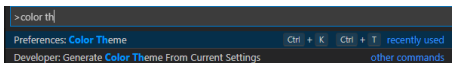
# Let us begin by changing color theme and the icons theme

Using the Command Palette (Ctrl-Shift-P), you choose Preferences: Color Theme.

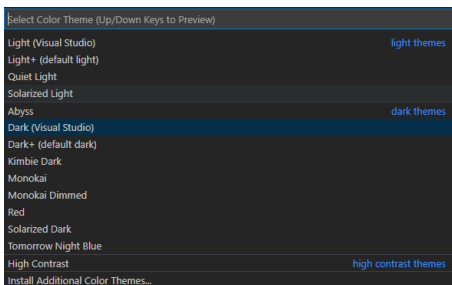


# Let us begin by changing color theme and the icons theme

Using the Command Palette (Ctrl-Shift-P), you choose Preferences: Color Theme.

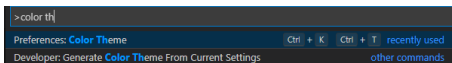


You can choose from a list of preinstalled themes, or decide to install additional color themes.

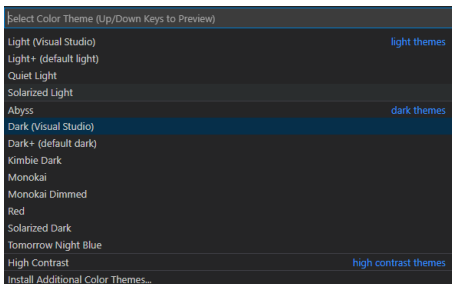


# Let us begin by changing color theme and the icons theme

Using the Command Palette (Ctrl-Shift-P), you choose Preferences: Color Theme.

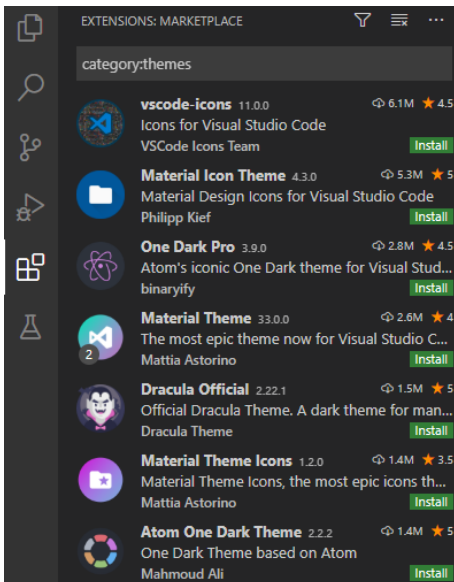


You can choose from a list of preinstalled themes, or decide to install additional color themes.



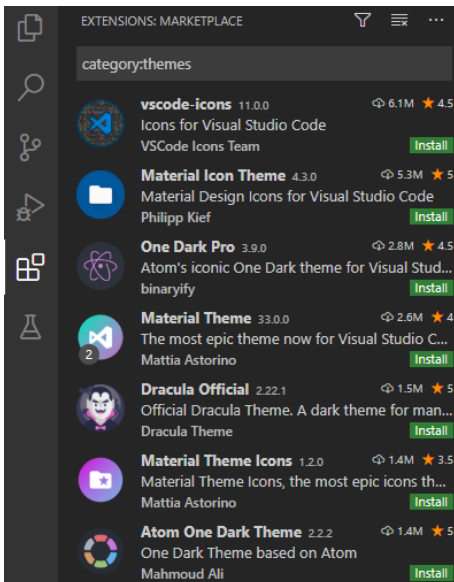
# Let us begin by changing color theme and the icons theme

If you choose to install additional color themes, it takes you to the extensions market place, with the results filtered with category: themes.



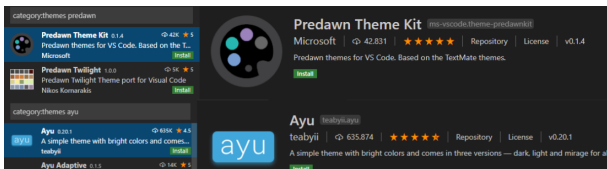
# Let us begin by changing color theme and the icons theme

If you choose to install additional color themes, it takes you to the extensions market place, with the results filtered with category: themes.



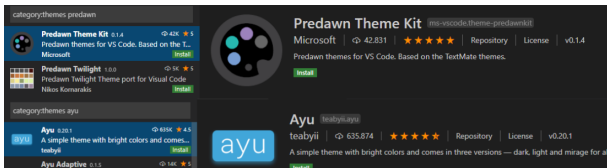
# Let us begin by changing color theme and the icons theme

I like the Predawn theme and the Ayu theme for icons. I search for them in the extension marketplace:

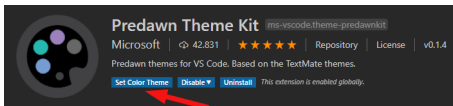


# Let us begin by changing color theme and the icons theme

I like the Predawn theme and the Ayu theme for icons. I search for them in the extension marketplace:



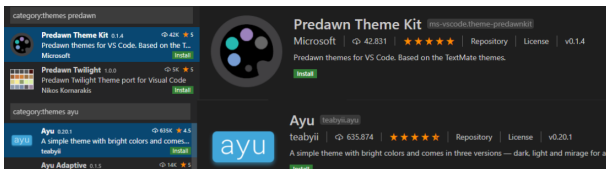
I first install predawn, and click on “Select Color Theme”.



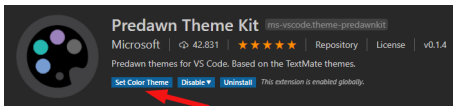


# Let us begin by changing color theme and the icons theme

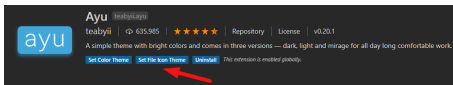
I like the Predawn theme and the Ayu theme for icons. I search for them in the extension marketplace:



I first install predawn, and click on “Select Color Theme”.



And after installing Ayu, I only click on “Set File Icon Theme”.



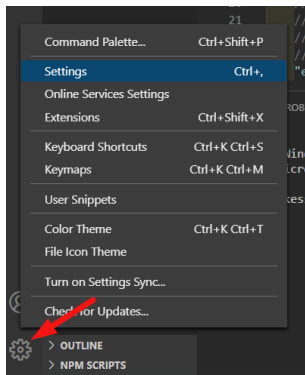
# Visual Studio Code is highly customizable

There are many many aspects of Visual studio Code that are configurable. These are controlled through the values of variables that are stored in json files with the name of settings.json.

# Visual Studio Code is highly customizable

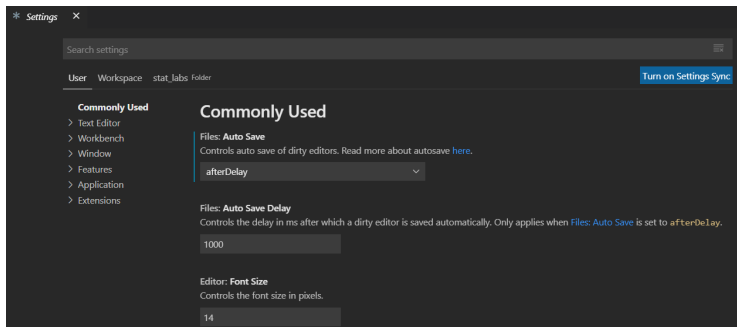
There are many many aspects of Visual studio Code that are configurable. These are controlled through the values of variables that are stored in json files with the name of settings.json.

To visualize and modify the settings, you click the wheel in the bottom left of the editor.



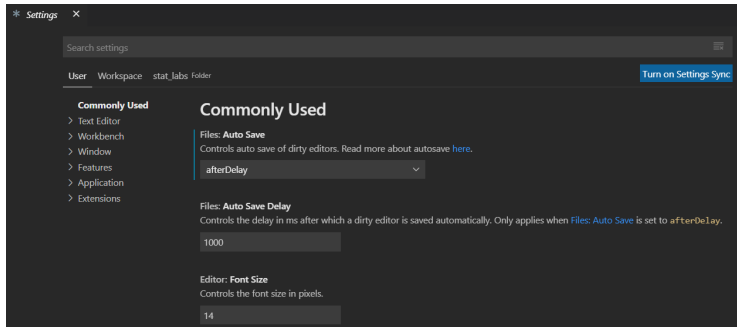
# Visual Studio Code is highly customizable

You then have the possibility to search for settings. Notice the three tabs, “User”, “Workspace” or “Folder”, to specify the context to which the modified settings should apply.



# Visual Studio Code is highly customizable

You then have the possibility to search for settings. Notice the three tabs, “User”, “Workspace” or “Folder”, to specify the context to which the modified settings should apply.

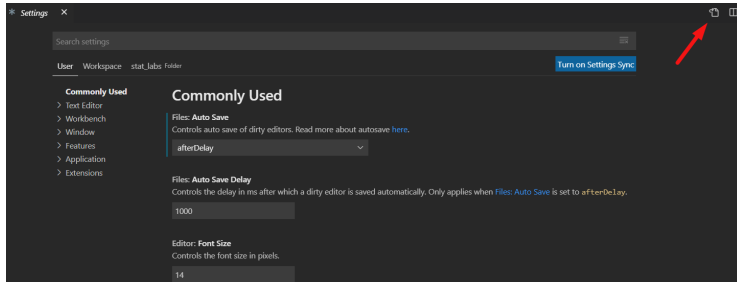


This is the “ui” mode of editing settings

We could also see all the settings variables in json form (key: value)

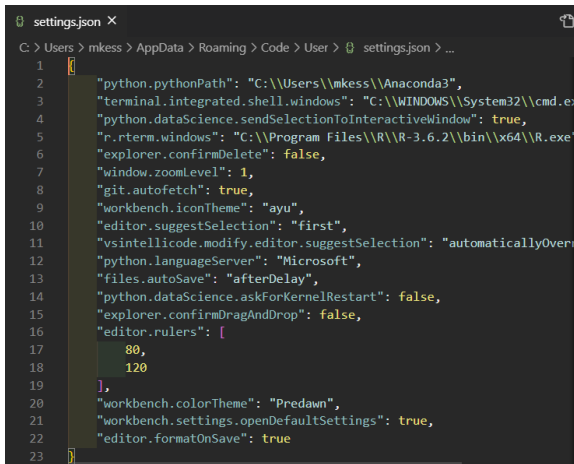
# Visual Studio Code is highly customizable

To switch to the “json” mode of editing settings, click on the upper right icon



# Visual Studio Code is highly customizable

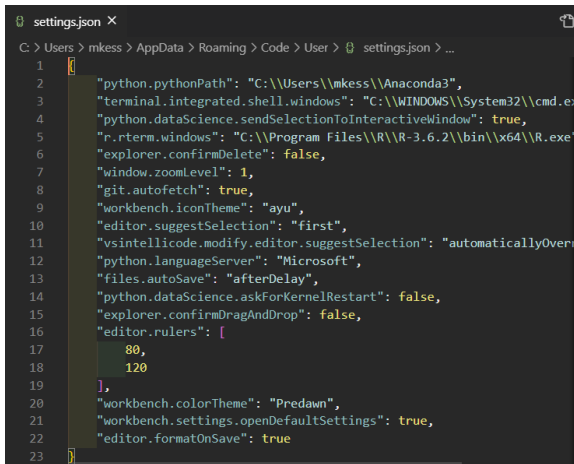
The format is key: value, where the key is the name of each setting variable.

A screenshot of a Visual Studio Code editor window showing a settings.json file. The window title is 'settings.json' with a close button. The breadcrumb path is 'C: > Users > mkess > AppData > Roaming > Code > User > settings.json > ...'. The file content is a JSON object with various settings. Line numbers 1 through 23 are visible on the left. The settings include paths for Python, terminal shell, R terminal, and explorer, as well as boolean flags for sendSelectionToInteractiveWindow, confirmDelete, autofetch, askForKernelRestart, confirmDragAndDrop, openDefaultSettings, and formatOnSave. There is also a list for editor.rulers with values 80 and 120. The editor has a light theme and a scrollbar on the right.

```
1  {
2    "python.pythonPath": "C:\\Users\\mkess\\Anaconda3",
3    "terminal.integrated.shell.windows": "C:\\WINDOWS\\System32\\cmd.e",
4    "python.dataScience.sendSelectionToInteractiveWindow": true,
5    "r.rterm.windows": "C:\\Program Files\\R\\R-3.6.2\\bin\\x64\\R.exe",
6    "explorer.confirmDelete": false,
7    "window.zoomLevel": 1,
8    "git.autofetch": true,
9    "workbench.iconTheme": "ayu",
10   "editor.suggestSelection": "first",
11   "vsintellicode.modify.editor.suggestSelection": "automaticallyOverr",
12   "python.languageServer": "Microsoft",
13   "files.autoSave": "afterDelay",
14   "python.dataScience.askForKernelRestart": false,
15   "explorer.confirmDragAndDrop": false,
16   "editor.rulers": [
17     80,
18     120
19   ],
20   "workbench.colorTheme": "Predawn",
21   "workbench.settings.openDefaultSettings": true,
22   "editor.formatOnSave": true
23 }
```

# Visual Studio Code is highly customizable

The format is key: value, where the key is the name of each setting variable.

A screenshot of the Visual Studio Code editor showing the settings.json file. The file is open in a dark-themed editor window with a title bar that says 'settings.json'. The file path is visible in the breadcrumb: 'C: > Users > mkess > AppData > Roaming > Code > User > settings.json > ...'. The JSON content is as follows:

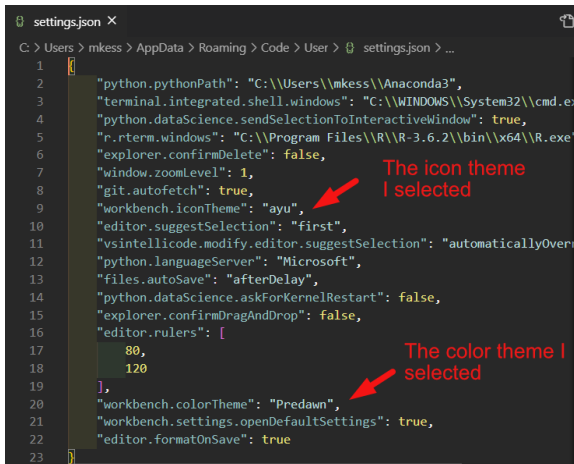
```
1  {
2    "python.pythonPath": "C:\\Users\\mkess\\Anaconda3",
3    "terminal.integrated.shell.windows": "C:\\WINDOWS\\System32\\cmd.exe",
4    "python.dataScience.sendSelectionToInteractiveWindow": true,
5    "r.rterm.windows": "C:\\Program Files\\R\\R-3.6.2\\bin\\x64\\R.exe",
6    "explorer.confirmDelete": false,
7    "window.zoomLevel": 1,
8    "git.autofetch": true,
9    "workbench.iconTheme": "ayu",
10   "editor.suggestSelection": "first",
11   "vsintellicode.modify.editor.suggestSelection": "automaticallyOverwrite",
12   "python.languageServer": "Microsoft",
13   "files.autoSave": "afterDelay",
14   "python.dataScience.askForKernelRestart": false,
15   "explorer.confirmDragAndDrop": false,
16   "editor.rulers": [
17     80,
18     120
19   ],
20   "workbench.colorTheme": "Predawn",
21   "workbench.settings.openDefaultSettings": true,
22   "editor.formatOnSave": true
23 }
```

If you hover over a variable, you get a description and a pen icon appears which allows you to edit its value.



# Visual Studio Code is highly customizable

We can see the two variables that reflect our selection of icons and color themes:



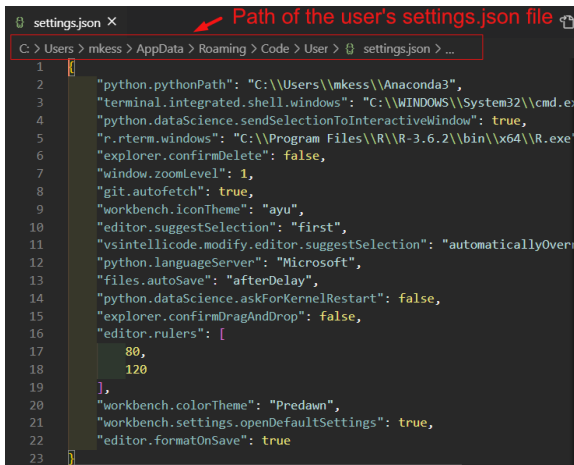
The image shows a screenshot of the Visual Studio Code settings editor. The file being edited is `settings.json`, located at `C:\Users\mkess\AppData\Roaming\Code\User\settings.json`. The file contains a JSON object with various settings. Two settings are highlighted with red arrows and text:

- `"workbench.iconTheme": "ayu",` is highlighted with a red arrow pointing to it from the text "The icon theme I selected".
- `"workbench.colorTheme": "Predawn",` is highlighted with a red arrow pointing to it from the text "The color theme I selected".

```
1  {
2    "python.pythonPath": "C:\\Users\\mkess\\Anaconda3",
3    "terminal.integrated.shell.windows": "C:\\WINDOWS\\System32\\cmd.exe",
4    "python.dataScience.sendSelectionToInteractiveWindow": true,
5    "r.terminal.windows": "C:\\Program Files\\R\\R-3.6.2\\bin\\x64\\R.exe",
6    "explorer.confirmDelete": false,
7    "window.zoomLevel": 1,
8    "git.autofetch": true,
9    "workbench.iconTheme": "ayu",
10   "editor.suggestSelection": "first",
11   "vsintellicode.modify.editor.suggestSelection": "automaticallyOverwrite",
12   "python.languageServer": "Microsoft",
13   "files.autoSave": "afterDelay",
14   "python.dataScience.askForKernelRestart": false,
15   "explorer.confirmDragAndDrop": false,
16   "editor.rulers": [
17     80,
18     120
19   ],
20   "workbench.colorTheme": "Predawn",
21   "workbench.settings.openDefaultSettings": true,
22   "editor.formatOnSave": true
23 }
```

# Visual Studio Code is highly customizable

You can see that you are in fact editing the user-level settings.json file:

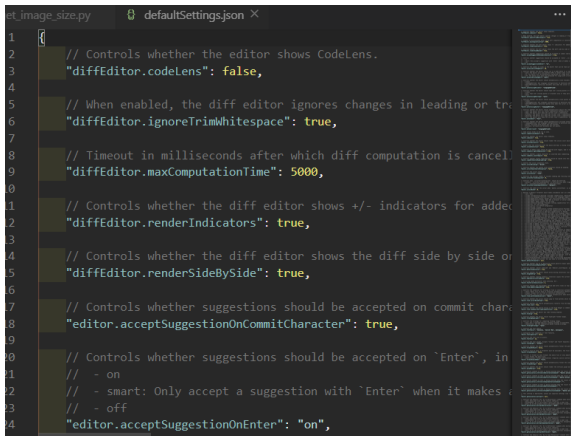


The screenshot shows the Visual Studio Code interface with the 'settings.json' file open. The file path is displayed in the breadcrumb navigation: 'C: > Users > mkess > AppData > Roaming > Code > User > settings.json > ...'. A red arrow points to this path with the text 'Path of the user's settings.json file'. The settings are listed in JSON format, including paths for Python, terminal, R, and explorer, as well as various UI and editor settings like 'window.zoomLevel', 'git.autofetch', 'workbench.iconTheme', 'editor.suggestSelection', 'vsintellicode.modify.editor.suggestSelection', 'python.languageServer', 'files.autoSave', 'python.dataScience.askForKernelRestart', 'explorer.confirmDragAndDrop', 'editor.rulers', 'workbench.colorTheme', 'workbench.settings.openDefaultSettings', and 'editor.formatOnSave'.

```
1  {
2    "python.pythonPath": "C:\\Users\\mkess\\Anaconda3",
3    "terminal.integrated.shell.windows": "C:\\WINDOWS\\System32\\cmd.e
4    "python.dataScience.sendSelectionToInteractiveWindow": true,
5    "r.rterm.windows": "C:\\Program Files\\R\\R-3.6.2\\bin\\x64\\R.exe"
6    "explorer.confirmDelete": false,
7    "window.zoomLevel": 1,
8    "git.autofetch": true,
9    "workbench.iconTheme": "ayu",
10   "editor.suggestSelection": "first",
11   "vsintellicode.modify.editor.suggestSelection": "automaticallyOverr
12   "python.languageServer": "Microsoft",
13   "files.autoSave": "afterDelay",
14   "python.dataScience.askForKernelRestart": false,
15   "explorer.confirmDragAndDrop": false,
16   "editor.rulers": [
17     80,
18     120
19   ],
20   "workbench.colorTheme": "Predawn",
21   "workbench.settings.openDefaultSettings": true,
22   "editor.formatOnSave": true
23 }
```

# Visual Studio Code is highly customizable

If you want to see all the default Settings, type “Open Default Settings” in the Palette Command (Ctrl-Shift-P)

A screenshot of the Visual Studio Code editor interface. The top bar shows two tabs: 'get\_image\_size.py' and 'defaultSettings.json'. The 'defaultSettings.json' tab is active, displaying a list of JSON key-value pairs. The code is syntax-highlighted, with keys in quotes and values in lowercase. Comments are present for several settings, explaining their function. The settings include 'diffEditor.codeLens', 'diffEditor.ignoreTrimWhitespace', 'diffEditor.maxComputationTime', 'diffEditor.renderIndicators', 'diffEditor.renderSideBySide', 'editor.acceptSuggestionOnCommitCharacter', and 'editor.acceptSuggestionOnEnter'. The editor has a dark theme, and the right sidebar shows a file explorer with various folders and files.

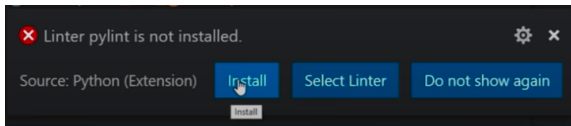
```
1 // Controls whether the editor shows CodeLens.  
2 "diffEditor.codeLens": false,  
3  
4 // When enabled, the diff editor ignores changes in leading or trailing whitespace.  
5 "diffEditor.ignoreTrimWhitespace": true,  
6  
7 // Timeout in milliseconds after which diff computation is cancelled.  
8 "diffEditor.maxComputationTime": 5000,  
9  
10 // Controls whether the diff editor shows +/- indicators for added/removed lines.  
11 "diffEditor.renderIndicators": true,  
12  
13 // Controls whether the diff editor shows the diff side by side or on top of each other.  
14 "diffEditor.renderSideBySide": true,  
15  
16 // Controls whether suggestions should be accepted on commit character.  
17 "editor.acceptSuggestionOnCommitCharacter": true,  
18  
19 // Controls whether suggestions should be accepted on `Enter`, in three modes:  
20 // - on: Always accept a suggestion on `Enter`.  
21 // - smart: Only accept a suggestion with `Enter` when it makes sense.  
22 // - off: Never accept a suggestion on `Enter`.  
23 "editor.acceptSuggestionOnEnter": "on",  
24
```

It is a huge list of key:value pairs, but you may find it useful to search for a variable, read the description, and copy, paste it and edit it in your own settings.json.

## What is linting?

Linting gives on the fly hints about ill written code, by marking possible mistakes.

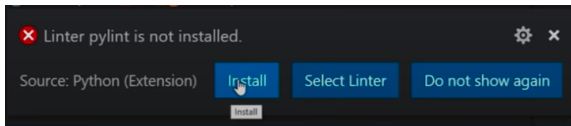
To enable it, use the Command Palette and type “Run Linting”, a dialog box will pop up, choose “Install”.



## What is linting?

Linting gives on the fly hints about ill written code, by marking possible mistakes.

To enable it, use the Command Palette and type “Run Linting”, a dialog box will pop up, choose “Install”.

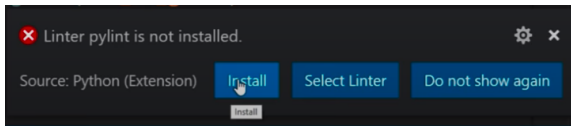


When asked, always use Conda to install the packages.

## What is linting?

Linting gives on the fly hints about ill written code, by marking possible mistakes.

To enable it, use the Command Palette and type “Run Linting”, a dialog box will pop up, choose “Install”.



When asked, always use Conda to install the packages.

**Note:** if you are using a virtual environment (for example with Miniconda)

Pylint will install in the virtual environment, which means you will have to install it again if you change environment.

## Style guide

We will try to be consistent with the accepted guidelines about code formatting, to improve readability.

## Style guide

We will try to be consistent with the accepted guidelines about code formatting, to improve readability.

*Good coding style is like correct punctuation: you can manage without it, but it sure makes things easier to read.*  
*Hadley Wickham.*



## Style guide

We will try to be consistent with the accepted guidelines about code formatting, to improve readability.

*Good coding style is like correct punctuation: you can manage without it, but it sure makes things easier to read.*

*Hadley Wickham.*

One of the [Python Enhancement Proposals](https://www.python.org/dev/peps/pep-0008/) documents the Style guide for Python Code. It is the famous PEP8 document

<https://www.python.org/dev/peps/pep-0008/>

It is a recommended reading.

## Style guide

We will try to be consistent with the accepted guidelines about code formatting, to improve readability.

*Good coding style is like correct punctuation: you can manage without it, but it sure makes things easier to read.*

*Hadley Wickham.*

One of the [Python Enhancement Proposals](https://www.python.org/dev/peps/pep-0008/) documents the Style guide for Python Code. It is the famous PEP8 document

<https://www.python.org/dev/peps/pep-0008/>

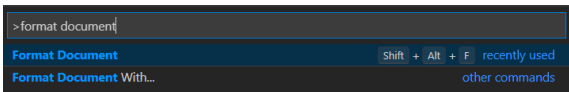
It is a recommended reading.

## Code formatters

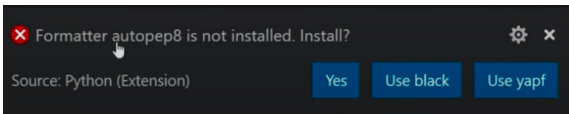
Code formatters exist that help converting badly formatted code into more readable one, some of them targeting the compliance with PEP8.

# Install a formatter in Visual Studio Code

Open a .py file or create one, and in the Command Palette , type “Format document”



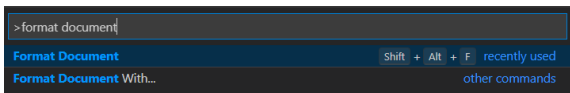
A dialog box pops up:



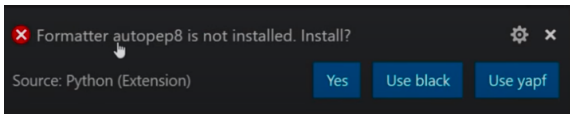
You can say yes to install autopep8, or you may choose either black or yapf. (When asked, use Conda to install packages)

# Install a formatter in Visual Studio Code

Open a .py file or create one, and in the Command Palette , type “Format document”



A dialog box pops up:



You can say yes to install autopep8, or you may choose either black or yapf. (When asked, use Conda to install packages)

Remember to format your files from time to time with Format Document or Shift + Alt + F