

# Python for Web Developers

## Learning Journal

### Objective

We find that the students who do particularly well in our courses are those who practice metacognition. **Metacognition** is the art of thinking about thinking; developing a deeper understanding of your own thought processes. With the help of this Learning Journal, you'll broaden your metacognitive knowledge and skills by reflecting on what you learn in this course.

Thanks to this Learning Journal, when you finish the course you'll have a complete and detailed record of your learning journey and progress over time. We really recommend that you take the time to complete this Journal; students do better in CF courses and in the working world as a result!

### Directions

First, complete the pre-work section before you start your course. Then, once you've begun learning, take time after each Exercise to return to this Journal and respond to the prompts.

There will be 3 to 5 prompts per Exercise, and we recommend spending about 10 to 15 minutes in total answering them. Don't overthink it—just write whatever comes to mind!

Also make sure that, once you've started filling this document in, you upload it as a deliverable on the platform. This is so that your mentor can also see your Journal and how you're progressing over time. Don't worry though—what you write here won't affect how you're graded for the Exercise tasks. The learning journal is mostly for you and your self-evaluation!

# Pre-Work: Before You Start the Course

Reflection questions (to complete before your first mentor call)

- What experiences have you had with coding and/or programming so far? What other experiences (programming-related or not) have you had that may help you as you progress through this course?

- At the community college I went to I took Java, C# and Web Development(HTML, CSS, JS, php) which I need more practice.

I worked as a tax officer that helped me to be more detail oriented and a better problem solver.

Since October-2023, I have been attending this course, learning more of JS, React, Angular, Nonerelational Database Mongo, APIs, TDD and more.

- What do you know about Python already? What do you want to know?
  - I don't have any experience with Python worth to mention.

- What challenges do you think may come up while you take this course? What will help you face them? Think of specific spaces, people, and times of day of week that might be favorable to your facing challenges and growing. Plan for how to solve challenges that arise.

- I believe that I may encounter version compatibility issues and problems or bugs caused by wrong indentations.

Remember, you can always refer to Exercise 1.4 of the Orientation course if you're not sure whom to reach out to for help and support.

## Exercise 1.1: Getting Started with Python

Learning Goals

- Summarize the uses and benefits of Python for web development  
Python is a beginner-friendly language known for its simplicity and readability, making it an excellent choice for those starting their programming journey. It's versatile in web development, data analysis,

automation, and game development. A vast community and extensive libraries provide support and resources, providing a smooth learning curve.

- Prepare your developer environment for programming with Python

### Step 1: Install Python

Check whether you have Python installed on your computer. Type in your terminal;

```
python --version
```

```
python2 --version
```

```
python3 --version
```

```
alexisabner@alexiss-air ~ % python3 --version
Python 3.12.4
```

### Step 2: Set up a new virtual environment and name it "cf-python-base".

```
alexisabner@alexiss-air ~ % pip3 install virtualenvwrapper
Collecting virtualenvwrapper
  Using cached virtualenvwrapper-6.1.0-py3-none-any.whl.metadata (5.1 kB)
Collecting virtualenv (from virtualenvwrapper)
  Using cached virtualenv-20.26.3-py3-none-any.whl.metadata (4.5 kB)
Collecting virtualenv-clone (from virtualenvwrapper)
  Using cached virtualenv_clone-0.5.7-py3-none-any.whl.metadata (2.7 kB)
Collecting stevedore (from virtualenvwrapper)
  Using cached stevedore-5.2.0-py3-none-any.whl.metadata (2.3 kB)
Collecting pbr!=2.1.0,>=2.0.0 (from stevedore->virtualenvwrapper)
  Using cached pbr-6.0.0-py2.py3-none-any.whl.metadata (1.3 kB)
Collecting distlib<1,>=0.3.7 (from virtualenv->virtualenvwrapper)
  Using cached distlib-0.3.8-py2.py3-none-any.whl.metadata (5.1 kB)
Collecting filelock<4,>=3.12.2 (from virtualenv->virtualenvwrapper)
  Using cached filelock-3.15.4-py3-none-any.whl.metadata (2.9 kB)
Collecting platformdirs<5,>=3.9.1 (from virtualenv->virtualenvwrapper)
  Using cached platformdirs-4.2.2-py3-none-any.whl.metadata (11 kB)
Using cached virtualenvwrapper-6.1.0-py3-none-any.whl (22 kB)
Using cached stevedore-5.2.0-py3-none-any.whl (49 kB)
Using cached virtualenv-20.26.3-py3-none-any.whl (5.7 MB)
Using cached virtualenv_clone-0.5.7-py3-none-any.whl (6.6 kB)
Using cached distlib-0.3.8-py2.py3-none-any.whl (468 kB)
Using cached filelock-3.15.4-py3-none-any.whl (16 kB)
Using cached pbr-6.0.0-py2.py3-none-any.whl (107 kB)
Using cached platformdirs-4.2.2-py3-none-any.whl (18 kB)
Installing collected packages: distlib, virtualenv-clone, platformdirs, pbr, filelock, virtualenv, stevedore, virtualenvwrapper
Successfully installed distlib-0.3.8 filelock-3.15.4 pbr-6.0.0 platformdirs-4.2.2 stevedore-5.2.0 virtualenv-20.26.3 virtualenv-clone-0.5.7 virtualenvwrapper-6.1.0
```

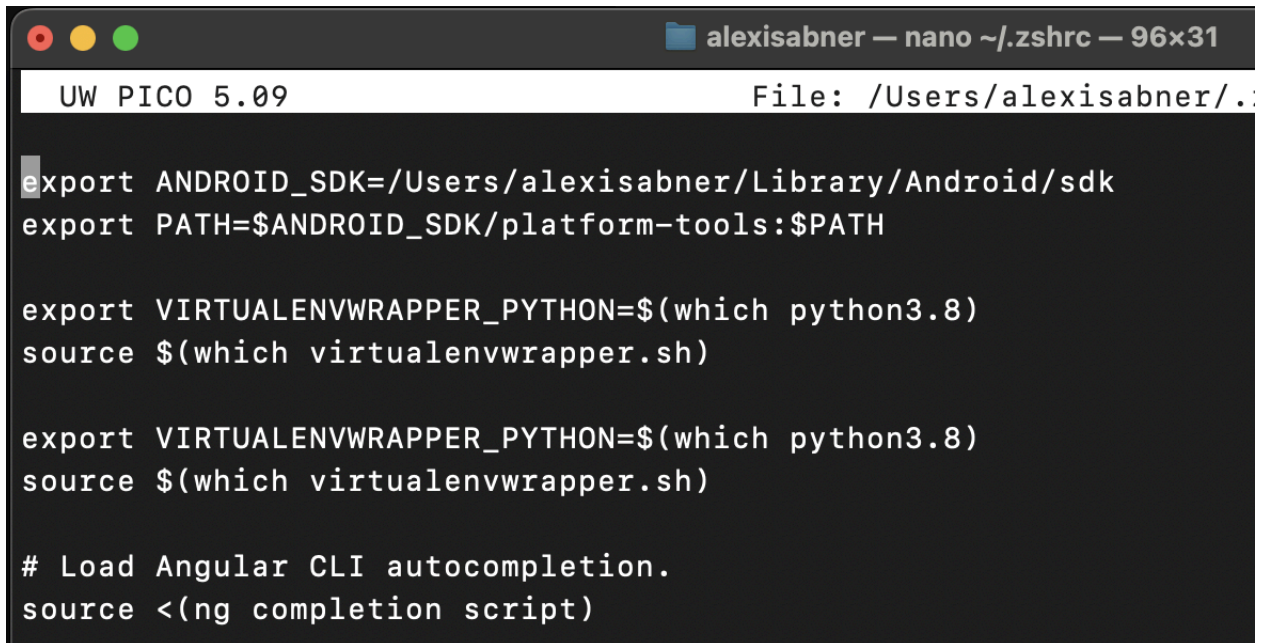
Creating backup;

```
alexisabner@alexiss-air ~ % sudo cp ~/.zshrc ~/.zshrc-backup
[Password:
alexisabner@alexiss-air ~ % ls -la ~/.zshrc-backup
-rw-r--r--@ 1 root  staff  264 Jul 30 20:15 /Users/alexisabner/.zshrc-backup
```

Verifying the location of the “virtualenvwrapper.sh” file:

```
alexisabner@alexiss-air ~ % which virtualenvwrapper.sh
/opt/homebrew/bin/virtualenvwrapper.sh
```

Modify the shell startup file



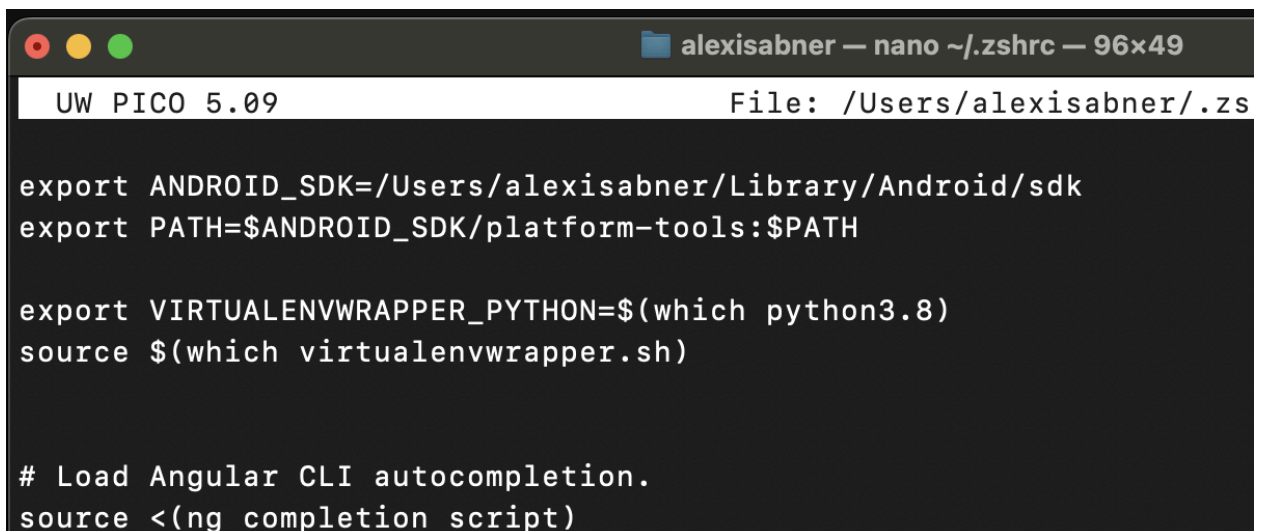
```
alexisabner — nano ~/.zshrc — 96x31
UW PICO 5.09 File: /Users/alexisabner/.

export ANDROID_SDK=/Users/alexisabner/Library/Android/sdk
export PATH=$ANDROID_SDK/platform-tools:$PATH

export VIRTUALENVWRAPPER_PYTHON=$(which python3.8)
source $(which virtualenvwrapper.sh)

export VIRTUALENVWRAPPER_PYTHON=$(which python3.8)
source $(which virtualenvwrapper.sh)

# Load Angular CLI autocompletion.
source <(ng completion script)
```



```
alexisabner — nano ~/.zshrc — 96x49
UW PICO 5.09 File: /Users/alexisabner/.zs

export ANDROID_SDK=/Users/alexisabner/Library/Android/sdk
export PATH=$ANDROID_SDK/platform-tools:$PATH

export VIRTUALENVWRAPPER_PYTHON=$(which python3.8)
source $(which virtualenvwrapper.sh)

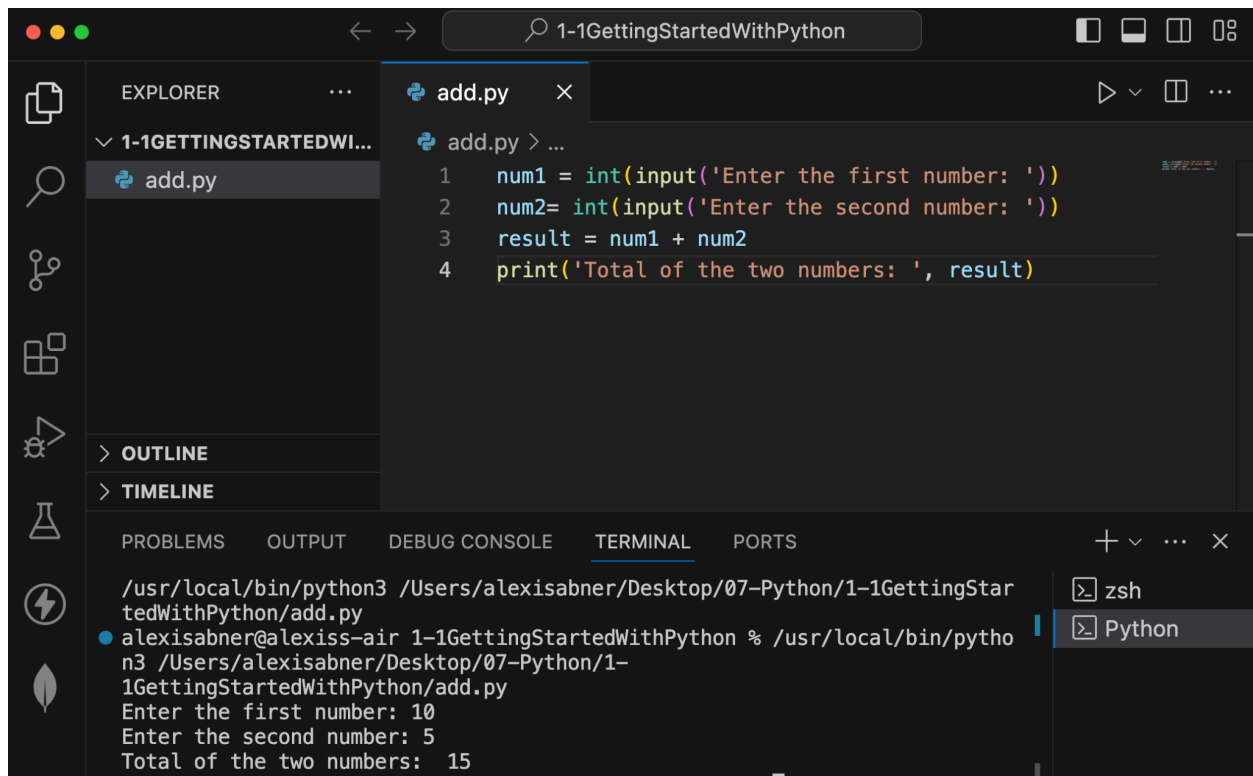
# Load Angular CLI autocompletion.
source <(ng completion script)
```

## Creating a new Virtual Environment

```
alexisabner@alexiss-air ~ % mkvirtualenv cf-python-base
created virtual environment CPython3.11.3.final.0-64 in 215ms
  creator CPython3Posix(dest=/Users/alexisabner/.virtualenvs/cf-python-base, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/Users/alexisabner/Library/Application Support/virtualenv)
  added seed packages: pip==24.1.2, setuptools==70.2.0, wheel==0.43.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-base/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-base/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-base/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-base/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-base/bin/get_env_details
(cf-python-base) alexisabner@alexiss-air ~ %
```

## 3. Install VSCode

VSCode has been installed at the beginning of this course.



The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer panel on the left shows a file named `add.py` under a workspace named `1-1GETTINGSTARTEDWI...`. The main editor area displays the contents of `add.py`:

```
1 num1 = int(input('Enter the first number: '))
2 num2 = int(input('Enter the second number: '))
3 result = num1 + num2
4 print('Total of the two numbers: ', result)
```

The bottom panel shows the TERMINAL view. The terminal output is as follows:

```
/usr/local/bin/python3 /Users/alexisabner/Desktop/07-Python/1-1GettingStar
tedWithPython/add.py
alexisabner@alexiss-air 1-1GettingStartedWithPython % /usr/local/bin/pytho
n3 /Users/alexisabner/Desktop/07-Python/1-
1GettingStartedWithPython/add.py
Enter the first number: 10
Enter the second number: 5
Total of the two numbers: 15
```

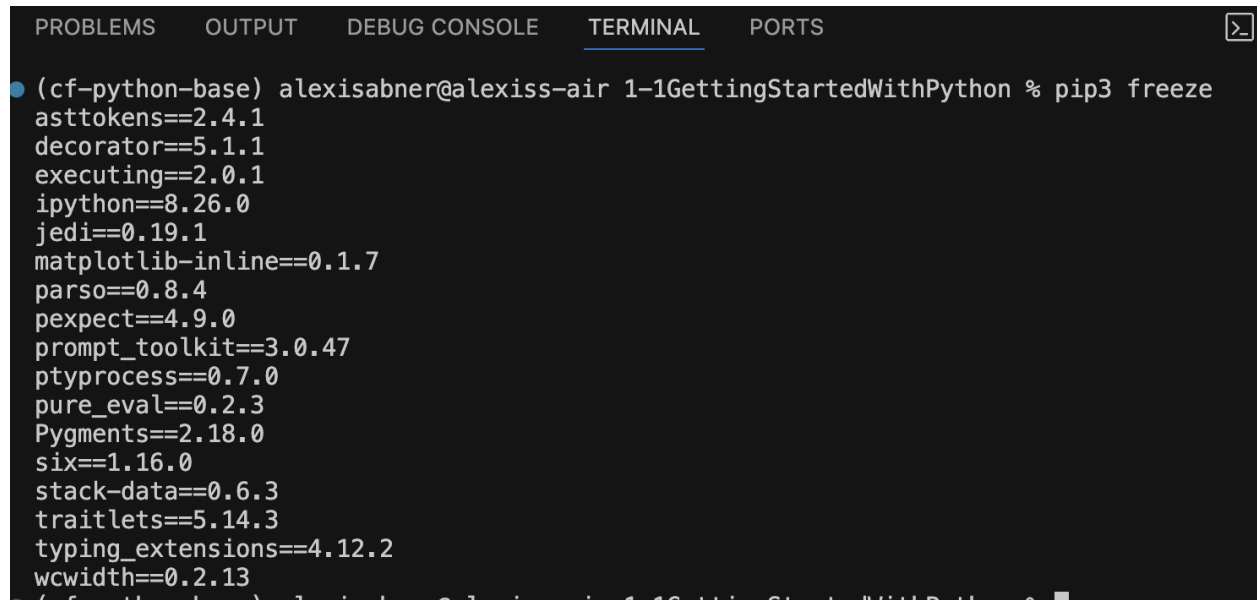
On the right side of the terminal panel, there is a dropdown menu showing the selected interpreter as `Python`, with `zsh` also visible.



## Setting up an IPython shell in the virtual environment “cf-python-base”.

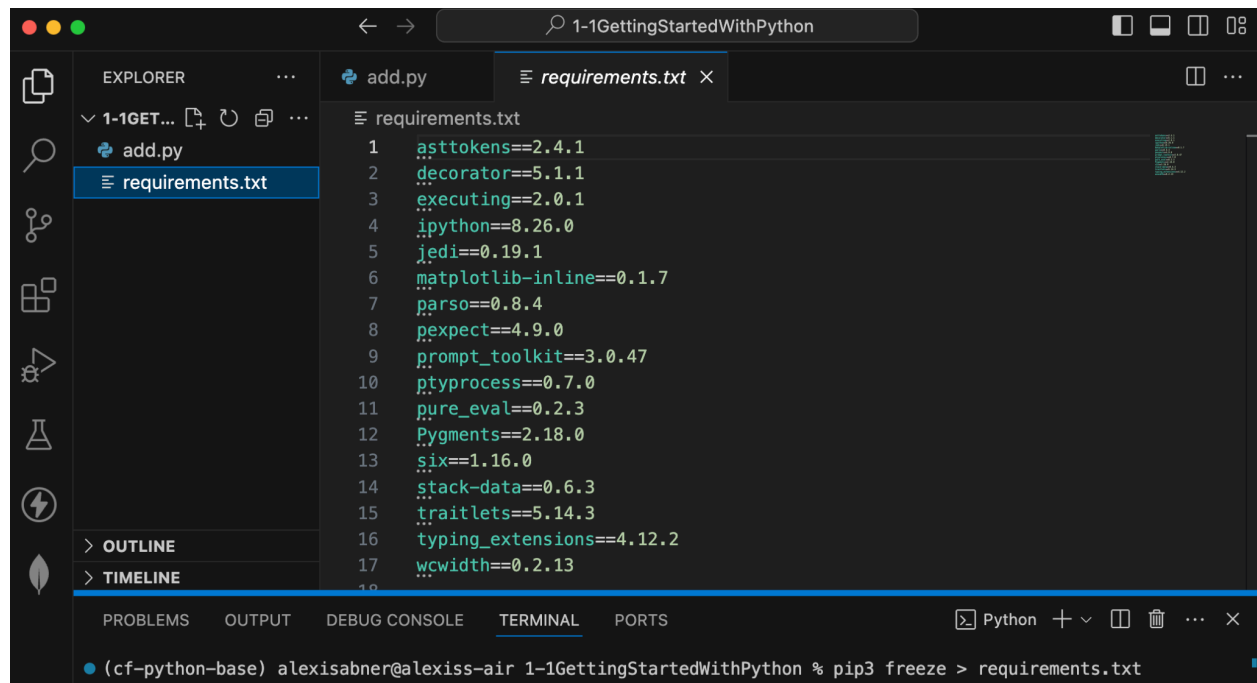
```
alexisabner — zsh — 96x46
[(cf-python-base) alexisabner@alexis-air ~ % pip3 install ipython
Collecting ipython
  Using cached ipython-8.26.0-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: decorator in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from ipython) (5.1.1)
Requirement already satisfied: jedi>=0.16 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from ipython) (0.19.1)
Requirement already satisfied: matplotlib-inline in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from ipython) (0.1.7)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from ipython) (3.0.47)
Requirement already satisfied: pygments>=2.4.0 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from ipython) (2.18.0)
Requirement already satisfied: stack-data in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from ipython) (0.6.3)
Requirement already satisfied: traitlets>=5.13.0 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from ipython) (5.14.3)
Requirement already satisfied: typing-extensions>=4.6 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from ipython) (4.12.2)
Requirement already satisfied: pexpect>4.3 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from ipython) (4.9.0)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from jedi>=0.16->ipython) (0.8.4)
Requirement already satisfied: ptyprocess>=0.5 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from pexpect>4.3->ipython) (0.7.0)
Requirement already satisfied: wcwidth in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from prompt-toolkit<3.1.0,>=3.0.41->ipython) (0.2.13)
Requirement already satisfied: executing>=1.2.0 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from stack-data->ipython) (2.0.1)
Requirement already satisfied: asttokens>=2.1.0 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from stack-data->ipython) (2.4.1)
Requirement already satisfied: pure-eval in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from stack-data->ipython) (0.2.3)
Requirement already satisfied: six>=1.12.0 in ./virtualenvs/cf-python-base/lib/python3.11/site-packages (from asttokens>=2.1.0->stack-data->ipython) (1.16.0)
Using cached ipython-8.26.0-py3-none-any.whl (817 kB)
Installing collected packages: ipython
Successfully installed ipython-8.26.0
```

## Requirements File



The terminal window shows the output of the 'pip3 freeze' command. The output lists various Python packages and their versions, including asttokens, decorator, executing, ipython, jedi, matplotlib-inline, parso, pexpect, prompt\_toolkit, ptyprocess, pure\_eval, Pygments, six, stack-data, traitlets, typing\_extensions, and wcwidth.

```
(cf-python-base) alexisabner@alexis-air 1-1GettingStartedWithPython % pip3 freeze
asttokens==2.4.1
decorator==5.1.1
executing==2.0.1
ipython==8.26.0
jedi==0.19.1
matplotlib-inline==0.1.7
parso==0.8.4
pexpect==4.9.0
prompt_toolkit==3.0.47
ptyprocess==0.7.0
pure_eval==0.2.3
Pygments==2.18.0
six==1.16.0
stack-data==0.6.3
traitlets==5.14.3
typing_extensions==4.12.2
wcwidth==0.2.13
```



The VS Code editor shows the 'requirements.txt' file in the Explorer pane. The file contains the same list of packages and versions as the terminal output. The file is named 'requirements.txt' and is located in the '1-1GettingStartedWithPython' directory. The editor also shows the 'add.py' file in the Explorer pane. The terminal pane at the bottom shows the command 'pip3 freeze' and the output 'requirements.txt'.

```
1 asttokens==2.4.1
2 decorator==5.1.1
3 executing==2.0.1
4 ipython==8.26.0
5 jedi==0.19.1
6 matplotlib-inline==0.1.7
7 parso==0.8.4
8 pexpect==4.9.0
9 prompt_toolkit==3.0.47
10 ptyprocess==0.7.0
11 pure_eval==0.2.3
12 Pygments==2.18.0
13 six==1.16.0
14 stack-data==0.6.3
15 traitlets==5.14.3
16 typing_extensions==4.12.2
17 wcwidth==0.2.13
```

## Creating a new virtual environment named "cf-python-copy"

```
alexisabner — -zsh — 96x22
(cf-python-base) alexisabner@alexiss-air ~ % mkvirtualenv cf-python-copy
created virtual environment CPython3.12.4.final.0-64 in 212ms
  creator CPython3Posix(dest=/Users/alexisabner/.virtualenvs/cf-python-copy, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/Users/alexisabner/Library/Application Support/virtualenv)
    added seed packages: pip==24.1
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-copy/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-copy/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-copy/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-copy/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/alexisabner/.virtualenvs/cf-python-copy/bin/get_env_details
```



## Installing the packages from cf-python-base requirements.txt

```
alexisabner — zsh — 127x61

(cf-python-copy) alexisabner@alexis-air ~ % pip3 install -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt
Collecting asttokens==2.4.1 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 1))
  Using cached asttokens-2.4.1-py2.py3-none-any.whl.metadata (5.2 kB)
Collecting decorator==5.1.1 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 2))
  Using cached decorator-5.1.1-py3-none-any.whl.metadata (4.0 kB)
Collecting executing==2.0.1 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 3))
  Using cached executing-2.0.1-py2.py3-none-any.whl.metadata (9.0 kB)
Collecting ipython==8.26.0 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 4))
  Using cached ipython-8.26.0-py3-none-any.whl.metadata (5.0 kB)
Collecting jedi==0.19.1 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 5))
  Using cached jedi-0.19.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting matplotlib-inline==0.1.7 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 6))
  Using cached matplotlib-inline-0.1.7-py3-none-any.whl.metadata (3.9 kB)
Collecting parso==0.8.4 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 7))
  Using cached parso-0.8.4-py2.py3-none-any.whl.metadata (7.7 kB)
Collecting pexpect==4.9.0 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 8))
  Using cached pexpect-4.9.0-py2.py3-none-any.whl.metadata (2.5 kB)
Collecting prompt_toolkit==3.0.47 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 9))
  Using cached prompt_toolkit-3.0.47-py3-none-any.whl.metadata (6.4 kB)
Collecting ptyprocess==0.7.0 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 10))
  Using cached ptyprocess-0.7.0-py2.py3-none-any.whl.metadata (1.3 kB)
Collecting pure_eval==0.2.3 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 11))
  Using cached pure_eval-0.2.3-py3-none-any.whl.metadata (6.3 kB)
Collecting Pygments==2.18.0 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 12))
  Using cached pygments-2.18.0-py3-none-any.whl.metadata (2.5 kB)
Collecting six==1.16.0 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 13))
  Using cached six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
Collecting stack-data==0.6.3 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 14))
  Using cached stack-data-0.6.3-py3-none-any.whl.metadata (18 kB)
Collecting traitlets==5.14.3 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 15))
  Using cached traitlets-5.14.3-py3-none-any.whl.metadata (10 kB)
Collecting typing_extensions==4.12.2 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 16))
  Using cached typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Collecting wcwidth==0.2.13 (from -r /Users/alexisabner/Desktop/07-Python/1-1GettingStartedWithPython/requirements.txt (line 17))
  Using cached wcwidth-0.2.13-py2.py3-none-any.whl.metadata (14 kB)
Using cached asttokens-2.4.1-py2.py3-none-any.whl (27 kB)
Using cached decorator-5.1.1-py3-none-any.whl (9.1 kB)
Using cached executing-2.0.1-py2.py3-none-any.whl (24 kB)
Using cached ipython-8.26.0-py3-none-any.whl (817 kB)
Using cached jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
Using cached matplotlib-inline-0.1.7-py3-none-any.whl (9.9 kB)
Using cached parso-0.8.4-py2.py3-none-any.whl (103 kB)
Using cached pexpect-4.9.0-py2.py3-none-any.whl (63 kB)
Using cached prompt_toolkit-3.0.47-py3-none-any.whl (386 kB)
Using cached ptyprocess-0.7.0-py2.py3-none-any.whl (13 kB)
Using cached pure_eval-0.2.3-py3-none-any.whl (11 kB)
Using cached pygments-2.18.0-py3-none-any.whl (1.2 MB)
Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
```

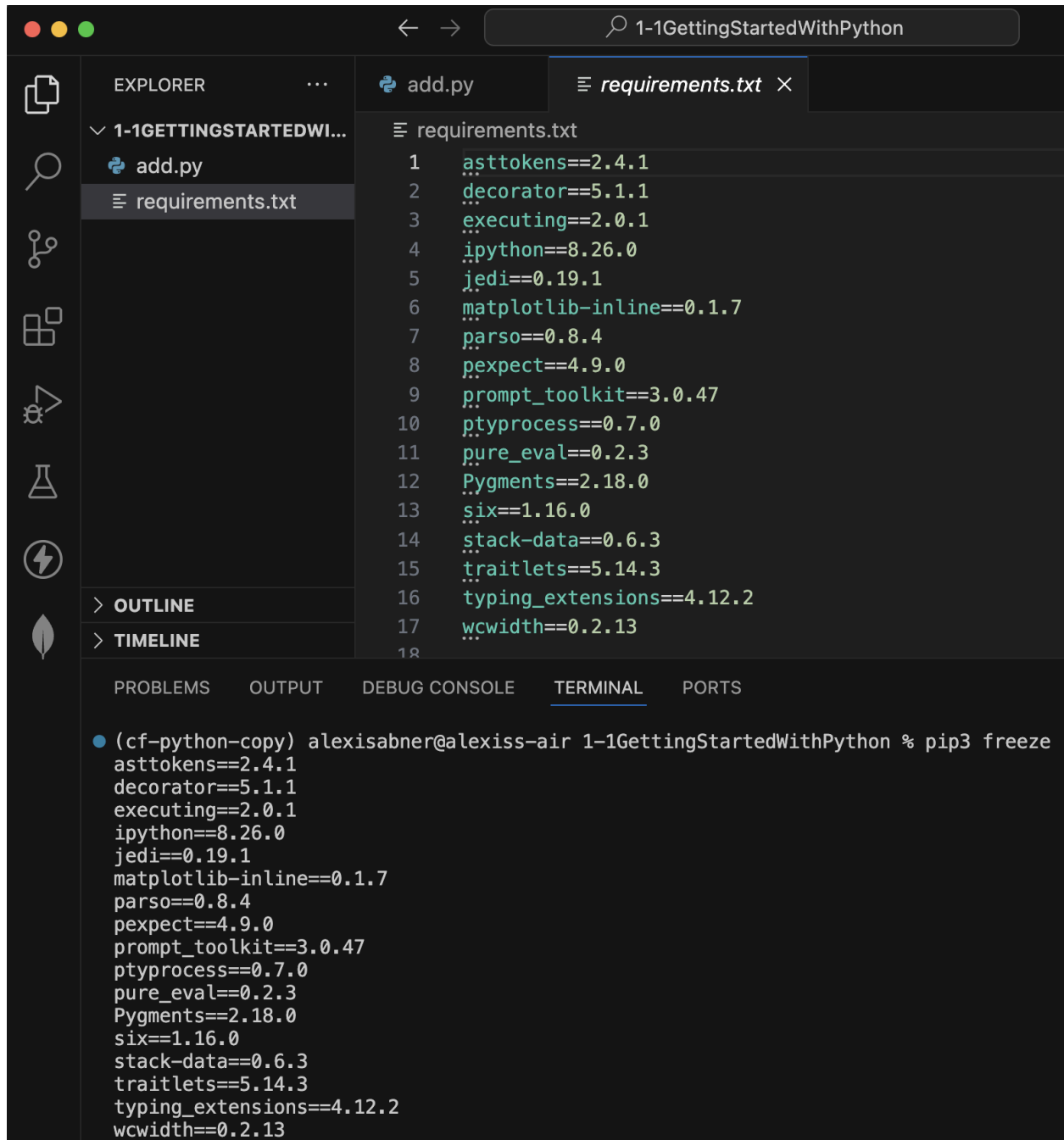
```
(cf-python-copy) alexisabner@alexiss-air ~ % pip list
```

Package	Version
---------	---------

-----	-----
-------	-------

asttokens	2.4.1
decorator	5.1.1
executing	2.0.1
ipython	8.26.0
jedi	0.19.1
matplotlib-inline	0.1.7
parso	0.8.4
pexpect	4.9.0
pip	24.1
prompt_toolkit	3.0.47
ptyprocess	0.7.0
pure_eval	0.2.3
Pygments	2.18.0
six	1.16.0
stack-data	0.6.3
traitlets	5.14.3
typing_extensions	4.12.2
wcwidth	0.2.13

```
(cf-python-copy) alexisabner@alexiss-air ~ %
```



The image shows a VS Code editor window with the file explorer on the left, the editor in the center, and the terminal at the bottom. The file explorer shows a project named '1-1GETTINGSTARTEDWI...' with files 'add.py' and 'requirements.txt'. The editor shows the contents of 'requirements.txt', which lists various Python packages and their versions. The terminal shows the output of the command 'pip3 freeze', which lists the same packages and versions as the requirements.txt file.

```
requirements.txt
1 asttokens==2.4.1
2 decorator==5.1.1
3 executing==2.0.1
4 ipython==8.26.0
5 jedi==0.19.1
6 matplotlib-inline==0.1.7
7 parso==0.8.4
8 pexpect==4.9.0
9 prompt_toolkit==3.0.47
10 ptyprocess==0.7.0
11 pure_eval==0.2.3
12 Pygments==2.18.0
13 six==1.16.0
14 stack-data==0.6.3
15 traitlets==5.14.3
16 typing_extensions==4.12.2
17 wcwidth==0.2.13
18

TERMINAL
• (cf-python-copy) alexisabner@alexiss-air 1-1GettingStartedWithPython % pip3 freeze
asttokens==2.4.1
decorator==5.1.1
executing==2.0.1
ipython==8.26.0
jedi==0.19.1
matplotlib-inline==0.1.7
parso==0.8.4
pexpect==4.9.0
prompt_toolkit==3.0.47
ptyprocess==0.7.0
pure_eval==0.2.3
Pygments==2.18.0
six==1.16.0
stack-data==0.6.3
traitlets==5.14.3
typing_extensions==4.12.2
wcwidth==0.2.13
```

## Reflection Questions

- In your own words, what is the difference between frontend and backend web development? If you were hired to work on backend programming for a web application, what kinds of operations would you be working on?

- Frontend is the UI part of an application. It is where the user interacts with the app.

Backend is where users retrieving, processing data, performing calculations, managing their accounts.

If I were hired as a backend developer, I would be creating APIs, testing them.

- Imagine you're working as a full-stack developer in the near future. Your team is asking for your advice on whether to use JavaScript or Python for a project, and you think Python would be the better choice. How would you explain the similarities and differences between the two languages to your team? Drawing from what you learned in this Exercise, what reasons would you give to convince your team that Python is the better option? (Hint: refer to the Exercise section "The Benefits of Developing with Python")

- Learning curve of Python is not so challenging.
- It uses indentation and clear syntax, similar to JavaScript, helps in writing clean and error-resistant code.
- Dynamic typing allows the developers to change types without errors.
- It has a very strong supporting community

- Now that you've had an introduction to Python, write down 3 goals you have for yourself and your learning during this Achievement. You can reflect on the following questions if it helps you. What do you want to learn about Python? What do you want to get out of this Achievement? Where or what do you see yourself working on after you complete this Achievement?

- I want to learn the basics of Python syntax and structure,
- I want to build a solid foundation in finding solutions,
- I want to create and complete the project successfully..