Python Package in PyPi

ADVANCED PYTHON

Group A
KHALID ALNASSER
ALI ALQAWAEEN
MOHAMMED ALJAMED
BASIL ALFAKHER
HADI ALSINAN
DURRAH ALZAMIL
HUSSAIN ALHAJJAJ

Creating and publishing a python package (Windows)

Tools required:

- GIT – Version Management System

Git is a version management system, allowing the team to keep track of code changes. The team uses Github to host the package repository.

Repo: https://github.com/khalidnass/medium-first-package

To start, you will need to create an account that will then be granted access to update and modify the repo contents

To install Github, follow one of the following approaches

- Github desktop application - https://desktop.github.com/
- GIT tools from https://git-scm.com/download/win

You can test the installation with the following commands:

"git --version"

```
C:\Users\HussainH>git --version
git version 2.33.0.windows.2
```

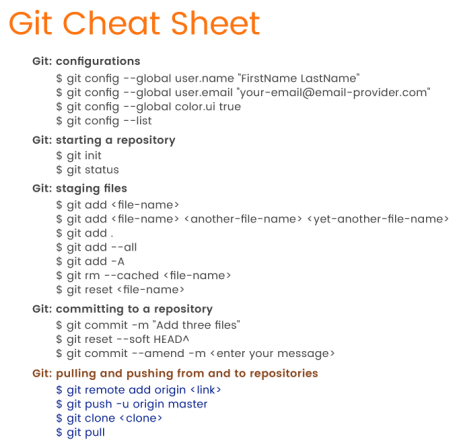
Here is the command to download the package from the github repository to your local machine:

"git clone <repo address>"

```
C:\Users\HussainH>git clone https://github.com/khalidnass/medium-first-package
Cloning into 'medium-first-package'...
remote: Enumerating objects: 72, done.
remote: Counting objects: 100% (72/72), done.
remote: Compressing objects: 100% (50/50), done.
remote: Total 72 (delta 27), reused 50 (delta 14), pack-reused 0
Receiving objects: 100% (72/72), 10.04 KiB | 934.00 KiB/s, done.
Resolving deltas: 100% (27/27), done.
```

Once you complete your changes, you can use the following commands to push the change to the git repo (git add -> git commit -m <describe the change> -> git push)

```
C:\Users\HussainH\medium-first-package>git add .

C:\Users\HussainH\medium-first-package>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   primelib/setup.py


C:\Users\HussainH\medium-first-package>git commit -m "update metadata"
[master 07d2fae] update metadata
 1 file changed, 4 insertions(+), 4 deletions(-)

C:\Users\HussainH\medium-first-package>git push
```



We recommend that you familiarize yourself with other git commands as well:

- https://www.youtube.com/watch?v=USjZcfj8yxE
- https://education.github.com/git-cheat-sheet-education.pdf

To understand git in-depth, here is a lecture from MIT that covers git and the concepts of version management in depth:

- https://missing.csail.mit.edu/2020/version-control/

- ## Python environment

We recommend that the team all use a similar python environment to ensure that no compatibility or library version issues.

Download Anaconda from the https://www.anaconda.com/products/individual

Anaconda comes with a base environment, an environment file with specific library version will be shared with the team separately.

- ## Twine and Wheel libraries

These are required package and upload to pypi

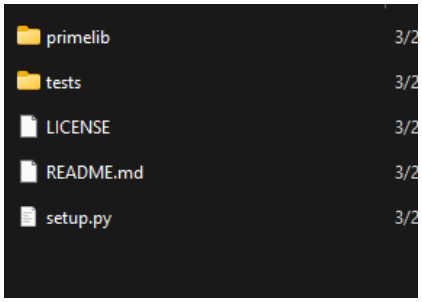
- **install wheel using "pip install wheel"**

```
Collecting wheel
  Downloading wheel-0.37.1-py2.py3-none-any.whl (35 kB)
Installing collected packages: wheel
Successfully installed wheel-0.37.1
```

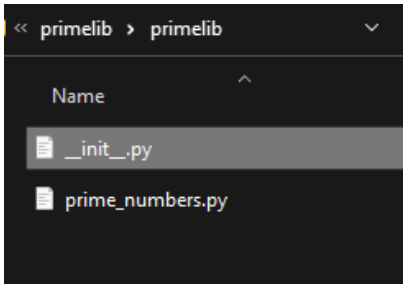
- **Install twine using "pip install twine"**

```
C:\Users\HussainH\medium-first-package\primelib>pip install twine
Collecting twine
  Downloading twine-3.8.0-py3-none-any.whl (36 kB)
Collecting rfc3986>=1.4.0
  Downloading rfc3986-2.0.0-py2.py3-none-any.whl (31 kB)
Collecting pkginfo>=1.8.1
  Downloading pkginfo-1.8.2-py2.py3-none-any.whl (26 kB)
```

The anatomy of a python package:



- primelib: main folder for the package
- Tests: contains python unit tests
- LICENSE: contains the license under which the package is distributed
- README.md: this file shows up in the repo page of github and should have a description of the package an how to use it
- setup.py: has metadata/configuration about the package and its requirement

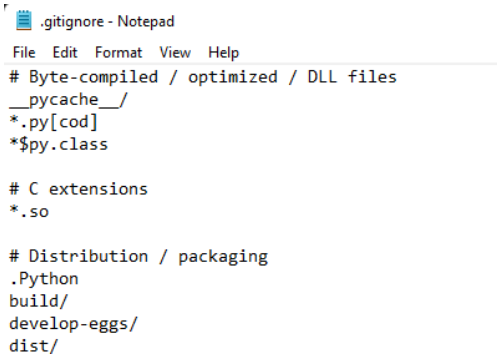


- The __init__.py is a special file that indicates to python that this folder is a package

setup.py file contains metadata, including the description, author, and classifiers (classifiers are meant to help users of pypi repositories find the package).

```python
from setuptools import setup, find_packages

with open("README.md", "r") as f:
    long_description = f.read()

setup(
    name="primelib-ie",
    version="0.0.1",
    author="group A",
    author_email="groupA@gmail.com",
    description="A small package to work with prime numbers",
    long_description=long_description,
    long_description_content_type="text/markdown",
    url="https://github.com/khalidnass/medium-first-package",
    packages=find_packages(),
    classifiers=[
        "Programming Language :: Python :: 3",
        "License :: OSI Approved :: MIT License",
        "Operating System :: OS Independent",
    ]
)
```

.gitignore This file contains a list of files that should **NOT** be uploaded to github, this includes temp files (e.g. generated by vscode), make **sure that all files that contain sensitive information for example are listed here**, they should not be uploaded to the public repo.

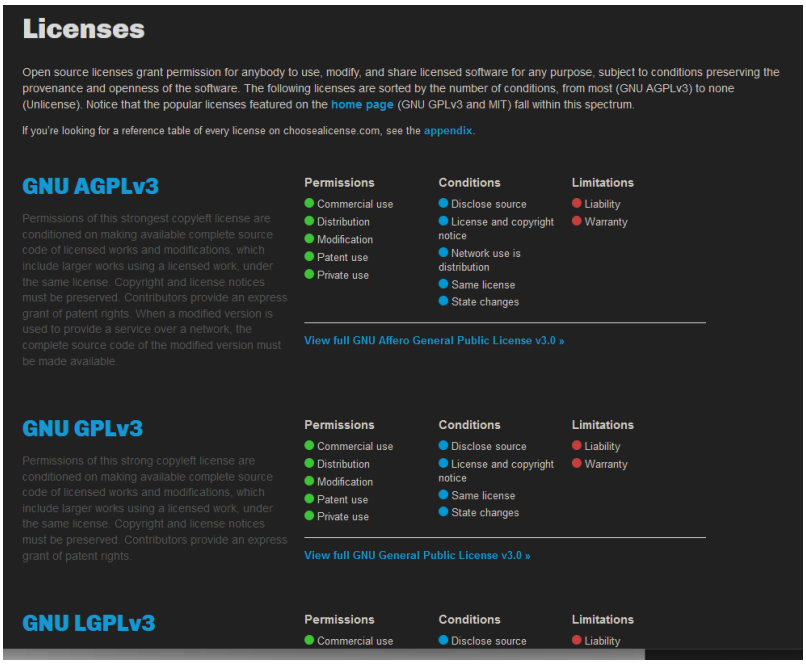



```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
```

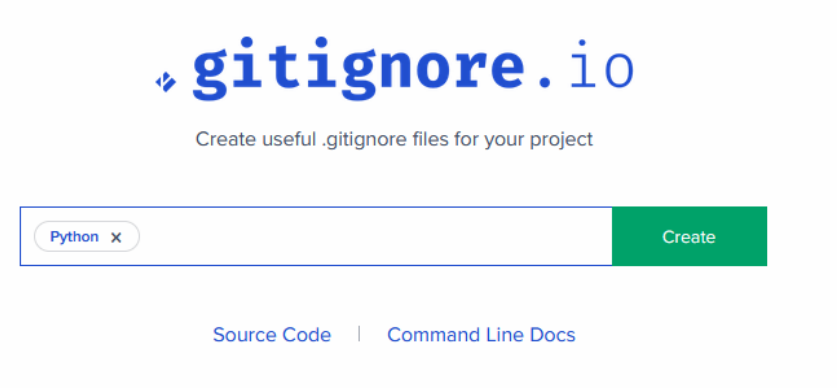
For more resources or information on python packages:

- [comprehensive walkthrough on python packages]

- [Website to help choose a distribution license]



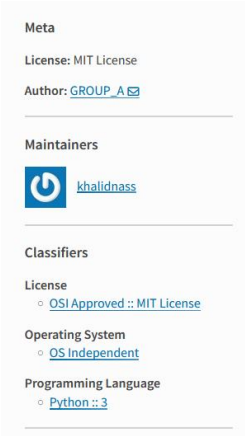
- [Template gitignore file for python projects]



Pypi and testPypi:

Since our package is not ready for production, we only publish to the testPyPi repository (for this you need to setup an account in pypi)

The package is available here at https://test.pypi.org/project/primelib-IE/ and the metadata (from setup.py) can be found on the site too.



Steps followed when publishing:

- **Create the source distribution:**

we use the sdist command as follows:

```
C:\Users\HussainH\medium-first-package\primelib>python setup.py sdist
running sdist
```

Once the command runs, a new folder "dist" is created with a tarball of the source code of the package:

```
medium-first-package\primelib\dist

        .
        ..
 2,337 primelib-ie-0.0.1.tar.gz
```

- **Generate the python wheel file:**

use this command "python setup.py bdist_wheel sdist"

```
C:\Users\HussainH\medium-first-package\primelib>python setup.py bdist_wheel sdist
running bdist_wheel
running build
running build_py
```

A new wheel file will now show up in the dist folder

```
\medium-first-package\primelib\dist

        .
        ..
  2,325 primelib-ie-0.0.1.tar.gz
  2,831 primelib_ie-0.0.1-py3-none-any.whl
```

- **Checking and uploading:**

We check if our package is ready for pypi with twine using "twine check dist/*"

```
C:\Users\HussainH\medium-first-package\primelib>twine check dist/*
Checking dist\primelib_ie-0.0.1-py3-none-any.whl: PASSED
Checking dist\primelib-ie-0.0.1.tar.gz: PASSED
```

To actually upload to the testpypi repository, use the "twine upload -r testpypi dist/*" command and enter your credentials

```
C:\Users\HussainH\medium-first-package\primelib>twine upload -r testpypi dist/*
Uploading distributions to https://test.pypi.org/legacy/
Enter your username:
```

Download and test the package:

Install using “pip install -i https://test.pypi.org/simple/ primelib-IE==0.0.1”

```
>pip install -i https://test.pypi.org/simple/ primelib-IE==0.0.1
istribution -qdm (f:\development\anaconda\envs\aramco_ml2\lib\site-packages)
istribution -qdm (f:\development\anaconda\envs\aramco_ml2\lib\site-packages)
/test.pypi.org/simple/
.1
files.pythonhosted.org/packages/f1/4b/7704210037009e2b5cdb63f3118f0430120525
```

Example usage:

```
>>> from primelib.prime_numbers import is_prime
>>> is_prime(5)
True
>>> is_prime(4)
False
>>>
```

Run the tests if you make any changes using pytest, run the command inside the tests folder

```
(generative_image_torch) D:\IE\Advance Python\IE_GROUP_A_THREAD_9_2\tests>pytest
============================= test session starts ==============================
platform win32 -- Python 3.8.2, pytest-5.4.2, py-1.8.1, pluggy-0.13.1
rootdir: D:\IE\Advance Python\IE_GROUP_A_THREAD_9_2
plugins: hypothesis-5.11.0, arraydiff-0.2, astropy-header-0.1.2, doctestplus-0.5.0, openfiles-0.5.0, remotedata-0.3.2
collected 2 items

test_prime.py ..                                                         [100%]

============================= warnings summary =================================
tests/test_prime.py::test_false
tests/test_prime.py::test_false
tests/test_prime.py::test_true
tests/test_prime.py::test_true
  D:\Users\khali\anaconda3\envs\generative_image_torch\lib\site-packages\pytest_remotedata\plugin.py:65: DeprecationWarning: distutils Version classes are d
eprecated. Use packaging.version instead.
    if LooseVersion(pytest.__version__) < LooseVersion("3.6"):

-- Docs: https://docs.pytest.org/en/latest/warnings.html
======================= 2 passed, 4 warnings in 0.08s =======================
```

Update the tests if necessary by updating the “tests/test_prime.py” file

```
from primelib.prime_numbers import is_prime

def test_false():
    assert is_prime(10) is False

def test_true():
    assert is_prime(11) is True
```

For more robust testing to ensure that the package runs other python environments or versions, we plan to use: tox library

What is tox?

tox is a generic virtualenv management and test command line tool you can use for:

- checking that your package installs correctly with different Python versions and interpreters
- running your tests in each of the environments, configuring your test tool of choice
- acting as a frontend to Continuous Integration servers, greatly reducing boilerplate and merging CI and shell-based testing.

Basic example

First, install `tox` with `pip install tox`. Then put basic information about your project and the test environments you want your project to run in into a `tox.ini` file residing right next to your `setup.py` file:

```
# content of: tox.ini , put in same dir as setup.py
[tox]
envlist = py27,py36

[testenv]
# install pytest in the virtualenv where commands will be executed
deps = pytest
commands =
    # NOTE: you can run any command line tool here - not just tests
    pytest
```

You can also try generating a `tox.ini` file automatically, by running `tox-quickstart` and then answering a few simple questions.

To sdist-package, install and test your project against Python2.7 and Python3.6, just type:

```
tox
```