**What is a Network**

Suppose your friend sends you a postcard message to you. Postcard containing message send by your friend travels over certain defined routes, it passes through some regional post office and reaches to the local post office and finally reaches to your home.

Similarly data send by one computer reaches another computer over the internet through a defined path, that defined path is called as Network. Data send can be in the form of image, video, document etc.

**TCP-IP Model / OSI Model**

**History of Computer Industry**

Years ago there were two leading computer equipment manufacturer in the industry IBM and Digital Equipment Corporation. As both were the strong competitors of each other they were manufacturing the equipments which were not compatible with each other.

So in one organization if one department is using IBM equipments and other department is using digital electronic equipment, both cannot share the data due to incompatibility.

So there was a need to set certain standard set of rule (Protocols) for device manufacturer so that the devices of the different manufacturer can share data with each other.

The rules or protocols that is mandatory for every equipment manufacturer to be followed in order to have device compatibility for sharing data is known as OSI model. [OSI- Open System Interconnect]. There was another competent model known as TCP-IP model developed. Since OSI model is just a reference model, TCP-IP model become the industry standards.

**Layer Approach of OSI model/TCP-IP model.**

**Application of OSI Model**

There are different departments in a company such as Maintenance, Accounts, HR etc. So if you have problem with your salary you know where to contact i.e Accounts Department. If you know that one of your machines is not working properly you contact to Maintenance department. Similarly if the Network Programmer knows these layers of OSI Model or TCP-IP model and their functioning, so during Troubleshooting he can locate the layer which is not working properly or malfunctioning and rectify the problem easily.

**Layer diagram of OSI and TCP-IP model**

OSI-Model TCP-IP Model

|  |
| --- |
| **7. Application** |
| **6.Presentation** |
| **5.Session** |
| **4.Transport** |
| **3.Network** |
| **2.Data Link** |
| **1.Physical** |

|  |
| --- |
| **Application** |
| **Transport** |
| **Internet** |
| **Network Interface** |

**Application Layer**

This is the seventh layer in the OSI-Model. Application layer is nothing to do with the applications installed on your system (Computer). Applications those who want to communicate over network at that time application layer is invoked. If you have an application that do not communicate over any type of network, then by running that application, application layer will not be invoked. Thus application layer communicate with the Network aware applications.

**Presentation Layer**

This is the Second Layer in the OSI-Model. The presentation layer is a layer at which the data is converted in the form suitable for the network to carry. If it is image or video or any document etc., at presentation layer this data is converted into the network specific format.

**Session Layer [*Create and Maintained Session*]**

This is the Third layer in the OSI-Model. When user login to a dynamic website, session for that user is created and is maintained till user logout. At the same time different user from different location login to the same website, session of each and every user is maintained separately. So if one of the user logout its session is closed and it does not affect the session of the other logged in users.

Similarly when different application on the system (Different Browser) tries to access the network, individual session for each application is started and maintained in this session layer. So if one application is closed it does not harm the other session as it is maintained separately.

**Transport Layer.**

This is the Third layer in the OSI-Model. Information from the above three layer- Application, Presentation and Session layer is passed to the transport layer. Transport layer breaks the information into small segments.

DATA Segment 1 + Segment 2

TH Segment 1 TH Segment 2

This layer adds Header to the Segment and Send it to Next Layer.

Transport Layer performs Two Function:

* Selecting the Type of Communication

1. Whether to have reliable communication.
2. Whether to have Unreliable communication

* To assign Port Number.

*Selecting the Type of communication*

When application requires reliable communication, transport layer uses TCP [Transmission Control Protocol] and when it requires unreliable communication it uses UDP [User Datagram protocol]. By word unreliable it does not mean it is unsafe. The only difference between the reliable and Unreliable communication is, in reliable communication the sender receive the acknowledgement of the data received from the receiver and in unreliable there is no acknowledgement from the receiver. Thus unreliable communication is faster as compared to the reliable communication. An example of unreliable communication is watching live cricket match, live streaming etc.

Many time when your are watching some live event, if there is some problem in the data transmission, you see a green dot on the screen, that green dot says that information for that pixel is not received and since it is a unreliable communication, the receiver do not send the acknowledgement of that non-received information to the sender.

*Assigning a Port Number*

Port number is the Number attached to the IP address. Transport layer attach two type of Port number.

* Random source Port Number
* Port Number of the destination.

For a web server the default port number is 80. If the data is to be send to the web server, then transport layer attach a port number 80 to the IP address. Thus transport layer attaché a destination port number to the IP address and pass it to network layer. The combination of IP address and port number is called as Socket.

**IP ADDRESS + PORT NUMBER = SOCKET**

Transport layer also add the Source Port number to the Socket.

**What is the use of Source port Number?**

If suppose that there are two applications running and it points to the same IP address, when reply comes from the IP address, transport layer must know to direct which information to which application, it does it by using the source port number.

**Network Layer.**

Data or Segment from the transport layer is passed to the network layer. Network layer adds the header to the segment. When network layer header is added to the data or segment coming from transport layer, that information is called as Packet.

Segment from transport Network

Layer + Header = PACKET

Network layer is also responsible for finding the best logical path to reach to the destination IP address.

NH TH Segment1 NH TH Segment2

**Data Link [**MAC addressing and Error Checking**]**

When the packet comes from Network layer to Data link layer, Data link layer adds header to the packets. Data link unit which contains the packet and Data link header is called as Frame.

**PACKET + DATA LINK HEADER = FRAME**

NH TH DH Segment 1 NH TH DH Segment 2

This layer is responsible for MAC addressing. MAC stands for Media Access Controller. MAC address is a hardware address. Every Network Interface card has its MAC address that cannot be changed. Error checking is done at this layer. If there is certain error in travelling the data from Physical Layer to Data link layer [Reverse Flow], data link mechanism uses certain error checking mechanism to find the Error.

**Physical Layer** [Actual Data Transfer, Data Unit: Bit]

This layer contains wires, cables, hardware ports etc. At this layer Data received from Data link layer is converted into bits i.e 1’s and 0’s. (Machine language)

*What are Headers in Transport, Network and Data Link Layer?*

Over the layer Data is passed from one layer to another layer. Data consist of two parts: Header and Body. Headers contain the protocols (rules) for that layer for transmitting the data to other layer.

**IP address**

House or building in which we stay has a unique address due to which people can send you the message or can come to your place.

Similarly if two device or computer wants to send and receive data over a network, it need address, this address is called as IP address. In short address of the computer is called as IP address.

According to IP V2, IP address is define as 32 bit number e.g.: 192.168.0.102. Number between the dots can range from 0 to 255.

**IP- Internet Protocols**

Types of IP Address

Public IP: This IP is provided by Internet Service Provider (ISP)

Private IP: IP address of Router.

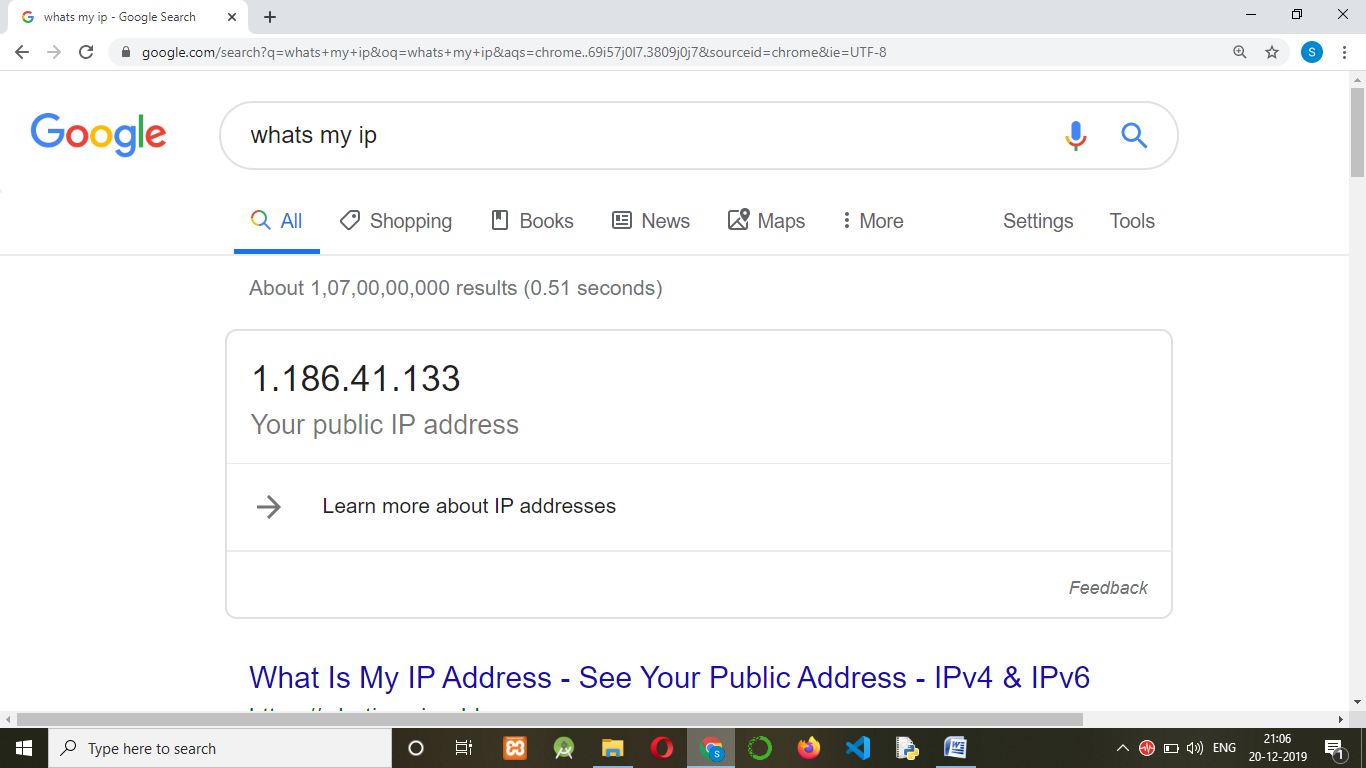
IP address

Public IP Private IP

Static Dynamic Static Dynamic

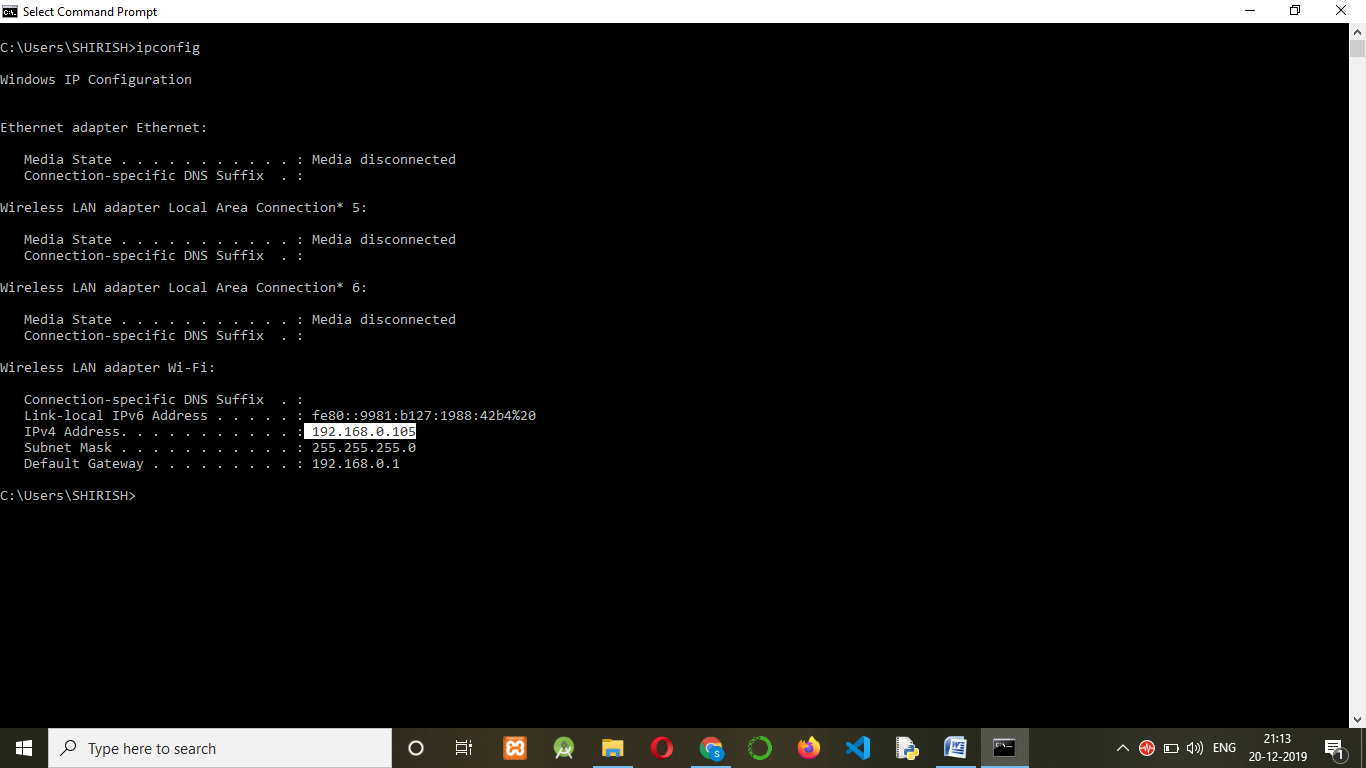
To know Public IP address:

* Open your Browser
* Type “What’s my IP” in the browser, you can see the Public IP address.



**To know Private IP:**

* Open cmd Prompt
* For windows type ‘ipconfig’, you get your private IP address.



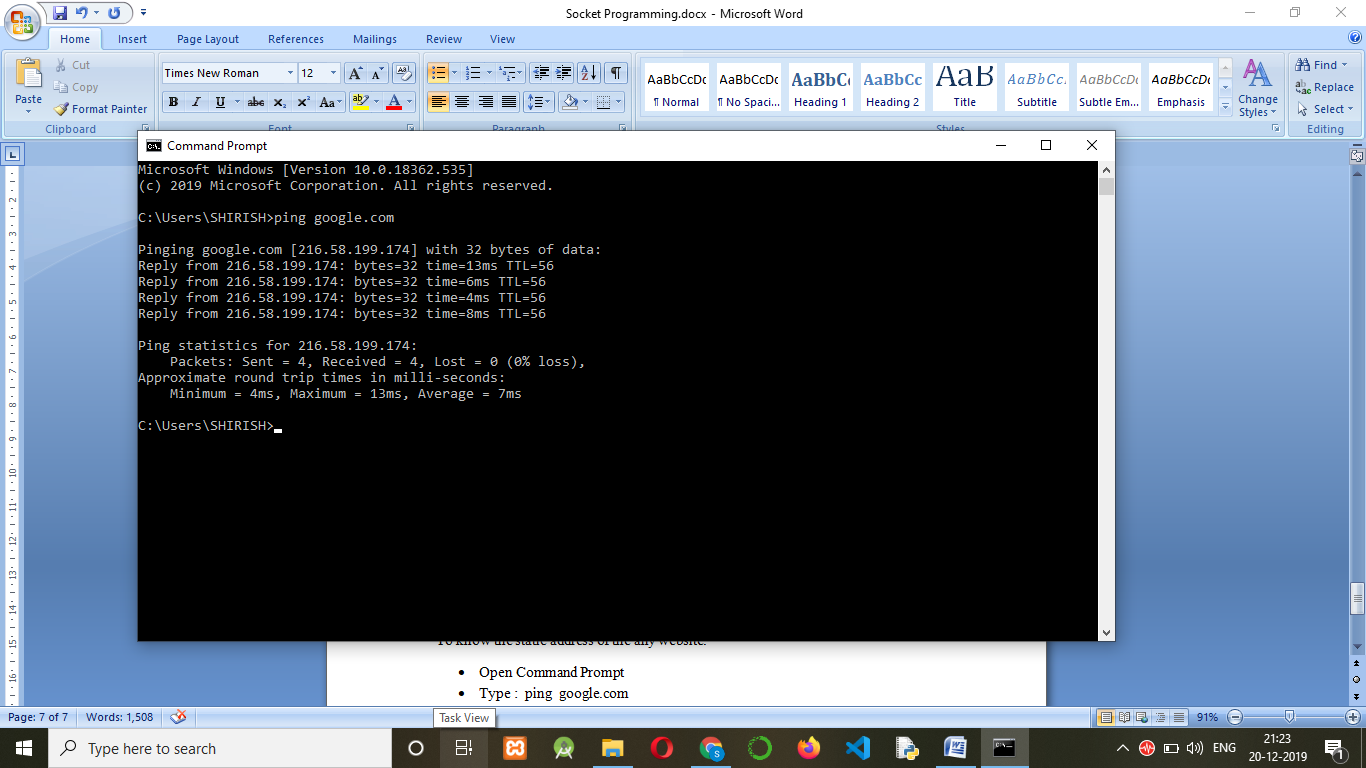
* For Mac and Linux open Terminal
* Type: ifconfig

**Static IP Address:**

Website and Servers has static IP addresses.

To know the static address of the any website.

* Open Command Prompt
* Type : ping google.com
* You get the IP address of the Google.com. Copy that IP address and paste in the browser, you will get www.google.com



**Port**

You can refer your Building **address** as the **IP Address** and your **Room** as a **Port.**

**Building address= IP address**

Room 1 Room 2

Port 1 Port 2

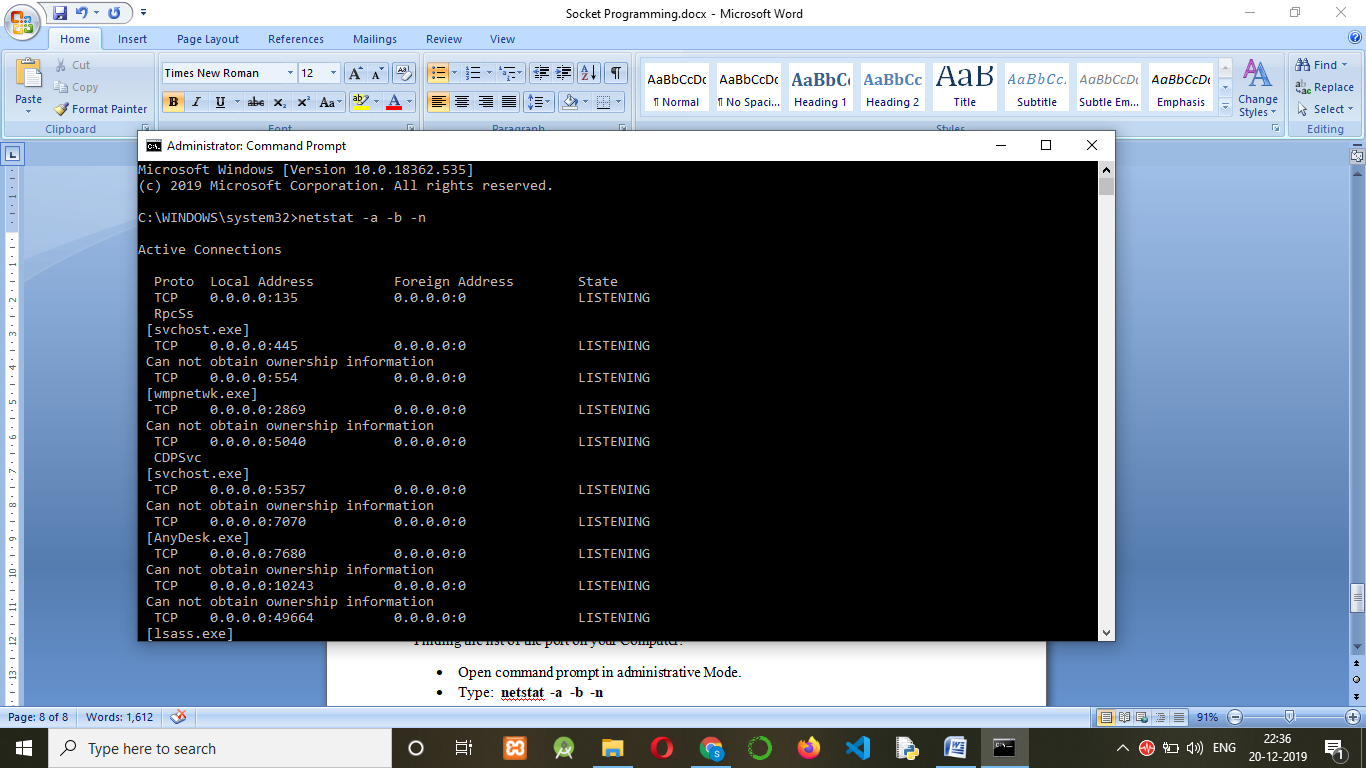
Room 3 Room 4

Port 3 Port 4

For a one device (Computer) to communicate with the other device (Computer) it not only require the IP address but also require PORT

Finding the list of the port on your Computer:

* Open command prompt in administrative Mode.
* Type: **netstat -a -b -n**

****

**List of some of the common Ports**

|  |  |  |
| --- | --- | --- |
| **PORT** | **PROTOCOL** | **FUNCTION** |
| 80 | HTTP | Web Pages |
| 443 | HTTPS | Secure Web Pages |
| 20 | FTP | File Transfer |
| 25 | SMTP | Sending Mail |

Type in the Browser.

* [www.google.com:80](http://www.google.com:80) : You will see the Google page.
* [www.google.com:443](http://www.google.com:443): Your will see the Google Page
* [www.google.com:20](http://www.google.com:20) : You will get the Message “**Site can’t be Reached**”

Socket Module in Python

**Importing Socket Module**

For socket Programming we need certain methods, these methods are available in the built-in Socket Module in python. So before starting socket programming, socket module must be imported in the file.

*import socket*

**Creating Socket Module object or instance**

In order to invoke or call the built in methods available in the socket module, we need to create socket object. Object is created by calling socket () constructor in the socket module.

syntax:

s = socket. socket(family name,type)

family name = socket.AF\_INET (default) type = socket.SOCK\_STREAM [TCP]

type=socket.SOCK\_DGRAM [UDP]

*s = socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)*

**Creating a Socket**

As discussed socket contains an IP address and Port Number. With the help of bind() method a socket is created by binding IP address and Port Number.

For localhost IP address= 127.0.0.1 and Port Number= 1234

Note: You can assign any port number except the default port number such as port number 80 for web-server, 443 for secure web-server, 25 SMPT for sending mail.

Parameters to the bind method must be send in the form of tuple, since tuple are immutable.

syntax:

ModuleObject.bind((“IP address”, Port number))

*s.bind ((“127.0.0.1”, 1234))*

**Listen Method**

In client-server communication, server must be always in listening mode i.e everytime server checks whether a request is coming from any client or not. Listen method is used for server to be in alert mode for any request coming from client.

syntax:

ModuleObject.listen (n) where n = number of clients connections, connecting with the server.

*s.listen (5) # server will be communicating with the 5 clients*

**Connect Method**

In client server, client must connect to the server. connect () method is used to connect to the server.

Moduleobject.connect ((“Server IP address”, Server Port No))

*s.connect ((“127.0.0.1”, 1234))*

**accept() Method**

When a client sends a request to the server, server uses accept () method to accept a request from the client.

Accept method return two objects:

1. One is the connection object of the client.
2. Information of the client i.e client IP address and Port Number.

Connection object is required to communicate with the client.

*conn, add=s.accept()*

*conn* object contains connection to the client and *add* contains client information i.e IP address of the client and Port Number.

**Sending Message**

For sending message from Server to Client and client to Server send () method is used.

syntax:

send(bytes(message , encoding=’utf-8’))

**Receiving Message**

To receive message recv() method is used.

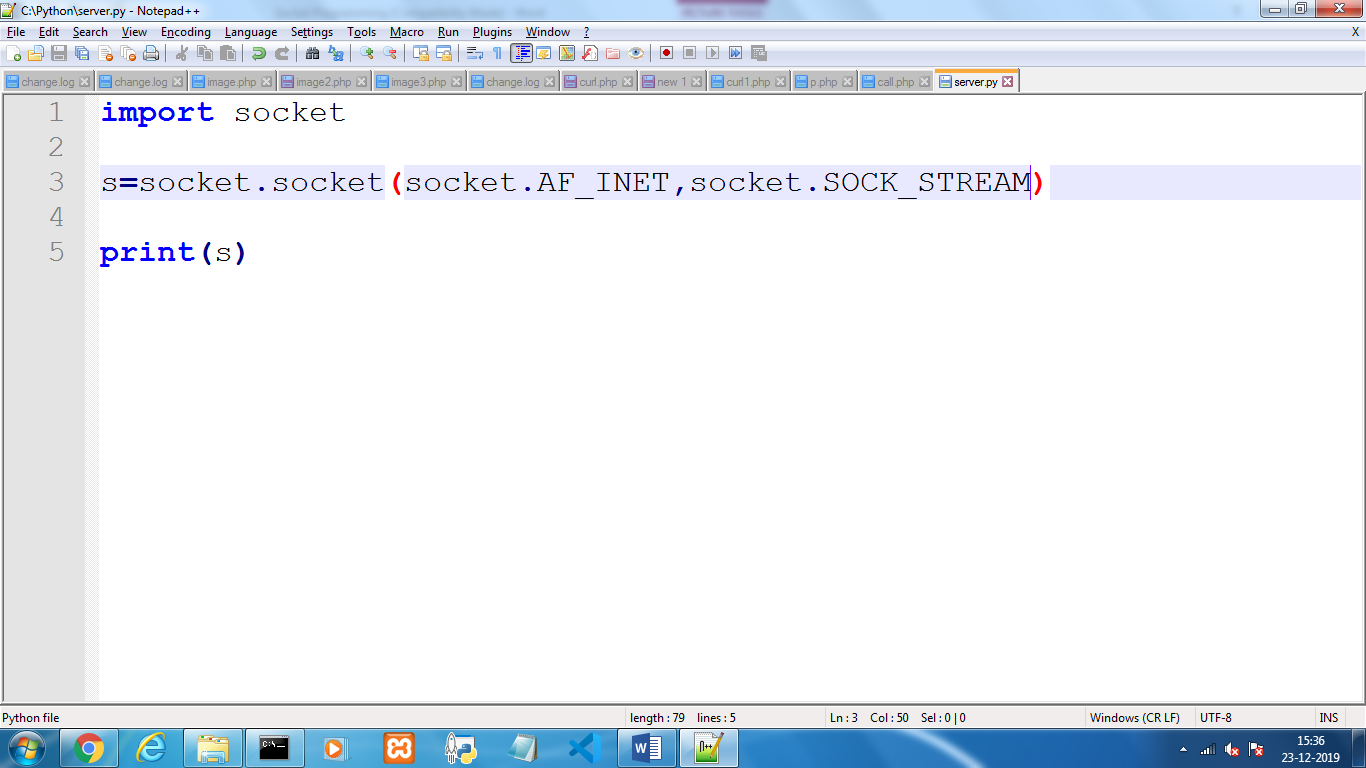
recv(n) n= number of bytes e.g n=1024

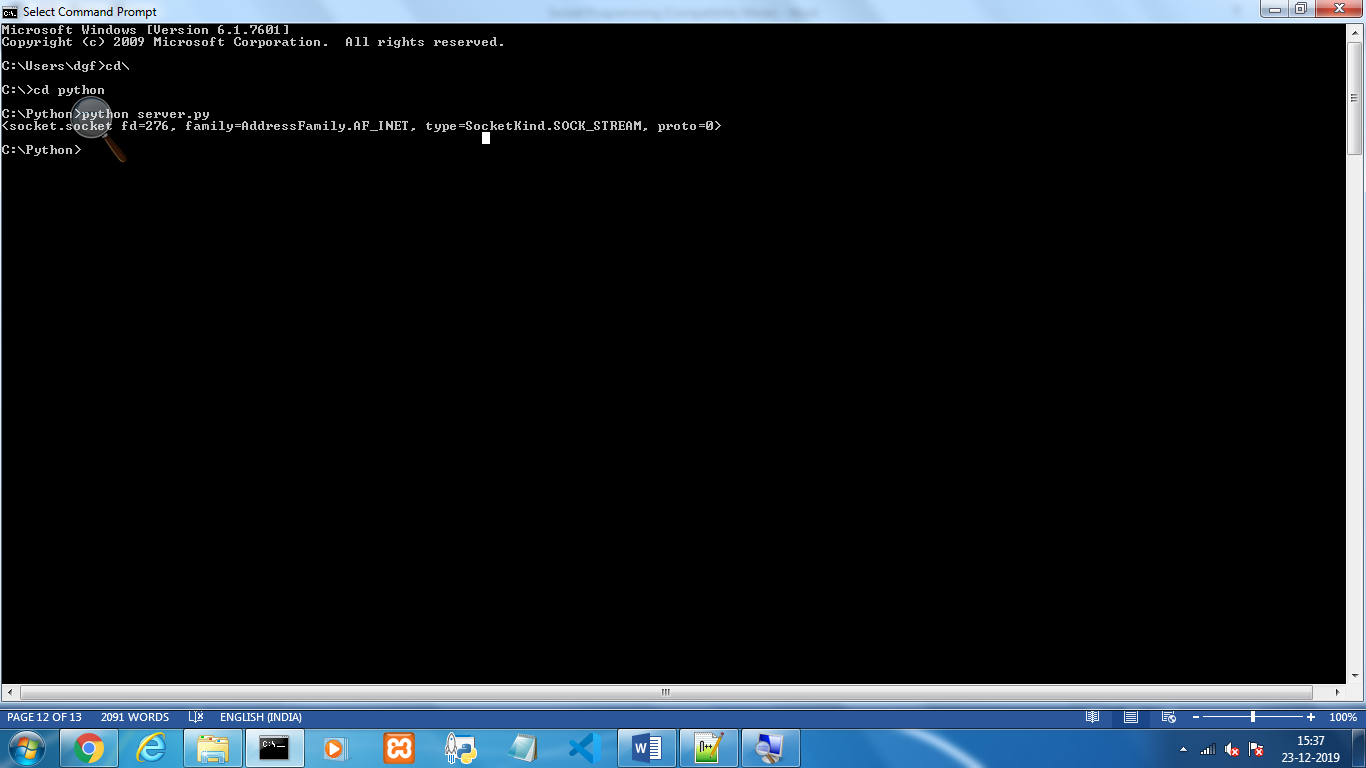
**CHATING APPLICATION**

1. Create two files a) server.py b) client.py

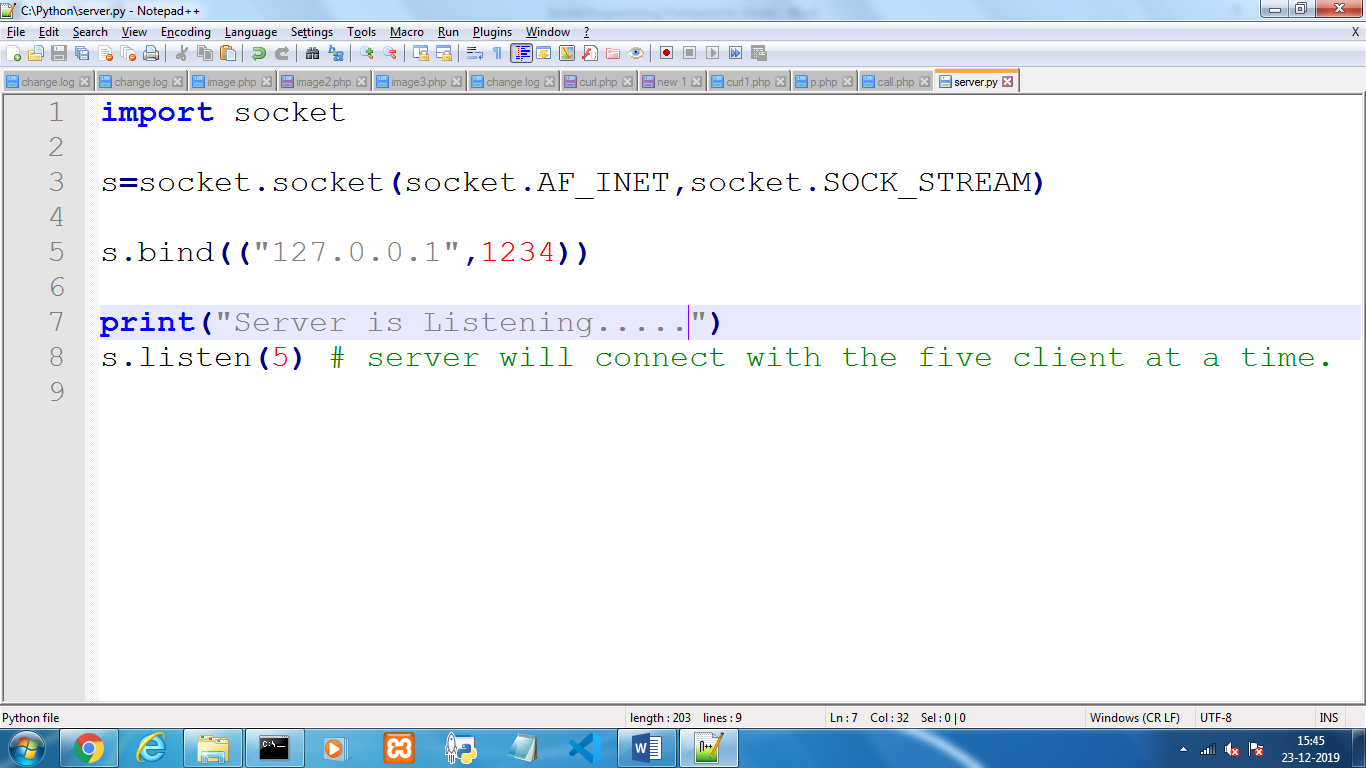
**Server.py**

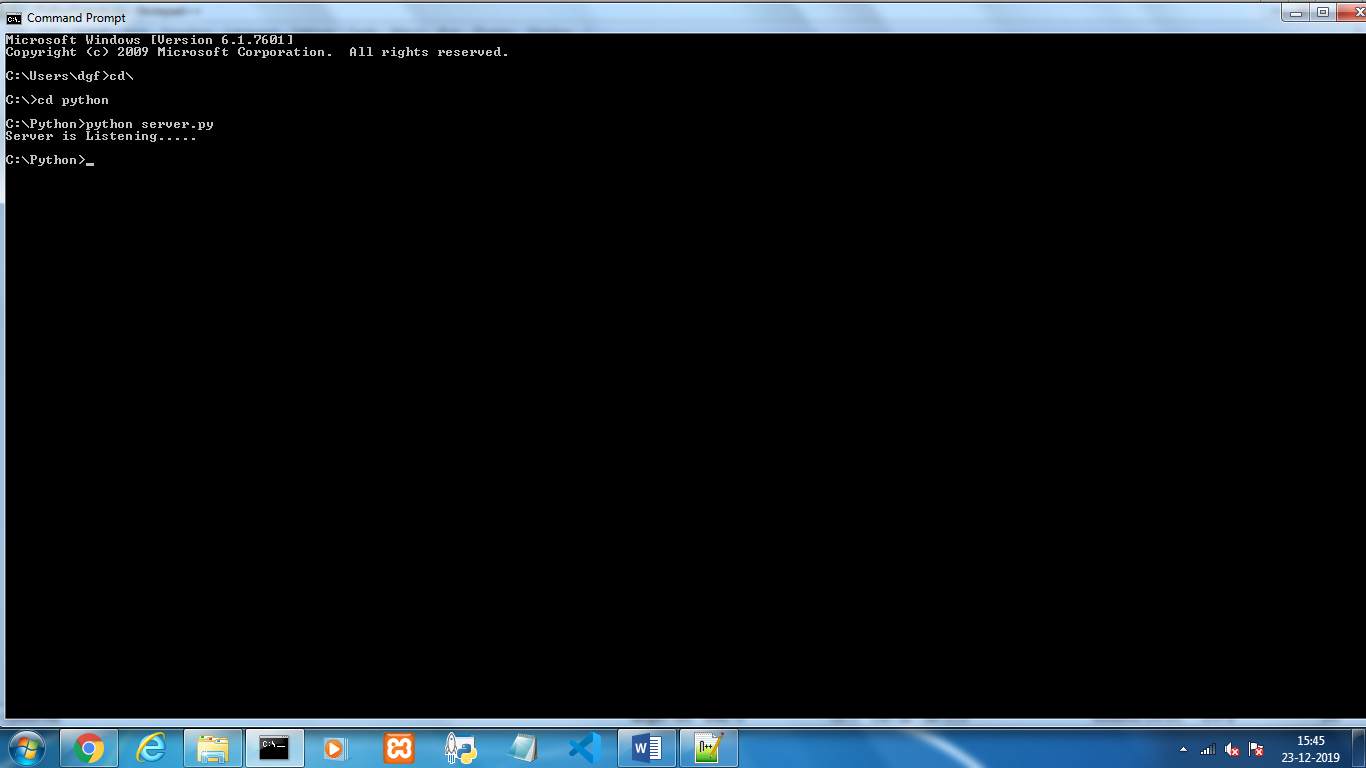
1. Import socket module.
2. Create a socket object.





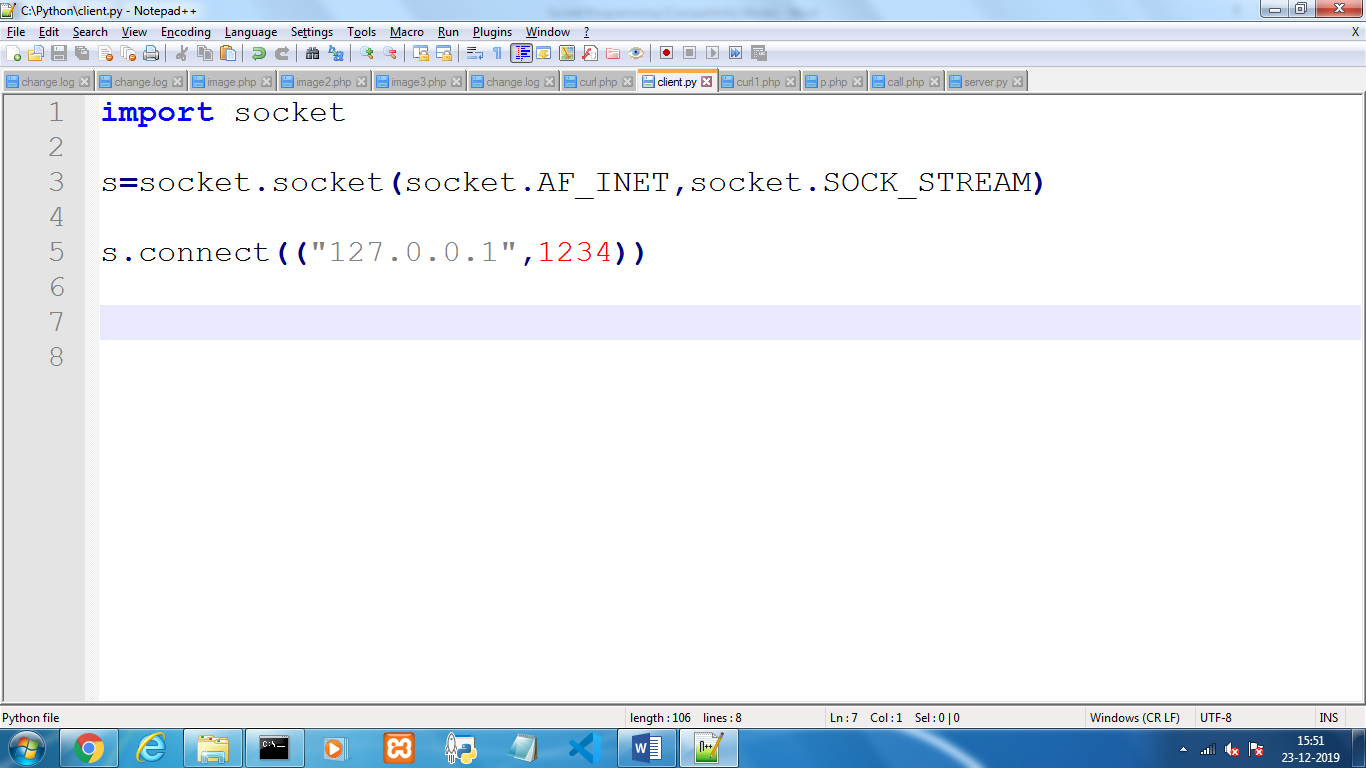
1. Create socket by binding IP address and Port number.
2. Now the server socket is ready, start listening for the connections from the Client.





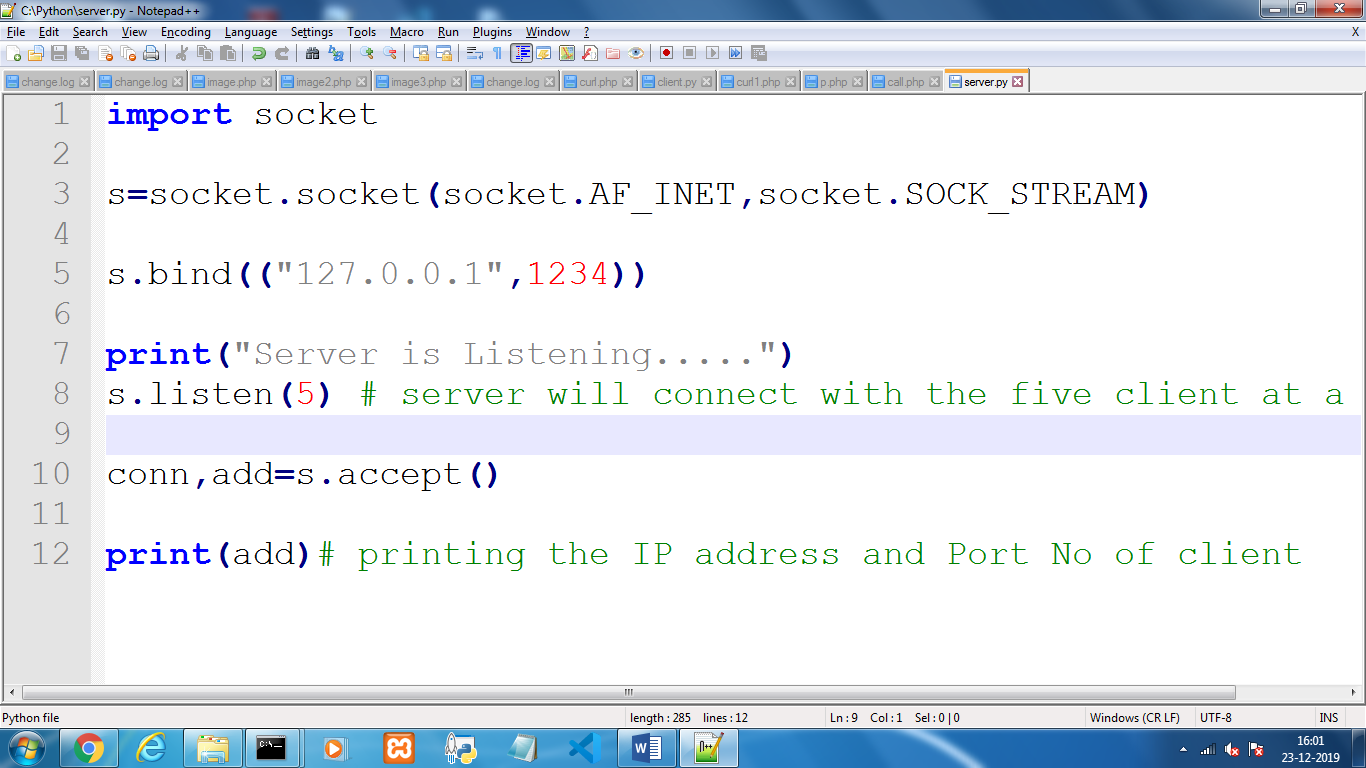
**Client.py**

1. Import socket module.
2. Create a socket object.
3. Connect with the server with the help of connect() method.

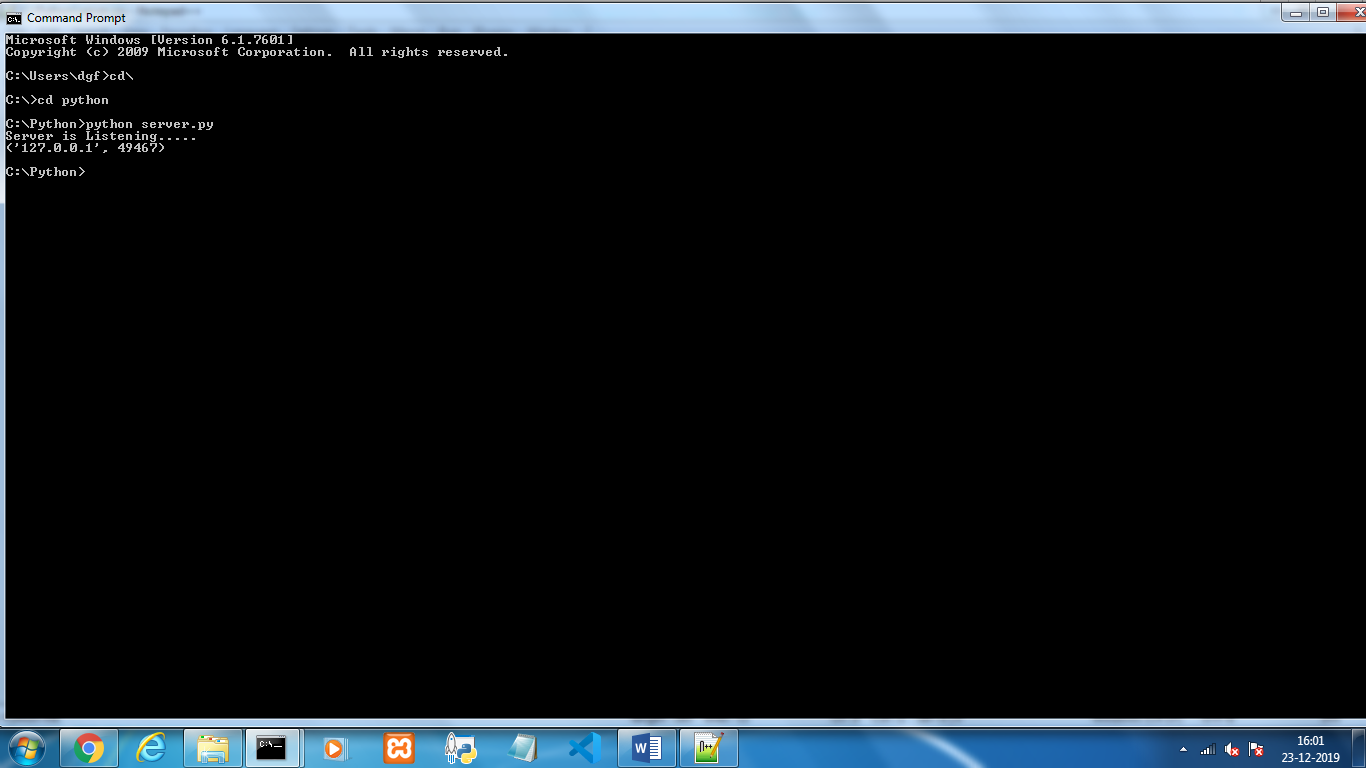


1. When you try to connect with the server with the connect () method, sever accept the connection by using accept () method.

**Server.py**

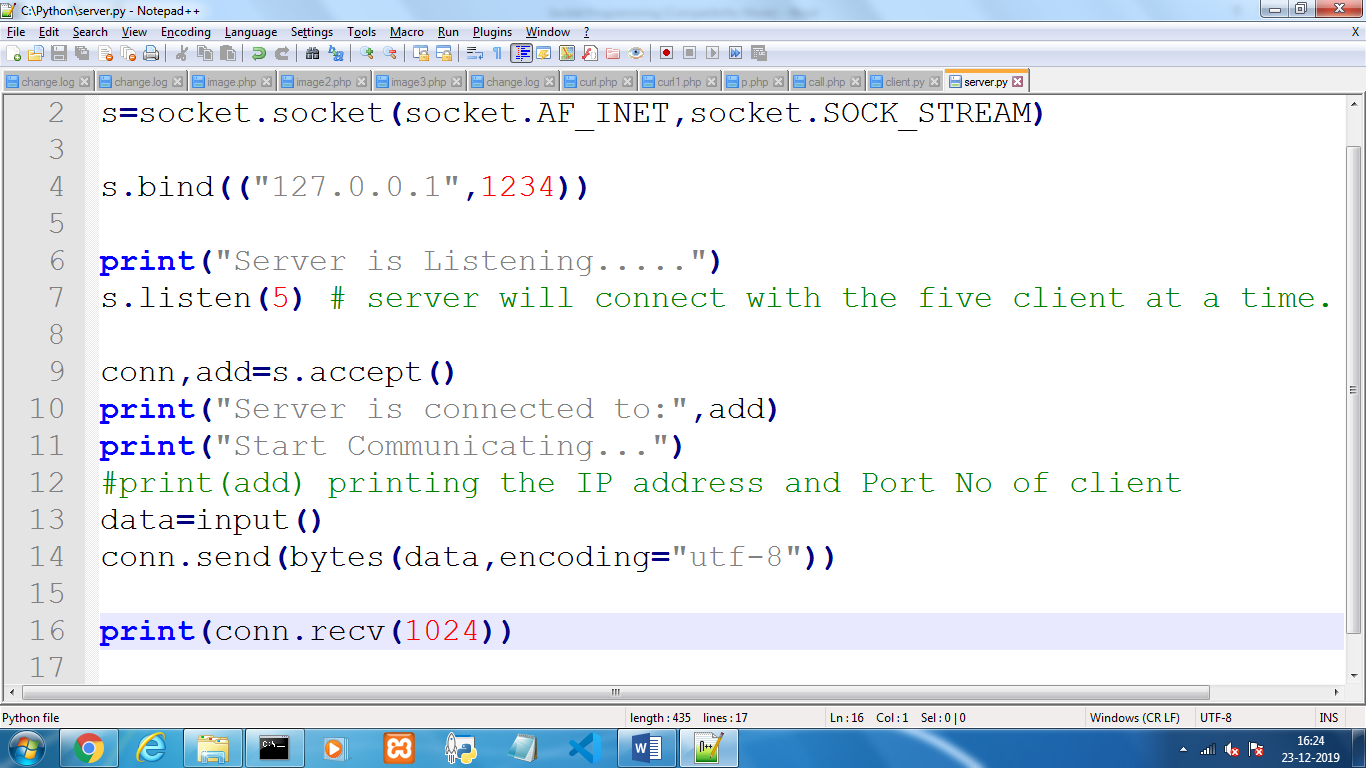


accept() method after accepting the connection returns the two object: a) connection object (conn) require for communicating with that particular client. b) IP address and Port number of the client. [Try to print the add variable]



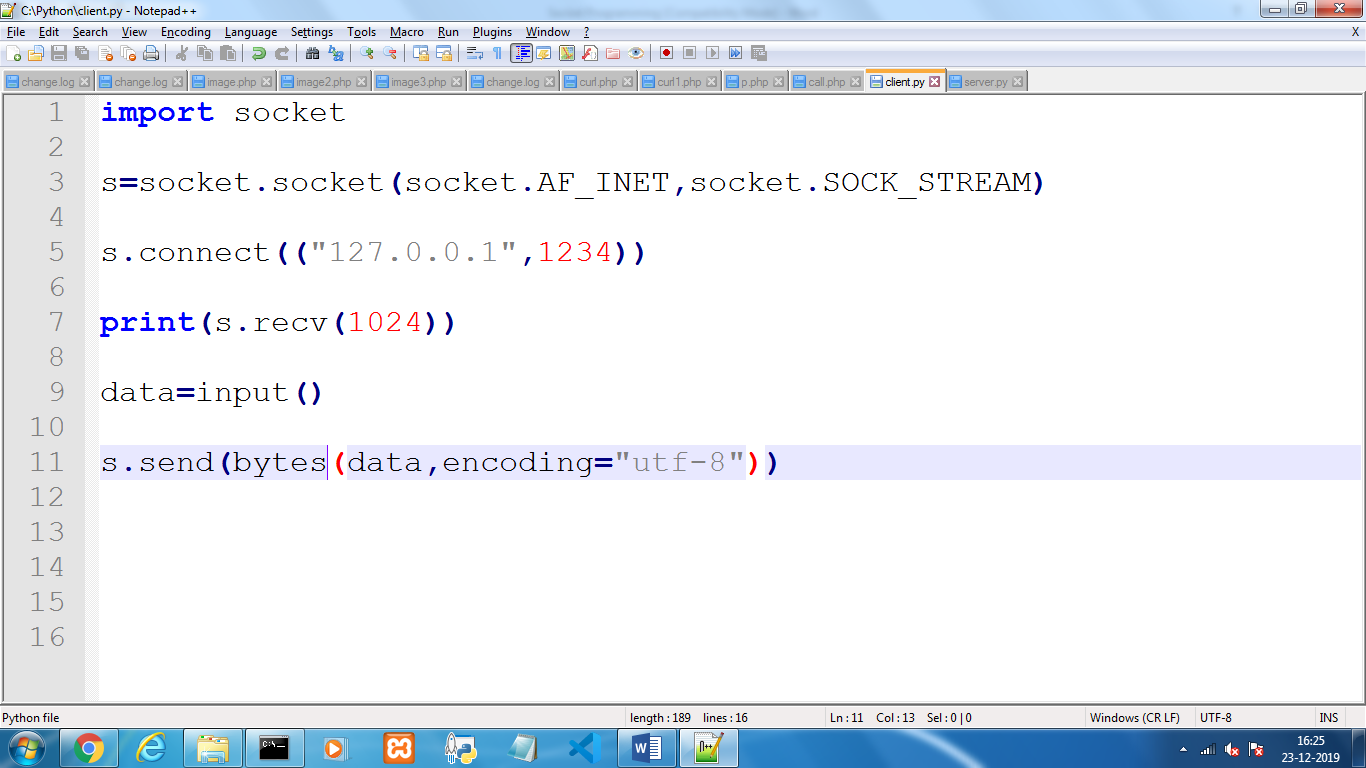
Since server and the client are the localhost so both will have same IP address, but with the different port number: 49467

1. Send message to the server by using send() method.
2. Receive message from the client with the help of recv() method.



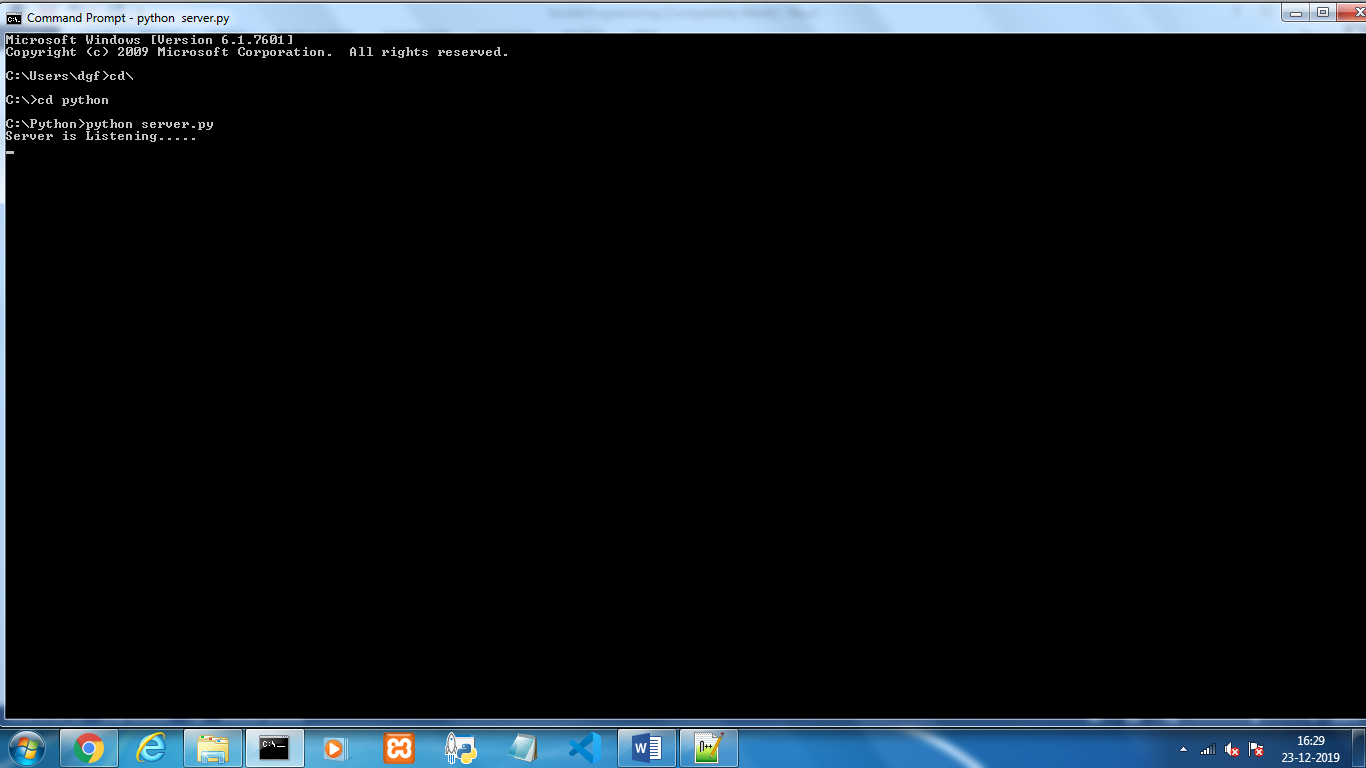
**Client.py**

1. Receive message send by the server with recv() method.
2. Send message to the server with send() method.

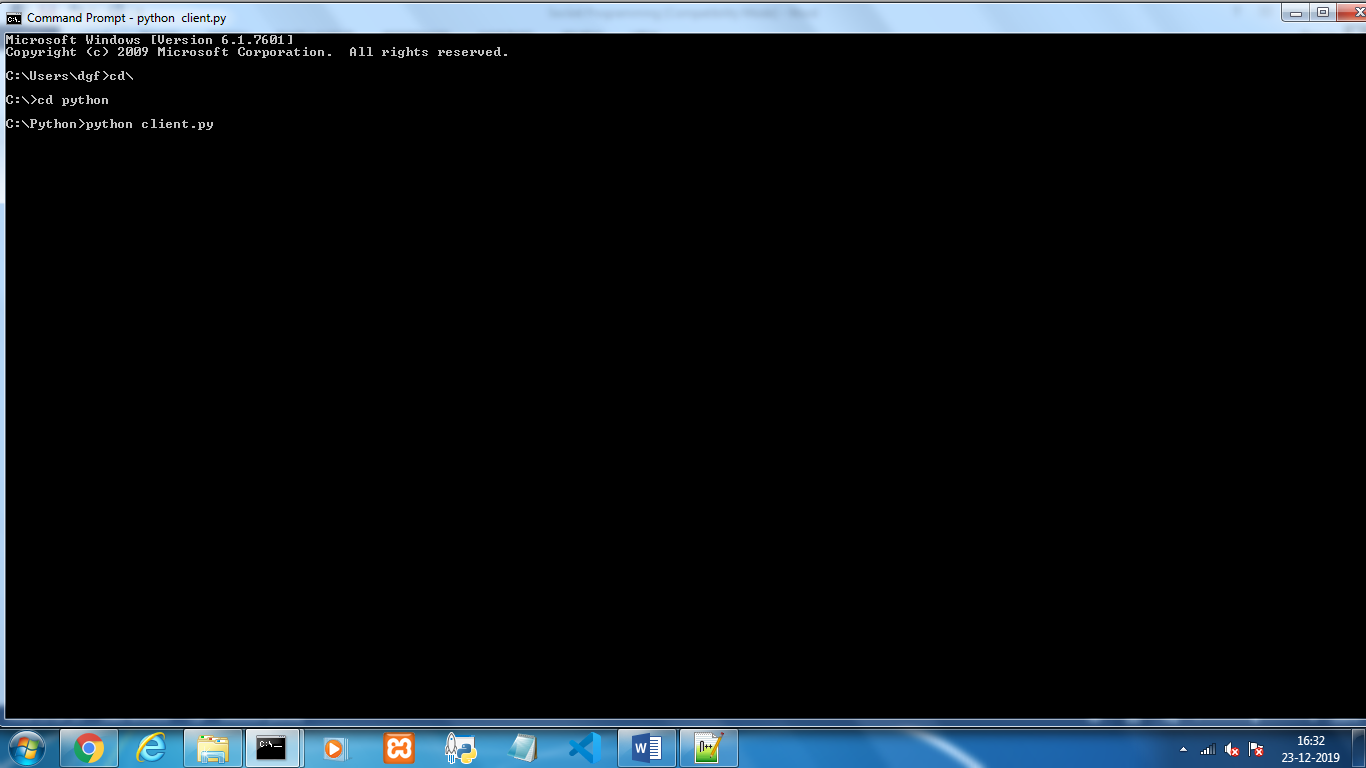


Final Output:

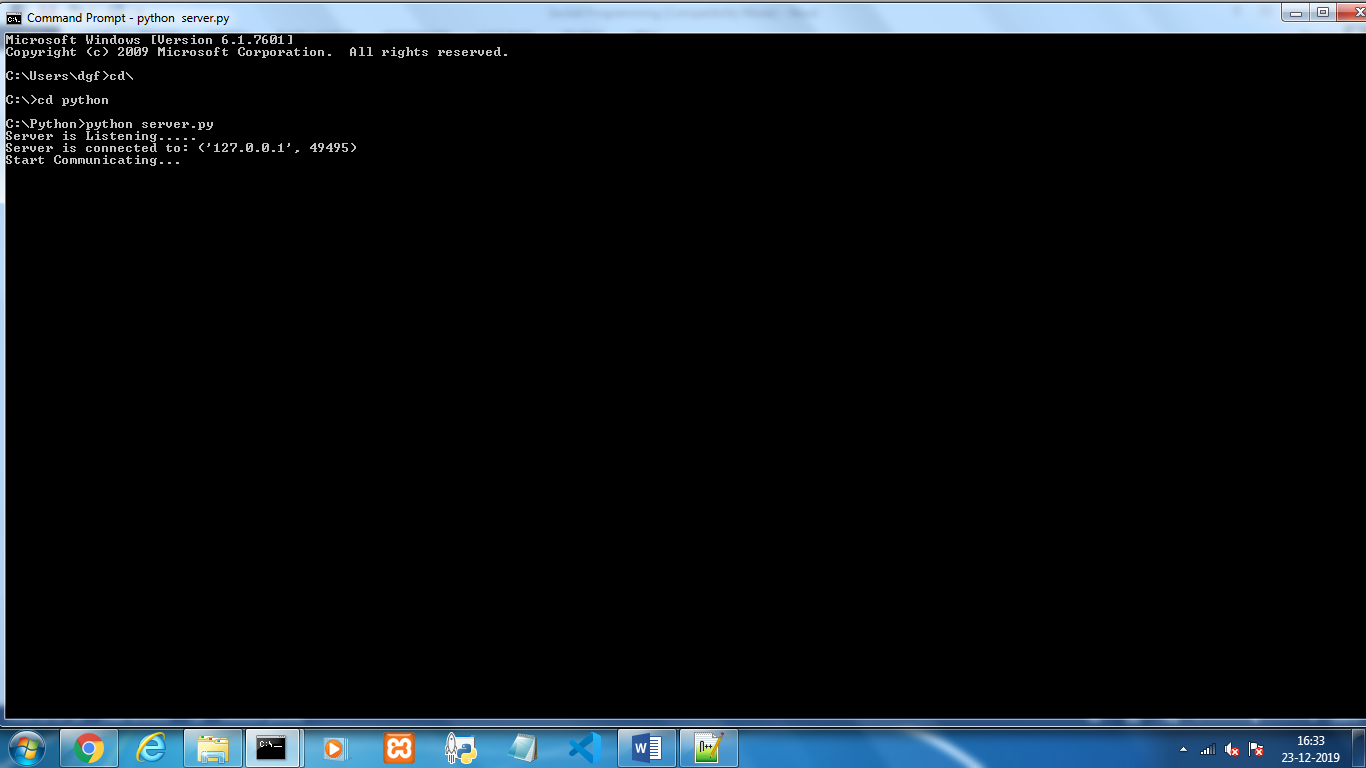
1. Start Executing server.py script



1. Start Executing client.py on another cmd Prompt



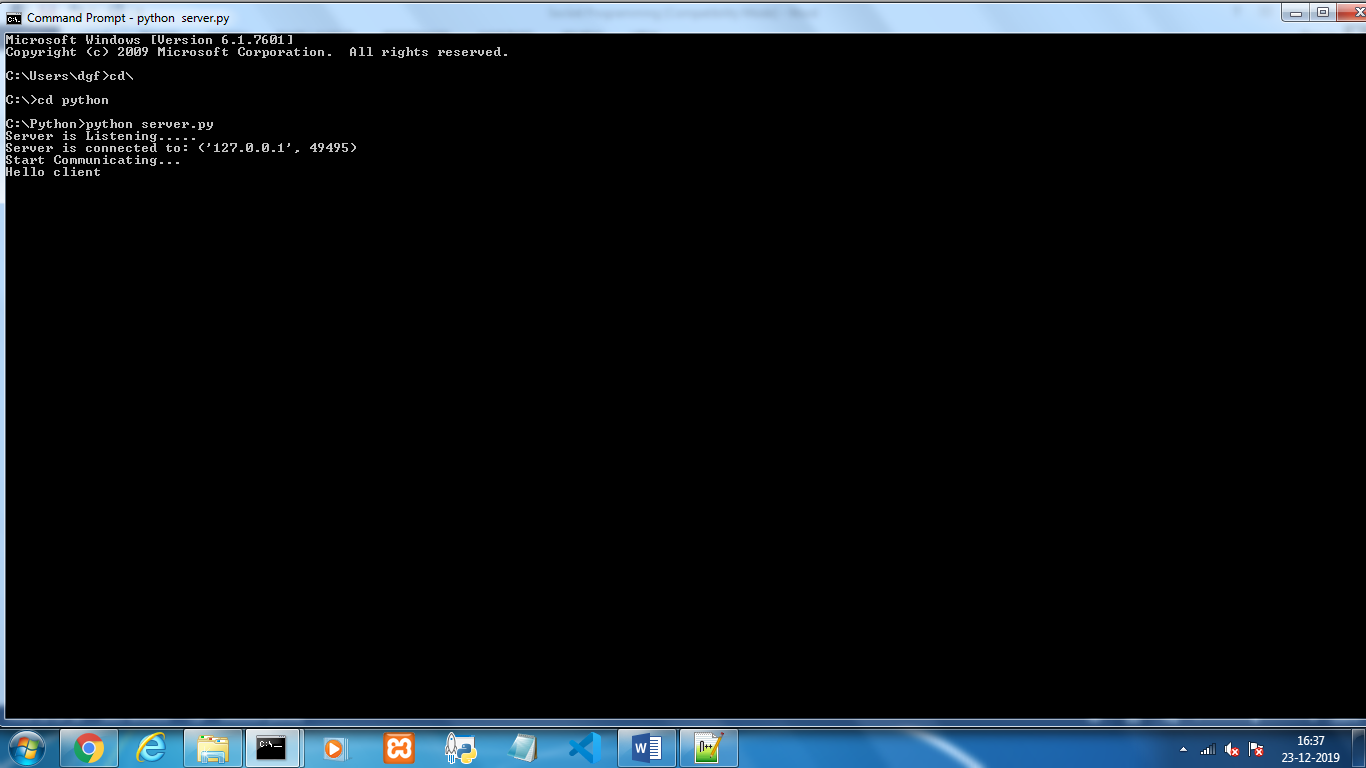
1. Now client is also connected to the server, see the server side cmd prompt.



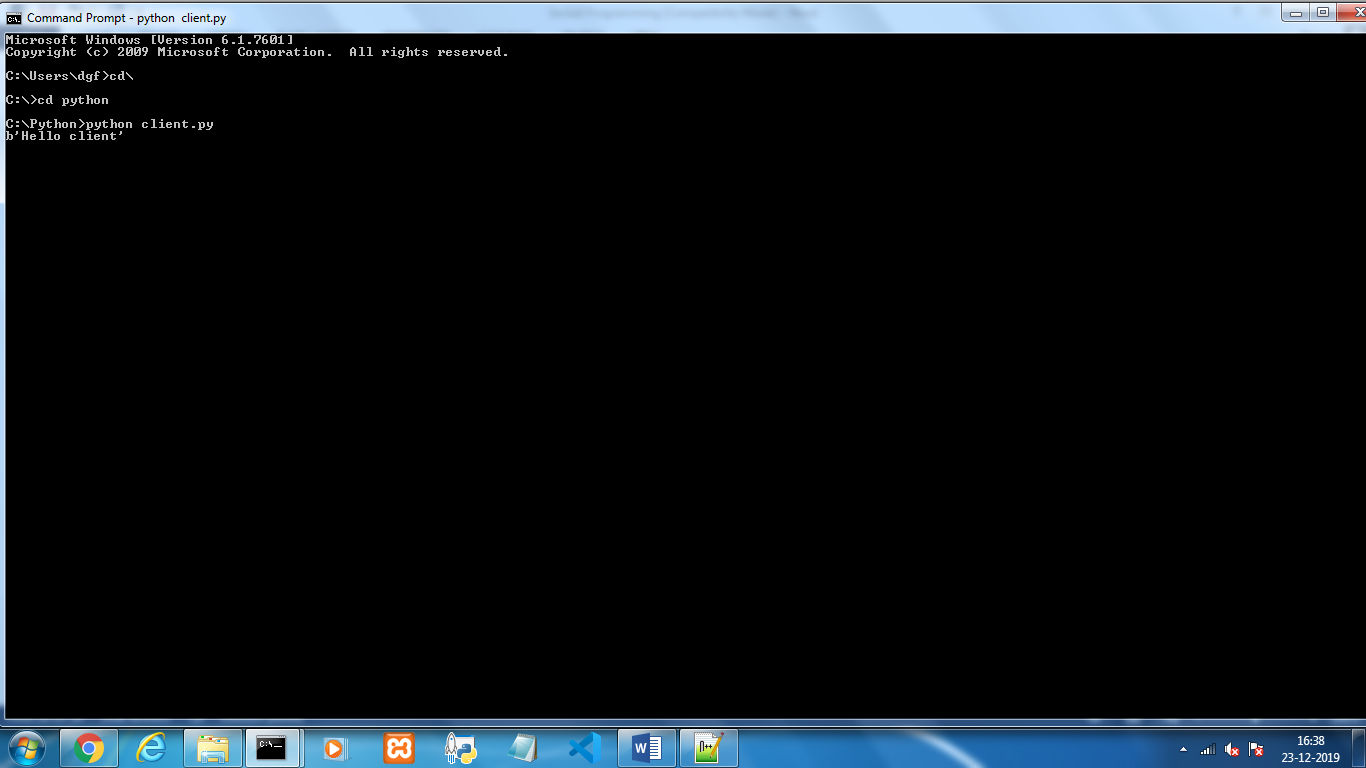
1. You can see message that server is connected to the Client on the port 49495.

*Note: Port number is a randomly selected number so it may vary during every execution*

1. Send message to the client.

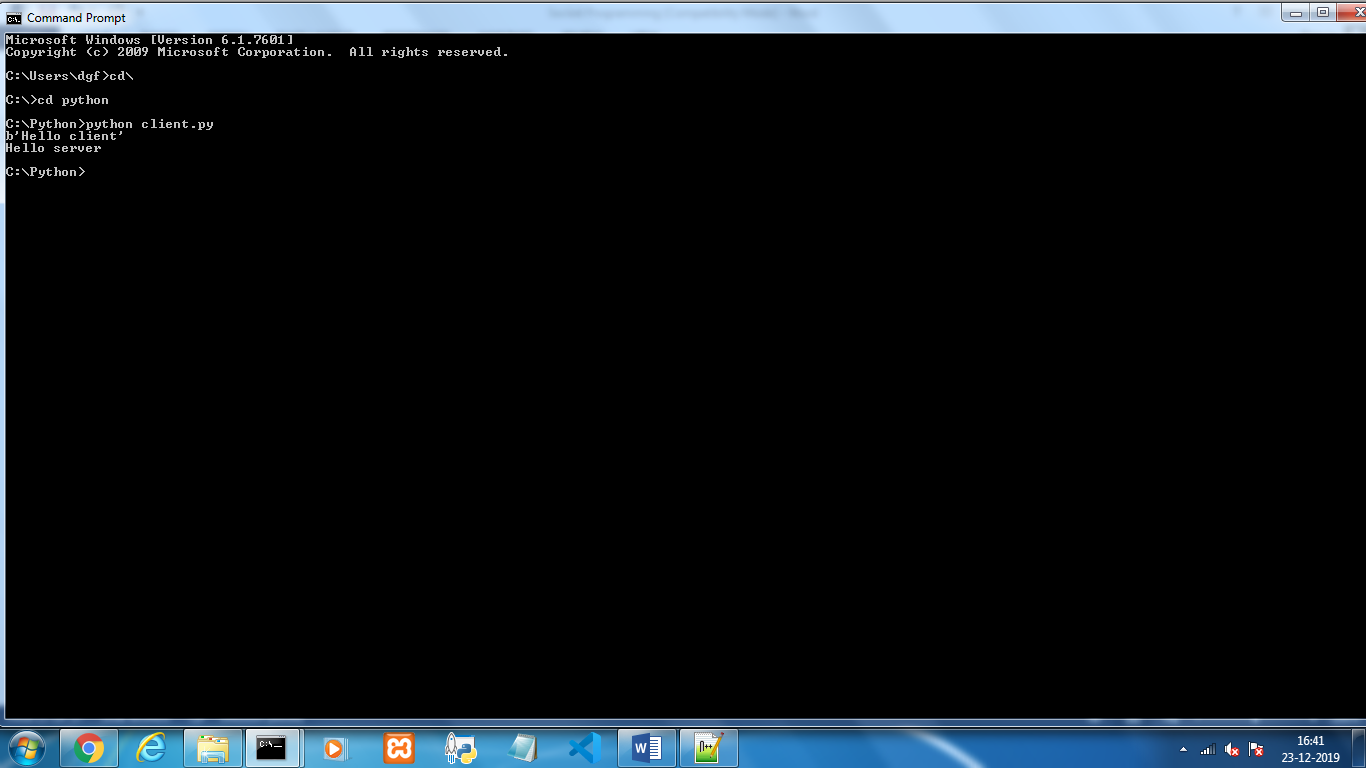


1. On client cmd prompt you get this message.

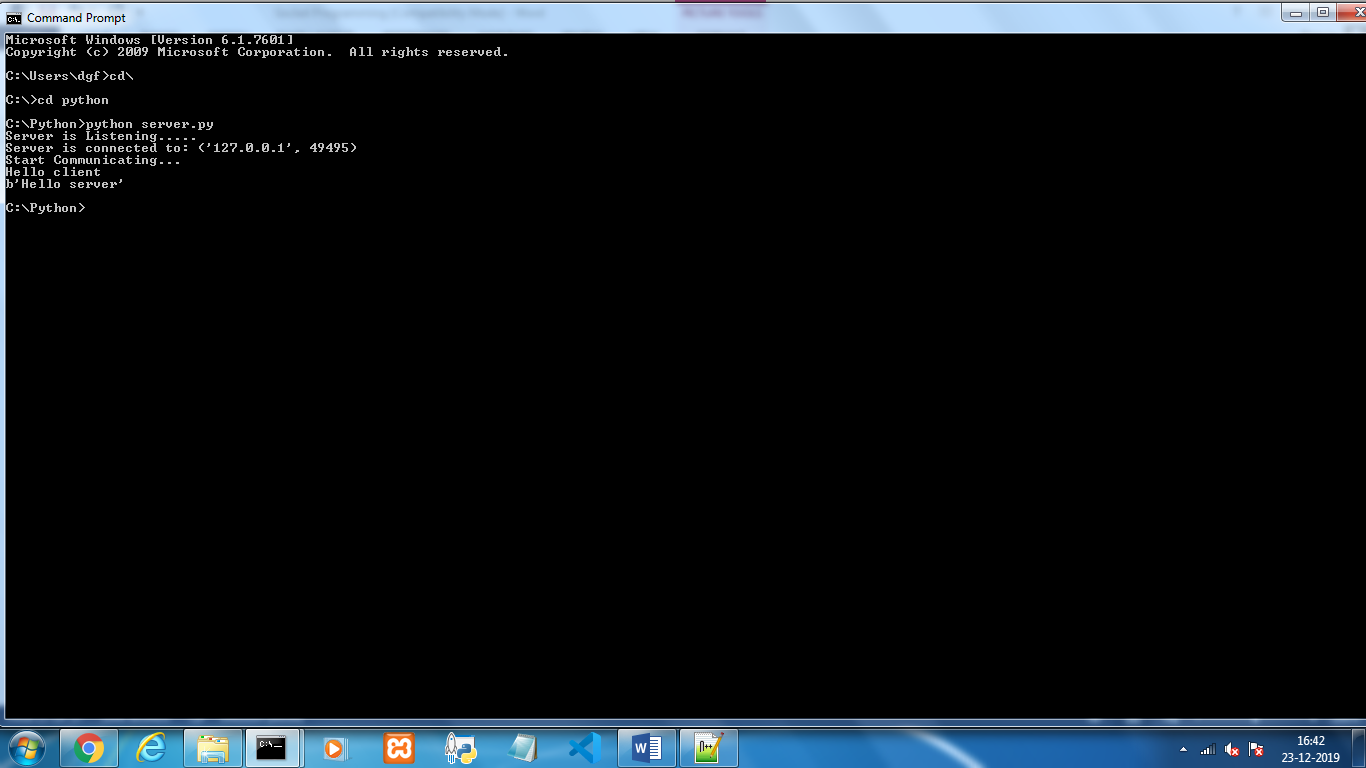


1. Type a message on client cmd prompt and send to server.

Client cmd Prompt

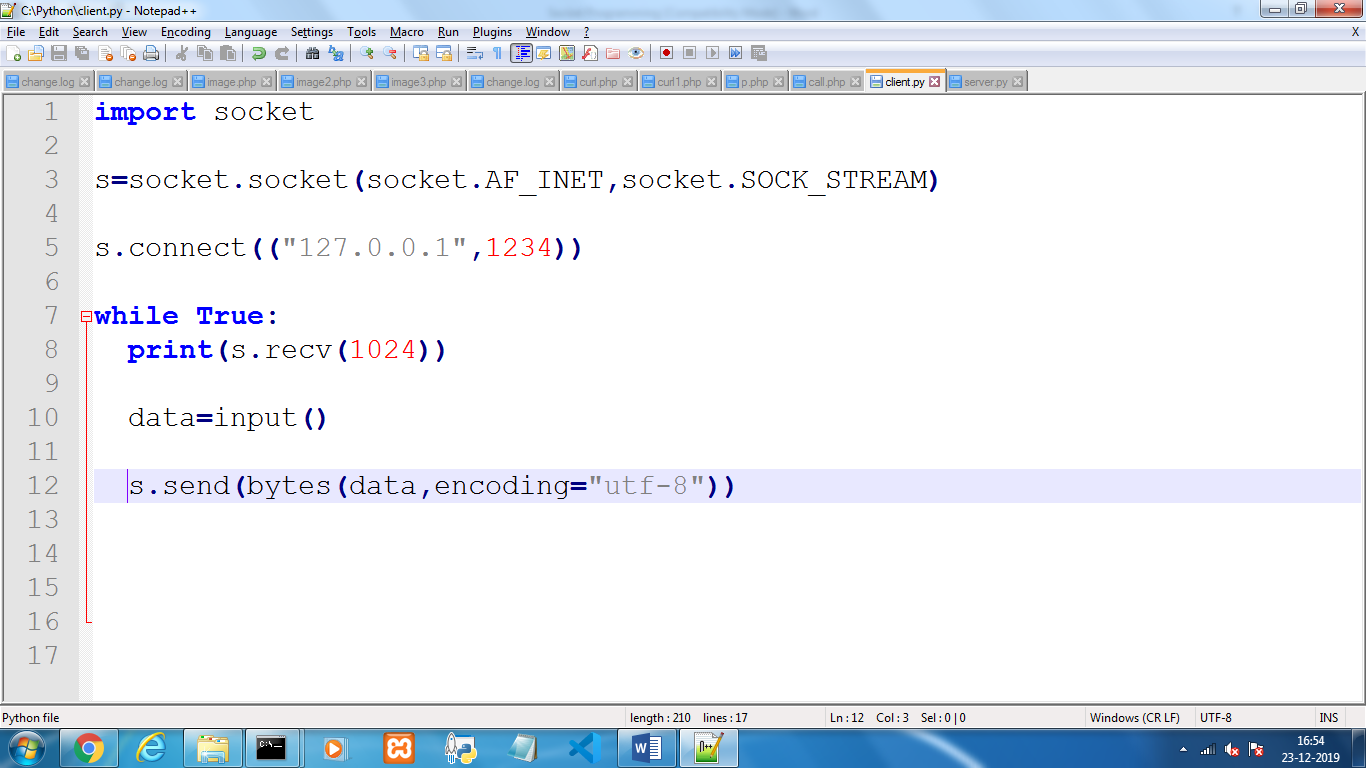


Server cmd Prompt



For continues chatting, jus enclosed your chatting logic inside infinite while or for loop and enjoy the chatting…

**Client.py**



**Server.py**

