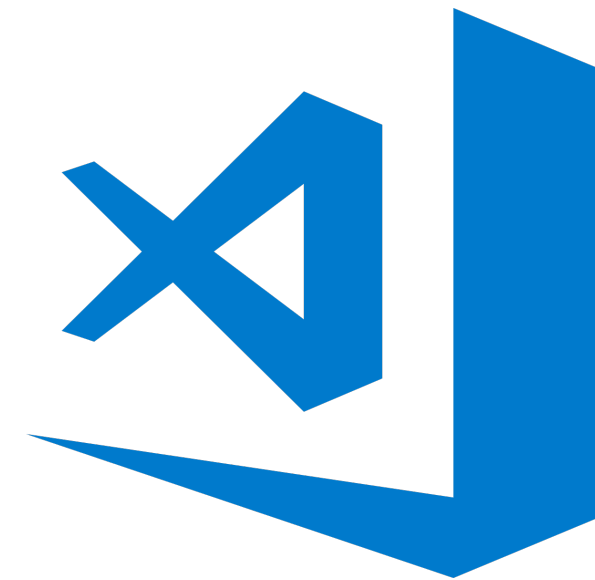


Creating virtual environment in Visual Studio Code for Python on macOS

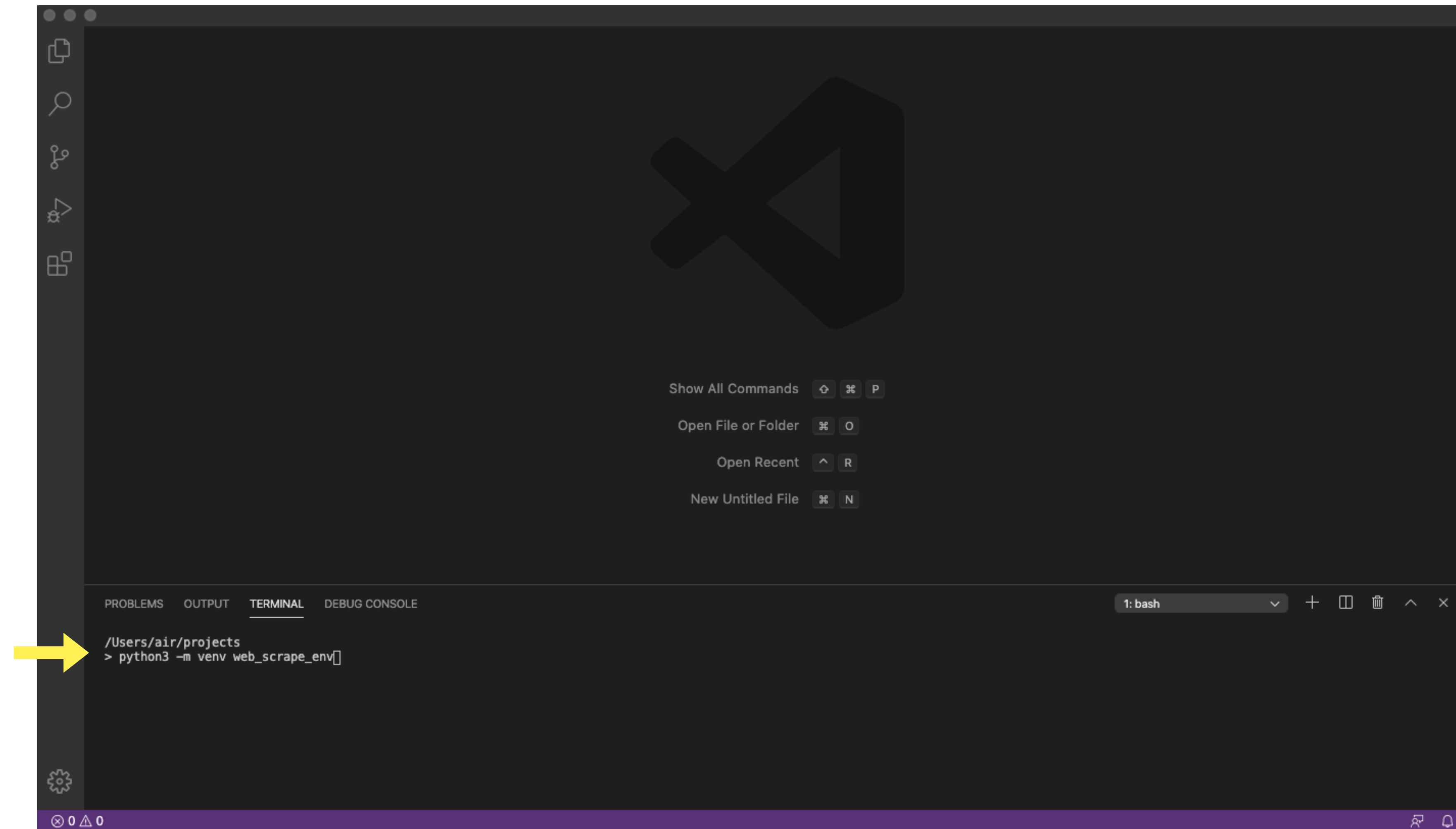


Will assume you have the following knowledge and your machine the following softwares:

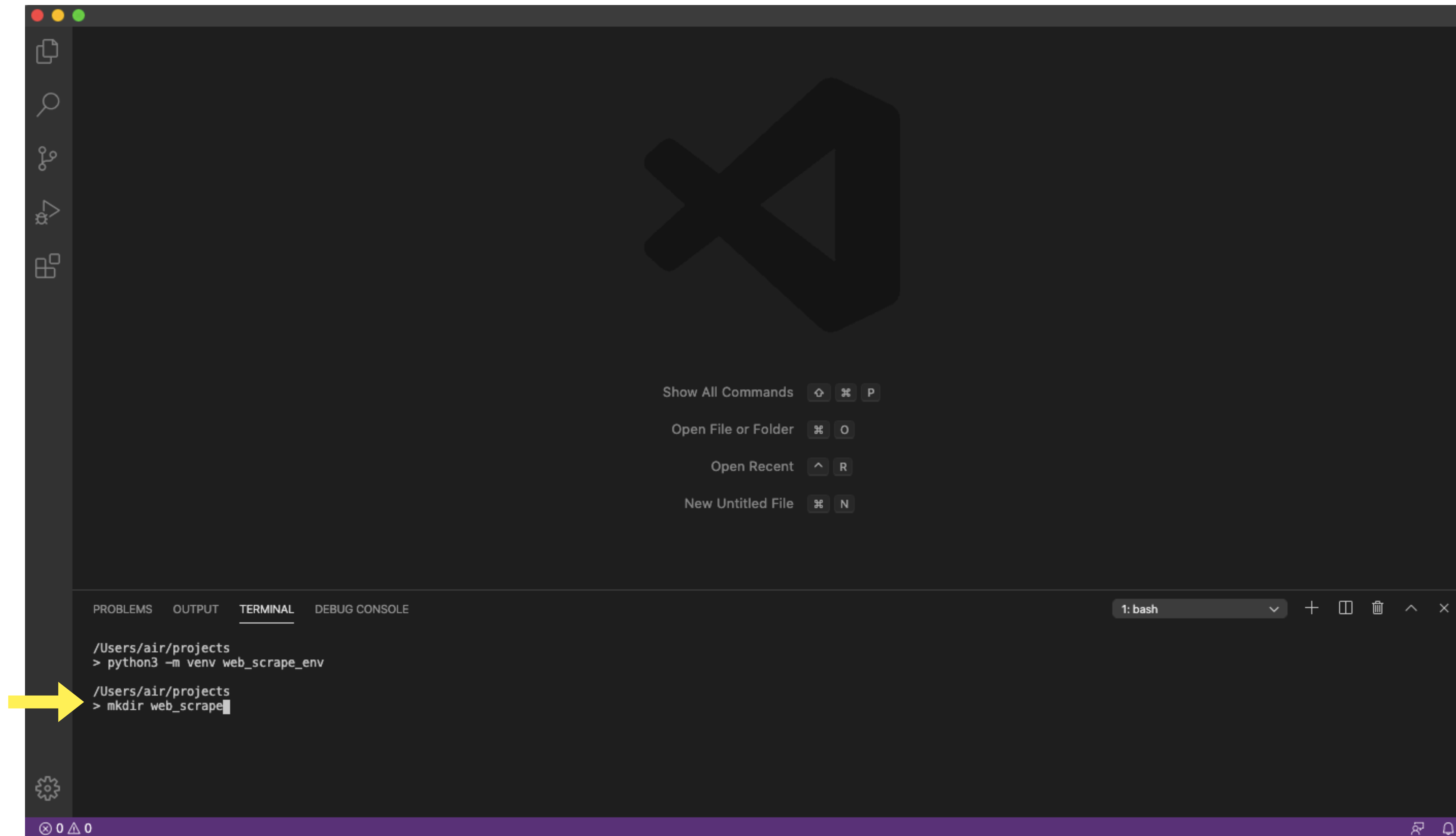
- Visual Studio Code (VSCode) with `<python>` and `<doraemon-runner>` extensions
- Python3
- Confident to use VS Code regarding folder (create/open), terminal (open/close), files (create/open/edit), install dependencies (when required), configuration (command palette)
- Know how to create folders and navigate them via terminal
- Know how to run a Python application `<python3 "file_name.py">`

Create a virtual environment to develop a Python application using VSCode

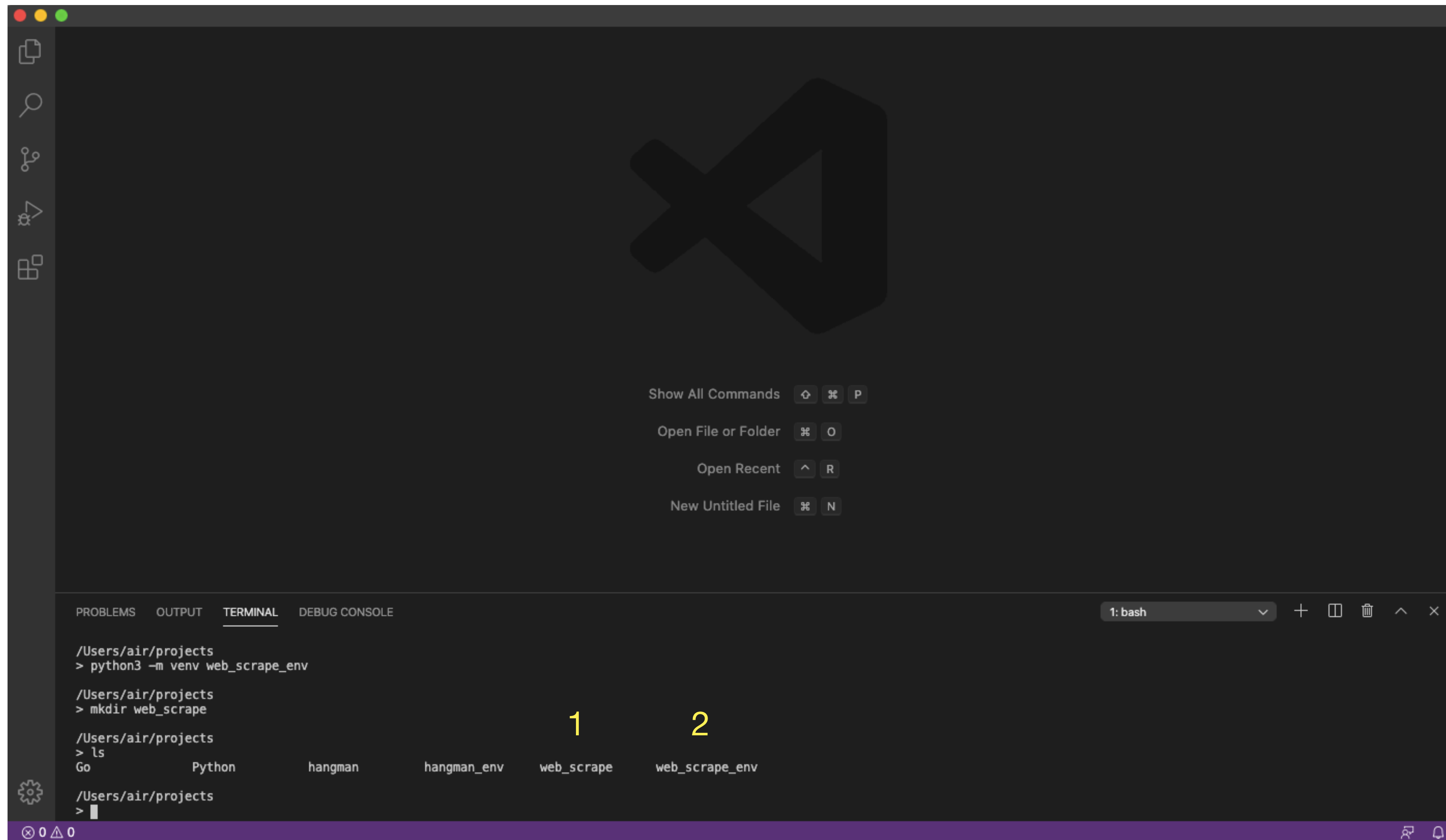
- Environment name: '*web_scrape_env*'
- Project folder: '*web_scrape*'
- Project file for test the environment: '*app.py*'
- Packages to install: '*BeautifulSoup4*' and '*Requests*'



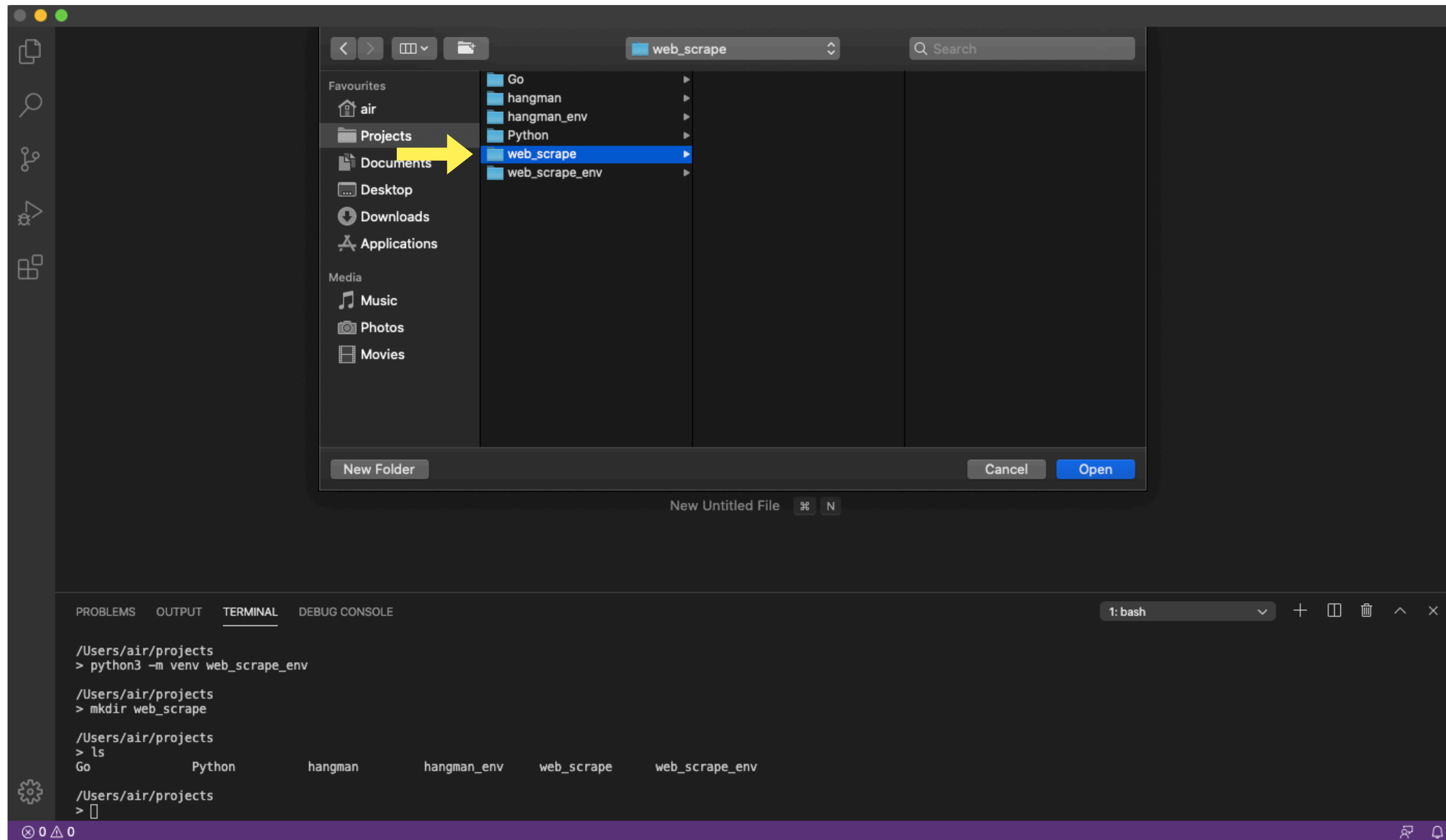
► in VSCode terminal (^`) create the virtual environment `<python3 -m venv web_scrape_env>`



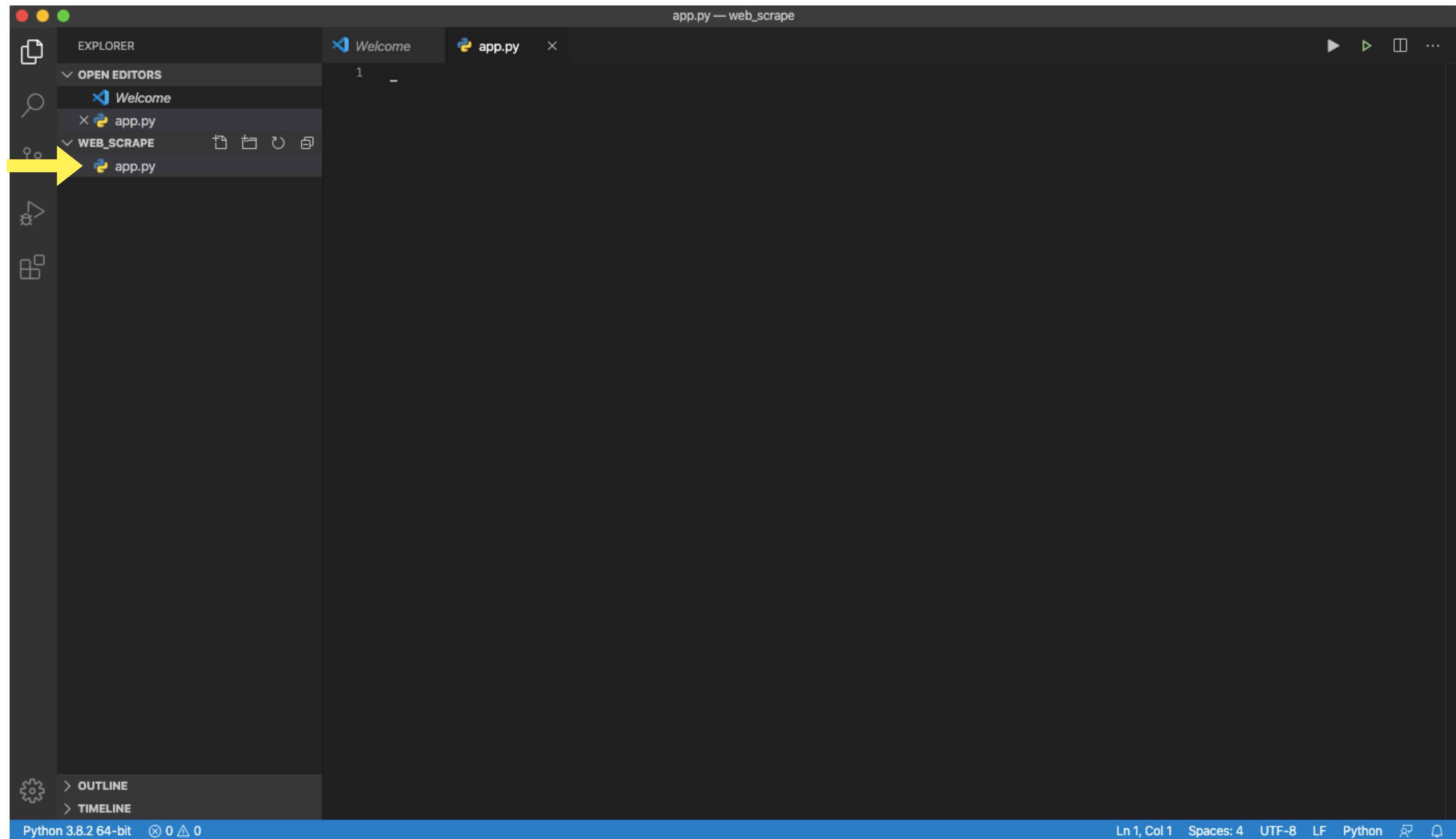
► create the folder 'web_scrape' <mkdir web_scrape>



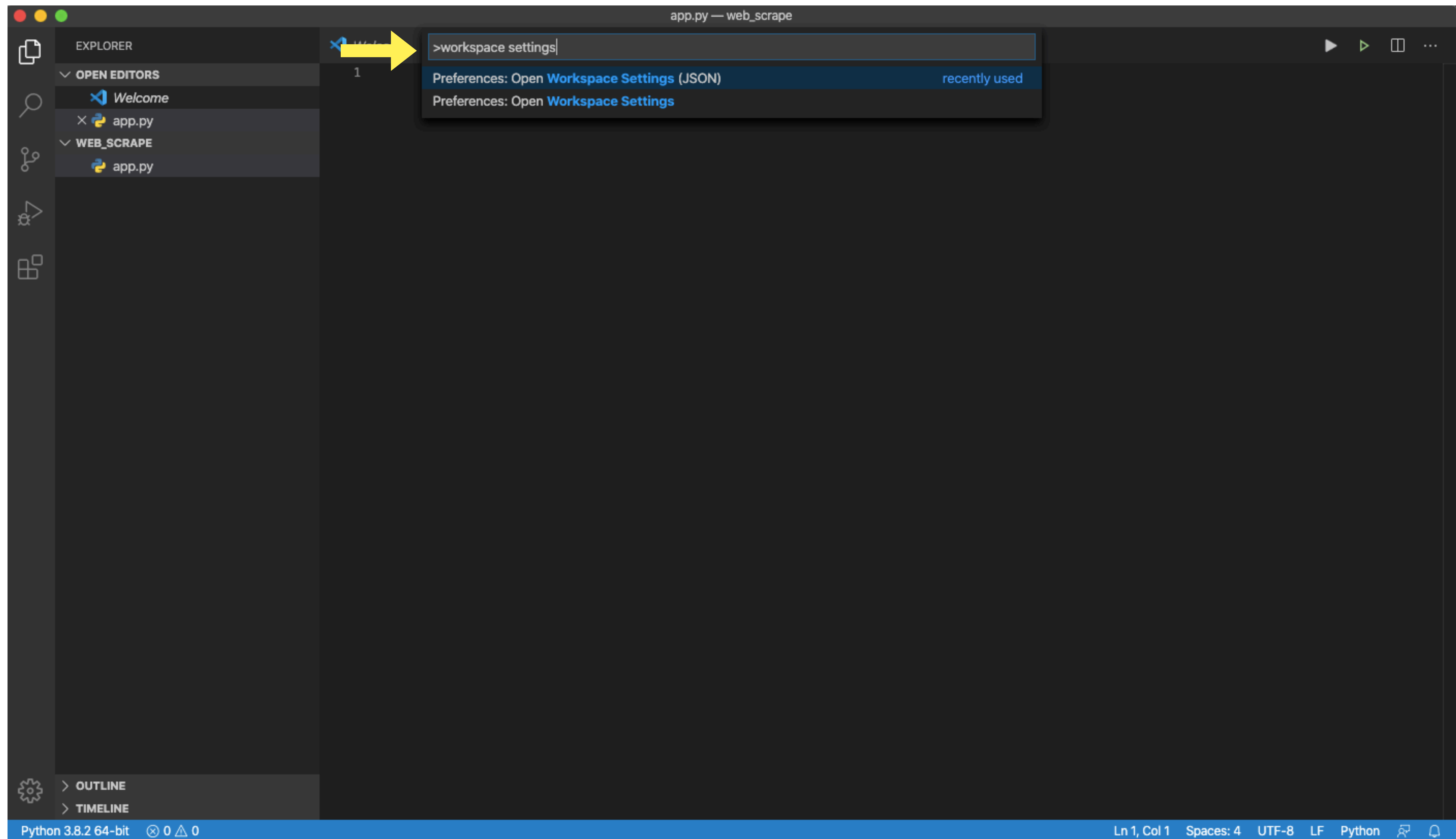
► both folders have been created (1)'web_scrape' and (2)'web_scrape_env'



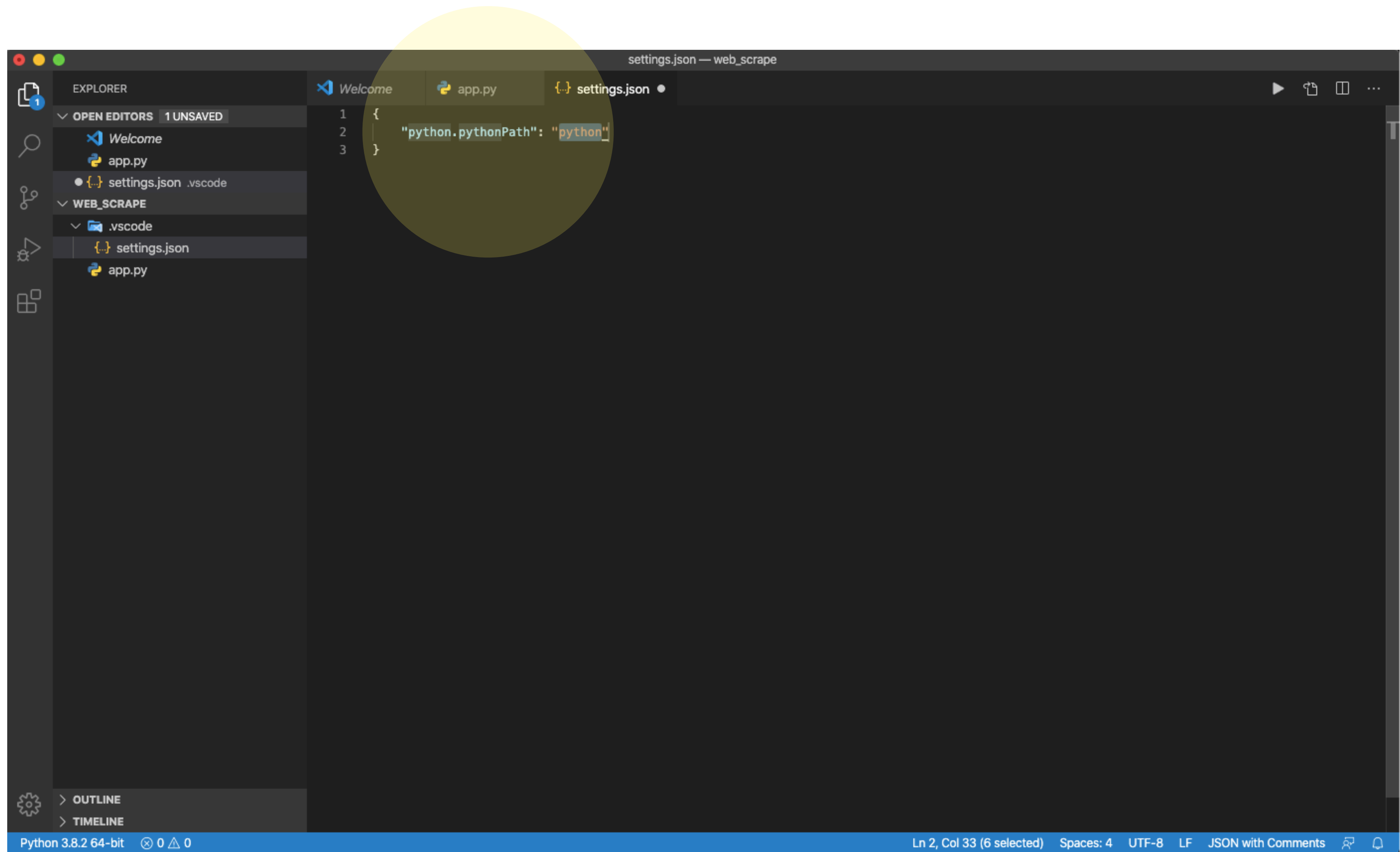
► open (`⌘O`) the folder project 'web_scrape'



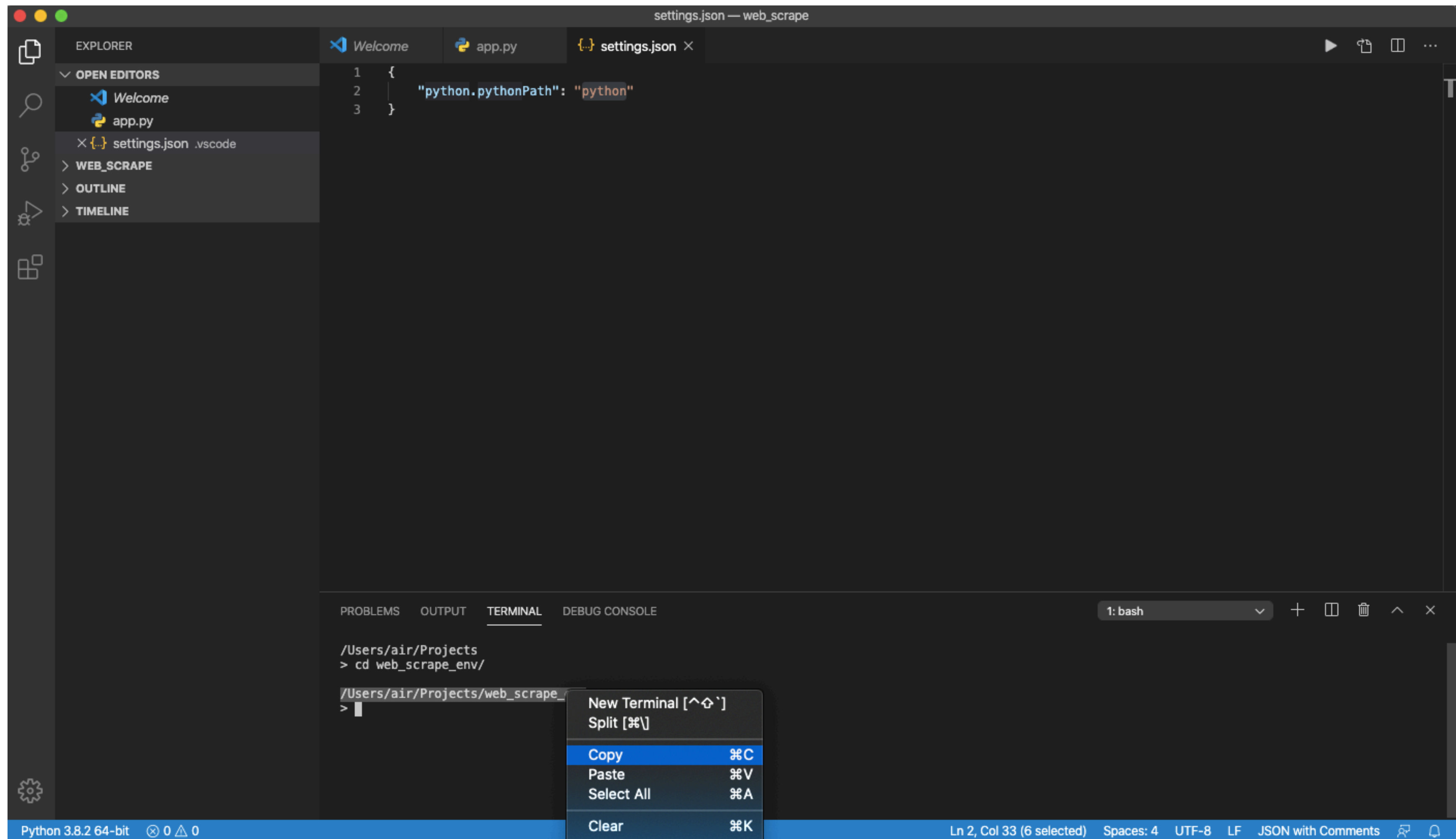
► create (⌘N) a new file 'app.py'



► open 'Workspace Settings (JSON)' on Command Palette ( ⌘P)

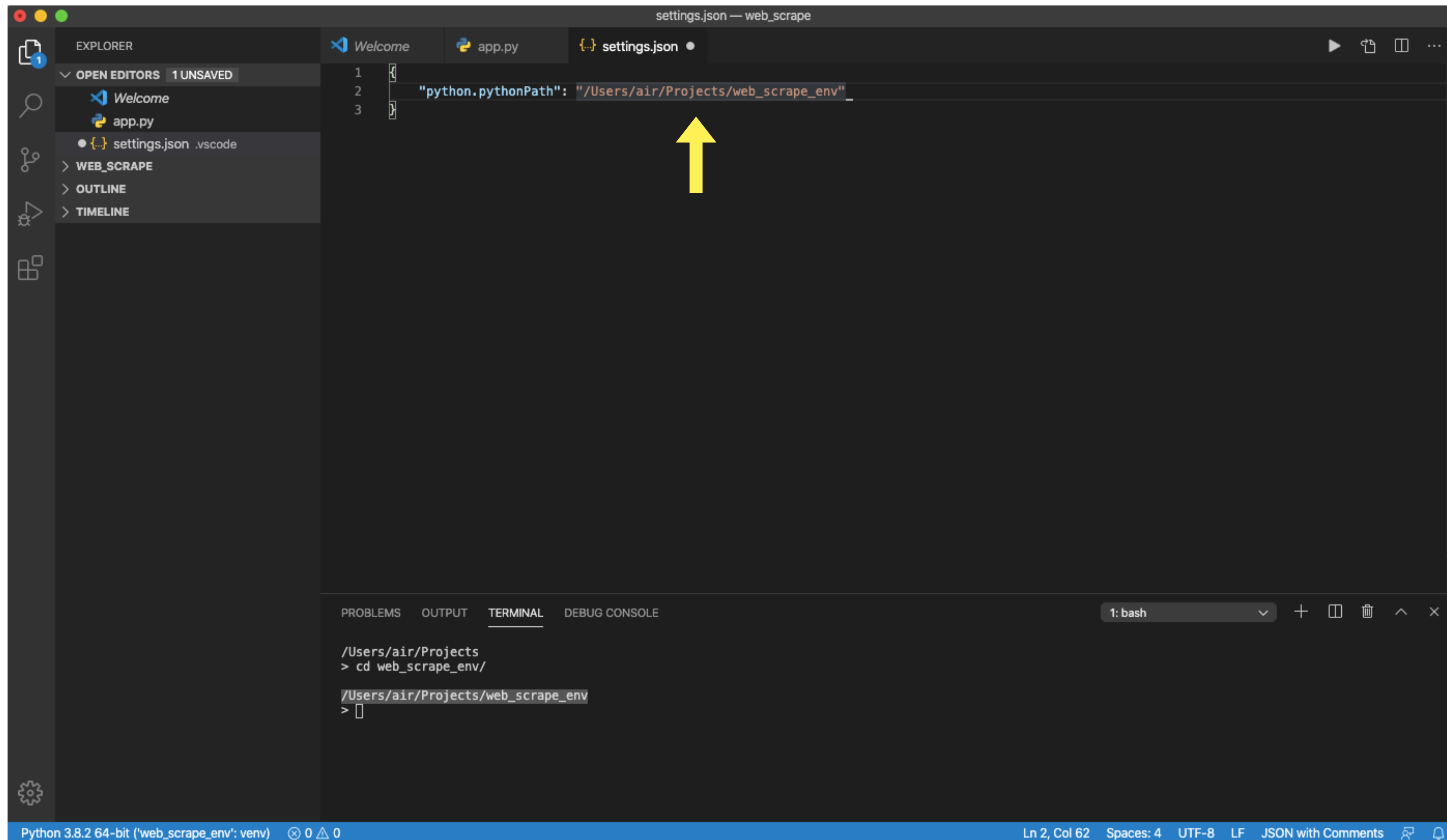


► type between curly brackets “python-pythonPath”: “python”

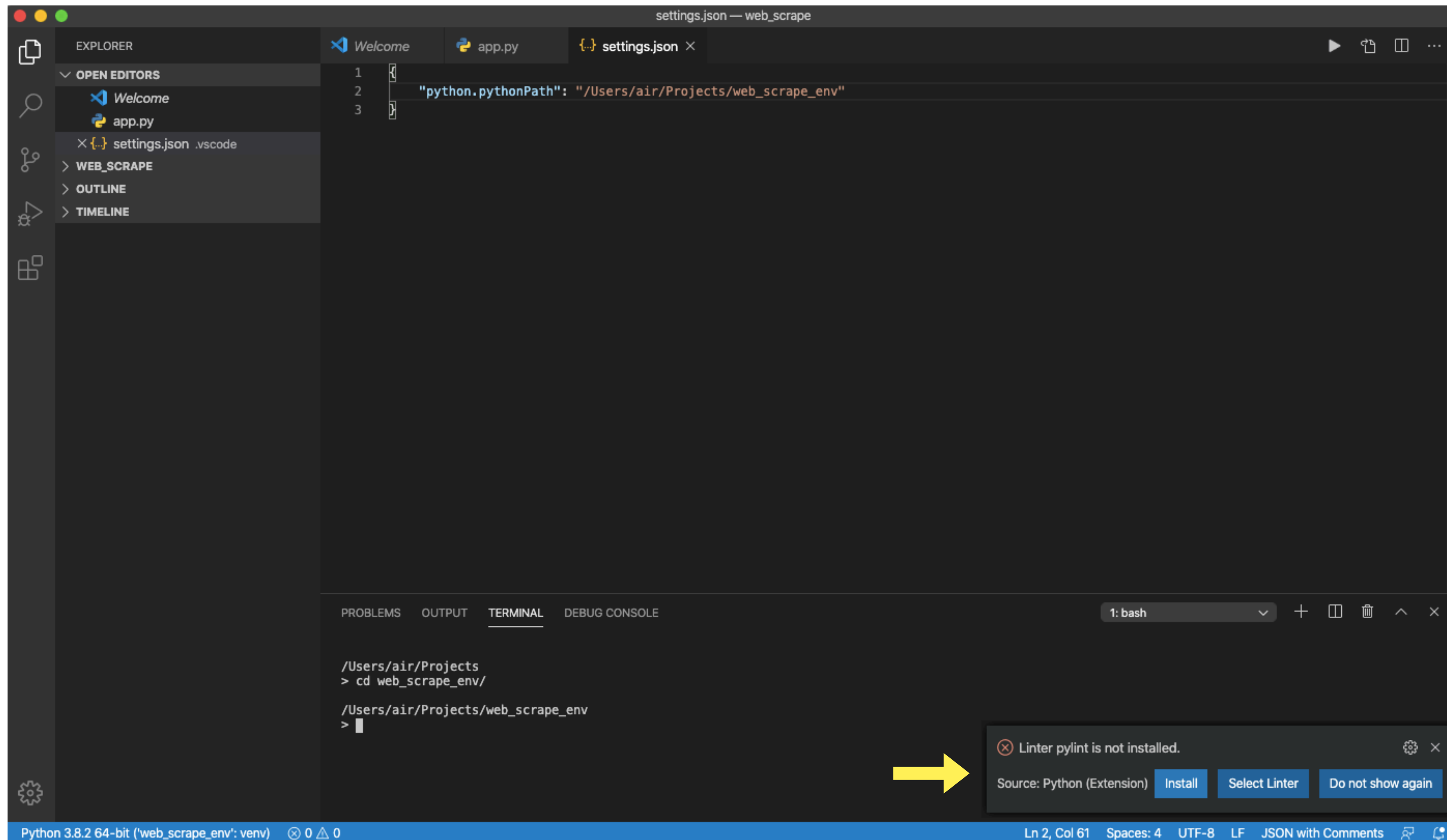


► open the terminal (`^``), change to the environment folder then copy its path

Giovanni Dalla Vecchia - 20/06/2020
VSCoDe_Python_Venv - v1



► replace 'python' to the path of the virtual environment 'web_scrape_env'



► if you get a message to install 'Linter' just confirm. Linter is a tool to analyse the source code

Giovanni Dalla Vecchia - 20/06/2020
VSCode_Python_Venv - v1

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the project structure with 'WEB_SCRAPE' selected.
- OPEN EDITORS:** Lists 'Welcome', 'app.py', and 'settings.json'.
- settings.json:** Contains the configuration `"python.pythonPath": "/Users/air/Projects/web_scrape_env"`.
- TERMINAL:** Displays the output of a pip install command, including package collection and installation details. A yellow arrow points to the terminal output.
- STATUS BAR:** Shows 'Python 3.8.2 64-bit ('web_scrape_env': venv)' and other details.

```
Using cached https://files.pythonhosted.org/packages/23/84/323c2415280bc4fc880ac5050dddfb3c8062c2552b34c2e512eb4aa68f79/wrapt-1.11.2.tar.gz
Collecting lazy-object-proxy==1.4.* (from astroid<2.4,>=2.3.0->pylint)
Collecting six~=1.12 (from astroid<2.4,>=2.3.0->pylint)
Using cached https://files.pythonhosted.org/packages/65/eb/1f97cb97bfc2390a276969c6fae16075da282f5058082d4cb10c6c5c1dba/six-1.14.0-py2.py3-none-any.whl
Installing collected packages: wrapt, lazy-object-proxy, six, astroid, mccabe, isort, pylint
Running setup.py install for wrapt ... done
Successfully installed astroid-2.3.3 isort-4.3.21 lazy-object-proxy-1.4.3 mccabe-0.6.1 pylint-2.4.4 six-1.14.0 wrapt-1.11.2
WARNING: You are using pip version 19.2.3, however version 20.0.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(web_scrape_env)
/Users/air/Projects/web_scrape
>
```

► now your project is pointing to the virtual environment 'web_scrape_env'

What is that mean 'point to the Virtual Environment web_scrape_env'?

- The project will have access to the packages inside the environment
- There is no need to install packages globally for all your projects
- Imagine if you have an application that uses a package version 1.0.1 with a specific function/method, but the other version let is say 2.3.5 works completely different. You have two options: change all your code or avoid any upgrade to the new version. But if you have a version 1.0.1 in your environment v1.0.1 you do not need to worry, unless you install the new version 2.3.5.
- DO NOT forget to change your virtual environment when create a new application.

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows the project structure with files 'Welcome', 'app.py', and 'settings.json'. The main editor area displays the 'settings.json' file with the following content:

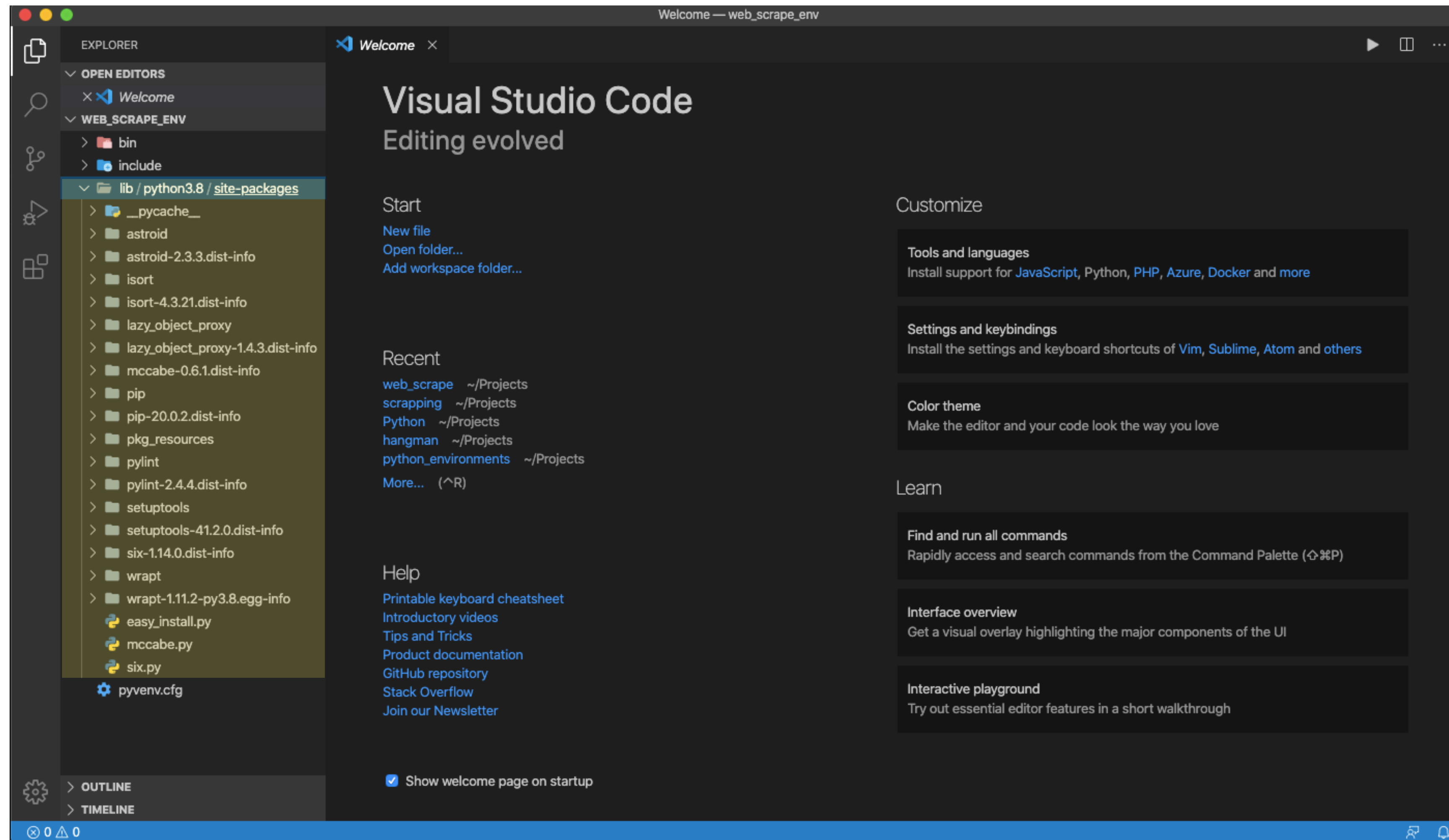
```
1 {  
2   "python.pythonPath": "/Users/air/Projects/web_scrape_env"  
3 }
```

Below the editor, the TERMINAL panel is active, showing the output of the 'pip list' command in the 'web_scrape_env' virtual environment. A green circle highlights the terminal output:

```
(web_scrape_env)  
/Users/air/Projects/web_scrape  
> pip list  
Package            Version  
-----  
astroid             2.3.3  
isort               4.3.21  
lazy-object-proxy   1.4.3  
mccabe              0.6.1  
pip                 20.0.2  
pylint              2.4.4  
setuptools          41.2.0  
six                 1.14.0  
wrapit              1.11.2  
(web_scrape_env)  
/Users/air/Projects/web_scrape  
>
```

The status bar at the bottom indicates the Python 3.8.2 64-bit environment is active, and the current file is 'settings.json' at line 2, column 62.

► enter in the terminal the command 'pip list'. That lists the packages installed in the virtual environment



► location of all the packages available on your environment 'web_scrap_env'

Giovanni Dalla Vecchia - 20/06/2020
VSCoDe_Python_Venv - v1

The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer panel on the left shows a project named 'web_scrape' with a file 'app.py' open in the editor. The terminal at the bottom is active, showing the following commands and output:

```
/Users/air/Projects
> source /Users/air/Projects/web_scrape_env/bin/activate 1
(web_scrape_env)
/Users/air/Projects
> pip install bs4 requests 2
```

The status bar at the bottom indicates the Python 3.8.2 64-bit interpreter is selected, and the file is at Line 1, Column 1.

► on your project (1) activate virtual environment and (2) install the packages ‘beautiful soup (bs4)’ and ‘requests’

The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer sidebar on the left shows the project structure with 'app.py' open in the editor. The terminal window at the bottom displays the following commands and output:

```
/Users/air/Projects
> source /Users/air/Projects/web_scrape_env/bin/activate
(web_scrape_env)
/Users/air/Projects
> pip install bs4 requests
Collecting bs4
  Using cached bs4-0.0.1.tar.gz (1.1 kB)
Collecting requests
  Using cached requests-2.23.0-py2.py3-none-any.whl (58 kB)
Collecting beautifulsoup4
  Using cached beautifulsoup4-4.9.0-py3-none-any.whl (109 kB)
Requirement already satisfied: certifi>=2017.4.17 in ./web_scrape_env/lib/python3.8/site-packages (from requests) (2020.4.5.1)
Requirement already satisfied: chardet<4,>=3.0.2 in ./web_scrape_env/lib/python3.8/site-packages (from requests) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in ./web_scrape_env/lib/python3.8/site-packages (from requests) (1.25.9)
Requirement already satisfied: idna<3,>=2.5 in ./web_scrape_env/lib/python3.8/site-packages (from requests) (2.9)
Requirement already satisfied: soupsieve>1.2 in ./web_scrape_env/lib/python3.8/site-packages (from beautifulsoup4->bs4) (2.0)
Installing collected packages: beautifulsoup4, bs4, requests
  Running setup.py install for bs4 ... done
Successfully installed beautifulsoup4-4.9.0 bs4-0.0.1 requests-2.23.0
(web_scrape_env)
/Users/air/Projects
>
```

A yellow arrow points from the 'OUTLINE' tab in the sidebar to the terminal window.

► now both packages are available to be used on your project

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left shows a project named 'web_scrape' containing a file 'app.py'. The main editor window displays the contents of 'app.py', which is a Python script. The script has five lines of code: two import statements and two print statements. A yellow number '1' is placed to the right of the first print statement. Below the editor, the TERMINAL panel is active, showing the command prompt for a virtual environment named 'web_scrape_env'. The command 'python app.py' has been entered, and a yellow number '2' is placed to the right of the command. The status bar at the bottom indicates the Python version is 3.8.2 64-bit, and the current file is 'app.py' at line 5, column 28.

```
app.py — web_scrape
```

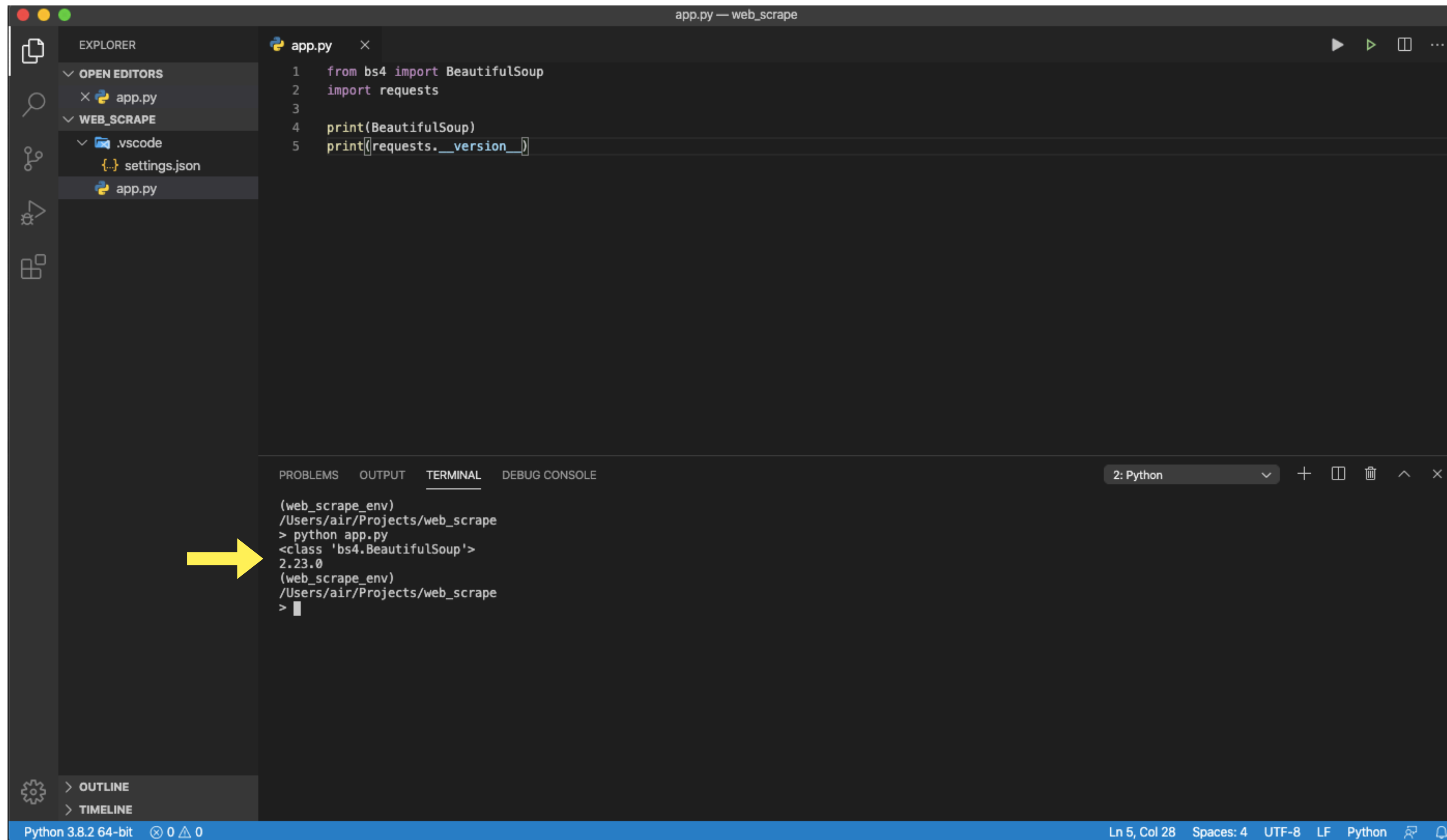
```
1 from bs4 import BeautifulSoup
2 import requests
3
4 print(BeautifulSoup)
5 print(requests.__version__)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 2: Python

```
(web_scrape_env)
/Users/air/Projects/web_scrape
> python app.py
```

Python 3.8.2 64-bit 0 0 Ln 5, Col 28 Spaces: 4 UTF-8 LF Python

► type the code (1) above inside the 'app.py' then (2) run 'python app.py'



The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named 'web_scrape' containing a file 'app.py'. The main editor area displays the contents of 'app.py', which is a Python script with the following code:

```
1 from bs4 import BeautifulSoup
2 import requests
3
4 print(BeautifulSoup)
5 print(requests.__version__)
```

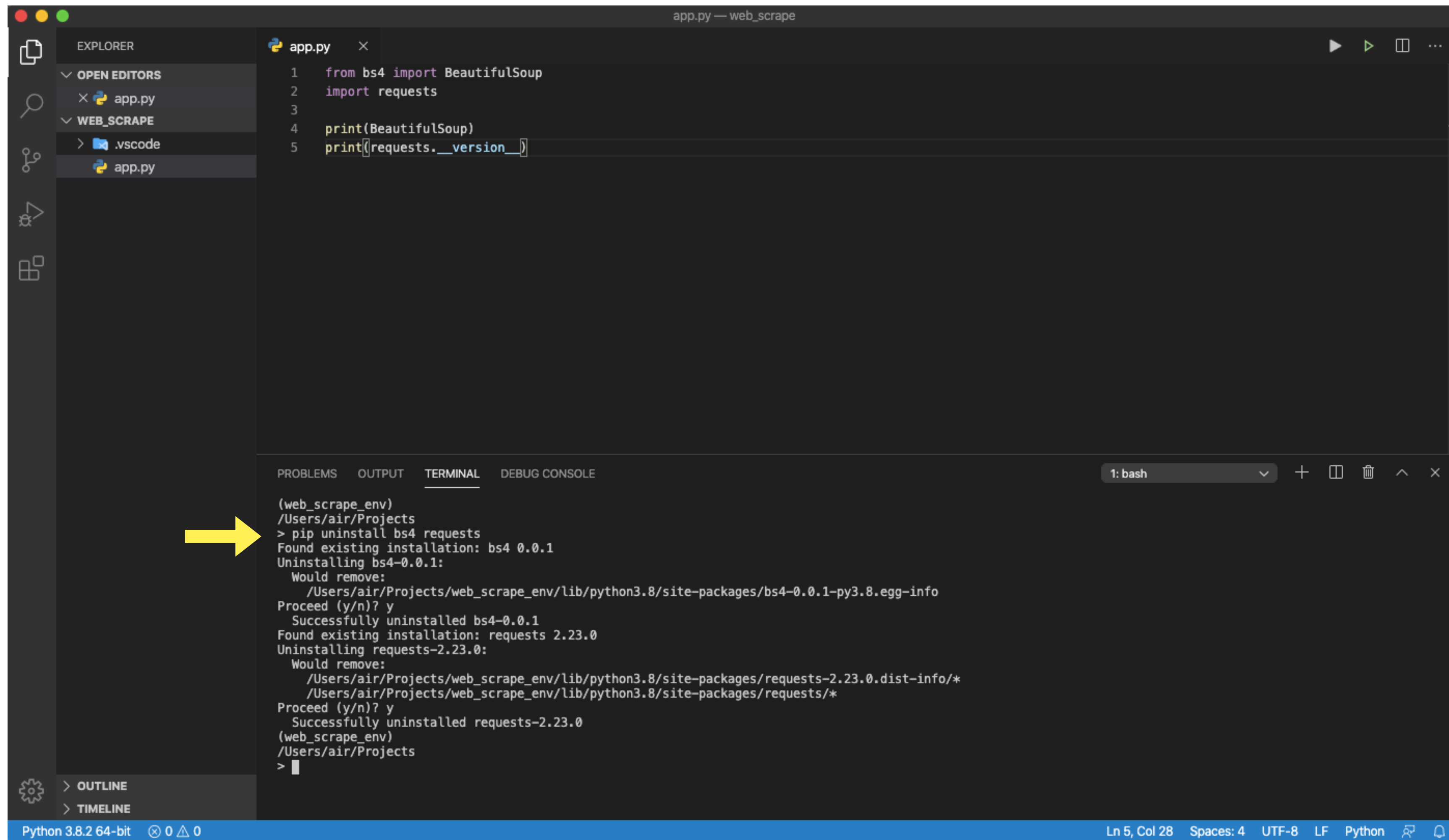
Below the editor, the TERMINAL panel is active, showing the output of running the script. A yellow arrow points to the output. The terminal text is as follows:

```
(web_scrape_env)
/Users/air/Projects/web_scrape
> python app.py
<class 'bs4.BeautifulSoup'>
2.23.0
(web_scrape_env)
/Users/air/Projects/web_scrape
> 
```

The status bar at the bottom indicates the Python 3.8.2 64-bit environment is active, and the current cursor position is at Line 5, Column 28.

► if all goes well you should see the following return on your prompt

Giovanni Dalla Vecchia - 20/06/2020
VSCoDe_Python_Venv - v1



The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left shows the project structure with 'app.py' open. The main editor area displays the content of 'app.py', which imports BeautifulSoup and requests, and prints their versions. The Terminal panel at the bottom shows the execution of 'pip uninstall bs4 requests' within a virtual environment, with output confirming the successful uninstallation of both packages. A yellow arrow points to the terminal output.

```
app.py — web_scrape
```

```
1 from bs4 import BeautifulSoup
2 import requests
3
4 print(BeautifulSoup)
5 print(requests.__version__)
```

```
(web_scrape_env)
/Users/air/Projects
> pip uninstall bs4 requests
Found existing installation: bs4 0.0.1
Uninstalling bs4-0.0.1:
  Would remove:
    /Users/air/Projects/web_scrape_env/lib/python3.8/site-packages/bs4-0.0.1-py3.8.egg-info
Proceed (y/n)? y
  Successfully uninstalled bs4-0.0.1
Found existing installation: requests 2.23.0
Uninstalling requests-2.23.0:
  Would remove:
    /Users/air/Projects/web_scrape_env/lib/python3.8/site-packages/requests-2.23.0.dist-info/*
    /Users/air/Projects/web_scrape_env/lib/python3.8/site-packages/requests/*
Proceed (y/n)? y
  Successfully uninstalled requests-2.23.0
(web_scrape_env)
/Users/air/Projects
> 
```

Python 3.8.2 64-bit 0 0 Ln 5, Col 28 Spaces: 4 UTF-8 LF Python

► if you need to uninstall the packages run the command 'pip uninstall bs4 requests'

Thanks very much