

# java

## 28강\_네트워킹

네트워크에 연결되어 있는 대상 사이에 데이터를 전송하는 방법에 대해서 살펴봅니다.

---

28-1 : 네트워크 데이터 입력 및 출력

28-2 : 소켓(Socket)

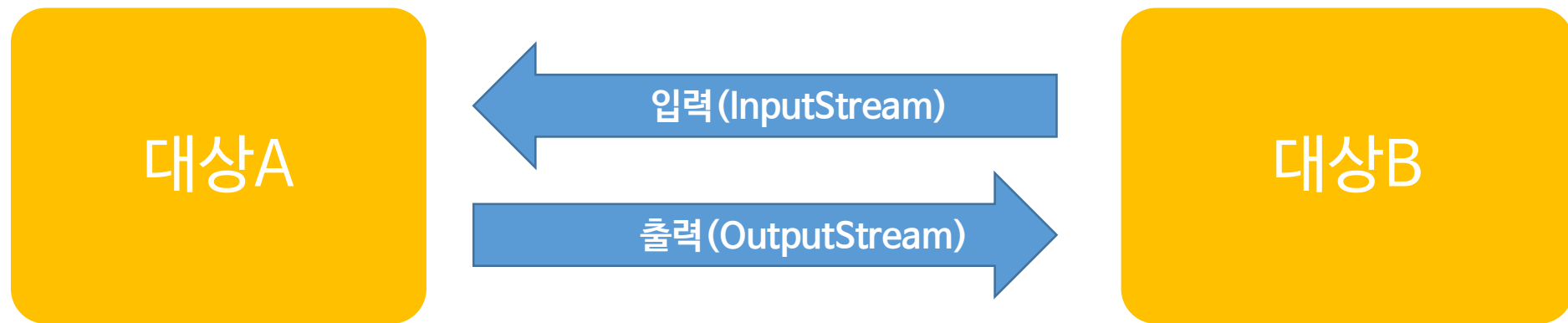
28-3 : Socket 클래스

28-4 : Client와 Server 소켓(Socket)

28-5 : 양방향 통신

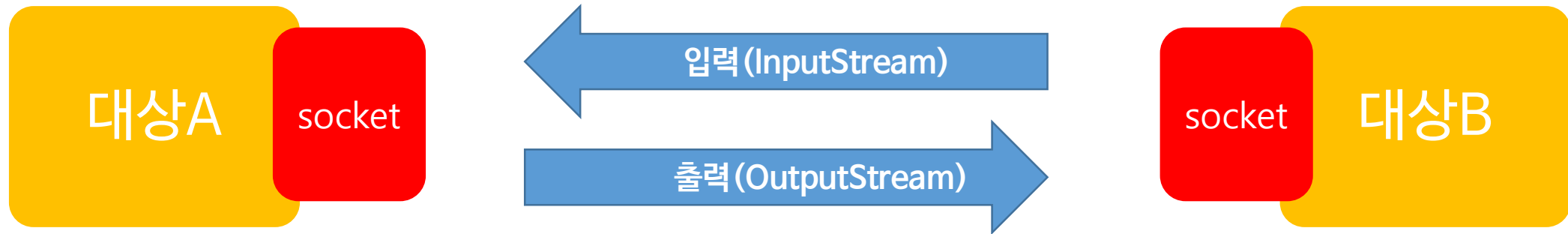
## 28-1 : 네트워크 데이터 입력 및 출력

네트워크 대상(객체) 사이에 입/출력(InputStream, OutputStream)를 이용해서 데이터를 입력하고 출력한다.



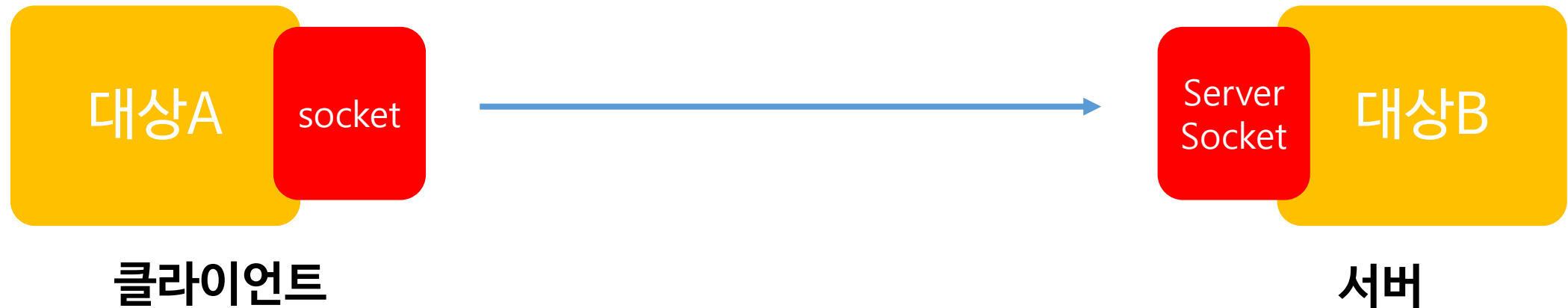
## 28-2 : 소켓(Socket)

네트워크상에서 데이터를 주고받기 위한 장치이다.

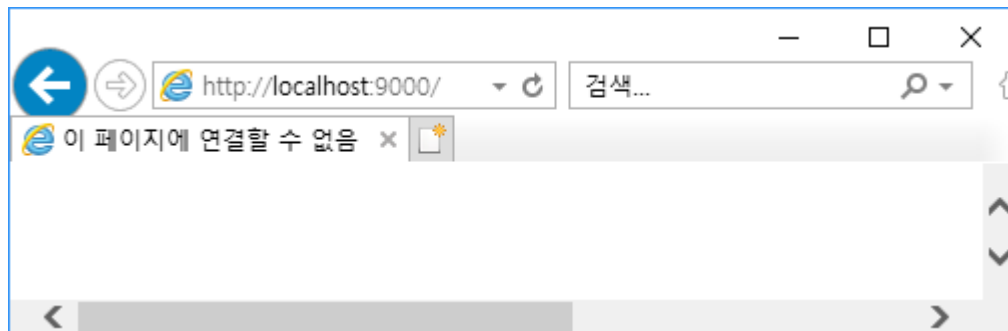


## 28-3 : Socket 클래스

서버는 클라이언트를 맞을 준비를 하고 있다가 클라이언트의 요청에 반응한다.



http://localhost:9000



```
serverSocket = new ServerSocket(9000);
System.out.println("클라이언트 맞을 준비 완료~~");
```

```
socket = serverSocket.accept();
System.out.println("클라이언트 연결~~");
System.out.println("socket: " + socket);
```

```
Problems @ Javadoc Declaration Console
<terminated> MainClass (3) [Java Application] C:\Program Files
클라이언트 맞을 준비 완료~~
클라이언트 연결~~
socket: Socket[addr=/0:0:0:0:0:0:0:0:
```

## 28-4 : Client와 Server 소켓(Socket)

서버에 ServletSocket를 준비하고 클라이언트에서 Socket를 이용해서 접속한다.

```
Socket socket = null;

try {
    socket = new Socket("localhost", 9000);
    System.out.println("서버 연결~~");
    System.out.println("socket: " + socket);
}
```



```
ServerSocket serverSocket = null;
Socket socket = null;

try {
    serverSocket = new ServerSocket(9000);
    System.out.println("클라이언트 맞을 준비 완료~~");

    socket = serverSocket.accept();
    System.out.println("클라이언트 연결~~");
    System.out.println("socket: " + socket);
}
```

Problems Javadoc Declaration Console

<terminated> MainClassSocket [Java Application] C:\Program Files\Java\jre1.8.0\_181\bin\javaw.exe (2018. 7. 30. 오후 6)

서버 연결~~

socket: Socket[addr=localhost/127.0.0.1,port=9000,localport=49987]



<terminated> MainClassServerSocket (3) [Java Application] C:\Program Files\Java\jre1.8.0\_181\bin\java

클라이언트 맞을 준비 완료~~

클라이언트 연결~~

socket: Socket[addr=/127.0.0.1,port=49987,localport=9000]

## 28-5 : 양방향 통신

클라이언트와 서버는 InputStream, OutputStream을 이용해서 양방향 통신을 할 수 있다.

```
try {  
    socket = new Socket("localhost", 9000);  
    System.out.println("서버 연결 완료~~");  
  
    outputStream = socket.getOutputStream();  
    dataOutputStream = new DataOutputStream(outputStream);  
  
    inputStream = socket.getInputStream();  
    dataInputStream = new DataInputStream(inputStream);  
  
    scanner = new Scanner(System.in);  
  
    while (true) {  
        System.out.println("메시지 입력~~");  
        String outMessage = scanner.nextLine();  
        dataOutputStream.writeUTF(outMessage);  
        dataOutputStream.flush();  
  
        String inMessage = dataInputStream.readUTF();  
        System.out.println("inMessage : " + inMessage);  
  
        if(outMessage.equals("STOP")) break;  
    }  
}
```



```
try {  
    serverSocket = new ServerSocket(9000);  
    System.out.println("클라이언트 맞을 준비 완료~~");  
  
    socket = serverSocket.accept();  
    System.out.println("클라이언트 연결~~");  
    System.out.println("socket: " + socket);  
  
    inputStream = socket.getInputStream();  
    dataInputStream = new DataInputStream(inputStream);  
  
    outputStream = socket.getOutputStream();  
    dataOutputStream = new DataOutputStream(outputStream);  
  
    while (true) {  
        String clientMessage = dataInputStream.readUTF();  
        System.out.println("clientMessage : " + clientMessage);  
  
        dataOutputStream.writeUTF("메시지 전송 완료~~");  
        dataOutputStream.flush();  
  
        if(clientMessage.equals("STOP")) break;  
    }  
}
```