

# Python using wheel

21 February 2025 09:35

## Understanding setuptools and wheel

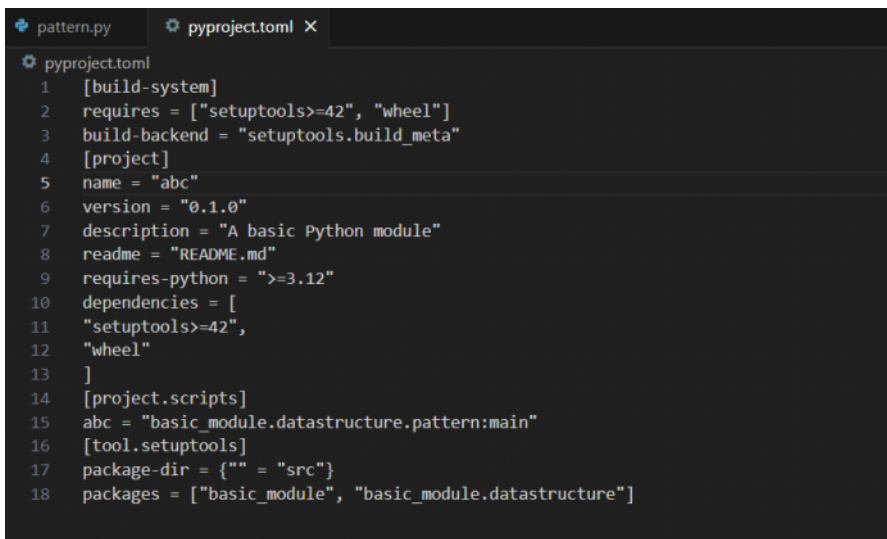
- **setuptools**: A package development and distribution tool used to configure and build Python projects.
- **wheel**: A binary packaging format that speeds up installations by avoiding recompilation.

To setup:

- 1) Create a directory where there won't be permission issues, avoid root directories
- 2) The structure of your directory is to be as follows

```
new1/                                # Project Root
|-- build/                            # (Generated after build)
|-- dist/                            # (Generated after build)
|-- src/                             # Source directory
|   |-- basic_module/               # Main module
|   |   |-- datastructure/          # Sub-module
|   |   |   |-- __init__.py
|   |   |   |-- pattern.py
|   |   |-- __init__.py
|-- pyproject.toml                  # Build system configuration
|-- README.md                      # Project documentation
```

- 3) And everything is to be done via ubuntu terminal and access vscode where you can use vscode for ubuntu instead
- 4) After created init.py files add pattern.py and pyproject.toml files
- 5) A .toml file (Tom's Obvious, Minimal Language) is a configuration file format used for defining settings in a structured and human-readable way. In Python packaging, pyproject.toml is used to specify project metadata and dependencies.



```
pyproject.toml
1 [build-system]
2 requires = ["setuptools>=42", "wheel"]
3 build-backend = "setuptools.build_meta"
4 [project]
5 name = "abc"
6 version = "0.1.0"
7 description = "A basic Python module"
8 readme = "README.md"
9 requires-python = ">=3.12"
10 dependencies = [
11     "setuptools>=42",
12     "wheel"
13 ]
14 [project.scripts]
15 abc = "basic_module.datastructure.pattern:main"
16 [tool.setuptools]
17 package-dir = {"" = "src"}
18 packages = ["basic_module", "basic_module.datastructure"]

pattern.py
```

And description readme file etc are editable and also name="abc" should be whatever the name you want to reference your .py code to be run by

- 6) Pattern.py is the code you want to test, sample one below

```
pattern.py x pyproject.toml
src > basic_module > datastructure > pattern.py
1 def main():
2     N=9
3     j=0
4     a=""
5     for i in range(1,N):
6         for j in range(1,i):
7             print(j,end=" ")
8             print("\n")
9 if __name__ == "__main__":
10     main()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - new1 + - ... ^

sandy@sandy:~/new1$ python3 src/basic_module/datastructure/pattern.py

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
```

Code should be running right to avoid complications

- 7) Then follow these 3 commands to finish the setup
- pip install --upgrade pip setuptools wheel build
  - pip install --upgrade pip setuptools wheel build --break-system-packages (incase above doesn't work)
  - python3 -m build --wheel

- 8) So now all u have to do is type the reference file name u gave aka "abc" here

```
sandy@sandy:~/new1$ abc

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
```

```
sandy@sandy:~$ cd new1
sandy@sandy:~/new1$ code .
sandy@sandy:~/new1$ abc

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

```
sandy@sandy:~$ cd new1  
sandy@sandy:~/new1$ code .  
sandy@sandy:~/new1$ abc
```

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
1 2 3 4 5 6  
1 2 3 4 5 6 7
```

U can see the changes reflected in both ubuntu terminal and vscode.