Socket Module: A logicacl connection bw a client and a server

Get IP Address

In [1]:

```
import socket

host='www.youtube.com'

try:
    addr=socket.gethostbyname(host)
    print('IP Address=',addr)

except:
    print('The website does not exist')
```

```
IP Address= 172.217.166.46
```

URL: Accessing different parts of URL

In [2]:

```
import urllib.parse

url='http://www.iitk.ac.in/doaa/courses-of-study'

tpl=urllib.parse.urlparse(url)

print(tpl)

print("sceme=",tpl.scheme)
print("net location=",tpl.netloc)
print("Parameters=",tpl.params)
print("Port no.=",tpl.port)
print("Total URL=",tpl.geturl)
```

```
ParseResult(scheme='http', netloc='www.iitk.ac.in', path='/doaa/courses-of-st
udy', params='', query='', fragment='')
sceme= http
net location= www.iitk.ac.in
Parameters=
Port no.= None
Total URL= <bound method ParseResult.geturl of ParseResult(scheme='http', net
loc='www.iitk.ac.in', path='/doaa/courses-of-study', params='', query='', fra
gment='')>
```

Downloading a webpage from net

copying the webpage: getting url and reading it into variable content then putting it into a html file

In [3]:

```
import urllib.request

try:
    file=urllib.request.urlopen("http://www.iitk.ac.in/")
    content=file.read()

except urllib.error.HTTPerror:
    print("webpage does not exist")
    exit()

f=open('webpage.html','wb')
f.write(content)
f.close()
```

Downloading an image from net

In [4]:

```
import urllib.request

url='http://www.iitk.ac.in/new/images/cards/counselling-service.jpg'

download=urllib.request.urlretrieve(url,'immage.png')
```

TCP/IP to create server which sends message to client

TCP: Transmission Control Protocol
Ip: Internet Protocol

Server Side

In [ ]:

```
import socket

host='localhost'
port = 5000

#create a socket at server side using TCP/IP protocol
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
#for IPV4 socket.AF_INET, for IPV6 socket.AF_INET6
#we can also use s=socket.socket(socket.AF_INET,socket.SOCK_STREAM) to use def
ault values

#bind the socket and port
s.bind((host,port))

# allow max 1 connection to the socket
s.listen(1)

#wait till client accepts connection
c, addr=s.accept()
#c is connection object for the server, addr is address of client

print('Connection from:',str(addr))

c.send(b"Hello there")
msg='See You Later'
c.send(msg.encode())

#disconnect from server
c.close()
```

Client Side

In [ ]:

```
import socket

#take the server name and port no.
host='localhost'
port=5000

#create a socket at server side
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)

#bind the socket and port
s.connect((host,port))

#recieve a message from server, 1024 B at a time
msg=s.recv(1024)

#recieve message string till msg in not empty
while msg:
    print("Recieved: ",msg.decode())
    msg=s.recv(1024)

s.close()
```

Scan the available ports

In [2]:

```
import socket

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)

server='www.iitk.ac.in'

def pscan(port):
    try:
        s.connect((server,port))
        return True
    except:
        return False

for x in range(0,50):
    #can go upto 68000 but scanning ports takes a lot of time
    if pscan(x):
        print('Your port',x,'is open')
    else:
        print('Your port',x,'is close')
```

```
Your port 0 is close
Your port 1 is close
Your port 2 is close
Your port 3 is close
Your port 4 is close
Your port 5 is close
Your port 6 is close
Your port 7 is close
Your port 8 is close
Your port 9 is close
Your port 10 is close
Your port 11 is close
Your port 12 is close
Your port 13 is close
Your port 14 is close
Your port 15 is close
Your port 16 is close
Your port 17 is close
Your port 18 is close
Your port 19 is close
Your port 20 is close
Your port 21 is close
Your port 22 is close
Your port 23 is close
Your port 24 is close
Your port 25 is close
Your port 26 is close
Your port 27 is close
Your port 28 is close
Your port 29 is close
Your port 30 is close
Your port 31 is close
Your port 32 is close
Your port 33 is close
Your port 34 is close
Your port 35 is close
Your port 36 is close
Your port 37 is close
Your port 38 is close
Your port 39 is close
Your port 40 is close
Your port 41 is close
Your port 42 is close
Your port 43 is close
Your port 44 is close
Your port 45 is close
Your port 46 is close
Your port 47 is close
Your port 48 is close
Your port 49 is close
```

```
In [ ]:
```