

# 네트워크(Network)





네트워크



# 네트워크

## 네트워크

- 여러 대의 컴퓨터를 통신 회선으로 연결한 것
- 홈 네트워크, 지역네트워크, 인터넷 등이 해당

## 네트워크의 목적

- 여러 개의 통신기기(컴퓨터, 휴대폰 등)들을 서로 연결하여 데이터를 손쉽게 주고 받거나 자원(프린터 등) 등을 공유하기 위함
- 빠른 데이터 교환



# 네트워크

## 서버와 클라이언트

- 네트워크로 연결된 컴퓨터 간의 관계를 역할(role)로 구분한 개념

## 서버

- 서비스를 제공하는 컴퓨터 또는 프로그램

## 클라이언트

- 서비스를 요청하여 사용하는(제공받는) 컴퓨터

※ 서비스를 제공하기 위해서는 서버용 프로그램이 있어야 하며, 서비스를 이용하기 위해서는 클라이언트용 프로그램이 각각 있어야 한다.



# 네트워크

## IP 주소

- 네트워크 상에서 통신기기들이 서로 인식하고 통신하기 위해 사용하는 고유한 주소
- 같은 네트워크상에서 동일한 IP주소를 여러 컴퓨터가 사용 할 수 없음
- 자바에서는 InetAddress 클래스를 통해서 관리

## port

- 같은 컴퓨터 내에서 프로그램을 식별하는 번호
- 클라이언트는 서버 연결 요청 시 IP주소와 port번호를 알아야 요청 가능

※ 서비스를 제공하기 위해서는 서버용 프로그램이 있어야 하며, 서비스를 이용하기 위해서는 클라이언트용 프로그램이 각각 있어야 한다.



# 네트워크

## 프로토콜

- 컴퓨터 간의 정보를 주고 받을 때의 통신방법에 대한 규약으로 접속이나, 전달방식, 데이터 형식, 검증 방법 등을 맞추기 위한 약속

## TCP

- 데이터의 전달의 신뢰성을 최대한 보장하기 위한 방식
- 연결지향형 통신
- 순차적으로 데이터를 전송하고 확인 및 오류 시 재전송
- HTTP, FTP등에 사용

## UDP

- 데이터의 빠른 전달을 보장하기 위해 사용
- 비 연결 지향형 통신
- 확인 및 재전송 작업이 없음
- 실시간 스트리밍 서비스 등에 사용





# Socket Programming



# Socket Programming

## 소켓(Socket)

- 서버와 클라이언트가 통신을 하기 위한 매개체

## 소켓 프로그래밍

- 소켓을 이용한 통신 프로그래밍
- 자바의 `java.net` 패키지를 통해 소켓프로그래밍을 지원

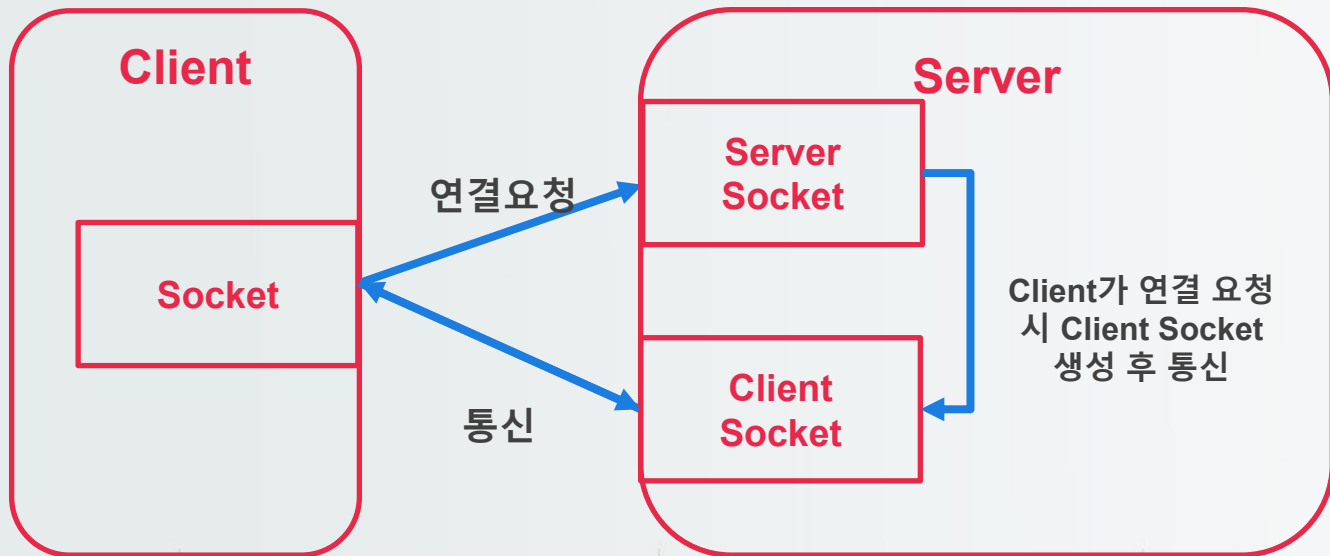




# Socket Programming

## TCP 소켓 프로그래밍

- 서버가 클라이언트의 연결 요청을 기다리다가 요청이 오면 연결하여 통신



# Socket Programming

## TCP 소켓 프로그래밍 순서(서버)

1. 서버에서 사용할 포트번호 지정
2. 서버용 소켓 객체 생성
3. 클라이언트 접속요청 대기
4. 접속요청이 오면 요청을 수락하고 해당 클라이언트에 대한 소켓객체 생성
5. 연결된 클라이언트와 입력 및 출력 스트림 생성
6. 보조스트림을 통한 성능 개선
7. 스트림을 통한 입력 및 출력
8. 통신 종료



# Socket Programming

## TCP 소켓 프로그래밍 순서(클라이언트)

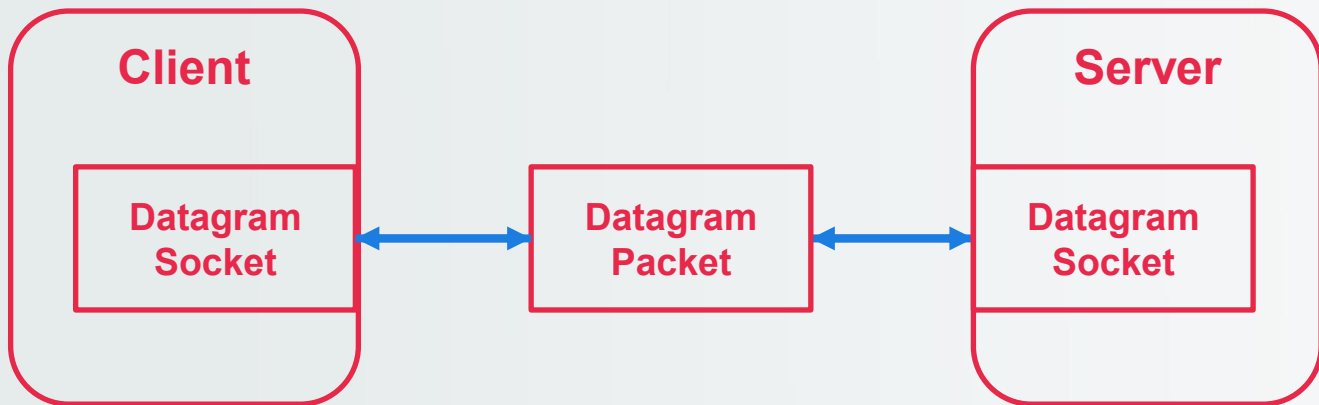
1. 서버의 IP와 서버의 port로 Socket객체 생성(서버에 요청)
2. 서버와의 입력 및 출력 스트림 생성
3. 보조스트림을 통한 성능개선
4. 스트림을 통한 입력 및 출력
5. 통신종료



# Socket Programming

## UDP 소켓 프로그래밍

- UDP는 연결 없이 DatagramSocket간에 DataPacket으로 변환된 데이터를 주고받음



# Socket Programming

## UDP 소켓 프로그래밍 순서(서버)

1. 서버에서 사용할 포트번호 지정
2. DatagramSocket 객체 생성
3. 연결할 클라이언트 IP 주소를 가진 InetAddress 객체 생성
4. 전송할 메시지를 byte[]로 변경
5. 전송할 메시지를 DatagramPacket 객체에 저장
6. 소켓 레퍼런스를 사용하여 전송
7. 소켓 종료



# Socket Programming

## UDP 소켓 프로그래밍 순서(클라이언트)

1. 서버가 보낸 메시지를 받을 `byte[]`을 준비
2. `DatagramSocket` 객체 생성
3. 메시지를 받을 `DatagramPacket` 객체 준비
4. 소켓레퍼런스를 사용하여 메시지 수신
5. `byte[]`로 받은 메시지를 `String`으로 바꾸어 사용
6. 소켓 종료

