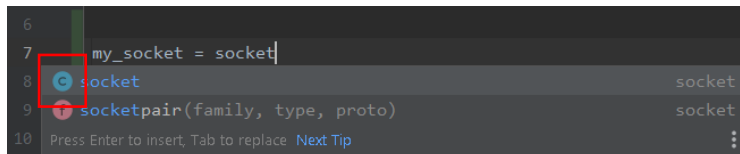


[Session 2] Network Programming Supplement

socket library

socket library provides several functions, constants, and classes for socket programming. In the pre-class lecture, we created socket with **socket()** function. Actually, the socket is an object of **class**, "socket" so in fact, it is an initializer of socket class.



```

6
7 my_socket = socket
8 socket
9 socketpair(family, type, proto)
10 Press Enter to insert, Tab to replace Next Tip

```

In this initializer, we usually pass two arguments to the method.

- **Address family:** Address family (IP address system), We can use IPv4(e.g. 172.168.10.24), or IPv6. AF_INET stands for IPv4, and AF_INET6 stands for IPv6
- **Type:** Data transmission type, we can use TCP(SOCK_STREAM) or UDP(SOCK_DGRAM).

After creating socket, sometimes we want to get or set our socket's property(option). So we use **getsockopt()** and **setsockopt()** method. (Usually we use SOL_SOCKET as its level)

- **getsockopt(level, name):** Get the value of option, "name"
- **setsockopt(level, name, value):** Set the option, "name" to "value"

In our class, I used **setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)**, this enables (True, 1) we can reuse the address.

bind(address) method binds the socket to "address", whose definition depends on address family. In our case (IPv4), address can be represented as 2-tuple, (host, port). We must bind only once for a socket. **host** is represented as a string of IP address (like "127.123.123.1"), but we can leave it blank ("") if we want to bind for all interface. Note that, usually **port numbers between 1 - 1023 are reserved for specific use**. We call these "Well-known port". So, **don't use these numbers** if possible.

listen(num) method makes the socket listen (be ready for accept). The number is called "backlog" and determines the length of queue for waiting client. If skipped, the number is set to some default value.

accept() method accepts the client who requested to connect, with **connect()**. It returns 2-tuple, (c_socket, address). c_socket is a new socket for the connection between the client and server. Until the connection is established, this method does not return: it means that, the program waits.

connect(address) method sends a request to server, in order to establish a connection. Like **bind()** method, the address is a 2-tuple with IP address and port number, in IPv4.

After connection is established, client still uses the socket we created, but server should use new socket, created by **accept()** method. (Why?) Anyway, we can send or receive data to / from each other.

- **send(data):** send data via the socket. Encode(e.g. **encode()** method) or some equivalent is needed, for sending data
- **recieve(bufsize):** Receive data via the socket. Program waits until some data is arrived (even if it was empty data), it reads at most (bufsize) bytes.