**NETWORKING**

**FROM URL, RETRIEVE THE HTML CODE / SAVE IN FILE**

**Code:**

import urllib.request

try:

url = urllib.request.urlopen(*"https://www.python.org/"*)

content = url.read()

url.close()

except urllib.error.HTTPError:

print(*"Page Not Found !!!! "*)

exit()

fl = open(*"python.html"*,*"wb"*)

fl.write(content)

fl.close()

**Output: (Screen Shot – File open in Web Browser) – No Console Output**

A screenshot of a computer

Description automatically generated

If url is wrongly entered as:

urllib.request.urlopen(*"https://www.python.og/"*)

instead of

urllib.request.urlopen(*"https://www.python.org/"*)

the output in the console will be:

Traceback (most recent call last):

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\urllib\request.py", line 1344, in do\_open

h.request(req.get\_method(), req.selector, req.data, headers,

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\http\client.py", line 1331, in request

self.\_send\_request(method, url, body, headers, encode\_chunked)

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\http\client.py", line 1377, in \_send\_request

self.endheaders(body, encode\_chunked=encode\_chunked)

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\http\client.py", line 1326, in endheaders

self.\_send\_output(message\_body, encode\_chunked=encode\_chunked)

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\http\client.py", line 1085, in \_send\_output

self.send(msg)

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\http\client.py", line 1029, in send

self.connect()

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\http\client.py", line 1465, in connect

super().connect()

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\http\client.py", line 995, in connect

self.sock = self.\_create\_connection(

^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\socket.py", line 828, in create\_connection

for res in getaddrinfo(host, port, 0, SOCK\_STREAM):

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\socket.py", line 963, in getaddrinfo

for res in \_socket.getaddrinfo(host, port, family, type, proto, flags):

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

socket.gaierror: [Errno 11001] getaddrinfo failed

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

File "C:\Users\kamal\OneDrive\Desktop\Python\Python Examples\t\_networkingEgs\urllibDemo.py", line 4, in <module>

url = urllib.request.urlopen("https://www.python.og/")

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\urllib\request.py", line 215, in urlopen

return opener.open(url, data, timeout)

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\urllib\request.py", line 515, in open

response = self.\_open(req, data)

^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\urllib\request.py", line 532, in \_open

result = self.\_call\_chain(self.handle\_open, protocol, protocol +

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\urllib\request.py", line 492, in \_call\_chain

result = func(\*args)

^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\urllib\request.py", line 1392, in https\_open

return self.do\_open(http.client.HTTPSConnection, req,

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\kamal\AppData\Local\Programs\Python\Python312\Lib\urllib\request.py", line 1347, in do\_open

raise URLError(err)

urllib.error.URLError: <urlopen error [Errno 11001] getaddrinfo failed>

**FROM URL, RETRIEVE THE IMAGECODE / SAVE IN FILE**

**Code:**

import urllib.request

url = *"https://www.python.org/static/img/python-logo.png"*

urllib.request.urlretrieve(url, *"python-logo.png"*)

**Output: (Screen Shot – File open in Web Browser) – No Console Output**

A screenshot of a computer

Description automatically generated

**SOCKET PROGRAMMING**

**STEPS INVOLVED IN SOCKET PROGRAMMING**

Establish a communication between the server and the client using the DCP IP Protocol, using Socket Programming.

First create a server by opening up a socket from the socket module.

Then we will bind it to a host or a machine and a port name and then the server will start listening on that port on the machine.

That is when the client can establish a connection by opening a socket connecting to the server that is running when the connection comes in.

From the client the request comes in, the server will accept the client connection and then the connection is established.

They both can send and receive messages at that point.

Finally when they are done, they can close the connection.

A close-up of a computer program

Description automatically generated

**CREATING A TCP/IP CLIENT**

**Code:**

import socket

host = *"localhost"*

port=4000

soc = socket.socket()

#CONNECT TO THE HOST AND PORT

soc.connect((host,port))

msg = soc.recv(1024)

while msg:

print(*"Recieved :"*, msg.decode())

msg = soc.recv(1024)

print()

soc.close()

**Output:**

Recieved : Hello, How are you?

Recieved : bye

**CREATE A TCP/IP SERVER**

**Code:**

import socket

host = *"localhost"*

port=4000

# AF\_INET --> INTERNET PROTOCOL VERSION 4

# FOR TCP/IP WE USE SOCK\_STREAM

sockVal = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

#BIND THE SOCKET TO A HOST AND PORT / BIND METHOD ACCEPTS A SET

sockVal.bind((host,port))

print(*"Server listening on port : "*, port)

#PARAMETER - NO OF CONNECTIONS TO CONNECT

sockVal.listen(1)

# ACCEPT A CLIENT CONNECTION WHEN A CLIENT TRIES TO CONNECT

#RETURN THE CONNECTION AND ALSO THE ADDRESS WHICH RETURNS FROM THE CLIENTS ADDRESS

con, addr = sockVal.accept()

print(*"Connection from : "*, str(addr))

#CONVERT THE STRING TO BINARY USING b IN FRONT OF THE STRING

con.send(*b"Hello, How are you? "*)

# OR

# ANOTHER METHOD TO CONVER TO BINAY

con.send(*"bye"*.encode())

#CLOSE THE CONNECTION

con.close()

**Output:**

Server listening on port : 4000

**CREATE A FILE SERVER**

**Code:**

import socket

host = *"localhost"*

port=6767

# AF\_INET --> INTERNET PROTOCOL VERSION 4

# FOR TCP/IP WE USE SOCK\_STREAM

sockVal = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

#BIND THE SOCKET TO A HOST AND PORT / BIND METHOD ACCEPTS A SET

sockVal.bind((host,port))

print(*"Server listening on port : "*, port)

#PARAMETER - NO OF CONNECTIONS TO CONNECT

sockVal.listen(1)

# ACCEPT A CLIENT CONNECTION WHEN A CLIENT TRIES TO CONNECT

#RETURN THE CONNECTION AND ALSO THE ADDRESS WHICH RETURNS FROM THE CLIENTS ADDRESS

con, addr = sockVal.accept()

print(*"Connection from : "*, str(addr))

#READ DATA FROM THE CLIENT

recFileName = con.recv(1024)

try:

flName = open(recFileName,*"rb"*)

flCont = flName.read()

con.send(flCont)

except FileNotFoundError:

con.send(*b"File does not exists"*)

finally:

flName.close()

#CLOSE THE CONNECTION

con.close()

**Output:**

Server listening on port : 6767

Connection from : ('127.0.0.1', 55819)

**CREATE A FILE CLIENT**

**Code:**

import socket

host = *"localhost"*

port=6767

soc = socket.socket()

#CONNECT TO THE HOST AND PORT

soc.connect((host,port))

#INPUT A FILE NAME

inputflName= input(*"Enter a fileName: "*)

soc.send(inputflName.encode())

fileContent=soc.recv(1024)

print(fileContent.decode())

soc.close()

**Output:**

Enter a fileName: python.html

<!doctype html>

<!--[if lt IE 7]> <html class="no-js ie6 lt-ie7 lt-ie8 lt-ie9"> <endif]-->

<!--[if IE 7]> <html class="no-js ie7 lt-ie8 lt-ie9"> <![endif]-->

<!--[if IE 8]> <html class="no-js ie8 lt-ie9"> <![endif]-->

<!--[if gt IE 8]><!--><html class="no-js" lang="en" dir="ltr"> <!--<![endif]-->

<head>

<!-- Google tag (gtag.js) -->

<script async src="https://www.googletagmanager.com/gtag/js?id=G-TF35YF9CVH"></script>

<script>

window.dataLayer = window.dataLayer || [];

function gtag(){dataLayer.push(arguments);}

gtag('js', new Date());

gtag('config', 'G-TF35YF9CVH');

</script>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<link rel="prefetch" href="//ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js">

<link rel="prefetch" href="//ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js">

<meta name="application-name" content="Python.org">

<meta name="msappli

**SENDING EMAILS**

Fist step is to create a message with body, subject, from and to information.

Once we have the message, we will use that message, create SMTP Server that will connect to a particular email server. Once its connected it will log in send a message and quit the connection.

![A blue square with black text

Description automatically generated

**Login to the Gmail Account**

A screenshot of a computer

Description automatically generated

**Search for the App Passords. On which give a app usage name and the password generated can be used.**

A screenshot of a computer

Description automatically generated

**App Password from Gmail – SMTP Server**

**jihm tuvr eidp yaas**

**Code:**

import smtplib

from email.mime.text import MIMEText

body=*"This is a test email from Python..."*

msg= MIMEText(body)

msg[*'From'*] = *"kamalrkumar@gmail.com"*

msg[*'To'*] = *"kamalrkumar@gmail.com"*

msg[*'Subject'*] = *"Test - Python"*

# OPEN UP AN SMTP SERVER CONNECTION

server = smtplib.SMTP(*'smtp.gmail.com'*,587)

#ENABLE A TLS CONNECTION - WILL ENABLE A SECURED CONNECTION SERVER

server.starttls()

# USER THE GMAIL ID AND FOR THE PASSWORD USE THE PASSWORD FROM APP PASSWORD FOR GMAIL.

server.login(*"kamalrkumar@gmail.com"*,*"jihmtuvreidpyaas"*)

server.send\_message(msg)

print(*"Mail Sent"*)

server.quit()

**Output (Console):**

Mail Sent

**Output (Email):**

A white background with black dots

Description automatically generated

**QUIZ**

A screenshot of a computer

Description automatically generated

A screenshot of a email

Description automatically generated