

Master VS Code for Python  
like a pro with these time-  
saving tips and shortcuts!"

Data  
Science  
Essentials

# VS code Python Tricks and Tips-Part1

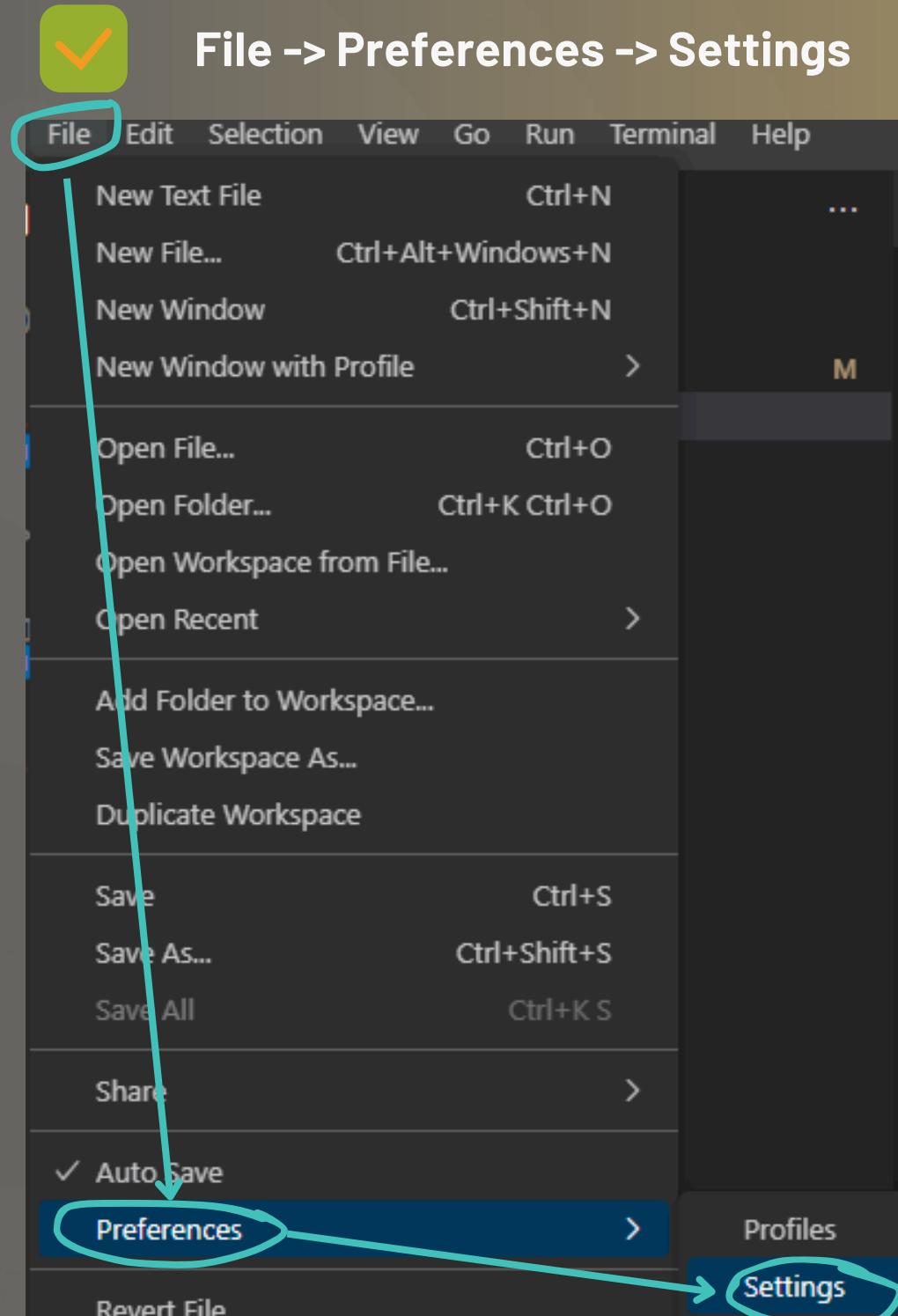


<https://www.linkedin.com/in/sidharthmahotra/>

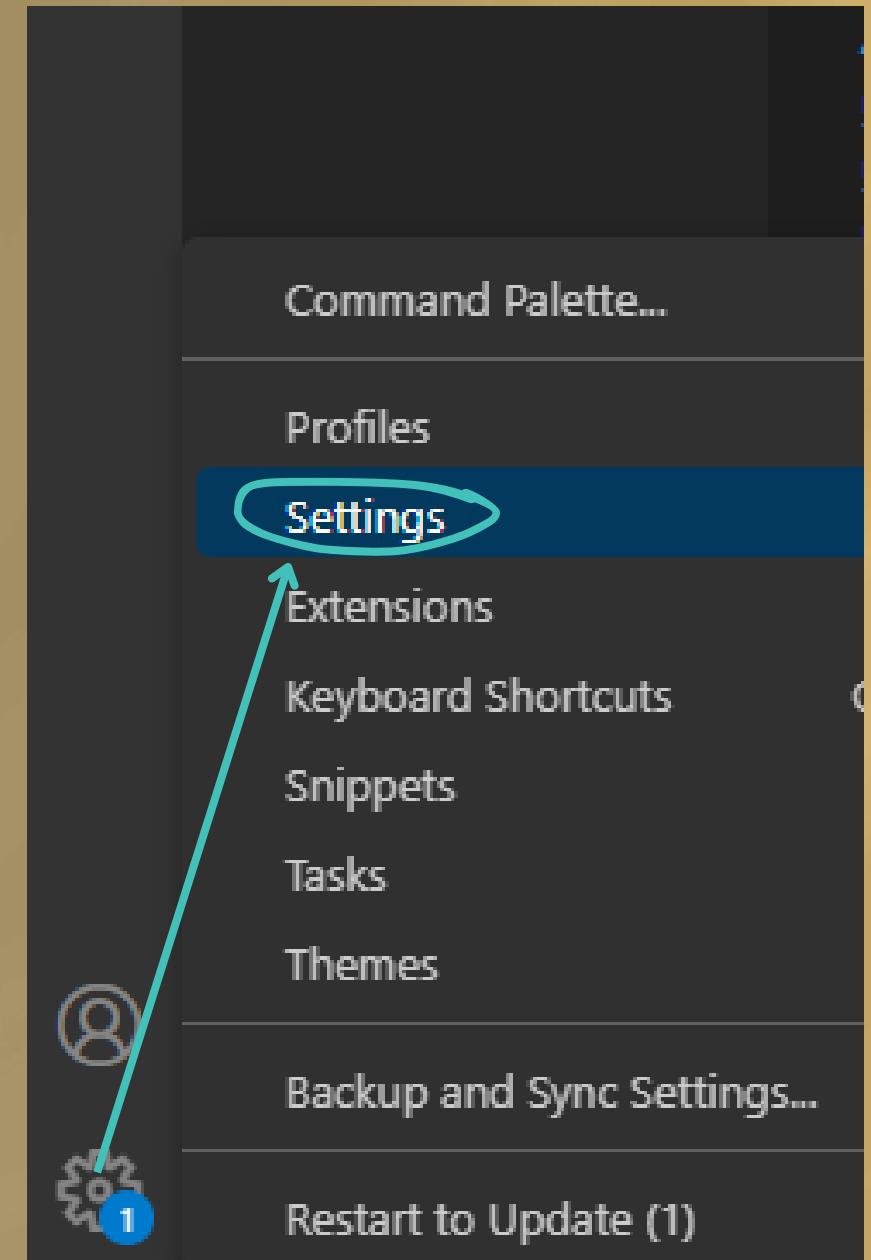
# VS code Python Tricks and Tips



## Accessing Settings



Shortcut : Access from Lower left corner



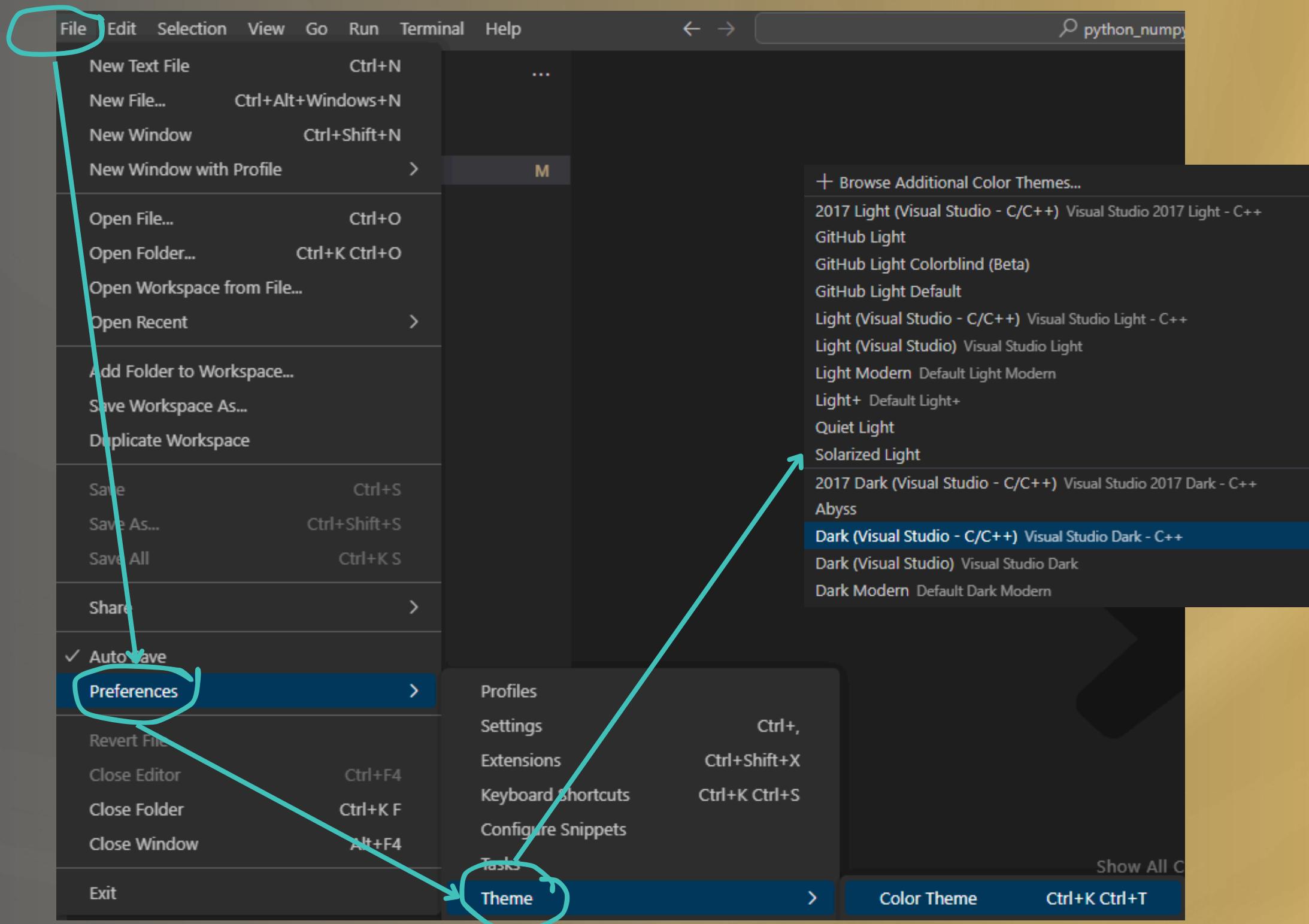
# VS code Python Tricks and Tips



## Choosing color theme



File -> Preferences -> Theme->**Color Theme**  
-> Choose from options available



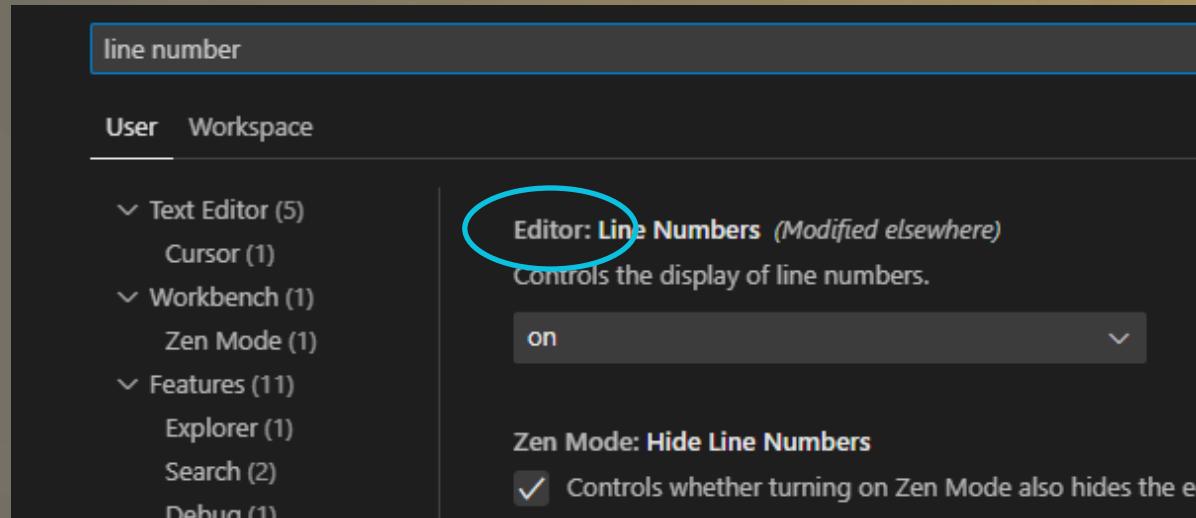
# VS code Python Tricks and Tips



## Enabling Line number in Python Script



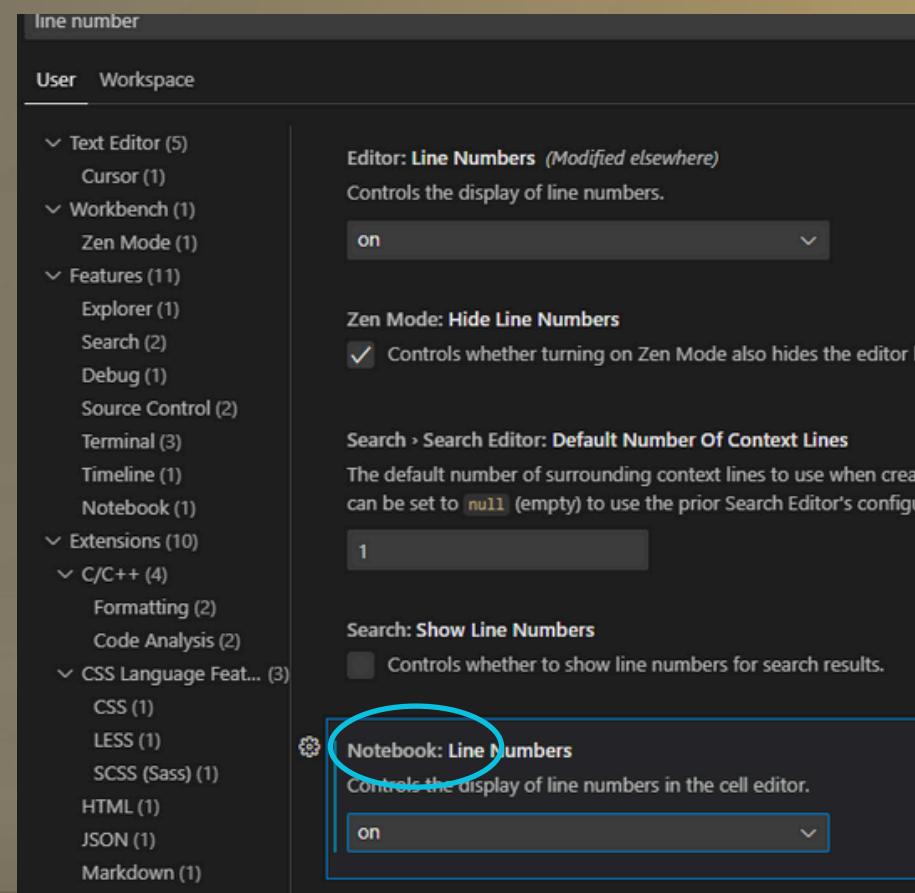
Settings-> Type Line number in Search Box  
->Editor Line Numbers -> Choose ON



## Enabling Line number in Jupyter notebook



Settings-> Type Line number in Search Box  
->NoteBook Line Numbers -> Choose ON



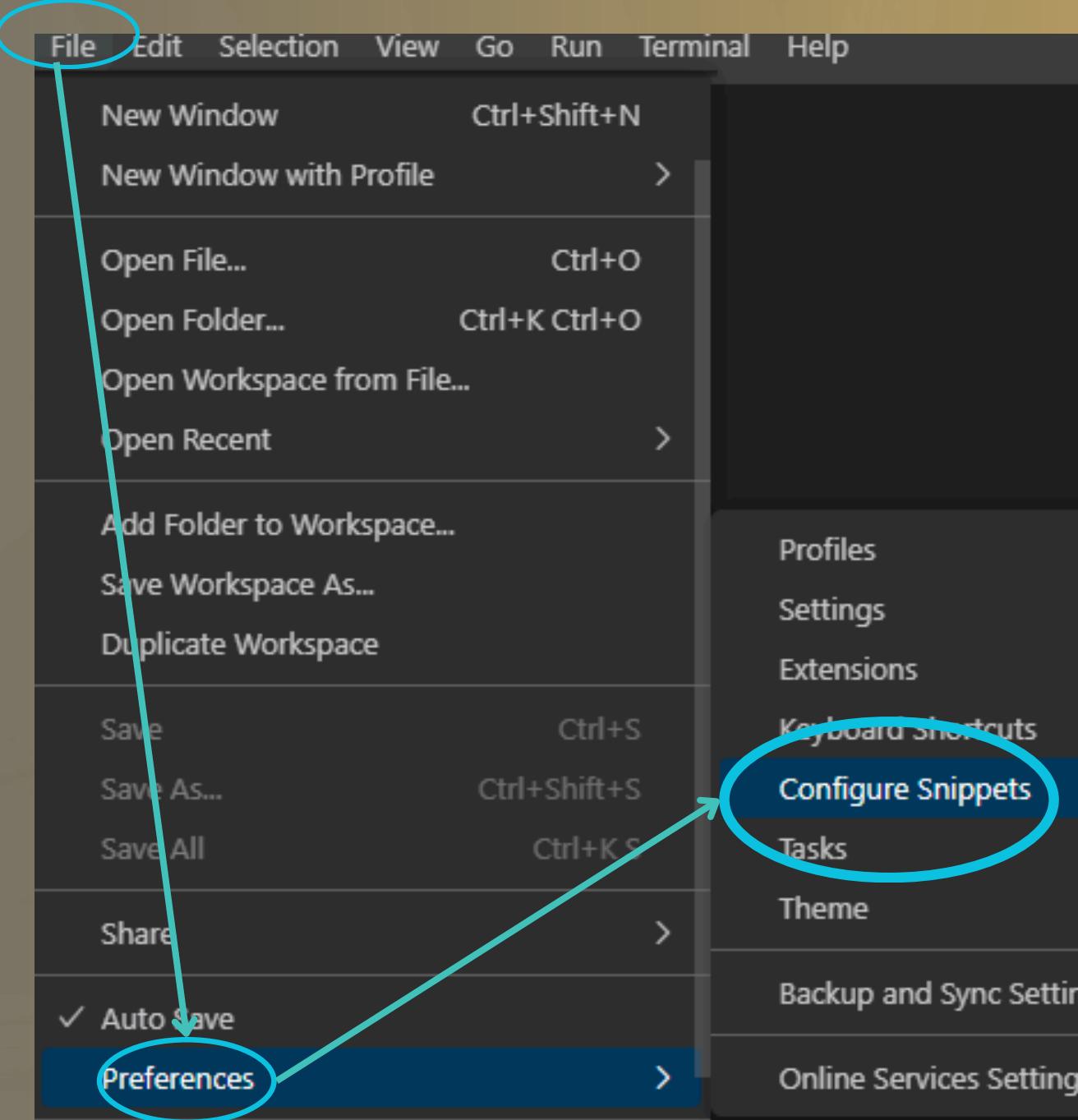
# VS code Python Tricks and Tips



## Code Snippets for repetitive tasks



File -> Preferences -> Theme->**Color Theme**  
-> Choose from options available

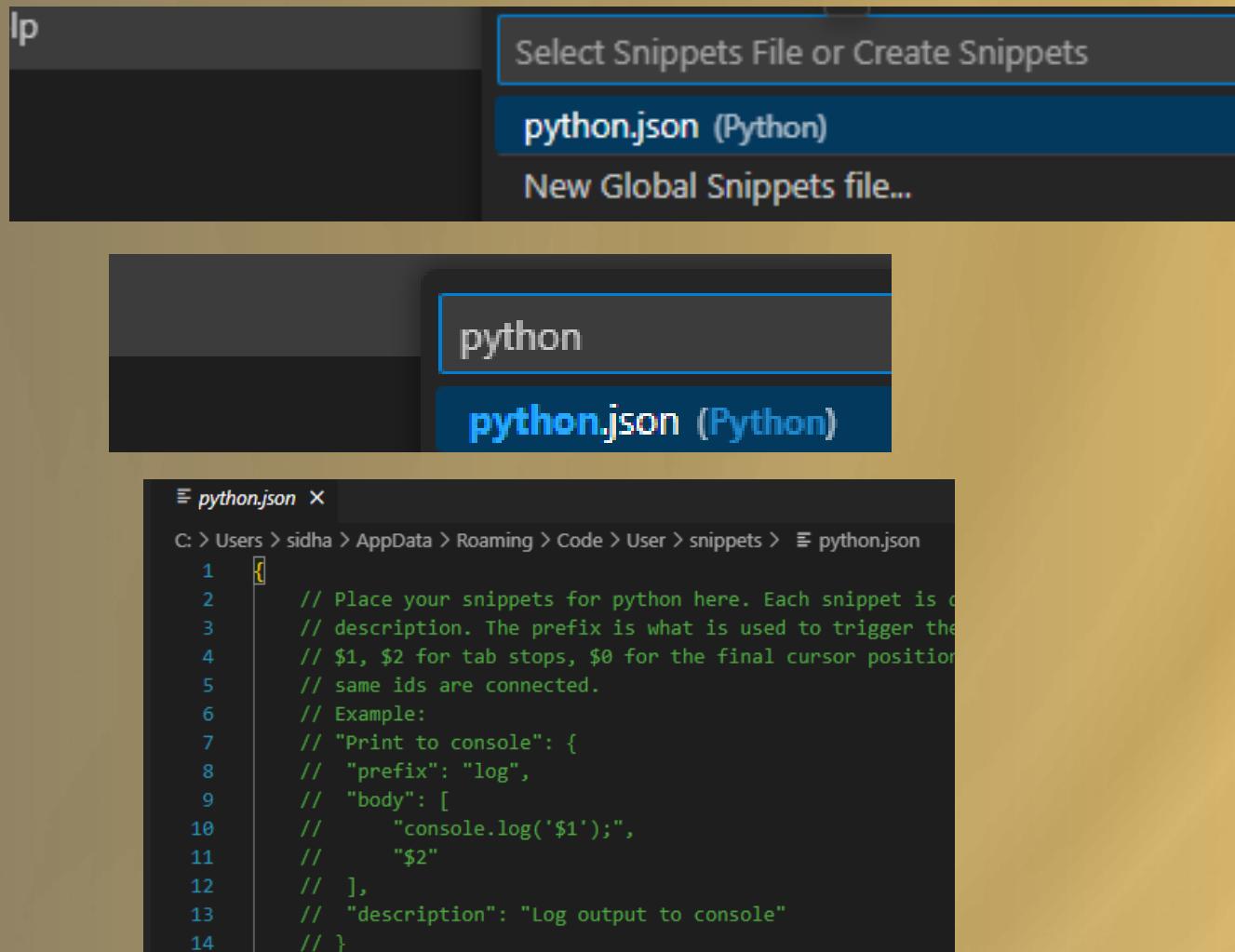


# VS code Python Tricks and Tips



**Code Snippets for repetitive tasks:  
Usage**

Type Python in Command Palette



Modify python json file to add snippet: e.g “numpy” for “import numpy as np”



Save the file ->Typing **numpy** will expand to **import numpy as np.**



# VS code Python Tricks and Tips



## Integrated Command terminal



Ability to run commands within terminal without leaving IDE

```
PS C:\ML\env\tf\repos\python_numpy_tutorials> ls
```

Directory: C:\ML\env\tf\repos\python\_numpy\_tutorials

Mode	LastWriteTime	Length	Name
-a---	12/17/2024 10:47 PM	29	LICENSE.md
-a---	12/21/2024 4:59 PM	22098	numpy_v1.ipynb
-a---	12/21/2024 9:48 PM	17014	numpy_v1.py
-a---	12/18/2024 4:09 PM	76612	numpy_v2.ipynb
-a---	12/18/2024 4:14 PM	14940	numpy_v2.py
-a---	12/18/2024 4:09 PM	46577	numpy_v3.ipynb
-a---	2/10/2021 3:48 PM	2947	numpy_v3.py
-a---	12/17/2024 10:47 PM	25	README.md

# VS code Python Tricks and Tips

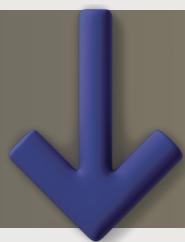


## Partial code run



Ability to run Specific portions of the code like cells in Jupyter notebook

Select a portion of the code



Shift + Enter

```
33
34  # ****Create a 1D array of numbers from 0 to
35  x1 = np.arange(10) # x1 = 0:9
36  print("x1.shape: " + str(x1.shape)) # (10,)
```



Execution Output

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

>>> x1 = np.arange(10) # x1 = 0:9
>>> print("x1.shape: " + str(x1.shape)) # (10,)
x1.shape: (10,)
```

# VS code Python Tricks and Tips



## Split Editor View

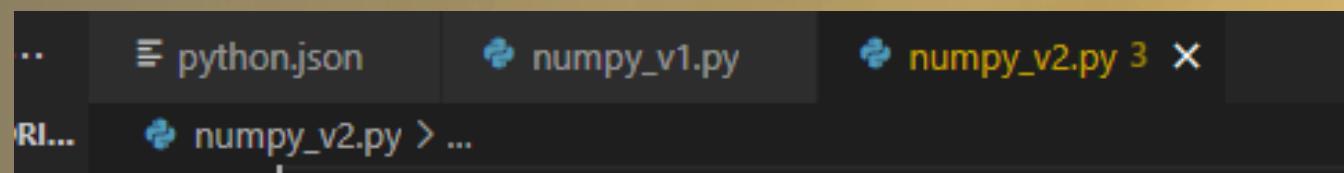


Work on multiple files side-by-side using split editor functionality (Ctrl+\).

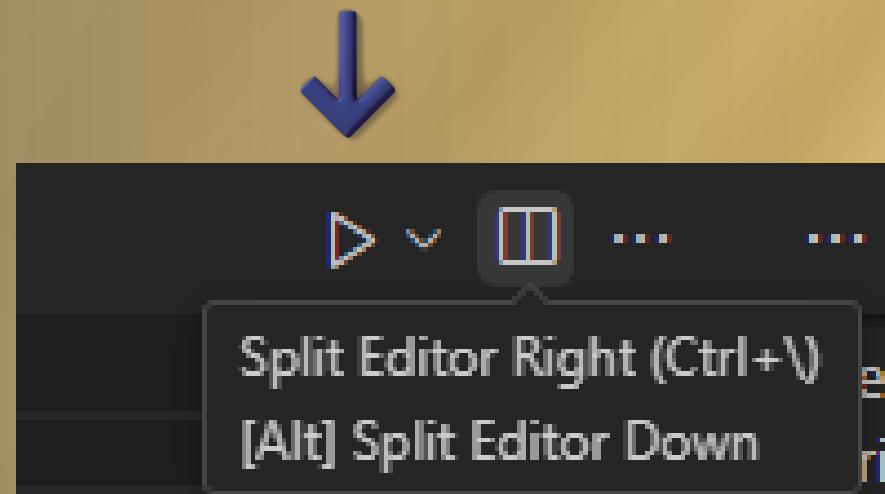


Especially helpful when writing unit tests alongside main code

**Two files opened in two tabs**



Press Ctrl + \ or The icon as shown



**Result as Split view of Chosen files**

```
python.json    numpy_v1.py ×
...
numpy_v1.py > ...
1  from __future__ import print_function
2  import numpy as np
3  from sklearn.preprocessing import
4  import random
5  import time
6
7  def chooseRandom1(x):
8      return np.random.choice(x,np.
```

```
...    numpy_v2.py 3 ×
...
numpy_v2.py > ...
1  from __future__ import
2  import os
3  import numpy as np
4  import numpy as np
5  import matplotlib.pyplot as plt
6
7  # Define the function
```

# VS code Python Tricks and Tips



## Peek Definitions



Peek at function/class etc definition without leaving the current file, saving time during code exploration

**Hover over a function or class and press Alt+F12**



```
impy_v1.py > chooseRandom2
from __future__ import print_function
import numpy as np
from sklearn.preprocessing import PowerTransformer
import random
import time

def chooseRandom1(x):
    return np.random.choice(x,np.size(x))
def chooseRandom2(x):
    return random.sample(x.tolist())
def chooseRandom3(x):
```

"PowerTransformer" is not accessed Pylance  
(class) PowerTransformer  
Apply a power transform featurewise to make data more Gaussian-like.  
Power transforms are a family of parametric, monotonic transformations that are applied to make data more Gaussian-like. This is useful for modeling issues related to heteroscedasticity (non-constant variance), or other situations where normality is desired.  
Currently, PowerTransformer supports the Box-Cox transform and the Yeo-Johnson transform. The optional parameter for stabilizing

# VS code Python Tricks and Tips



## Performance Boost



Improves performance when working on large projects

**Exclude unnecessary files from indexing by adding them to .vscode/settings.json**

```
json
"files.exclude": {
    "**/__pycache__": true,
    "**/*.pyc": true
}
```

# VS code Python Tricks and Tips



## Git Integration



Simplifies version control workflows

Use built-in Git tools to commit, push,  
pull, and resolve merge conflicts  
directly within VS Code's Source

Control tab

Changes  
numpy\_v1.ipynb

Discard Changes

