Ajay Agrawal

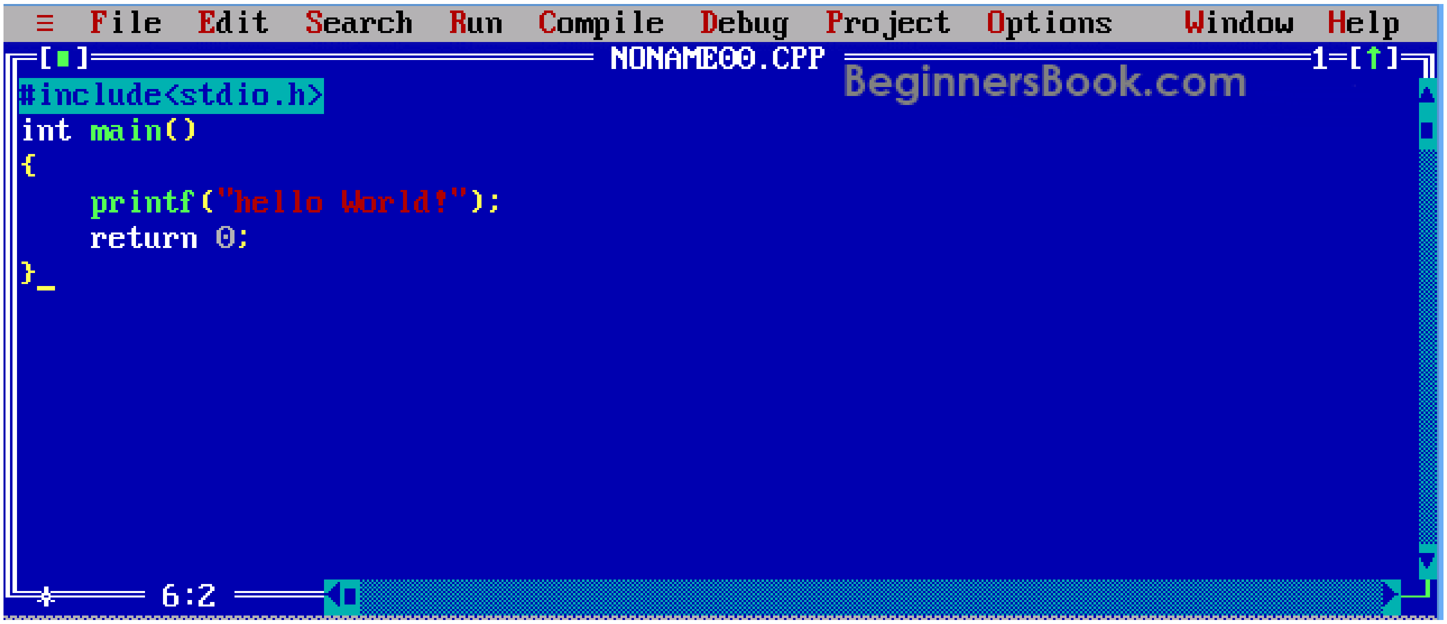
Ajay Agrawal FOLLOW

Proud to become Pythonista

How To Publish Your Own Python Package

Published Jan 20, 2020

Do you remember how exciting it was when created your first “Hello World” program in any language?


≡ File Edit Search Run Compile Debug Project Options Window Help
NONAME00.CPP
1
BeginnersBook.com
6:2

```
#include<stdio.h>
int main()
{
    printf("hello World!");
    return 0;
}_
```

Mine was like this one, I got my first chance to write the Hello World program in C language in my first year of Bachelor degree. but it was a magical feeling because I was able to command the system and able to print anything on the monitor.

Why I mentioned this story before the agenda because today also I felt the same. When I did it successfully. The difference is this time I did not have any mentor to teach me and it took me 8 trials to do it. E

SOME PRE-REQUISITES BEFORE WE START IT...

1. You should have Python3 installed on your machine.
2. Make sure you have the latest version of pip. Be sure by typing it on your terminal or command prompt

```
python -m pip install -upgrade pip
```

3. You should have installed wheel and twine library. If not then they can be easily installed by the following commands,

```
pip install wheel
```

```
pip install twine
```

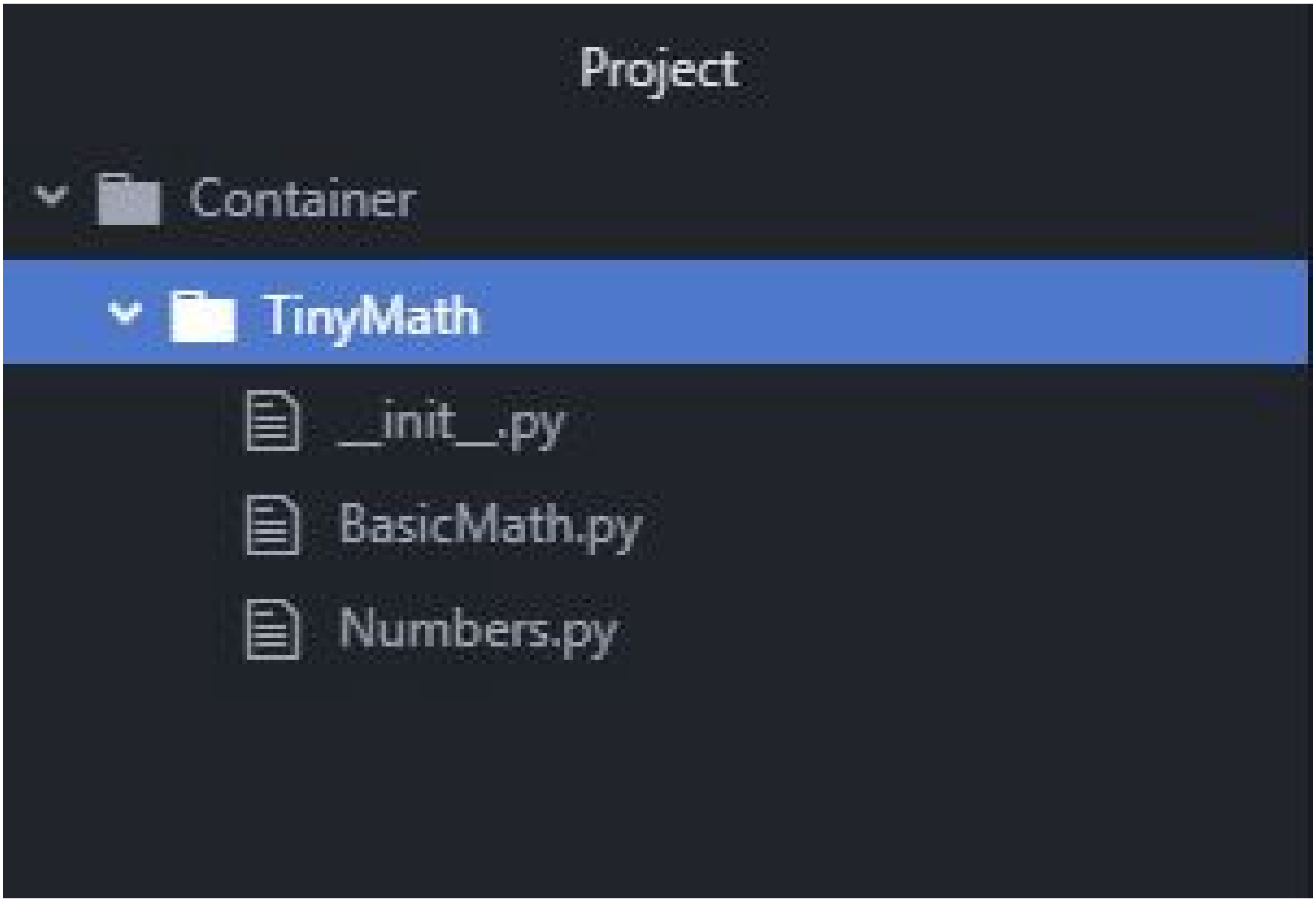
4. You should have a working GitHub account.

STEP 1: CREATE A SEPERATE EMPTY FOLDER TO CONTAIN OUR PACKAGE

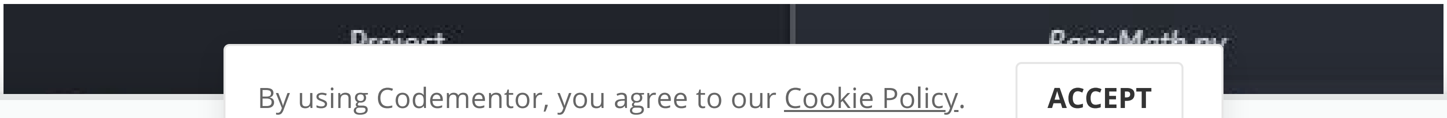
Create an empty folder on any location on your pc, you can use any name because it’s just a container and name of this folder will not be used in the process. Let us make it with the name “Container”.

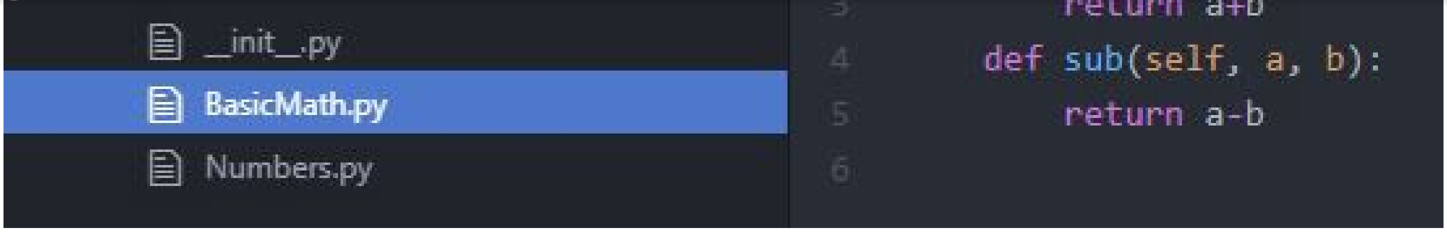
STEP2: JUMP INSIDE YOUR FOLDER AND WRITE YOUR PYTHON FILES

Jump inside the “Container folder” and create another folder, this will be your package name. So choose it wisely as I chose “TinyMath" . Jump inside and make some files which you want to upload as your library. After writing my files my file structure looks like this

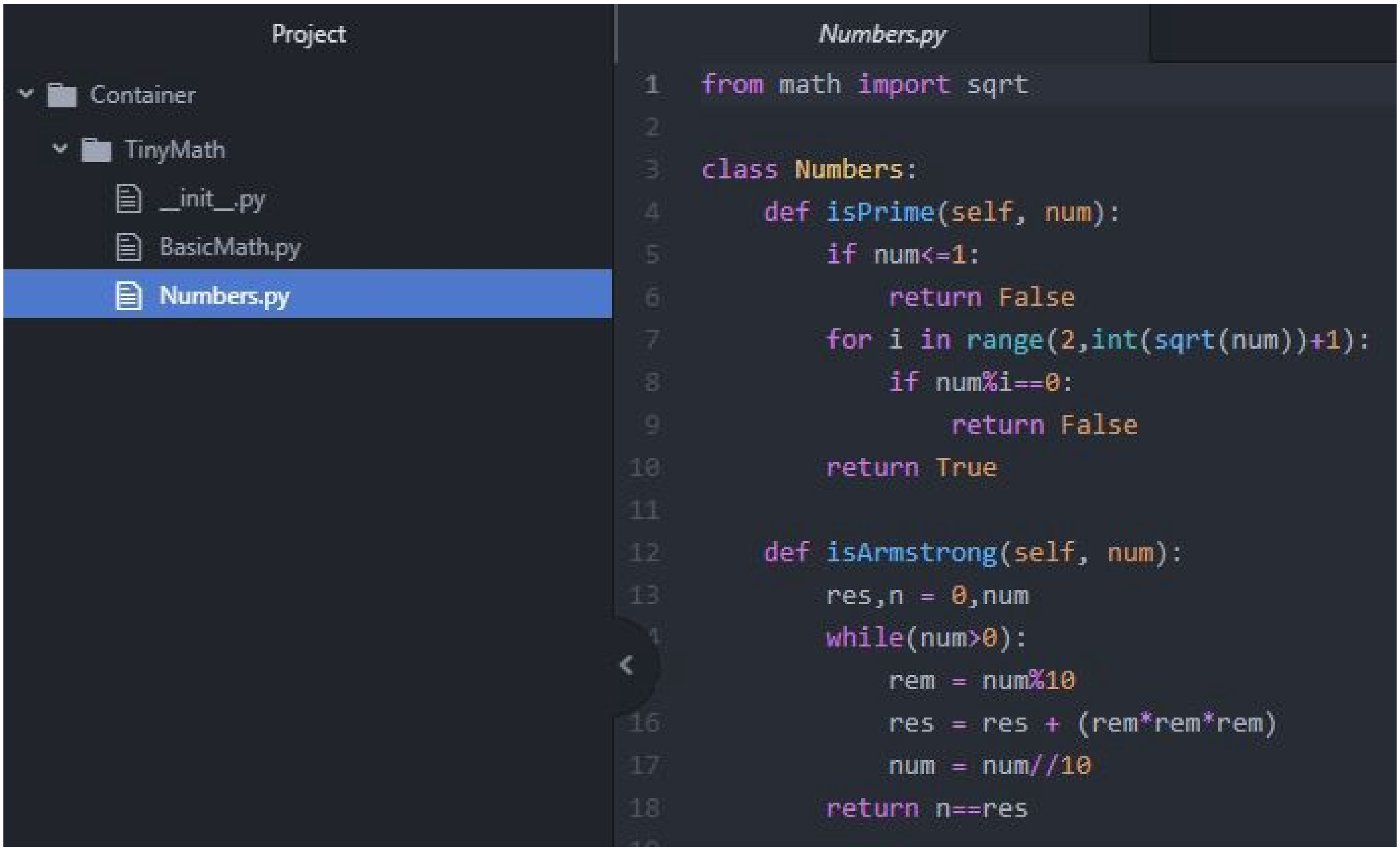


Now I will show all three files one-by-one what is inside them,

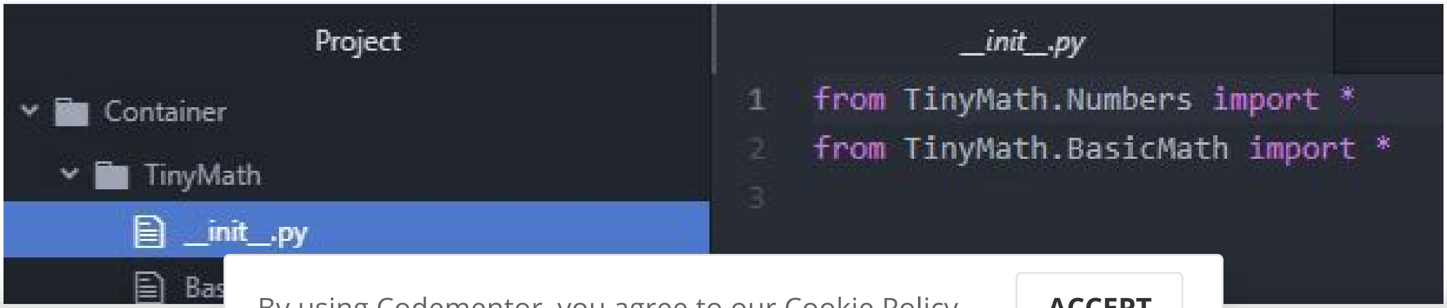




```
        return a+b
    def sub(self, a, b):
        return a-b

```



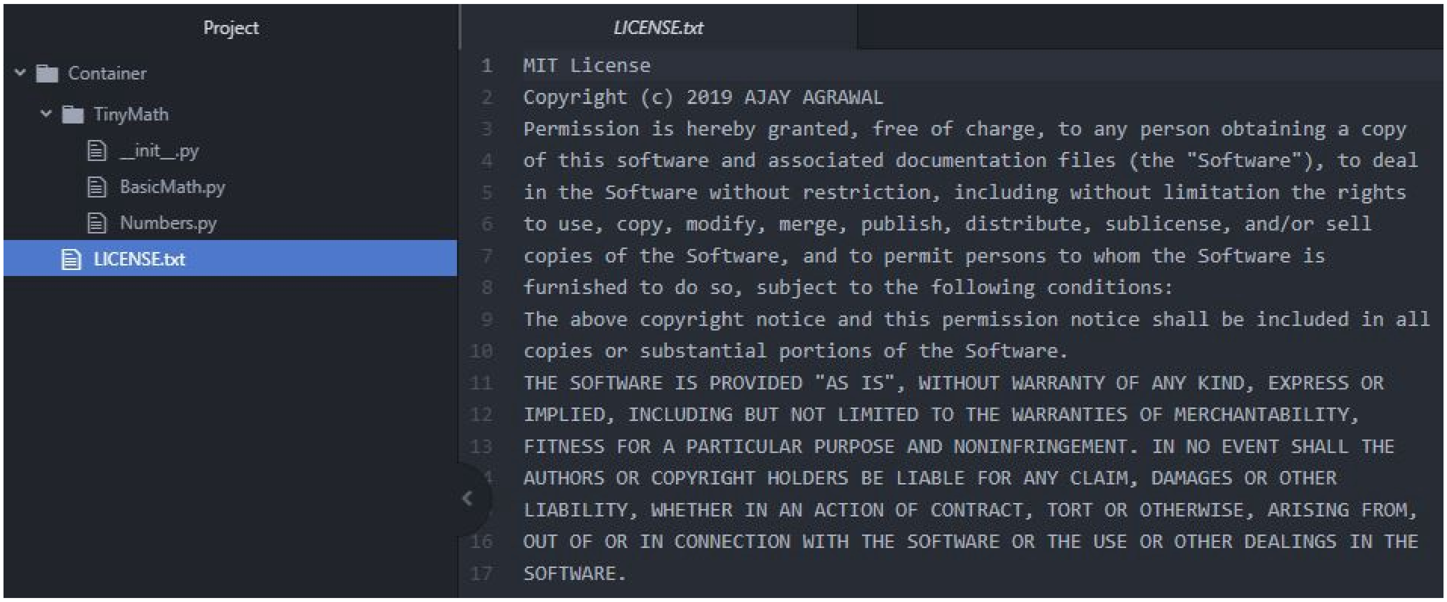
```
from math import sqrt

class Numbers:
    def isPrime(self, num):
        if num<=1:
            return False
        for i in range(2,int(sqrt(num))+1):
            if num%i==0:
                return False
        return True

    def isArmstrong(self, num):
        res,n = 0,num
        while(num>0):
            rem = num%10
            res = res + (rem*rem*rem)
            num = num//10
        return n==res
```



```
from TinyMath.Numbers import *
from TinyMath.BasicMath import *

```

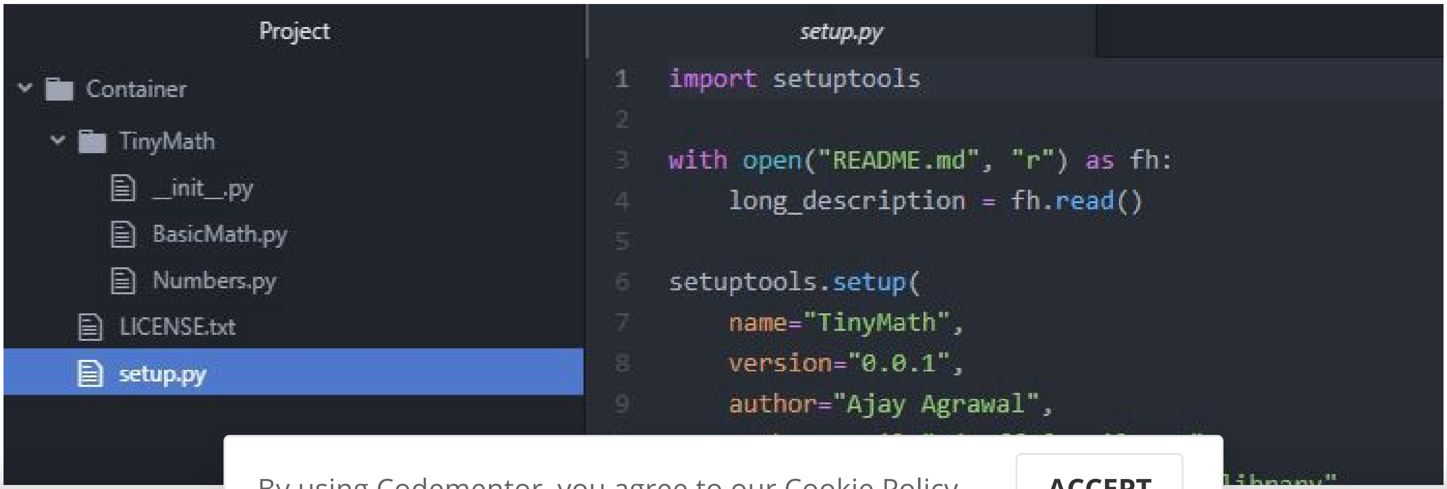
STEP 3: WRITE SOME OFFICIAL FILES

Now we have created enough content for the publishing. For that purpose, first of all, come out of TinyMath folder. We have to create some extra files to upload our package.

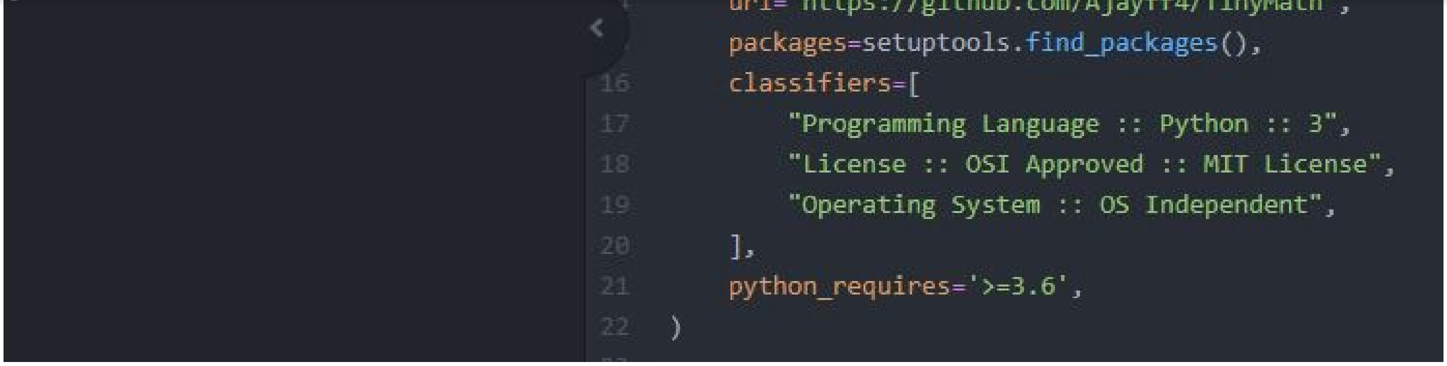


```
MIT License
Copyright (c) 2019 AJAY AGRAWAL
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

For this file just change the year, if it is not right to you and replace your name with mine.

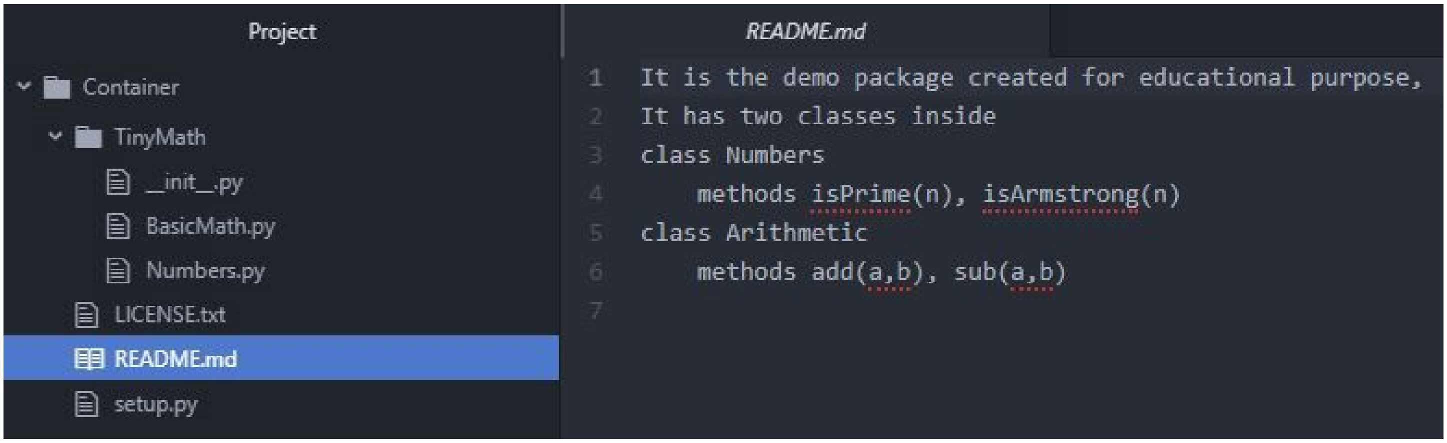


```python
import setuptools

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

setuptools.setup(
    name="TinyMath",
    version="0.0.1",
    author="Ajay Agrawal",
```



```
    url="https://github.com/Ajayff4/TinyMath",
    packages=setuptools.find_packages(),
    classifiers=[
        "Programming Language :: Python :: 3",
        "License :: OSI Approved :: MIT License",
        "Operating System :: OS Independent",
    ],
    python_requires='>=3.6',
)
```

For this file, change the package name from TinyMath to your package name, author name, author_email, description, and URL. I know we have yet not uploaded anything on GitHub. but please be with the flow, we will do it.

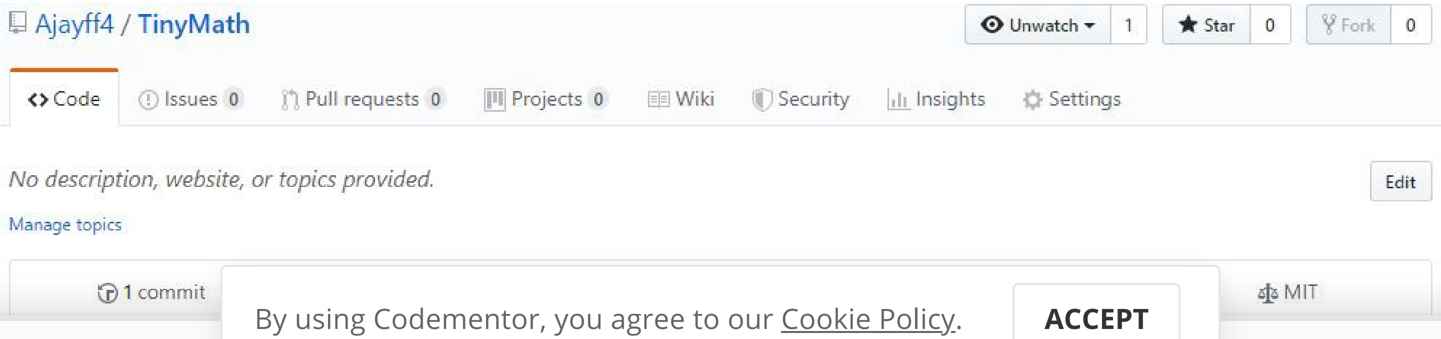


```
It is the demo package created for educational purpose,
It has two classes inside
class Numbers
    methods isPrime(n), isArmstrong(n)
class Arithmetic
    methods add(a,b), sub(a,b)
```

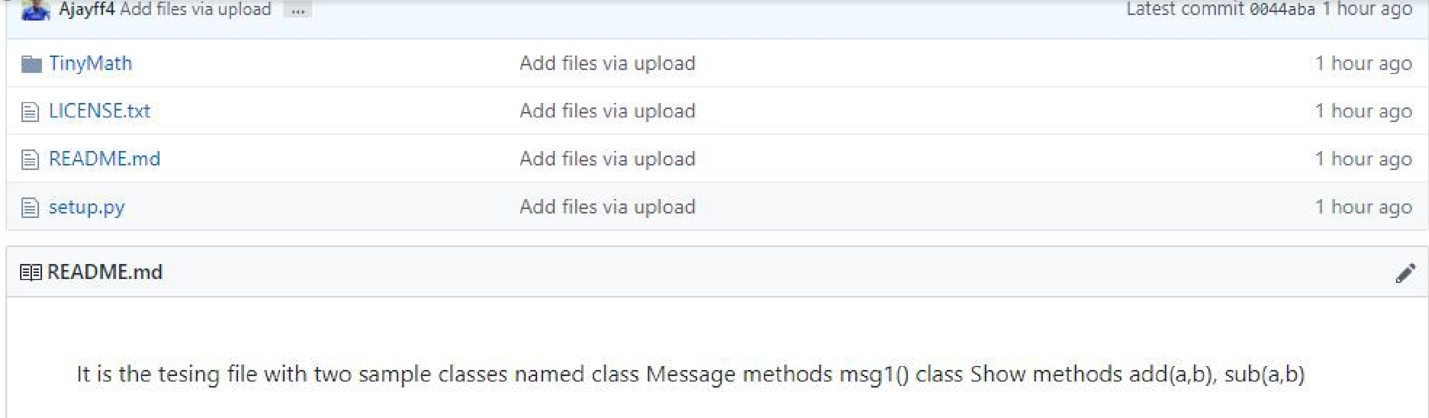
Here as expected you have to describe the sub-packages, classes, and methods defined inside. I'm not that much good on documentation. But you can do it better.

STEP 4: UPLOADING FILES IN GITHUB

Create the new repository on your GitHub and name of the repository should be the name of your package. In my case, it is **TinyMath**. Upload all files of Container folder and write something **commit message** and click on **Commit changes** button.



No description, website, or topics provided.


Ajayff4 Add files via upload ... Latest commit 0044aba 1 hour ago

TinyMath	Add files via upload	1 hour ago
LICENSE.txt	Add files via upload	1 hour ago
README.md	Add files via upload	1 hour ago
setup.py	Add files via upload	1 hour ago

README.md

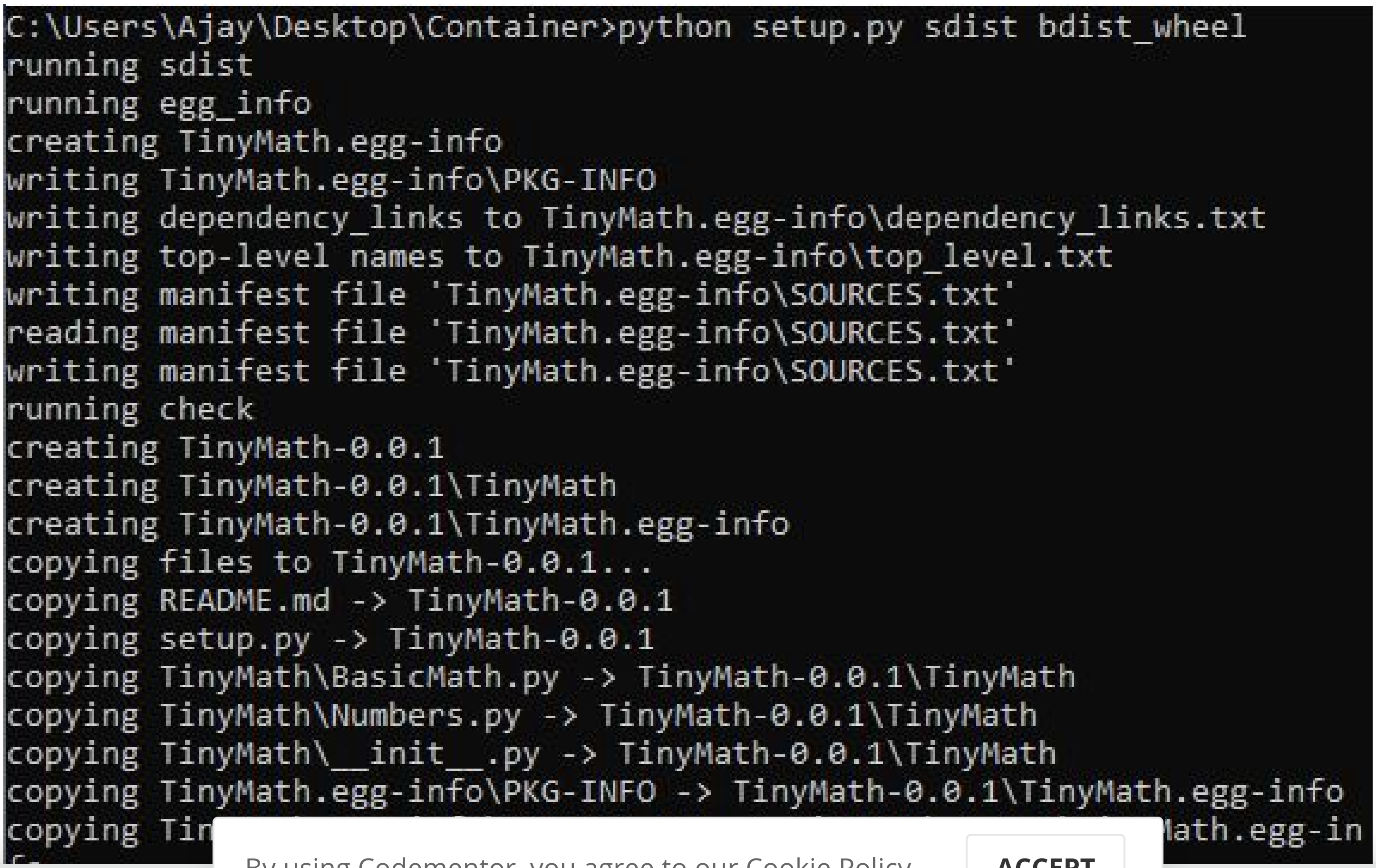
It is the tesing file with two sample classes named class Message methods msg1() class Show methods add(a,b), sub(a,b)

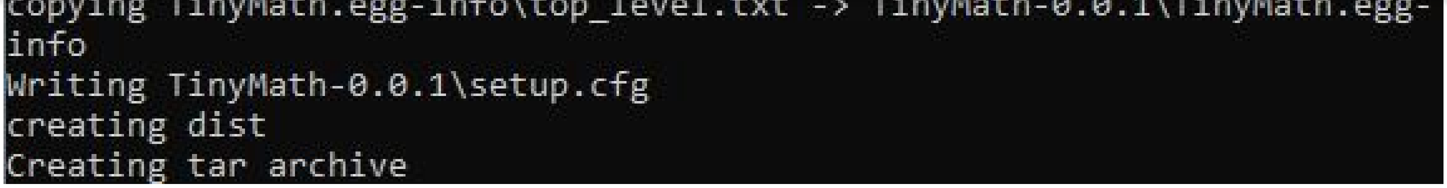
If you click on Clone or download button, then you can see the link that we used in setup.py in url field except .git and the end. This is for the future updates and improvements that you will make.

STEP 5: COMPILE THE setup.py FILE

```
python setup.py sdist bdist_wheel
```

use this command in Container folder.


```
C:\Users\Ajay\Desktop\Container>python setup.py sdist bdist_wheel
running sdist
running egg_info
creating TinyMath.egg-info
writing TinyMath.egg-info\PKG-INFO
writing dependency_links to TinyMath.egg-info\dependency_links.txt
writing top-level names to TinyMath.egg-info\top_level.txt
writing manifest file 'TinyMath.egg-info\SOURCES.txt'
reading manifest file 'TinyMath.egg-info\SOURCES.txt'
writing manifest file 'TinyMath.egg-info\SOURCES.txt'
running check
creating TinyMath-0.0.1
creating TinyMath-0.0.1\TinyMath
creating TinyMath-0.0.1\TinyMath.egg-info
copying files to TinyMath-0.0.1...
copying README.md -> TinyMath-0.0.1
copying setup.py -> TinyMath-0.0.1
copying TinyMath\BasicMath.py -> TinyMath-0.0.1\TinyMath
copying TinyMath\Numbers.py -> TinyMath-0.0.1\TinyMath
copying TinyMath\__init__.py -> TinyMath-0.0.1\TinyMath
copying TinyMath.egg-info\PKG-INFO -> TinyMath-0.0.1\TinyMath.egg-info
copying Ti                                               ath.egg-in
fo
```



```
copying TinyMath.egg-info\top_level.txt -> TinyMath-0.0.1\TinyMath.egg-info
Writing TinyMath-0.0.1\setup.cfg
creating dist
Creating tar archive
```

After compilation, you will see some new folders named like these…

build: build package information.
dist: Contains your .whl file. A **WHL file** is a package saved in the **Wheel** format, which is the standard built-package format used for **Python distributions**. You can directly install a .whl file using

```
pip install some_package.whl
```

on your system.
project.egg.info: An egg package contains compiled bytecode, package information, dependency links, and captures the info used by the setup.py test command when running tests.

Now we are ready to upload our package into PyPI(Python Package Index).

STEP 6: CREATE AN ACCOUNT IN pypi.org AND test.pypi.org

Why two accounts in two websites?

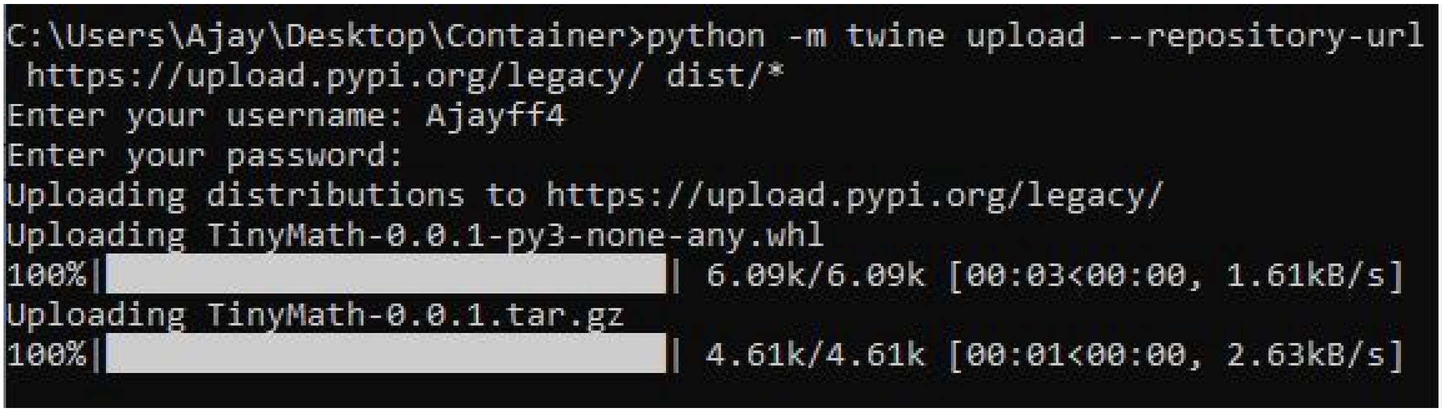
Because if you are only curious to know how to do it. Then you should work with test.pypi.org because it is the server which is getting used to upload the libraries which are made for educational purpose, curiosity, testing and so on. But for right now use pypi.org

We can easily create a new account in both servers. So make one for you I already have. Now finally you have to just fire another command and you are done uploading your package. Which is different for pypi.org and test.pypi.org

For pypi.org

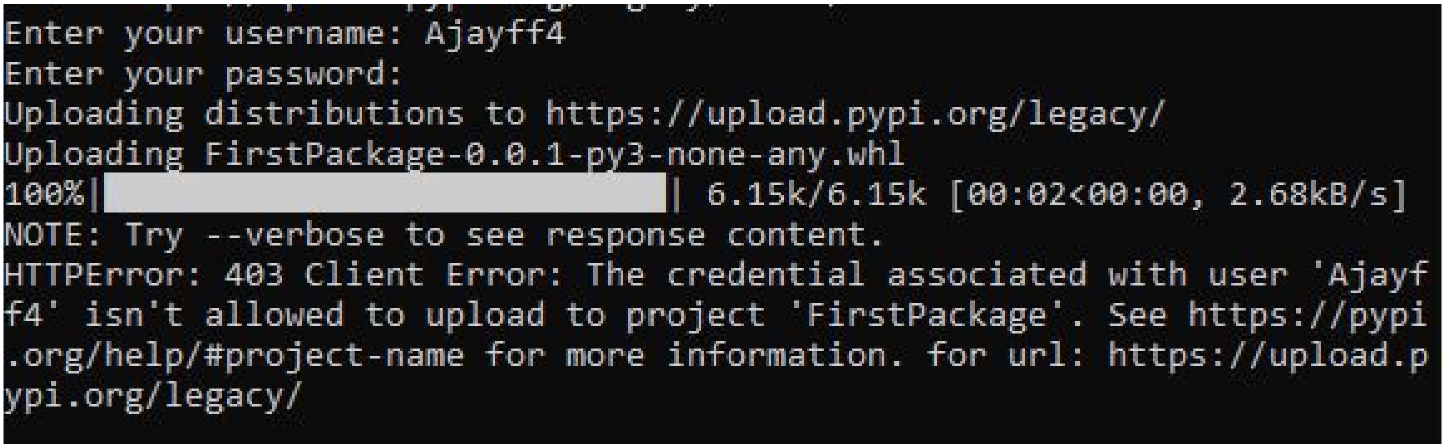
For test.pypi.org

```
python -m twine uplaod --repository-url https://test.pypi.org/legacy/ dist/*
```



```
C:\Users\Ajay\Desktop\Container>python -m twine upload --repository-url
 https://upload.pypi.org/legacy/ dist/*
Enter your username: Ajayff4
Enter your password:
Uploading distributions to https://upload.pypi.org/legacy/
Uploading TinyMath-0.0.1-py3-none-any.whl
100%|████████████████████████████| 6.09k/6.09k [00:03<00:00, 1.61kB/s]
Uploading TinyMath-0.0.1.tar.gz
100%|████████████████████████████| 4.61k/4.61k [00:01<00:00, 2.63kB/s]
```

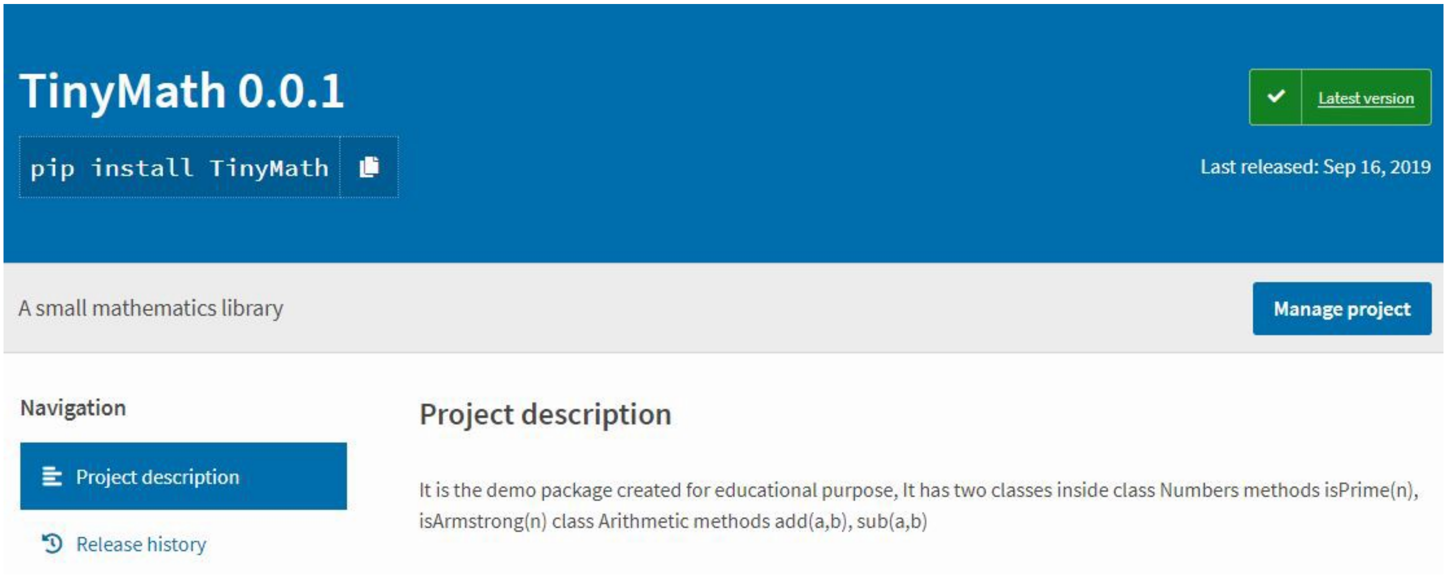
If you see the output like this, then Congratulations you have published your own python package on PyPI server.



```
Enter your username: Ajayff4
Enter your password:
Uploading distributions to https://upload.pypi.org/legacy/
Uploading FirstPackage-0.0.1-py3-none-any.whl
100%|████████████████████████████| 6.15k/6.15k [00:02<00:00, 2.68kB/s]
NOTE: Try --verbose to see response content.
HTTPError: 403 Client Error: The credential associated with user 'Ajayf
f4' isn't allowed to upload to project 'FirstPackage'. See https://pypi
.org/help/#project-name for more information. for url: https://upload.p
ypi.org/legacy/
```

You might get this error sometime, it appears when the package is uploaded by someone with the same name. So you have to just change the Package name in codes and change the folder name t

YOU HAVE USED



TinyMath 0.0.1

```
pip install TinyMath
```

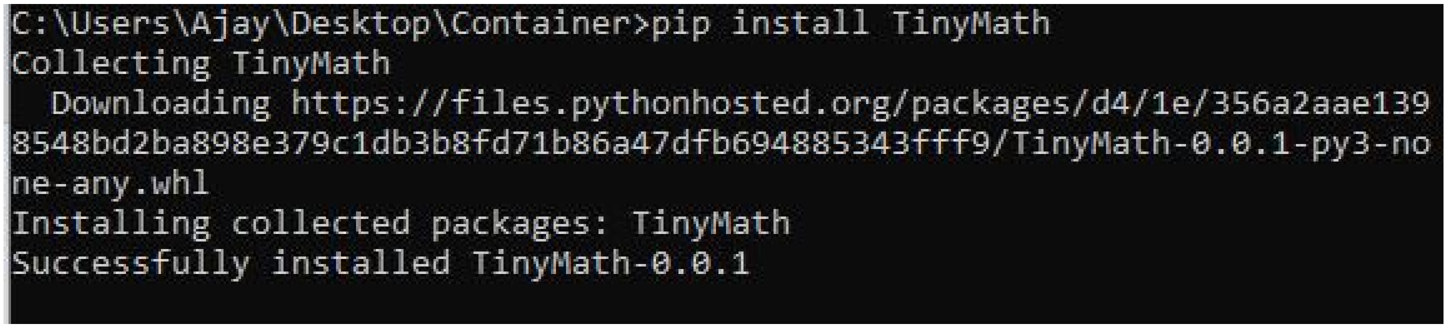
Last released: Sep 16, 2019

A small mathematics library

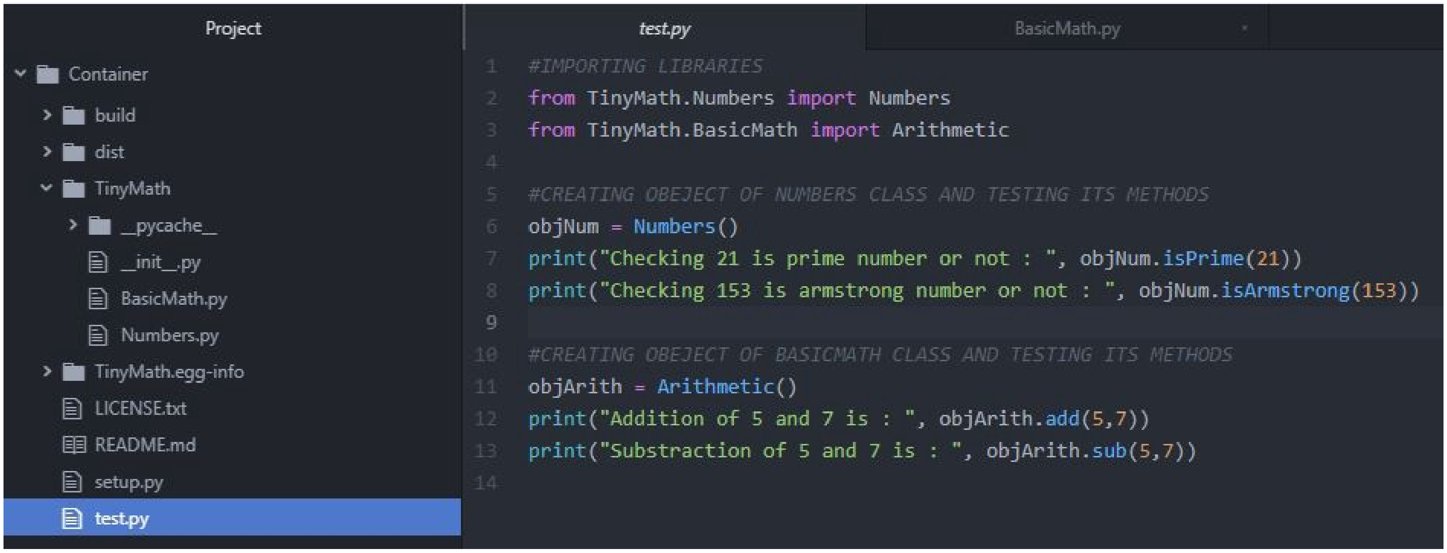
Project description

It is the demo package created for educational purpose, It has two classes inside class Numbers methods isPrime(n), isArmstrong(n) class Arithmetic methods add(a,b), sub(a,b)

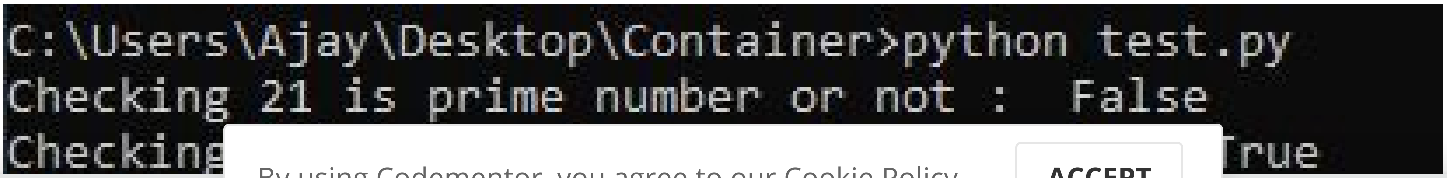
STEP 8: INSTALL YOUR PACKAGE ON YOUR PC AND TEST IT.



```
C:\Users\Ajay\Desktop\Container>pip install TinyMath
Collecting TinyMath
  Downloading https://files.pythonhosted.org/packages/d4/1e/356a2aae139
8548bd2ba898e379c1db3b8fd71b86a47dfb694885343fff9/TinyMath-0.0.1-py3-no
ne-any.whl
Installing collected packages: TinyMath
Successfully installed TinyMath-0.0.1
```



```python
#IMPORTING LIBRARIES
from TinyMath.Numbers import Numbers
from TinyMath.BasicMath import Arithmetic

#CREATING OBEJECT OF NUMBERS CLASS AND TESTING ITS METHODS
objNum = Numbers()
print("Checking 21 is prime number or not : ", objNum.isPrime(21))
print("Checking 153 is armstrong number or not : ", objNum.isArmstrong(153))

#CREATING OBEJECT OF BASICMATH CLASS AND TESTING ITS METHODS
objArith = Arithmetic()
print("Addition of 5 and 7 is : ", objArith.add(5,7))
print("Substraction of 5 and 7 is : ", objArith.sub(5,7))
```



```
C:\Users\Ajay\Desktop\Container>python test.py
Checking 21 is prime number or not :  False
Checking                                  True
```

STEP 9: RELEASE NEW VERSIONS

Repeat the STEP5 to STEP7 and make some changes in setup.py file if they required. Like version
It is the link to the GitHub repository that I created.
You can follow official documentation for more details here

Congratulations for your first python package publishing!!!

Thank you

Python 3.x Python Python package publish Pypi

Enjoy this post? Give **Ajay Agrawal** a like if it's helpful.

♡ 2 SHARE

Ajay Agrawal

Ajay Agrawal

Proud to become Pythonista

Love to make, experiment, create and design with Python. I have some experience in C and C++ also. Right now I am working on React-Redux as a web developer.

FOLLOW

Be the first to share your opinion

Leav

Ajay Agrawal **Enjoy this post?** ♡ 2

Ajay Agrawal **Enjoy this post?**

♡ 2