

# Socket 통신

컴퓨터네트워크 3주차

---

공과대학 5호관 633호  
데이터 네트워크 연구실

조교 : 황동준

## 실습

### 1. Python의 socket 모듈을 사용하여 양방향 채팅 구현

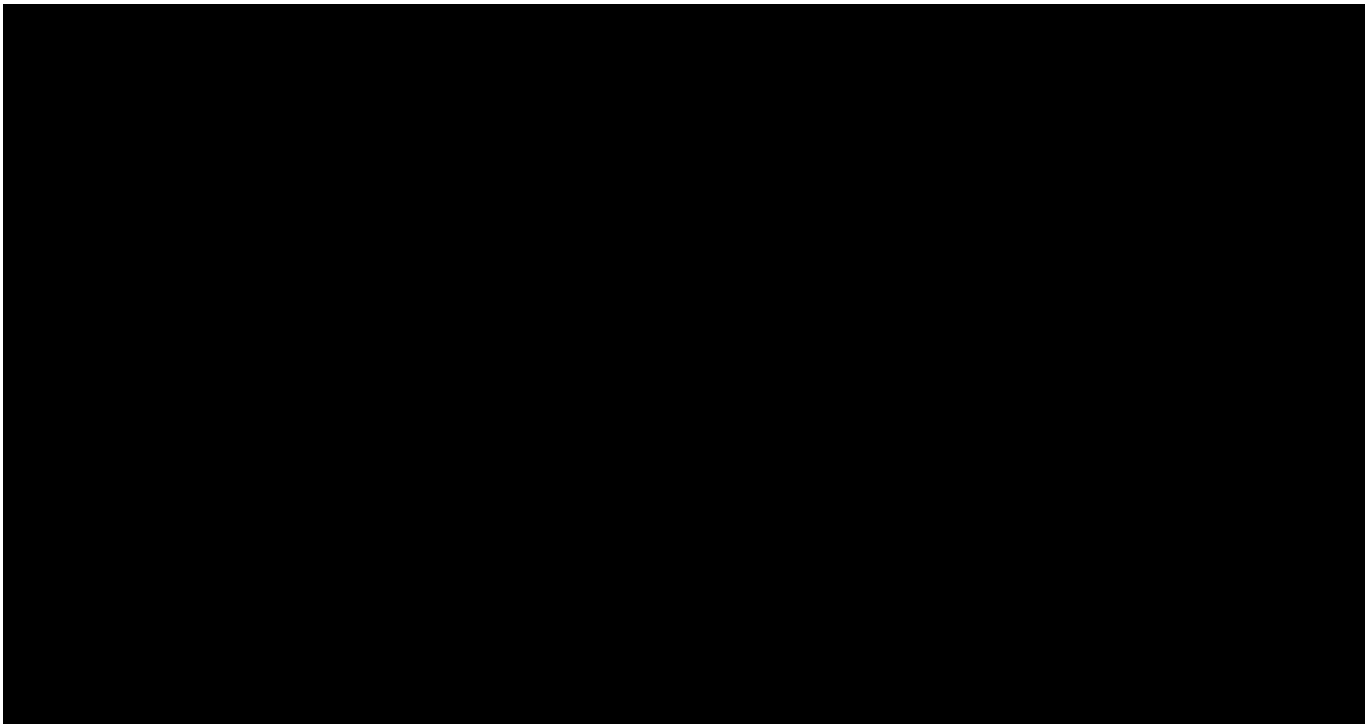
- socket 통신에 대한 이해

## 과제

### 1. Python의 socket 모듈을 통한 간단한 숫자 게임 구현 (10점)

- 서버와 클라이언트 사이의 socket 통신을 사용하여 숫자 게임을 진행
  - 서버 구현 (5점)
  - 클라이언트 구현 (3점)
  - Youtube 업로드 (2점)
    - Youtube의 경우 자신의 계정에 업로드하며, 일부 공개로 하여 링크 공유

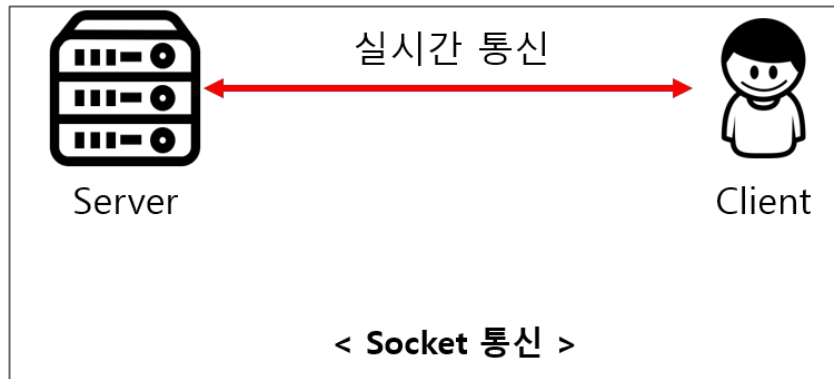
## 결과 영상



**실습 : Python의 socket 모듈을  
사용하여 양방향 통신 구현**

# Socket 통신

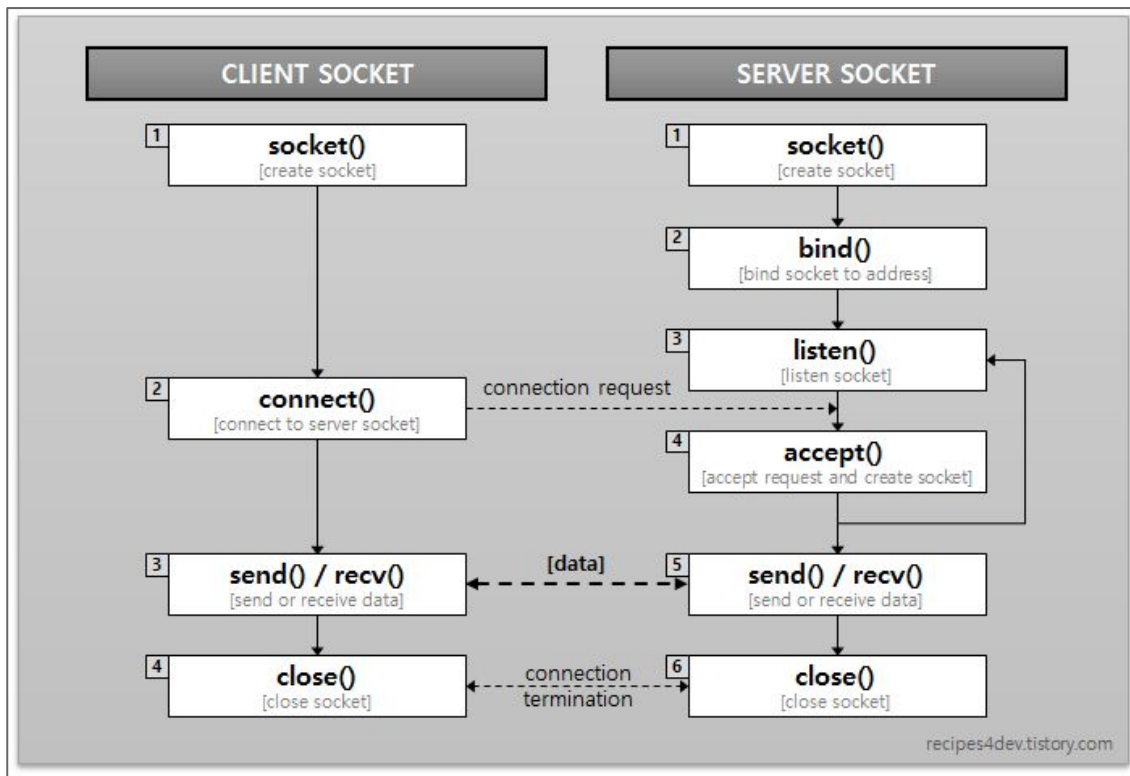
- Server와 Client가 계속 연결을 유지하는 양방향 통신
  - 실시간으로 데이터를 주고받는 상황이 필요한 경우 사용됨



- HTTP 통신과 달리 Server또한 Client로 통신을 보낼 수 있으며, 실시간 Streaming중계나 실시간 채팅, 온라인 게임등에 자주 사용됨

# python socket 통신

- TCP socket 통신
- server와 client의 통신을 진행하는 기능 확인
- server를 먼저 실행한 후, client를 실행
  - 1번부터 순서대로 실행됨



# python socket 통신

```
1 import socket
2
3 TCP_IP = '10.0.2.15'
4 TCP_PORT = 5001
5
6 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 sock.bind((TCP_IP, TCP_PORT))
8 sock.listen()
9
10 conn, addr = sock.accept()
11 print('Connection address : ' + str(addr))
12 while True:
13     data = conn.recv(1024)
14     if not data: break
15     print('Connection address : ' + data.decode())
16     conn.send(data)
17 conn.close()
```

socket-server.py

```
1 import socket
2
3 TCP_IP = '10.0.2.15'
4 TCP_PORT = 5001
5
6 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 sock.connect((TCP_IP, TCP_PORT))
8
9 message = 'hello server! I\'m client'
10 sock.send(message.encode())
11 data = sock.recv(1024)
12 sock.close()
13
14 print('received message : ' + data.decode())
```

socket-client.py

- IP 주소 : 각자 자신의 주소 사용할 것
- 이전 슬라이드 그림 참고하면서 코드 실습 진행해 볼 것

# python socket 통신 결과 화면

<pre>datanetwork@datanetwork-VirtualBox:~/socket\$ python3 socket-server.py Connection address : ('10.0.2.15', 54498) Connection address : hello server! I'm client</pre>	<pre>datanetwork@datanetwork-VirtualBox:~/socket\$ python3 socket-client.py received message : hello server! I'm client datanetwork@datanetwork-VirtualBox:~/socket\$</pre>
---	---

1. server와 client 파일을 실행
2. client가 보낸 메시지를 server가 전달 받음
3. server가 전달받은 메시지를 다시 client로 전송 후, 출력
4. client는 server로부터 전달받은 메시지를 출력



**과제 : Python의 socket 모듈을 통한  
간단한 숫자 게임 구현**

# 숫자 게임 규칙

1. Client 3명의 요청을 대기
2. 랜덤한 숫자 10개를 생성  
(범위 : 1 ~ 100)
3. Client에게 1번부터 10번까지의 숫자 중 선택 요청
4. 랜덤한 숫자 1개를 생성  
(범위 : -1 ~ 4) → 각 client마다 생성된 숫자를 부여
  - a. client마다 부여되는 랜덤 숫자는 모두 다름
5. multiply와 add 중 선택 요청
  - a. 사용자가 2번에서 선택했던 숫자와 5번에서 선택한 계산 방식을 사용하여 4번에서 생성된 숫자와 계산
6. 3명의 client의 결과를 비교하여 1등은 이겼다는 메시지, 나머지는 졌다는 메시지 전달

# 숫자 게임 규칙 - Client

1. Server로 연결 요청
2. Server로부터 1번부터 10번까지의 숫자 요청이 들어올 경우 선택하여 전송
3. Server로부터 multiply나 add에 대한 선택 요청이 들어올 경우 선택하여 전송

# 숫자 게임 구현시 유의사항

- 결과 영상과 동일하게 문구 출력이 되도록 할 것
  - 출력 문구를 다르게 하고 싶을 경우 보고서에 명시하여 다르게 진행 가능
  - 게임 순서는 동일하게 지킬 것!
- Client와 관련된 파일은 1개만 제작하여 server와 통신할 것
- socket 통신은 TCP/IP 소켓 통신을 사용 (UDP는 사용 금지)
- IP주소는 현 운영체제의 주소를 사용하며, Port번호는 5001번으로 고정
- multiprocessing, multithread, multiplexing등은 사용 금지
  - single process로 진행할 것

# 과제 제출

- 과제 제출 기한
  - 2021년 9월 30일 17:59 까지 사이버 캠퍼스에 제출
- 제출 파일 (.zip)
  - 학번\_server.py / 학번\_client.py (파일)
  - 보고서 (Youtube링크 포함)
- 딜레이 1일당 20%감점
  - ex) 10점 만점 기준, 1일 딜레이인 경우 8점
- 과제 **카피 적발시** 보여준사람 본사람 모두 0점처리
  - 카피 적발기준 : 과제 유사도 80%이상

# 유의사항

- 파일명 : **CN02\_03\_학번\_이름.zip**
  - python파일과 보고서를 함께 압축하여 제출
  - 보고서 : **PDF로 제출할것 (파일명 : 03\_학번\_이름.pdf)**
    - 과제 목표 (도출해야할 결과)
    - 코드 설명과 과제 해결 방법 (실행 결과 사진도 보고서에 첨부, 코드 및 설명 필요)
    - 과제 느낀점 및 하고싶은 말 (**선택사항**)
  - 형식 지켜지지 않을시 채점대상에서 제외 (보고서도 HWP, DOC은 채점대상에서 제외)
- 질문
  - 디스코드 컴퓨터네트워크 채널에서 질문을 받음