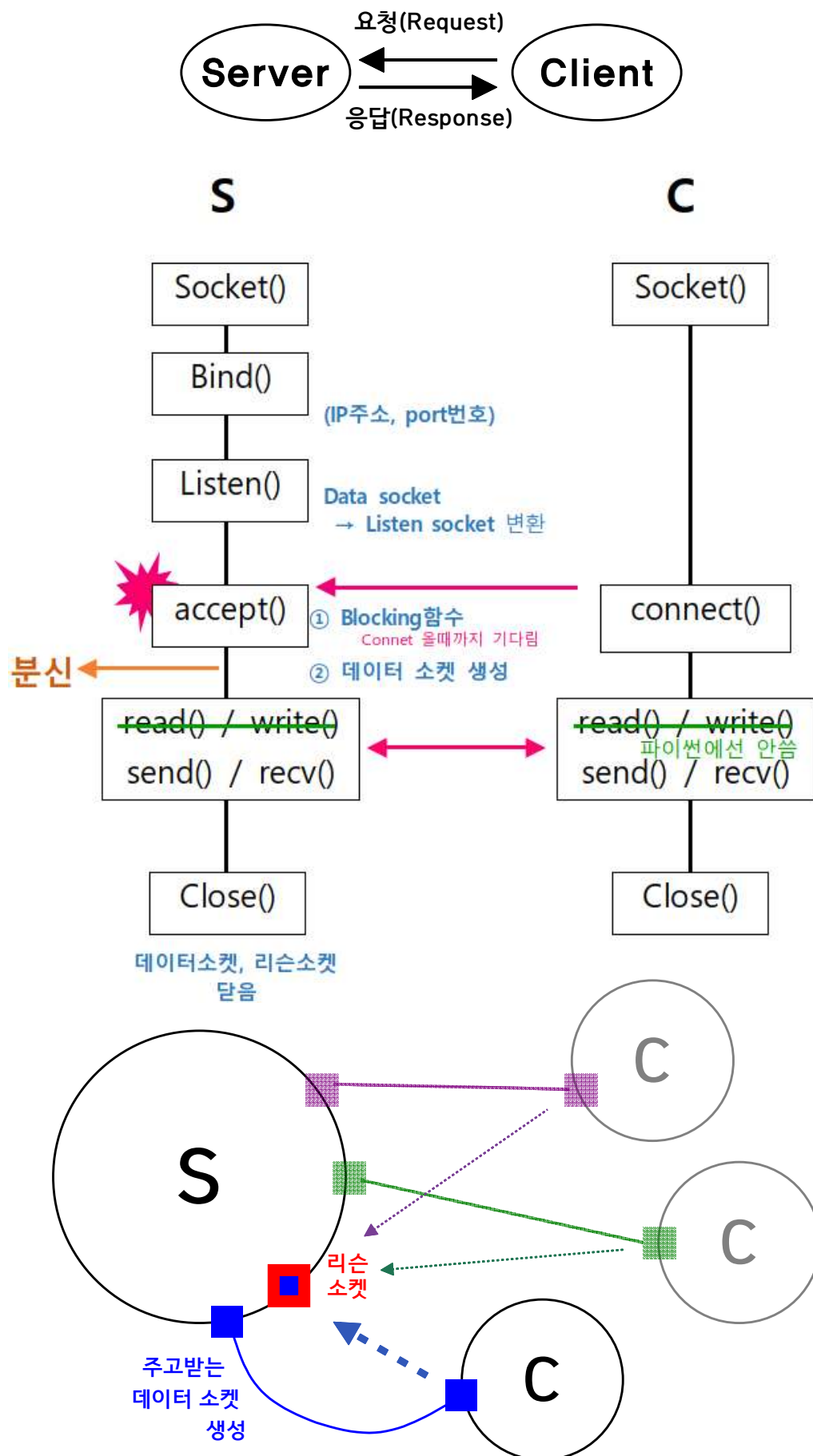


## TCP/IP socket통신



리슨소켓 - 다중 클라이언트 연결하기 위함

## 파이참 다운로드 & 설치

<https://www.jetbrains.com/ko-kr/pycharm/download/download-thanks.html?platform=windows&code=PCC>  
Community버전 설치

### ( 작업장소 )

D:\PyCharmWorkspace

### ( 파이썬 인터프리터 추가 )

File - Settings - Project:PycharmWorkspace

Python Interpreter - Show All - [+] - ok

### <교재>

- IoT 사물인터넷을 위한 **파이썬 네트워크 프로그래밍** (제 2 판)
- cmd창에서 Time\_server.py 가 무한루프돌면  
**Ctrl+C**를 눌러서 중단-서버종료 시킴

## [ Time\_server.py ] - p.193

```
import socket
import time

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# 네트워크 패밀리/ tcp-sockstream , ip-,,,
# tcp소켓-> af_inet, sock_stream
address = ("", 5000) # 서버ip = "" - NULL
sock.bind(address)
sock.listen(5) # backlog -> 동시접속자가 많으면 크기를 많이줌..

while True:
    data_socket, client_addr = sock.accept()
    print("connection requested from", client_addr)
    data_socket.send(time.ctime(time.time()).encode())
    # ctime-문자열로 바꾸어줌, encode-bytestream으로 바꿔줘야함
    data_socket.close()
    # 접속하면 현재시간을 알려주는 서버 # 클라이언트 순차 처리
```

## [ Time\_client.py ] - p.194

```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # tcp socket

server_address = ("127.0.0.1", 5000) # ip주소, port번호- 튜플형식
sock.connect(server_address) # 127.0.0.1 - loop-back 주소- 자기자신주소(localhost)

msg = sock.recv(1024).decode() # decode - encode의 반대
print("recv:", msg)
sock.close()
```

## [ echo\_server.py ] - p.195

```
from socket import *

sock = socket(AF_INET, SOCK_STREAM)
sock.bind(("", 2500)) # 포트번호- 현재 프로그램 찾아가는 내선번호 같은거,,
                    # 0~ 1024 : reserved / 80 : web
                    # 2Byte 사용=> 0~65535 / 2000 이후 번호 사용 권장
sock.listen(5)

while True:
    print("waiting for clients...")
    data_socket, client_addr = sock.accept()
    print("connected by ", client_addr)

    msg = data_socket.recv(1024).decode() # buffer size

    if msg:
        print("recv : ", msg)
        data_socket.send(msg.encode())

    data_socket.close()
```

## [ echo\_client.py ] - p.196

```
import socket

port = 2500
address = ("127.0.0.1", port)
BUFSIZE = 1024

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(address)

msg = input("Msg to send : ")
sock.send(msg.encode())

recv_msg = sock.recv(BUFSIZE).decode()
print("echo msg : ", recv_msg)
sock.close()
```

[ id\_name\_server.py ] - p.203

```
from socket import *

sock = socket(AF_INET, SOCK_STREAM)
sock.bind(("", 5000))
sock.listen(5)
table = {'20150001': 'Hong',
         '20150002': 'Sim',
         '20150003': 'Park'}

while True:
    print("waiting for clients...")
    data_socket, client_addr = sock.accept()
    print("connected by ", client_addr)

    num_input = data_socket.recv(1024).decode()

    if num_input:
        print("recv : ", num_input)

        if num_input in table.keys():
            name = table[num_input]
            print("send_name : '", name, "' to ", client_addr)
        else:
            name = "Noname"

        data_socket.send(name.encode())

    data_socket.close()
```

[ id\_name\_client.py ] - p.204

```
import socket

port = 5000
address = ("127.0.0.1", port)
BUFSIZE = 1024

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(address)

num_input = input("Student_ID to send : ")
sock.send(num_input.encode())

recv_name = sock.recv(BUFSIZE).decode()
print("ID : ", num_input, " Name : ", recv_name)
sock.close()
```

```
class TCPServer:
    def __init__(self, port): # 생성자- 객체 초기화(자동호출)
        import socket
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.bind(("", port)) # IP, port
        self.sock.listen(5)

    def __del__(self): # 소멸자
        self.sock.close()

    def accept(self):
        data_socket, client_addr = self.sock.accept()
        return data_socket, client_addr

if __name__ == '__main__':
    # name-모듈이름/ main-프로그램 최상위 진입점(시작점) 영역이름
    # 현재파일이 최상위 진입점인지, import된 모듈인지 확인하기위함.
    sock = TCPServer(2500)
    data_socket, client_addr = sock.accept()
    msg = data_socket.recv(1024).decode()
    print("recv : ", msg)

    data_socket.send(msg.encode())
    data_socket.close()
```

```
import my_server_module as mt # My TCP

port = 2500
sock = mt.TCPServer(port)

while True:
    data_socket, client_addr = sock.accept()
    print("connected by ", client_addr)
    msg = data_socket.recv(1024).decode()
    print("msg : ", msg)
    data_socket.send(msg.encode())
    data_socket.close()
```