



Packages in Python

Stat 159/259
October 26th, 2026

Where do we interact with packages?

Python Package Index (pypi)

Find, install and
with the

Type '/' to search proj

Publishing a

Conda

ANACONDA.ORG

Search for environments

CMD + K

Sign In

Sign Up

The screenshot displays the GitHub interface for the repository 'sequoiarose/salti'. The top navigation bar includes links for Code, Issues, Pull requests, Agents, Actions, Projects, Wiki, Security and quality, Insights, and Settings. A search bar is present on the right. The repository name 'salti' is shown with a 'Private' label. A notification banner states: 'Your main branch isn't protected. Protect this branch from force pushing or deletion, or require status checks before merging. View documentation.' Below this, the file explorer shows a list of files: coZinv, config, matrix_experiments, notebooks, notes, output, and secrets. The commit history table shows recent commits by 'sequoiarose' with descriptions like 'Merge branch 'main' of github.com:sequoiarose/salti', 'added bkg limits op', 'updated matrix exp', 'updated model test', 'notes', 'removed old output', and 'added new environment variables'. A 'Clone' dropdown menu is open, showing options for Local, Codespaces, Clone, HTTPS, SSH, GitHub CLI, and a web URL. The right sidebar contains 'Usage Instructions' for installing packages from the channel, including a 'Temporary' section with the command: `conda install --channel anaconda <PACKAGE>`. A line graph shows repository activity from September 2026 to October 2026.

Github

Using packages in python

1. Install the package:

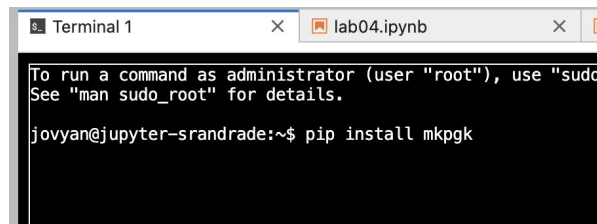
'pip install mypkg'

'conda install mypkg'

This can be skipped if you are using the package locally

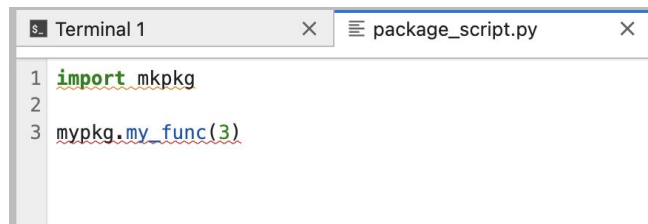
2. Load the package:

'import mkpg'



A terminal window titled 'Terminal 1' with a tab 'lab04.ipynb'. The terminal output shows a message: 'To run a command as administrator (user "root"), use "sudo". See "man sudo_root" for details.' followed by the command 'jovyan@jupyter-srandrade:~\$ pip install mkpgk'.

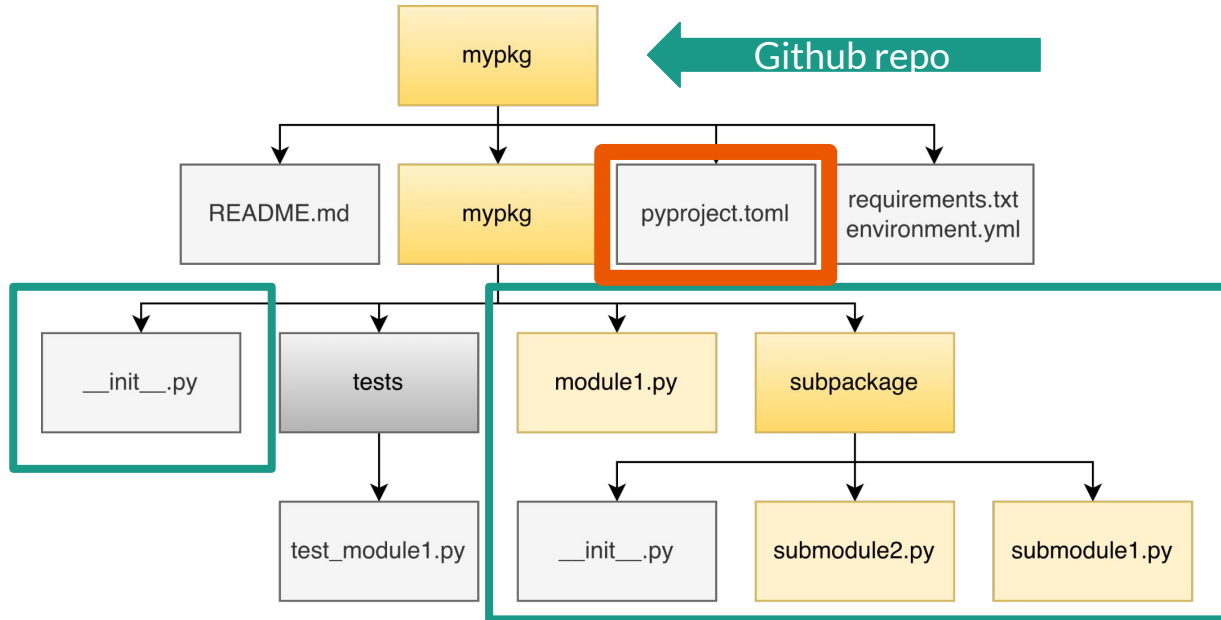
```
Terminal 1 lab04.ipynb
To run a command as administrator (user "root"), use "sudo".
See "man sudo_root" for details.
jovyan@jupyter-srandrade:~$ pip install mkpgk
```



A terminal window titled 'Terminal 1' with a tab 'package_script.py'. The terminal shows three lines of Python code: '1 import mkpkg', '2', and '3 mypkg.my_func(3)'.

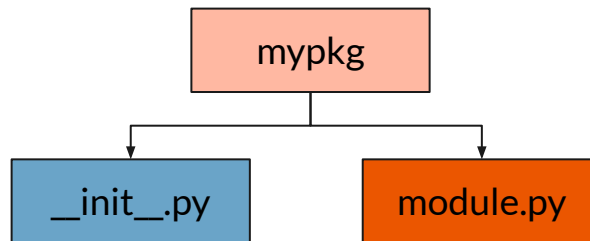
```
Terminal 1 package_script.py
1 import mkpkg
2
3 mypkg.my_func(3)
```

Python Package Structure



Minimal Components of a Local Python Package

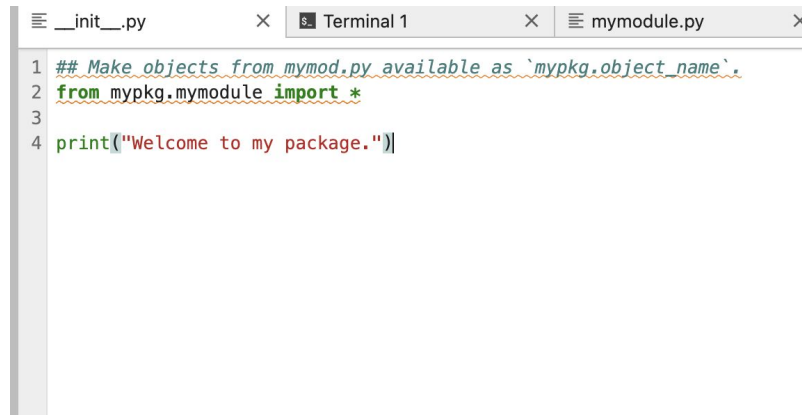
- Directory named after the package
- `__init__.py` within package directory
- Package code in a module





The `__init__.py` file

- Imports the code from the module to make it available
- Was strictly required prior to python 3.3 to import from a folder -> can be an empty file!
- Still recommended to always use and required for submodules and special package structure



```
__init__.py  Terminal 1  mymodule.py
1  ## Make objects from mymod.py available as `mypkg.object_name`.
2  from mypkg.mymodule import *
3
4  print("Welcome to my package.")
```



Additional Components: pyproject.toml

- To install a package using a package manager, you will need to add a `pyproject.toml`
- It stores metadata necessary for installation, including:
 - Requirements
 - Python version
 - Package version
 - Authors

```
1  [build-system]
2  requires = ["setuptools >= 70"]
3  build-backend = "setuptools.build_meta"
4
5  [project]
6  name = "mytoy"
7  authors = [
8      { name="Example Author", email="author@example.com" },
9  ]
10 description = "A Python library to make toys"
11 readme = "README.md"
12
13 requires-python = ">=3.10"
14 dependencies = [
15     "tqdm",
16 ]
17
18 license = "BSD-3-Clause"
19 license-files = ["LICEN[CS]E*"]
20 classifiers = [
21     "Programming Language :: Python :: 3",
22     "Operating System :: OS Independent"
23 ]
24
25 keywords = ["tools", "toys"]
26 dynamic = ["version"]
27
28 [project.urls]
29 Homepage = "https://github.com/fperez/mytoy"
30 Issues = "https://github.com/fperez/mytoy/issues"
31
32 [tool.setuptools]
33 packages = ["mytoy"]
34
35 [tool.setuptools.dynamic]
36 version = {attr = "mytoy.__version__"}
```



Additional Components: tests

- Tests are typically stored in the package directory under a tests directory
- Can use pytest or unittest or another testing framework
- Tests can be run automatically with github actions to generate badges

```
iovvana@iunvter-srandrade:~/stat159/class_demo$ cd mvtnv
```

pandas: A Powerful Python Data Analysis Toolkit

Testing	 
---------	---



Demo: Creating a Minimal Package from Scratch Locally

```
mkdir mypkg
```

```
cat << EOF > mypkg/__init__.py
```

```
## Make objects from mymod.py available as  
`mypkg.object_name`.
```

```
from mypkg.mymod import *
```

```
print("Welcome to my package.")
```

```
EOF
```



Demo: Creating a Minimal Package from Scratch Locally

```
cat << EOF > mypkg/mymod.py
```

```
x = 7
```

```
def myfun(val):
```

```
    print("The arg is: ", str(val), ".", sep = '')
```

```
EOF
```



Demo: Creating a Minimal Package from Scratch Locally

- Start python
- Run:

```
import mypkg
```

```
mypkg.x
```

```
mypkg.myfun(7)
```



Demo: Working with a Complete Package

See: <https://github.com/fperez/mytoy>

- `git clone https://github.com/fperez/mytoy`
- `cd mytoy`
- `pip install --user .`

Sharing packages

Python Package Index (pypi)

Find, install and
with the

Type '/' to search proj

Publishing a

Conda
ANACONDA.ORG

Search for environments

CMD + K

Sign In

Sign Up

The screenshot shows a GitHub repository page for 'sequoiarose / salti'. The repository is private and has 3 branches and 0 tags. A blue banner at the top states 'Your main branch isn't protected' and provides instructions on how to protect the branch. Below this, a file explorer shows the repository structure with folders like 'coZinv', 'config', 'matrix_experiments', 'notebooks', 'notes', 'output', and 'scripts'. A commit history table is visible, showing recent commits by 'sequoiarose' with messages like 'Merge branch 'main'' and 'added bkg limits op'. A dropdown menu is open over the 'Clone' button, showing options for cloning via HTTPS, SSH, or GitHub CLI, and also offering to open the repository in GitHub Copilot, GitHub Desktop, or download a ZIP file. On the right side of the repository page, there is a 'Usage Instructions' section for Conda, which includes a 'Temporary' tab and a 'Configured' tab. The 'Temporary' tab shows the command 'conda install --channel anaconda <PACKAGE>'. The 'Configured' tab shows a graph of package usage over time, with a peak in September 2020.

Github