Python les-materialen

# Network Programming: Features, Internet Modules & Networking Terminologies

In this article, we will learn about the essence of network programming using Python. For learning Python network programming, one must know about the following topics:

* Data encoding
* High-level client modules
* HTTP and web programming
* Programming using sockets
* Basic networking terms

There are two levels of network services in Python. These are:

* High-level access
* Low-level access

In the low-level access, we can use and access the socket support for the operating systems by using Python libraries. Programmers can also implement connection-less and connection-oriented protocols for performing network programming.

Programmers can access the application-level network protocols by using high-level access services. The standard library of Python has full support for protocols, encoding, and networking concepts.

## Socket

A ***socket*** is defined as an end-point in the flow of communication between any two programs or channels. The sockets are created by using a set of requests in programming, also called socket API (Application Programming Interface).

These sockets use various protocols for determining a connection for port-to-port communication. The main uses of protocols are:

IP addressing E-Mail FTP (File transfer protocol) Domain Name servers

**Domain**: It is a family of protocols that are used as the mechanism for transportation.

**Type**: It is the type of communication between two endpoints.

**Protocol**: It is used to identify a variant.

**Port**: It is a medium through which the server listens to the clients.

### A program for socket

Python has a socket method that sets up different sockets virtually. The syntax for the same is as follows:

s= socket.socket (socketFamily, type\_of\_the\_socket, protocol=value)

Explanation:

**socketFamily**: It is either AF\_UNIX or AF\_INET.

**type\_of\_the\_socket**: It is either SOCK\_STREAM or SOCK\_DGRAM.

**Protocol**: It is usually left out and defaulting to 0.

Methods to manage the connections:

* listen(): This method is used to establish and start TCP listeners.
* bind(): This method is used to bind-address to the socket.
* connect(): It is used to make a connection with the TCP server.
* accept(): It is used to make a TCP client connection.
* recv(): This method is used to receive messages.
* close(): It is used to close a socket.
* sendto(): This method is used to send UDP messages.
* Send(): This method is used to send messages.

## Networking Terminologies

**Let us quickly discuss the basic terms of networking:**

**Internet protocol:**Internet protocol is a set of rules and procedures to govern the flow of data. It has two significant protocols:

### User Datagram Protocol (UDP)

**The User Datagram Protocol is a connectionless protocol. Some properties of UDP are:**

* **Unreliable**: Whenever a User Datagram Protocol message is sent, we don’t have a way to know whether it has reached its destination or not. In the User Datagram protocol, there is no way of acknowledgment.
* **Not ordered**: We cannot predict the order of messages in which they arrive.

### Transmission control protocol (TCP)

TCP uses the concept of a handshake. In simple words, it is a way to ensure that the connection has been established between hosts, and now the data transfer can be started. TCP protocol requires us to build a network in the first place. Some properties of Transmission Control Protocol are:

* **Reliable**: Transmission Control Protocol manages the acknowledgment and timeout of the message. It makes several attempts to deliver the messages. The server also requests the lost parts again to get the lost messages.
* **Heavy-weight**: Transmission Control Protocol has three packets to set up a connection for the socket. These packets are:
* SYN
* SYN+ACK
* ACK

## Server and client application example

### A Simple Server

To write Internet servers, we use the socket function available in socket module to create a socket object. A socket object is then used to call other functions to setup a socket server.

Now call bind(hostname, port) function to specify a port for your service on the given host.

Next, call the accept method of the returned object. This method waits until a client connects to the port you specified, and then returns a connection object that represents the connection to that client.

#!/usr/bin/python # This is server.py file  
  
import socket # Import socket module  
  
s = socket.socket() # Create a socket object  
host = socket.gethostname() # Get local machine name  
port = 12345 # Reserve a port for your service.  
s.bind((host, port)) # Bind to the port  
  
s.listen(5) # Now wait for client connection.  
while True:  
 c, addr = s.accept() # Establish connection with client.  
 print  
 'Got connection from', addr  
 c.send('Thank you for connecting')  
 c.close() # Close the connection

### A Simple Client

Let us write a very simple client program which opens a connection to a given port 12345 and given host. This is very simple to create a socket client using Python’s socket module function.

The socket.connect(host\_name, port ) opens a TCP connection to hostname on the port. Once you have a socket open, you can read from it like any IO object. When done, remember to close it, as you would close a file.

The following code is a very simple client that connects to a given host and port, reads any available data from the socket, and then exits.

#!/usr/bin/python # This is client.py file  
  
import socket # Import socket module  
  
s = socket.socket() # Create a socket object  
host = socket.gethostname() # Get local machine name  
port = 12345 # Reserve a port for your service.  
  
s.connect((host, port))  
print  
s.recv(1024)  
s.close() # Close the socket when done

Now run this server.py in background and then run above client.py to see the result.

# Following would start a server in background.

$ python server.py

# Once server is started run client as follows:

$ python client.py

This would produce following result:

Got connection from (‘127.0.0.1’, 48437) Thank you for connecting

## Python Internet Modules

Protocol Name

The Function of the protocol

Port No.

Python Module associated

Gopher

Transfer of documents

70

Gopherlib, urllib

Telnet

Used for Command line

23

telnetlib

IMAP4

Used for fetching emails

143

impalib

POP3

Used for fetching emails

110

poplib

SMTP

Used for sending emails

25

smtlib

FTP

Used for file transfers

20

Ftblib, urllib

MNTP

Usenet news

119

mntplib

HTTP

Used for web pages

80

Httplib, urllib

## Wrapping up

These are the fundamental properties of Python as a networking language. Thus, Python is a general-purpose programming language. It is object-oriented and interactive. It uses English keywords frequently, which makes it easy to understand.

### How is Python utilized in networking?

Learning the usage of Python in networking is necessary for all the upcoming network engineers to build an excellent career in this field. The main use of Python is to build different scripts that can automate specific complex network configurations. Complete support to the network protocols is provided by the standard library of Python. Python is much more useful than other languages in networking because of the code simplicity. Task automation for all the complex tasks is made easy with the help of python programming. This is how Python is utilized in networking.

### What is meant by Python Network Programming?

The process of writing programs that could be used to communicate across the network with all the other programs is called Network Programming. In Python Network Programming, Python is used as the programming language for handling all the computer networking requirements. For instance, if you wish to create and run any local web server or directly download some files in your system from a URL, you can make use of Python Network Programming. Using Python, the entire network programming tasks become easy and simple. There are plenty of Python libraries to simplify the tasks for the programmers and software developers. For getting into python network programming, you need to be clear with the basics of writing codes in the python language. Once you have sound knowledge about the language, you can build a great career in this field.

### How is Python connected to the internet?

The python module named urllib is useful for connecting and opening URLs from the internet. Every URL action can be performed with the help of this library. You can even retrieve different forms of data from the internet with the help of Python by using the urllib library.Import urllib in the program