KH정보교육원

네트워크 프로그래밍

네트워크 프로그래밍

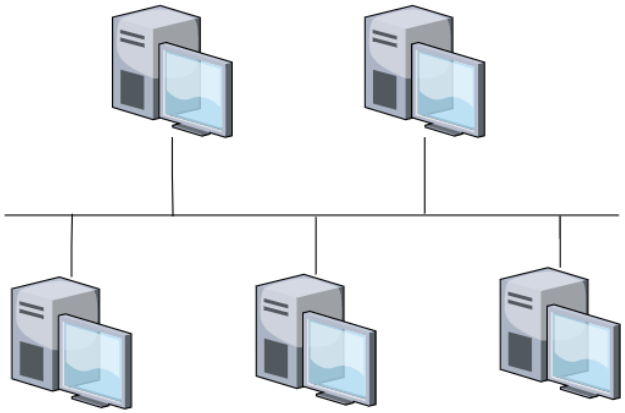
KH정보교육원

Network

하나 이상의 단말이 매개체를 통해 연결되어 통신이 가능한 상태

Network의 목적

- 여러 개의 통신기기(컴퓨터,휴대폰 등)들을 서로 연결하여 데이터를 손쉽게 주고 주고 받거나 또는 자원(프린터 등)등을 공유하기 위함. 추가적으로 빠른 데이터 교환 가능
- 자바에서 제공하는 java.net 패키지를 사용하면 어플리케이션끼리 데이터 통신 가능



네트워크 프로그래밍

KH정보교육원

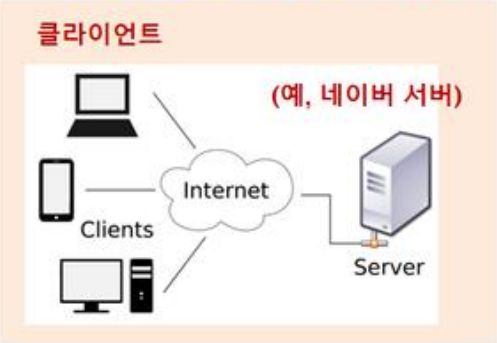
서버(Server)

다양한 서비스를 제공하는 컴퓨터 또는 프로그램

클라이언트(Client)

서비스를 요청해서 사용하는(제공받는) 컴퓨터 또는 프로그램

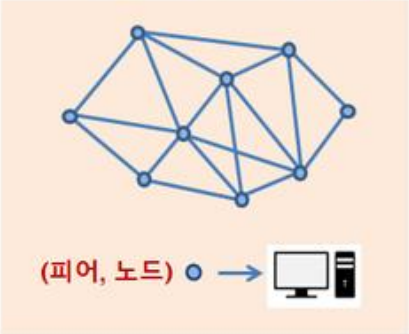
1. 클라이언트-서버 모델 : 중앙화



1) 서버: 서비스 제공 (DB 관리 & 처리)
   - 3자가 서버관리: 검열 가능
2) 클라이언트: 서비스를 표시하는 단말

그림출처:
https://upload.wikimedia.org/wikipedia/commons/c/c9/Client-server-model.svg

2. P2P 네트워크: 분산화



1) P2P 네트워크: 다수의 피어로 구성
   - 피어: 동일한 권한의 참가자
2) 암호화폐에서 피어의 역할
   - DB 관리 및 처리(서버 역할)
3) 3자(중개자)가 필요 없음

네트워크 프로그래밍

KH정보교육원

Protocol

원활한 통신을 하기 위해 미리 지정한 통신 규약.

컴퓨터 간의 정보를 주고 받을 때의 통신방법에 대한 규약으로 접속이나 전달방식

데이터 형식, 검증 방법 등을 맞추기 위한 약속

ex) HTTP, IP, FTP, TCP, UDP 등

IP(Internet Protocol)

전 세계 컴퓨터에 부여된 고유의 식별 주소

세상의 모든 컴퓨터(또는 네트워크 기기)는 중복되지 않는 IP 주소

IP 주소는 컴퓨터끼리 서로 통신하기 위한 '전화번호'

네트워크 프로그래밍

KH정보교육원

TCP(Transmission Control Protocol)

- 데이터 전달의 신뢰성을 최대한 보장하기 위한 방식
- 연결지향 : 데이터를 전달할 논리적인 연결을 먼저 구성 (3way-Handshake)
- 신뢰성 : 순차적으로 데이터를 전송함, 확인 응답 및 오류시 재전송

UDP(User Datagram Protocol)

- 데이터의 빠른 전달을 보장하기 위해 사용됨
- 빠른 속도 보장을 위해 대부분의 기능을 제한함
  - 비 신뢰성 : 확인 응답 및 재전송 작업이 없음
  - 비 연결 지향성 : 세션을 맺지 않음
- 빠른 데이터 전송을 위해 사용 됨 (실시간 스트리밍 서비스 등)

IP Address (IPv4)

- 인터넷에 연결된 컴퓨터(혹은 모바일 기기 등)에 할당된 IP 주소는 IPv4(IP version4) 형태 IPv4의 주소 체계는 32bit로 표기
- 최대 약 40억개의 서로 다른 주소를 부여 가능

IP Address (IPv6)

- IPv6의 주소 체계는 128bit로 표기, 16bit씩 8마디
- 최대 1조개의 서로 다른 주소를 부여 가능

네트워크 프로그래밍

KH정보교육원

Port

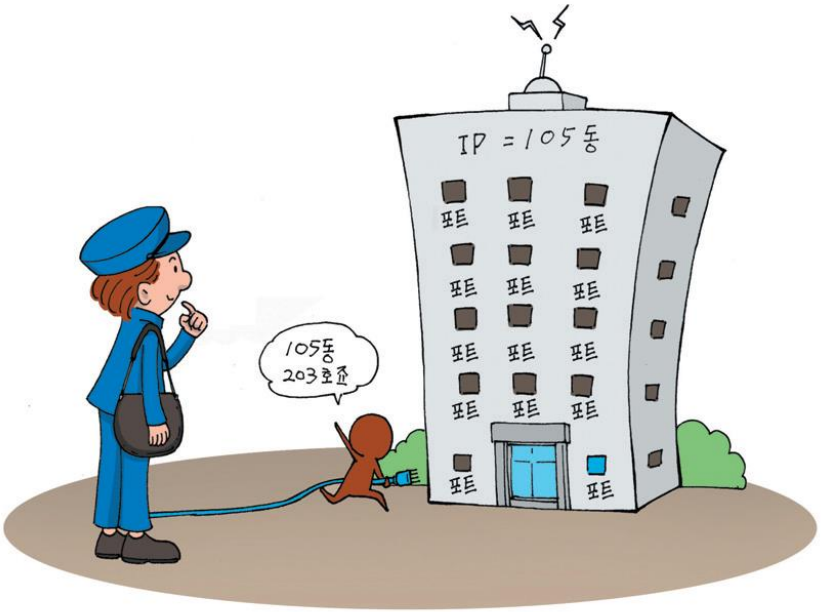
네트워크 상의 응용프로그램을 지정하는 번호(프로그램 식별 번호)

TCP, UDP는 각 각 0~65535 번의 Port 번호를 가지며 Port 번호를 가지고 서비스를 구분함

모든 응용프로그램은 하나 이상의 포트 생성 가능, 포트를 이용하여 상대방 응용프로그램과 데이터 교환

Well-known Port(0 ~ 1023)

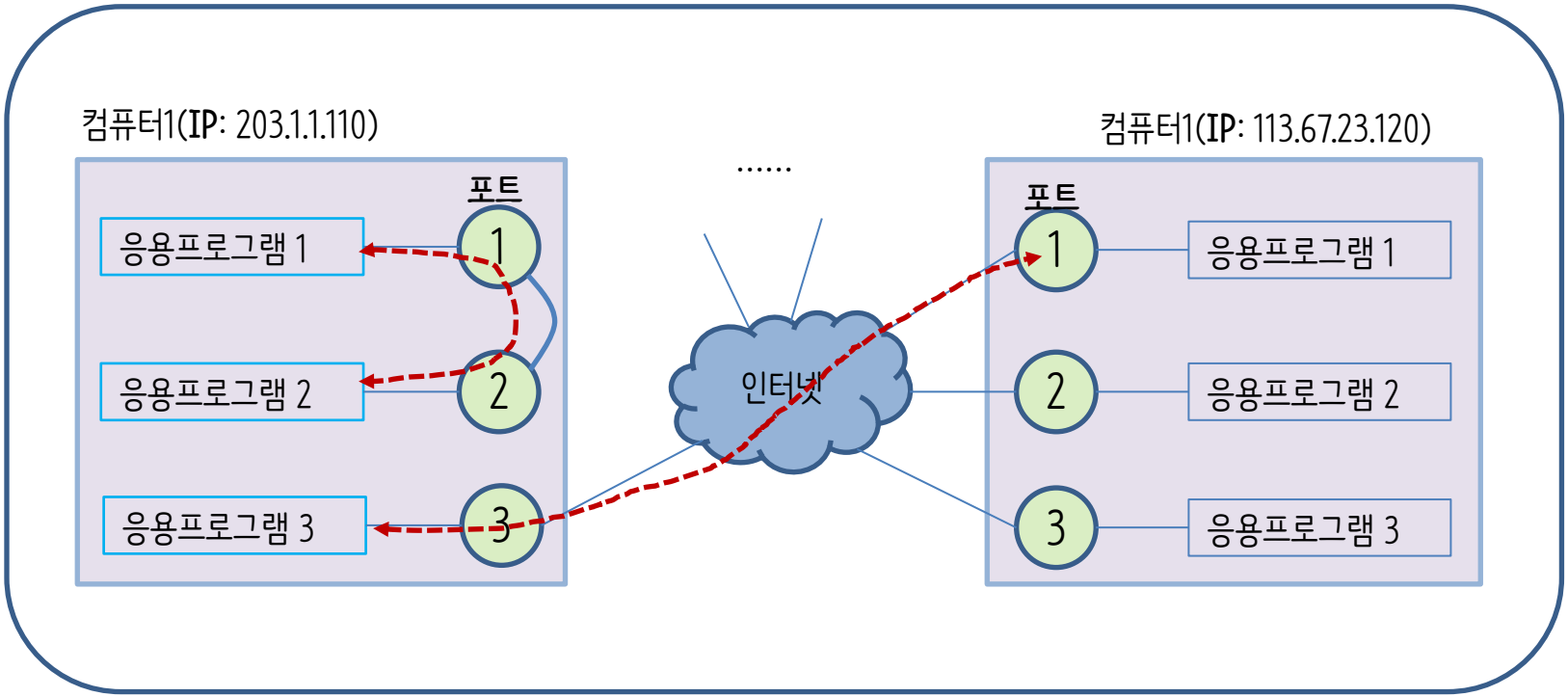
FTP(21), Telnet(23), 웹 서비스(80), DNS(53) 처럼 인터넷에서 자주 사용하는 응용 서비스는 모두 고정된 포트 번호를 사용



네트워크 프로그래밍

KH정보교육원

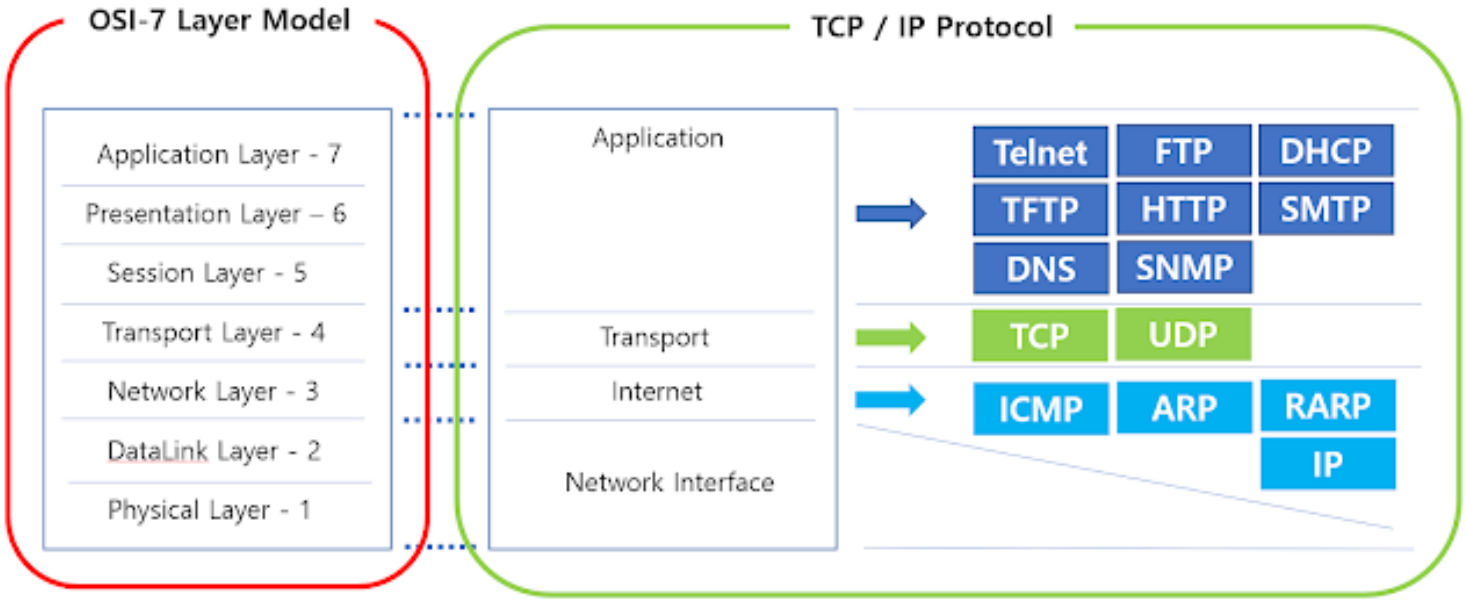
Port를 이용한 통신



KH정보교육원

Network Model

- 데이터를 만들때 최상의 효율성과 안정성을 보장하기 위해 만들어 놓은 구조 형태
- 통신이 일어나는 절차를 각 기능별로 모듈화시켜 만들어 놓은 계층적인 구조
- OSI 7 Layer 와 TCP/IP 모델이 있음



OSI 7 Layer

계층	설명
7계층 (Application)	사용자 인터페이스 계층으로 사용자의 명령을 받아주는 계층
6계층 (Presentation)	상위 계층에서 만들어진 데이터의 형태 표현 계층 (인코딩, 압축, 암호화 등)
5계층 (Session)	하위 계층과 상위 계층의 세션 연결 및 동기화를 하기 위한 계층
4계층 (Transport)	데이터 전송 방식 결정 계층 (TCP,UDP)
3계층 (Network)	종단 간 연결 보장 계층 (출발지와 목적지 주소 부여 계층, IP Address)
2계층 (Data Link)	Node 간 연결 보장 계층 (인접 장비에 접근하기 위한 정보 부여 계층 , MAC Address)
1계층 (Physical)	비트 형태의 신호를 패턴을 부여하여 전기적 신호로 변경하여 전송하는 계층

TCP/IP Model

계층	설명
4계층 (Application)	OSI 7 Layer에서 5~7 계층에 속함. 상위 계층이며 소프트웨어 계층이라고 함. 데이터 생성 계층으로 주로 소프트웨어 개발 분야에서 참조.
3계층 (Transport)	OSI 7 Layer에서 4계층에 속함. 하위 계층이며 하드웨어 계층이라고 함. 데이터 전달 계층으로 주로 네트워크 분야에서 참조
2계층 (Internet)	OSI 7 Layer에서 3계층에 속함 상동
1계층 (Network Interface)	OSI 7 Layer에서 1,2 계층에 속함 상동

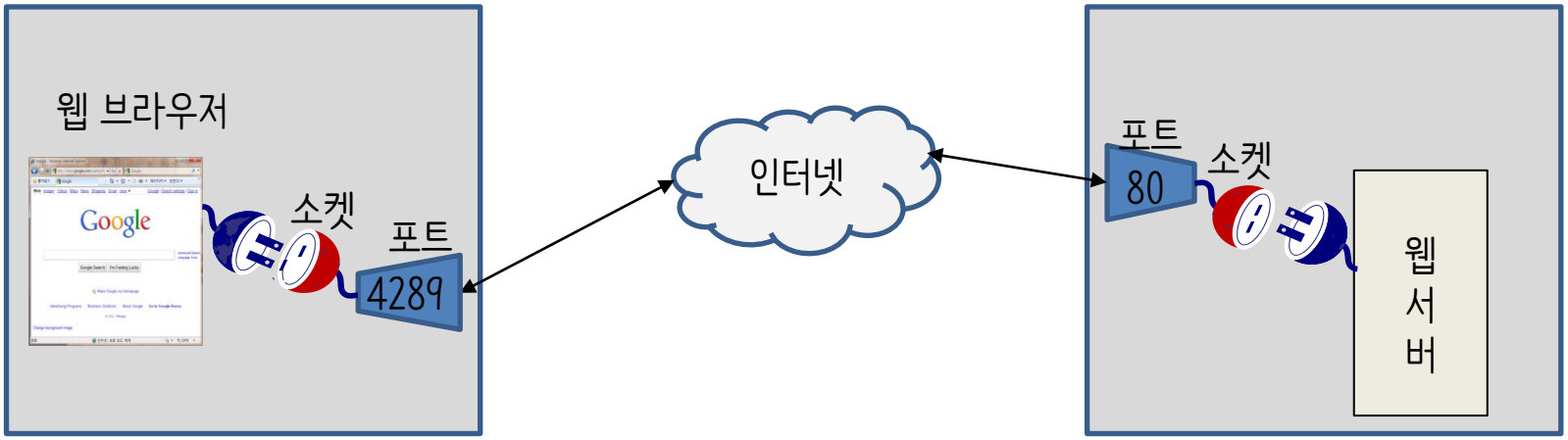
소켓(Socket) 프로그래밍

KH정보교육원

소켓(Socket)이란?

소켓이란 프로세스 간의 통신을 담당하며, InputStream과 OutputStream을 가지고 있으며 두가지 스트림을 통해 프로세스 간의 통신(입출력)이 이루어진다.

즉, 소켓이란 서버와 클라이언트가 통신을 하기 위한 매개체

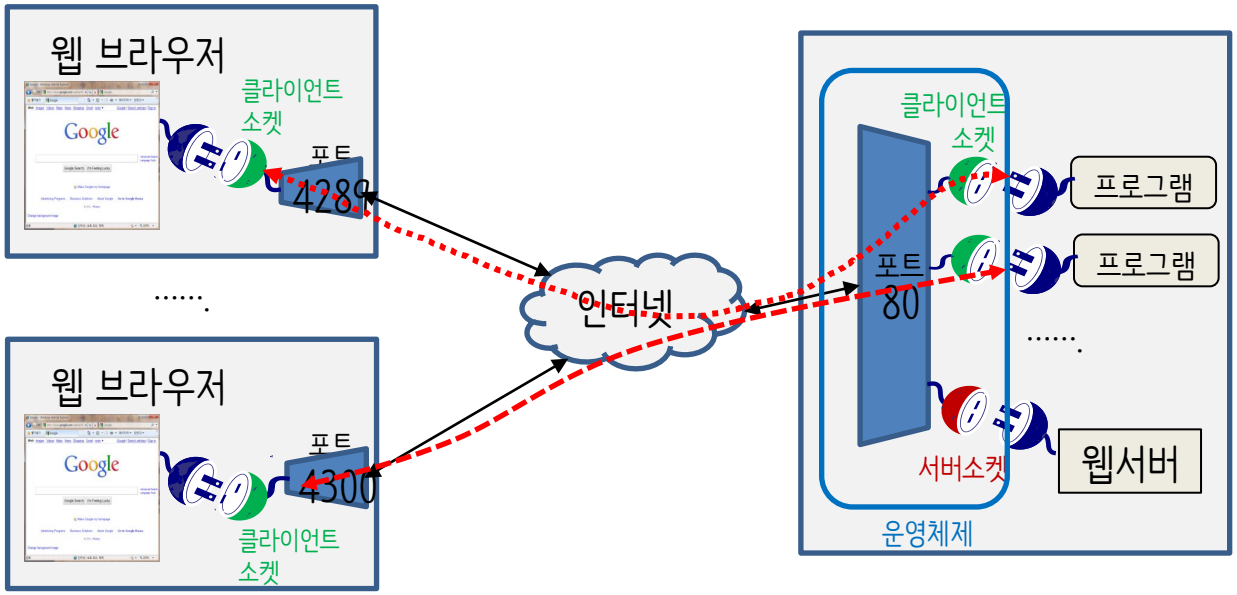


소켓(Socket) 프로그래밍

KH정보교육원

소켓(Socket) 프로그래밍란?

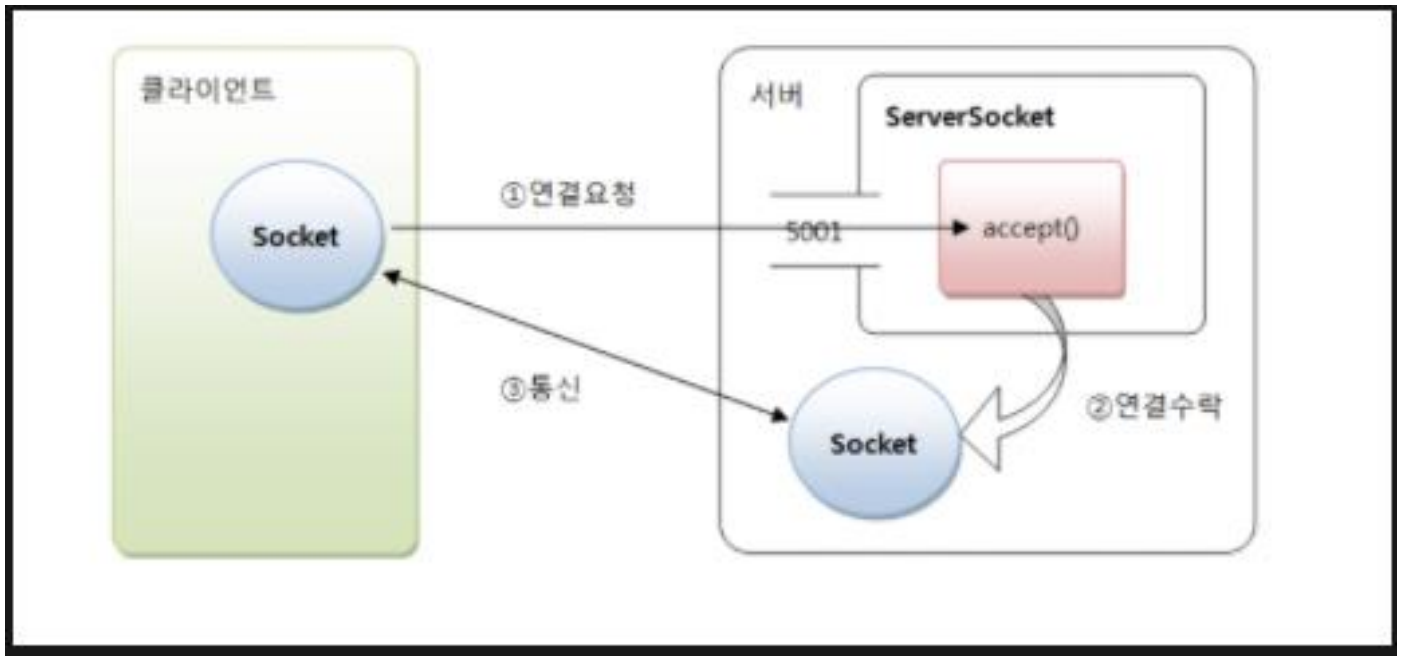
- TCP/IP 네트워크를 이용하여 쉽게 통신 프로그램을 작성하도록 지원하는 기반 기술
- 자바로 소켓 통신할 수 있는 라이브러리 지원
- 소켓(Socket)은 소켓끼리 데이터를 주고 받으며, 소켓은 특정 IP 포트 번호와 결합함.



소켓(Socket) 프로그래밍

KH정보교육원

소켓(Socket)과 서버소켓(Server Socket)



소켓(Socket) 프로그래밍

KH정보교육원

코드 작성

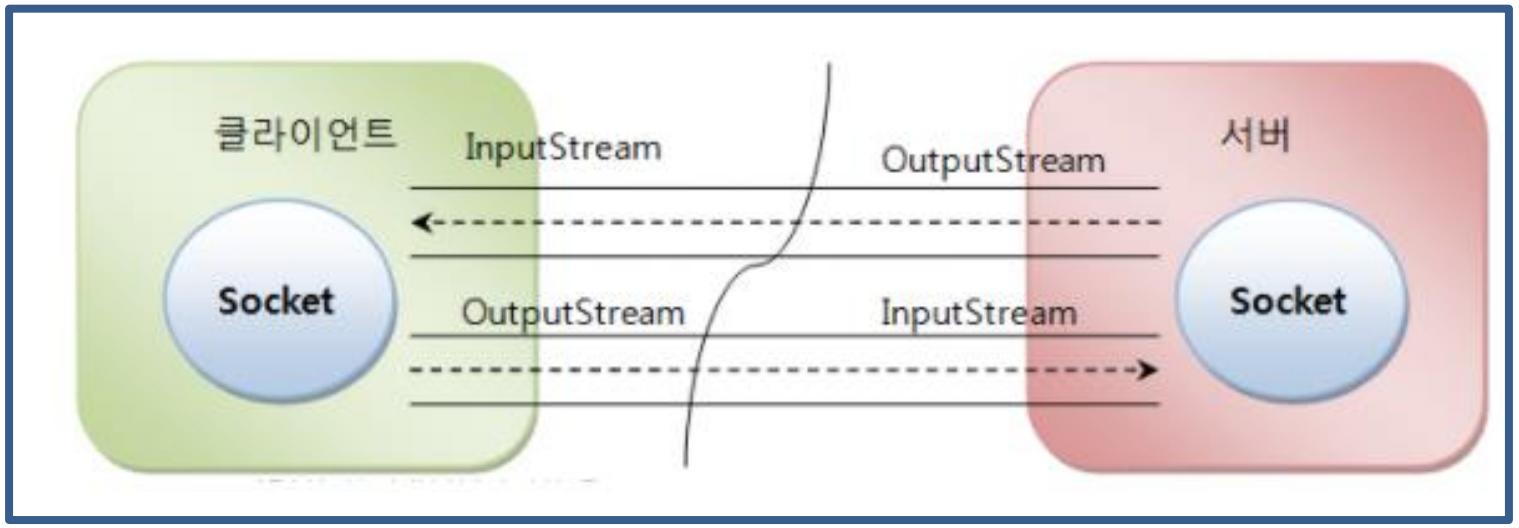
```
public class ClientProgram {
    public static void main(String [] args) {
        try {
            System.out.println("연결 요청");
            socket = new Socket("127.0.0.1", 4885);
            System.out.println("연결 성공!");
        }catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
public class ServerProgram {
    public static void main(String [] args) {
        ServerSocket serverSocket = null;
        try {
            serverSocket = new ServerSocket(4885);
            while(true) {
                System.out.println("연결 기다림");
                Socket socket = serverSocket.accept();
                System.out.println("연결 수락함");
            }
        }catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

소켓(Socket) 프로그래밍

KH정보교육원

소켓(Socket) 데이터 통신



소켓(Socket) 프로그래밍

KH정보교육원

코드 작성

```
// 입력 스트림 얻기
InputStream is = socket.getInputStream();
// 출력 스트림 얻기
OutputStream os = socket.getOutputStream();

// 데이터 받기
byte [] byteArr = new byte[100];
InputStream inputStream = socket.getInputStream();
int readByteCount = inputStream.read(byteArr);
String data = new String(byteArr, 0, readByteCount);

// 데이터 보내기
String data = "보낼 데이터";
byte [] byteArr = data.getBytes();
OutputStream outputStream = socket.getOutputStream();
outputStream.write(byteArr);
outputStream.flush();
```