Socket Programming

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

Sockets are the endpoints of a bidirectional communications channel. Sockets may communicate within a process, between processes on the same machine, or between processes on different continents.

Sockets may be implemented over a number of different channel types: Unix domain sockets, TCP, UDP, and so on. Socket programming is started by importing the socket library and making a simple socket.

```
s = socket.socket (socket_family, socket_type, protocol=0)
```

- **socket_family**: This is either AF_UNIX or AF_INET.
- **socket_type**: Defines the types of communication between the two end-points. It can have the following values.
    - SOCK_STREAM (for connection-oriented protocols e.g. TCP), or
    - SOCK_DGRAM (for connectionless protocols e.g. UDP).
- **Protocol**: leave this field or set this field to zero.

Server Socket Method

- **s.bind():**This method binds the socket to address (hostname, port number pair).
- **s.listen():**This method is used to listen to the connections associated with the socket.
- **s.accept():**This accept TCP client connection, waiting until connection arrives (blocking).

Client Socket Method

- **s.connect():**This method is used to connect the client to host and port and initiate the connection towards the server.

General Socket Methods

- **obj.recv():** Use this method to receive messages at endpoints when the value of the protocol parameter is TCP.
- **obj.send():** Apply this method to send messages from endpoints in case the protocol is TCP.
- **obj.recvfrom():** Call this method to receive messages at endpoints if the protocol used is UDP.
- **obj.sendto():** Invoke this method to send messages from endpoints if the protocol parameter is UDP.
- **obj.gethostname():** This method returns the hostname.
- **obj.close():** This method is used to close the socket. The remote endpoint will not receive data from this side.

Simple Server

```
import socket
s = socket.socket()
host = socket.gethostname()
port = 12345
s.bind((host, port))
s.listen(5)
while True:
   con, addr = s.accept()
   print 'Connecting with', addr
   con.send('Connection is successful')
   con.close()
```

Simple Client

```
import socket
s = socket.socket()
host = socket.gethostname()
port = 12345
s.connect((host, port))
print s.recv(1024)
s.close()
```