import dnspython as dns

import dns.resolver

result = dns.resolver.query('tutorialspoint.com', 'A')

for ipval in result:

print('IP', ipval.to\_text())

import dnspython as dns

import dns.resolver

result = dns.resolver.query('mail.google.com', 'CNAME')

for cnameval in result:

print ' cname target address:', cnameval.target

result = dns.resolver.query('mail.google.com', 'MX')

for exdata in result:

print ' MX Record:', exdata.exchange.text()

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

loookup\_continue = True

while loookup\_continue:

name = input('Enter the DNS name to resolve: ')

record\_type = input('Enter the query type

[A/MX/CNAME]: ')

answers = dns.resolver.query(name, record\_type)

if record\_type == 'A':

print('Got answer IP address: %s' %[x.to\_text() for x in answers])

elif record\_type == 'CNAME':

print('Got answer Aliases: %s' %[x.to\_text() for x in answers])

elif record\_type == 'MX':

for rdata in answers:

print('Got answers for Mail server records:')

print('Mailserver', rdata.exchange.to\_text(), 'has preference', rdata.preference)

print('Record type: %s is not implemented' %record\_type)

lookup\_more = input("Do you want to lookup more records? [y/n]: " )

if lookup\_more.lower() == 'n':

loookup\_continue = False

import ntplib

from time import ctime

HOST\_NAME = 'pool.ntp.org'

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

params = {}

client = ntplib.NTPClient()

response = client.request(HOST\_NAME)

print('Received time: %s' %ctime(response.tx\_time))

print('ref\_clock: ',ntplib.ref\_id\_to\_text(response.ref\_id,

response.stratum))

print('stratum: ',response.stratum)

print('last\_update: ', response.ref\_time)

print('offset: %f' %response.offset)

print('precision: ', response.precision)

print('root\_delay: %.6f' %response.root\_delay)

print('root\_dispersion: %.6f' %response.root\_dispersion)

list directory

from ftplib import FTP

def main():

ftp = FTP('ftp.ibiblio.org')

ftp.login()

ftp.cwd('pub/academic/biology/') # change to some other subject

entries = ftp.nlst()

ftp.quit()

print(len(entries), "entries:")

for entry in sorted(entries):

print(entry)

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

main()

Lấy vị trí

import googlemaps

from datetime import datetime

gmaps = googlemaps.Client(key='AIzaSyDlCJIftZj3jba1FOnbWtdY3wypgnKCQZk')

# Geocoding an address

geocode\_result = gmaps.geocode('University of Transport and Communications, Hanoi')

# Look up an address with reverse geocoding

reverse\_geocode\_result = gmaps.reverse\_geocode((40.714224, -73.961452))

# Request directions via public transit

now = datetime.now()

directions\_result = gmaps.directions("Hanoi", "Noi Bai Airport", mode="transit", departure\_time=now)

lấy vị tri 2

import urllib.request

import json

endpoint = 'https://maps.googleapis.com/maps/api/directions/json?'

api\_key = 'AIzaSyB\_ODI43-iEC8ZfTaeRX\_kaog1VdKmH8Ow'

origin = input("where are you?").replace(' ','+')

destination = input("where do you want to go").replace(' ','+')

nav\_request = 'origin=()&destination=()&key='.format(origin,destination,api\_key)

request = endpoint+nav\_request

response = urllib.request.urlopen(request).read()

directions = json.loads(response)

print(directions)

directions.keys()

routes = directions['routes']

routes.keys()

routes[0].keys() legs=routes[0]['legs']

routes[0]['legs'] legs[0]['distance']['text']

**Get welcome message (anonymous account)**

import ftplib

with ftplib.FTP('ftp.debian.org') as ftp:

print(ftp.getwelcome())

The with command will automatically close the connection to the server for Python 3 code.

**Directory listing**

import ftplib

with ftplib.FTP('ftp.debian.org') as ftp:

try:

ftp.login()

files = []

ftp.dir(files.append)

print(files)

except ftplib.all\_errors as e:

print('FTP error:', e)

**Send command**

import ftplib

with ftplib.FTP('ftp.debian.org') as ftp:

try:

ftp.login()

wdir = ftp.sendcmd('PWD')

print(ftplib.parse257(wdir))

wdir2 = ftp.pwd()

print(wdir2)

except ftplib.all\_errors as e:

print('FTP error:', e)

**Đổi thư mục**

import ftplib

with ftplib.FTP('ftp.debian.org') as ftp:

try:

ftp.login()

wdir = ftp.pwd()

print(wdir)

ftp.cwd('debian')

wdir2 = ftp.pwd()

print(wdir2)

except ftplib.all\_errors as e:

print('FTP error:', e)

**Tạo thư mục**

import ftplib

from contextlib import closing

with closing(ftplib.FTP('ftp.example.com')) as ftp:

try:

ftp.login('user7', 's$cret')

ftp.mkd('newdir')

files = []

ftp.retrlines('LIST', files.append)

for fl in files:

print(fl)

except ftplib.all\_errors as e:

print('FTP error:', e)

**Lấy kích thước file**

import ftplib

with ftplib.FTP('ftp.debian.org') as ftp:

try:

ftp.login()

# TYPE A for ASCII mode

ftp.sendcmd('TYPE I')

size = ftp.size('debian/ls-lR.gz')

print(size)

except ftplib.all\_errors as e:

print('FTP error:', e)

**Download text file**

import ftplib

import os

with ftplib.FTP('ftp.debian.org') as ftp:

file\_orig = '/debian/README'

file\_copy = 'README'

try:

ftp.login()

with open(file\_copy, 'w') as fp:

res = ftp.retrlines('RETR ' + file\_orig, fp.write)

if not res.startswith('226 Transfer complete'):

print('Download failed')

if os.path.isfile(file\_copy):

os.remove(file\_copy)

except ftplib.all\_errors as e:

print('FTP error:', e)

if os.path.isfile(file\_copy):

os.remove(file\_copy)

**Upload text file**

import ftplib

with ftplib.FTP('ftp.example.com') as ftp:

filename = 'README'

try:

ftp.login('user7', 's$cret')

with open(filename, 'rb') as fp:

res = ftp.storlines("STOR " + filename, fp

if not res.startswith('226 Transfer complete'):

print('Upload failed')

except ftplib.all\_errors as e:

print('FTP error:', e)

**Download binary file**

import os

from ftplib import FTP

def main():

if os.path.exists('patch8.gz'):

raise IOError('refusing to overwrite your patch8.gz file')

ftp = FTP('ftp.kernel.org')

ftp.login()

ftp.cwd('/pub/linux/kernel/v1.0')

with open('patch8.gz', 'wb') as f:

ftp.retrbinary('RETR patch8.gz', f.write)

ftp.quit()

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

main()

**Listing**

from ftplib import FTP

def main():

ftp = FTP('ftp.ibiblio.org')

ftp.login()

ftp.cwd('/pub/academic/astronomy/')

entries = []

ftp.dir(entries.append)

ftp.quit()

print(len(entries), "entries:")

for entry in entries:

print(entry)

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

main()