

# pip & Package Initialization

```
├── __init__.py
├── my_package1
│   ├── __init__.py
│   ├── my_module_1.py
│   └── my_module_2.py
└── my_package2
    ├── __init__.py
    ├── my_module_3.py
    └── my_module_4.py

2 directories, 7 files
→ upper_package git:(main) x
```

- First find out where your python installed
- Create your packages inside your python directory
- $\Rightarrow$  /opt/homebrew/lib/python3.9/site-packages
- keep in mind that you have to put a file named **init.py** in the folders you will create.

## Importing \* From a Package

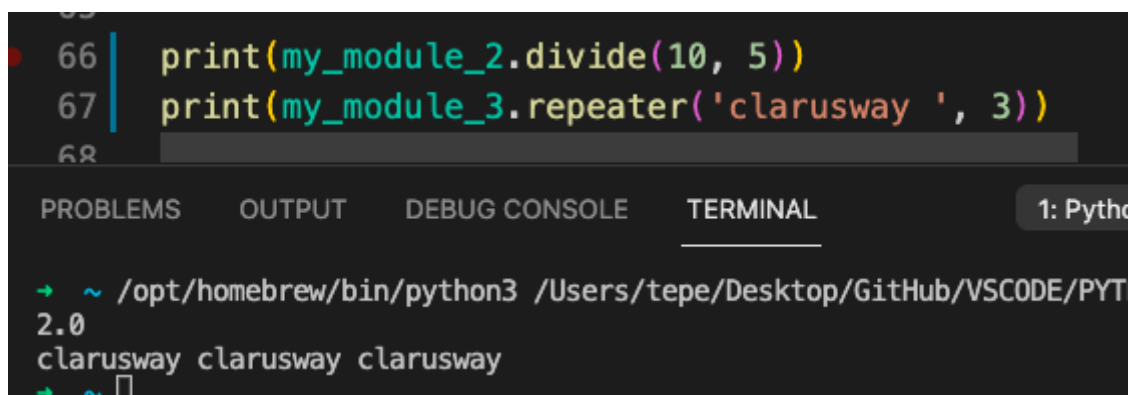
```

from upper_package import my_package1, my_package2
from upper_package.my_package1 import my_module_1, my_module_2
print(dir(my_package1))

from upper_package.my_package2 import my_module_3, my_module_4
print(dir(my_package2))

```

Now you can use module within packages you created.



The screenshot shows a code editor with two lines of Python code being executed. The first line calls `my_module_2.divide(10, 5)` and the second line calls `my_module_3.repeater('clarusway ', 3)`. Below the code, the terminal output shows the results: `2.0` and `clarusway clarusway clarusway`.

```

66 | print(my_module_2.divide(10, 5))
67 | print(my_module_3.repeater('clarusway ', 3))
68 |

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Python

```

→ ~ /opt/homebrew/bin/python3 /Users/tepe/Desktop/GitHub/VSCODE/PYT
2.0
clarusway clarusway clarusway
→ ~ 

```

## PIP commands

- `pip --version`
- `pip install my_package`
- `pip list`
- `pip show my_package`
- `pip uninstall my_package`

\*\* for macOS use prefix  $\Rightarrow$  `python3 -m`

for example :

- `python3 -m pip --version`
- `python3 -m pip list`
- `python3 -m pip install my_package`