# Python

# Anaconda SSL error

conda config --set ssl\_verify false

# Linux basic commands

find ./GFG -name sample.txt

# Os

## Assure path exist

def assure\_path\_exists(path):

dir = os.path.dirname(path)

if not os.path.exists(dir):

os.makedirs(dir)

# convert python



## .py to .c

* Install cython

pip install cython or pip3 install cython

* Create file > test.py

print("Convert python to cython")

* In cmd

>cython test.py

* It will create a test.c file

## .py to .pyc

* Create file > test.py

print("Convert python to cython")

* In cmd
* > python
* >>> import py\_compile
* >>> py\_compile.compile(‘test.py’)
* This will create test.pyc file

## .py to .pyx

* Create Pythagoras.py

import time

def count\_triples(limit):

result = 0

for a in range(1, limit + 1):

for b in range(a + 1, limit + 1):

for c in range (b + 1, limit + 1):

if c \* c > a \* a + b \* b:

break

if c \* c == (a \* a + b \* b):

result += 1

return result

if \_\_name\_\_ == '\_\_main\_\_':

start = time.time()

result = count\_triples(1000)

duration = time.time() - start

print(result, duration)

* Python Pythagoras.py
  + 881 9.497968435287476
* Create pyth\_triples.pyx

def count\_triples(limit):

result = 0

for a in range(1, limit + 1):

for b in range(a + 1, limit + 1):

for c in range (b + 1, limit + 1):

if c \* c > a \* a + b \* b:

break

if c \* c == (a \* a + b \* b):

result += 1

return result

* In cmd
* pip install cython
* create main.py

import time

import pyximport; pyximport.install()

import pyth\_triples

def main():

start = time.time()

result = pyth\_triples.count\_triples(1000)

duration = time.time() - start

print(result, duration)

if \_\_name\_\_ == "\_\_main\_\_":

main()

* python main.py
  + 881 6.446093320846558
* Create file setup.py

from distutils.core import setup

from Cython.Build import cythonize

setup(

ext\_modules = cythonize("pyth\_triples.pyx")

)

* python setup.py build\_ext –inplace
* it will create pyth\_triples.c, pyth\_triples.cp36-win\_amd64.pyd and a build folder
* edit main.py

import time

# import pyximport; pyximport.install()

import pyth\_triples

def main():

start = time.time()

result = pyth\_triples.count\_triples(1000)

duration = time.time() - start

print(result, duration)

if \_\_name\_\_ == "\_\_main\_\_":

main()

* python main.py
  + 881 6.341957330703735
* Edit pyth\_triples.pyx as

### Cdef declarations

list

cdef list foo = []

string

cdef char\* c\_string = NULL

float

cdef float x = 5.0

file

cdef FILE\* p

cdef int x,y,z

cdef char \*s

cdef float x = 5.2 (single precision)

cdef double x = 40.5 (double precision)

cdef list languages

cdef dict abc\_dict

cdef object thing

def count\_triples(limit):

cdef int result = 0

cdef int a = 0

cdef int b = 0

cdef int c = 0

for a in range(1, limit + 1):

for b in range(a + 1, limit + 1):

for c in range (b + 1, limit + 1):

if c \* c > a \* a + b \* b:

break

if c \* c == (a \* a + b \* b):

result += 1

return result

* Delete or move file > pyth\_triples.c, pyth\_triples.cp36-win\_amd64.pyd
* Python main.py
  + 881 0.03200125694274902
* python setup.py build\_ext –i
* python main.py
* in ubuntu >
* python setup.py build\_ext -i
* It will crate an .so file in the same directory

# Debugger

* import pdb
* pdb.pm()

## Jupyter notebook

from IPython.core.debugger import set\_trace

set\_trace()

# pickling

## save

import pickle

# take user input to take the amount of data

number\_of\_data = int(input('Enter the number of data : '))

data = []

# take input of the data

for i in range(number\_of\_data):

raw = input('Enter data '+str(i)+' : ')

data.append(raw)

# open a file, where you ant to store the data

file = open('important', 'wb')

# or

file = open('index.pickle', 'wb')

# dump information to that file

pickle.dump(data, file)

# close the file

file.close()

## load

import pickle

# open a file, where you stored the pickled data

file = open('important', 'rb')

# dump information to that file

data = pickle.load(file)

# close the file

file.close()

print('Showing the pickled data:')

cnt = 0

for item in data:

print('The data ', cnt, ' is : ', item)

cnt += 1



import pdb

def transform(x, y):

pdb.set\_trace()

z = x+ y

z = 5

x = 50

y = 60

pdb.set\_trace()

transform(5, 10)

print('z = ' + str(z))