Computer Network[Socket]

유명성

0. Python

Life is SHORT YOU Need PYTHON

0. Python

0.1 What is Python

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodename()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '    %s [label="%s' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s"];' % ast[1]
        else:
            print '"]'
    else:
        print '"];'
        children = []
        for  in n, childenumerate(ast[1:]):
            children.append(dotwrite(child))
        print ,'    %s -> {' % nodename
        for  in :namechildren
            print '%s' % name,
```

0. Python

0.2 Python's features

Dynamic typing 실행할 때 변수의 타입 검사

Script language 컴파일하지 않고 인터프리터가 코드를 직접 실행,
python은 속도를 위해 byte 코드 및 JIT 사용

Multi paradigm 절차적, 객체지향, 함수형, 관점형 프로그래밍 지원

Unlimited access 객체, 구조체 member에 무제한적 접근 가능

Everything is object python은 변수/함수 모두 객체

0. Python

0.2 Data Types

종류	설명	문법 예

`str`	문자열: 이뮤터블 방식의 일련의 유니코드 코드포인트.	`'Wikipedia'`
		`"Wikipedia"`
		`"""Spanning multiple lines"""`

`bytearray`	뮤터블(mutable) 방식의 일련의 바이트.	`bytearray(b'Some ASCII')`
		`bytearray(b"Some ASCII")`
		`bytearray([119, 105, 107, 105])`

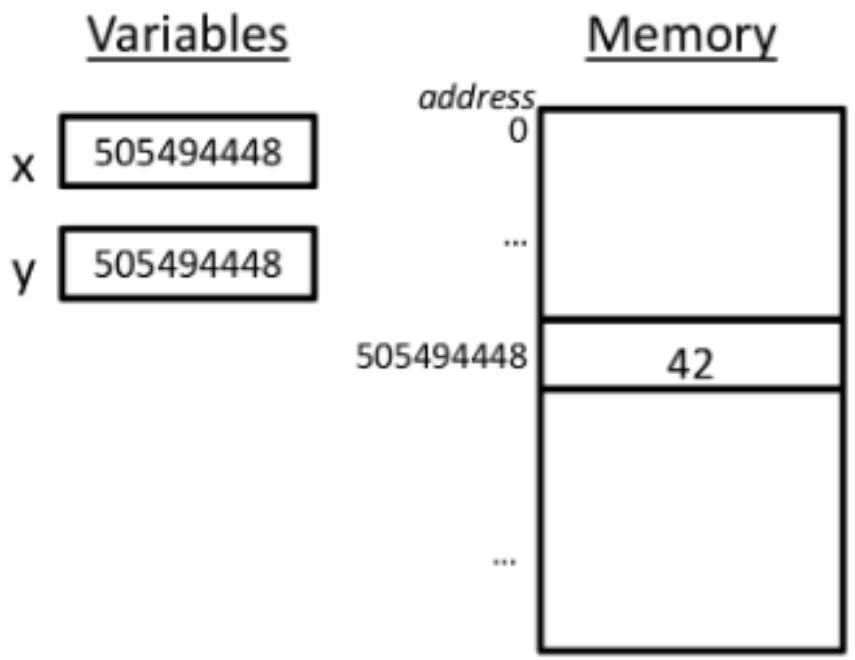
`bytes`	이뮤터블(immutable) 방식의 일련의 바이트.	`b'Some ASCII'`
		`b"Some ASCII"`
		`bytes([119, 105, 107, 105])`

`list`	뮤터블(mutable) 방식의 리스트. 혼합 형태를 포함할 수 있다.	`[4.0, 'string', True]`
`tuple`	이뮤터블(immutable) 방식. 혼합 형태를 포함할 수 있다.	`(4.0, 'string', True)`
`set`, `frozenset`	순서가 정해지지 않은 집합. 중복 허용 안 함. `frozenset` 은 이뮤터블(immutable)이다.	`{4.0, 'string', True}` `frozenset([4.0, 'string', True])`
`dict`	뮤터블(mutable) 방식의 연관 배열의 키와 값 쌍.	`{'key1': 1.0, 3: False}`
`int`	이뮤터블(immutable) 방식의 정수로서 크기는 무제한.	`42`
`float`	이뮤터블(immutable) 방식의 부동소수점 수 (시스템 정의 정밀도).	`3.1415927`
`complex`	이뮤터블(immutable) 방식의 복소수. (실수와 허수)	`3+2.7j`
`bool`	이뮤터블(immutable) 방식의 진리값.	`True` `False`

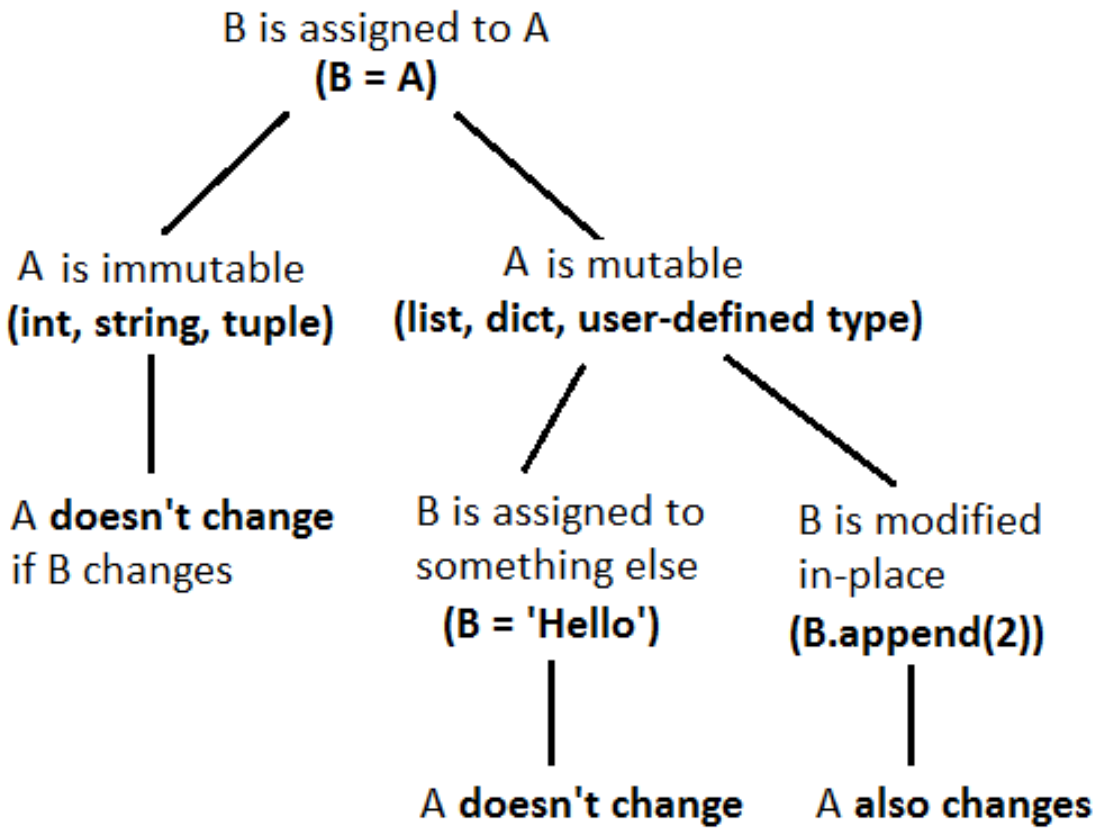
0. Python

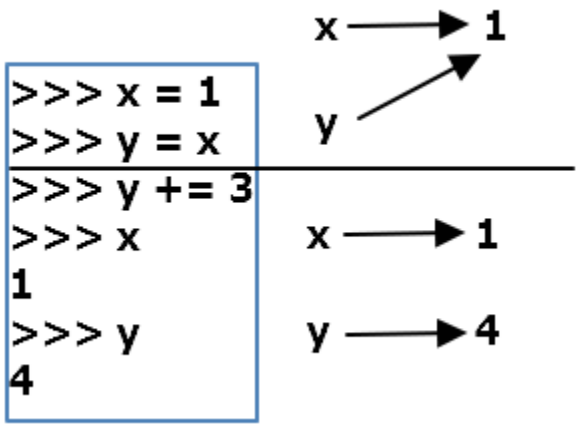
0.3 Variables

```
x = 42
y = x
```

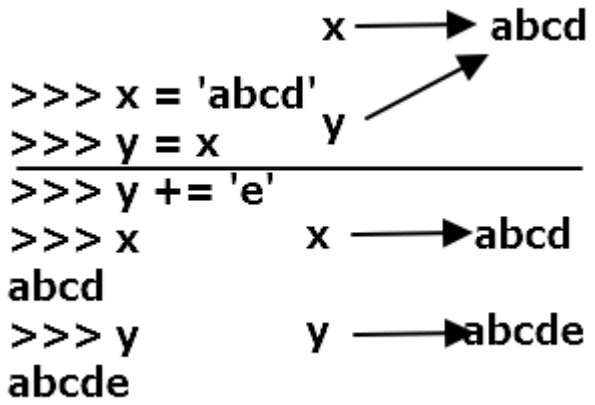


Mutable vs Immutable

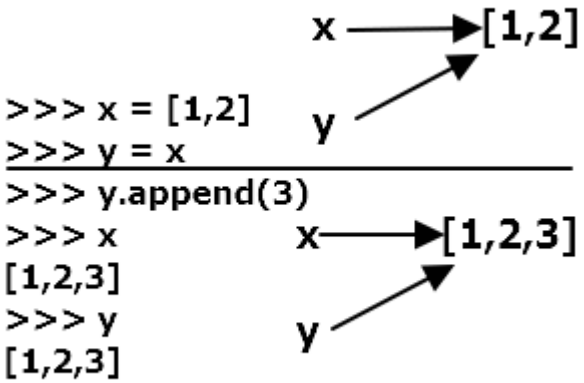




Number : immutable



String : immutable



List : Mutable

0. Python

0.4 Call by

Immutable : Like a call by value

```
>>> def test(a):
...     a += 10
...
>>> a = 1
>>> test(a)
>>> a
```

Mutable : Like a call by reference

```
>>> def test(a):
...     a.append('bear')
...
>>> b = ['teddy']
>>> test(b)
>>> b
['teddy', 'bear']
```

1. Assignment 1 Review

1. Assignment 1 Review

1.1 Command line Argument

```
if __name__ == "__main__":
    argv = sys.argv

    if(len(argv) < 5) or not ("-o" in argv and "-i" in argv):
        print("Wrong argvs : {}".format(argv))
        sys.exit()
```

명령행 인자

프로그램 시작 시 전달되는 인자
sys.argv는 String의 List이며 0번 원소는 실행파일 경로가 저장.
1번 원소부터 ‘ ’(공백)으로 구분된 인자들이 저장되어 있다.

1. Assignment 1 Review

1.2 List Slice and Comprehension

```
#arr = list(map(int, argv[4:])) you can also use the map function
arr = [int(x) for x in argv[4:]]
```

List comprehension

[대괄호] 안에서 표현식을 통해 List의 원소 생성
map 함수로 대체가능

List slice

[대괄호] 안에서 인덱스를 통해 List의 원소를 나눔

1. [n:m] = n번 원소부터 m-1번 원소
2. [n:] = n번 원소부터 끝까지
3. [:m] = 처음부터 m-1번 원소까지
4. [n:m:s] = n번부터 m-1번까지 s만큼 증가하며
5. [-1] : 마지막 원소
6. [-2:] : 맨 뒤에서 2개
7. [:-n] : 맨 뒤부터 n개 빼고 전부

1. Assignment 1 Review

1.3 Lambda

```
if argv[(argv.index("-o"))+1] == "A":
    quick_sort(arr, 0, len(arr), lambda x, y : x < y)
elif argv[(argv.index("-o"))+1] == "D":
    quick_sort(arr, 0, len(arr), lambda x, y : x > y)
```

Lambda(익명함수)

한 번 쓰고 heap에서 증발되어 되는 익명함수
가독성을 높이기 위해 사용되며, 메모리를 절약할 수 있다.
lambda 인자 : 표현식

```
>>> def hap(x, y):
...     return x + y
```

```
>>> (lambda x,y: x + y)(10, 20)
30
```

1. Assignment 1 Review

1.3 Lambda

```
10  def quick_sort(arr, left, right, key):
11      if left >= right:
12          return
13
14      pivot = arr[right-1]
15      i = left
16      j = right-2
17
18      while i <= j:
19          while(key(arr[i], pivot)):
20              i+=1
21          while(not arr[j] == pivot and not key(arr[j], pivot)):
22              j-=1
23
24          if(i<=j):
25              swap(arr, i, j)
26              i+=1; j-=1
27
28      swap(arr, i, right-1)
29      quick_sort(arr, left, i, key)
30      quick_sort(arr, i+1, right, key)
```

2. Network Socket

2. Network Socket

2.1 What is Socket

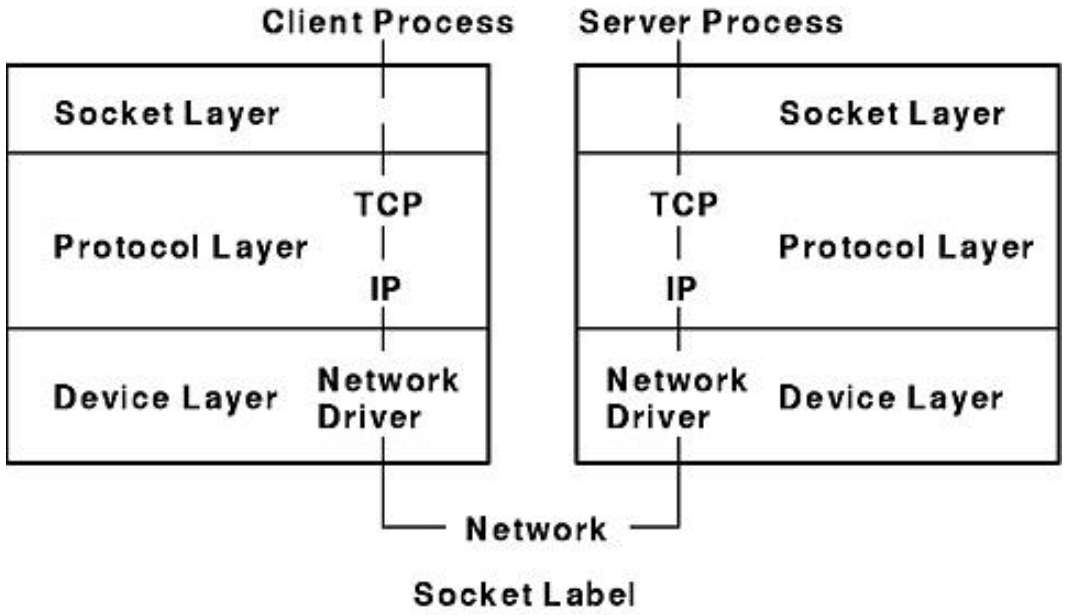
A socket is a logical endpoint of communication

2. Network Socket

2.1 What is Socket

Socket

BSD UNIX의 file을 통한 IPC(Inter Process Communication)을
확장해 원격지 호스트의 프로세스와 통신할 수 있도록 만든 인터페이스



2. Network Socket

2.2 Socket’s Features

Socket의 특징

- Socket 인터페이스 자체가 TCP/IP 표준은 아니다.
- 특정 운영체제 및 언어에 종속적으로 동작
- Server-Client 모델
- 양방향(two-way) 통신 모델

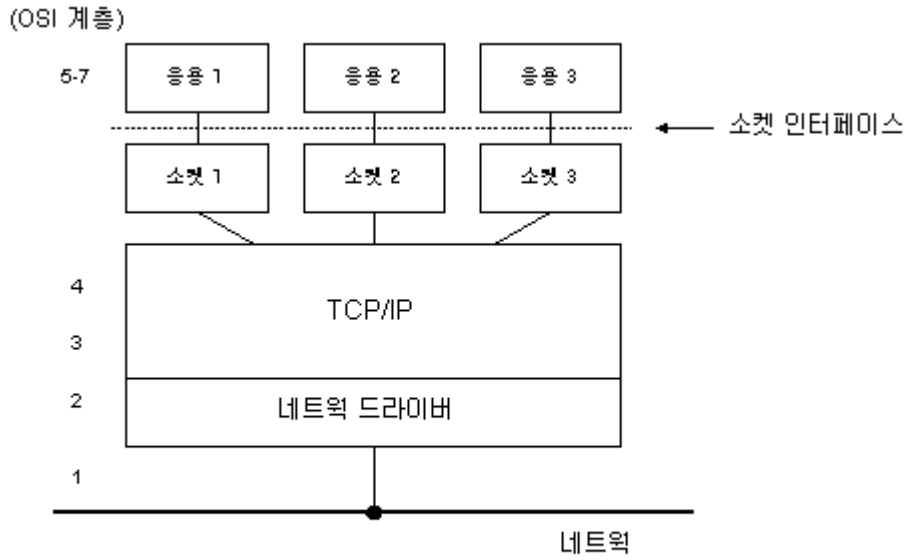
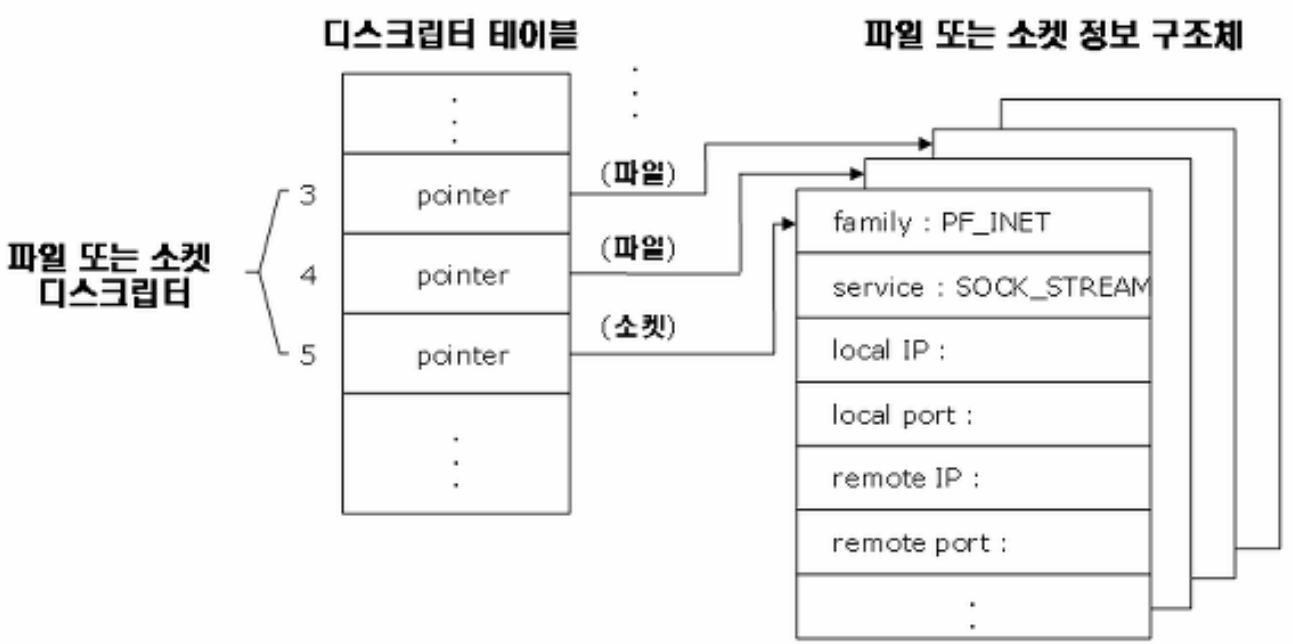


그림 2-1 소켓 인터페이스의 위치

Socket descriptor

File API와 유사하게 socket descriptor를 사용해 접근한다.

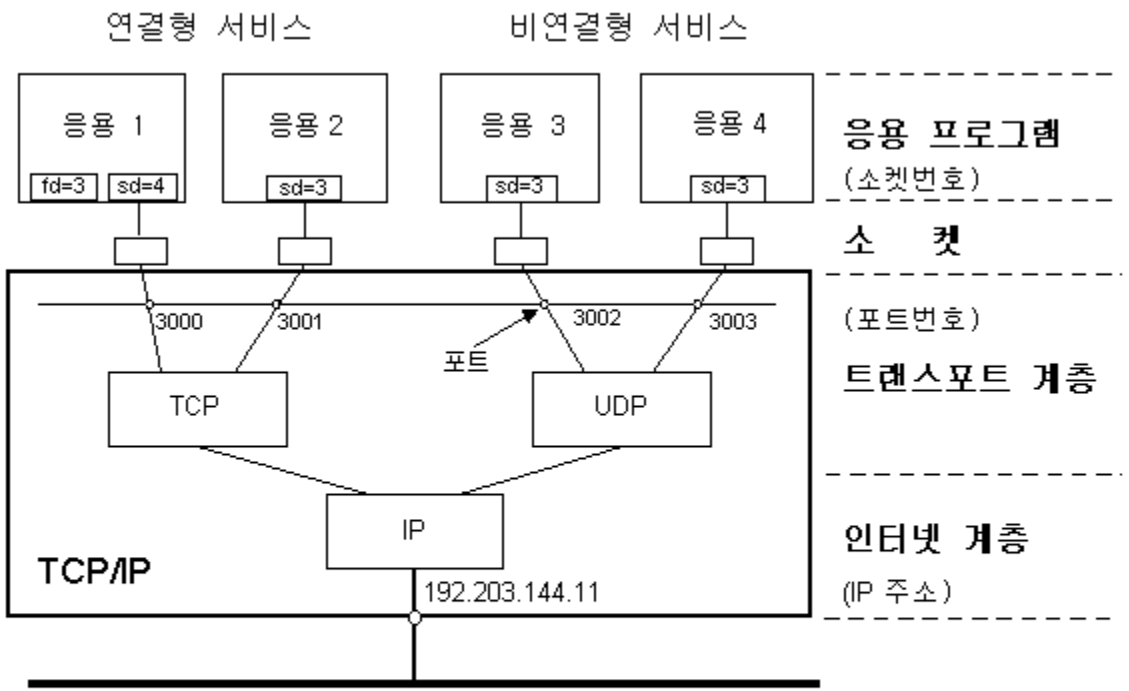


통신연관(Association)을 위한 5-Tuple

1. 프로토콜(TCP, UDP)
2. 자신의 IP 주소
3. 자신의 Port 번호
4. 상대방 IP 주소
5. 상대방 Port 번호

2. Network Socket

2.4 5-Tuple

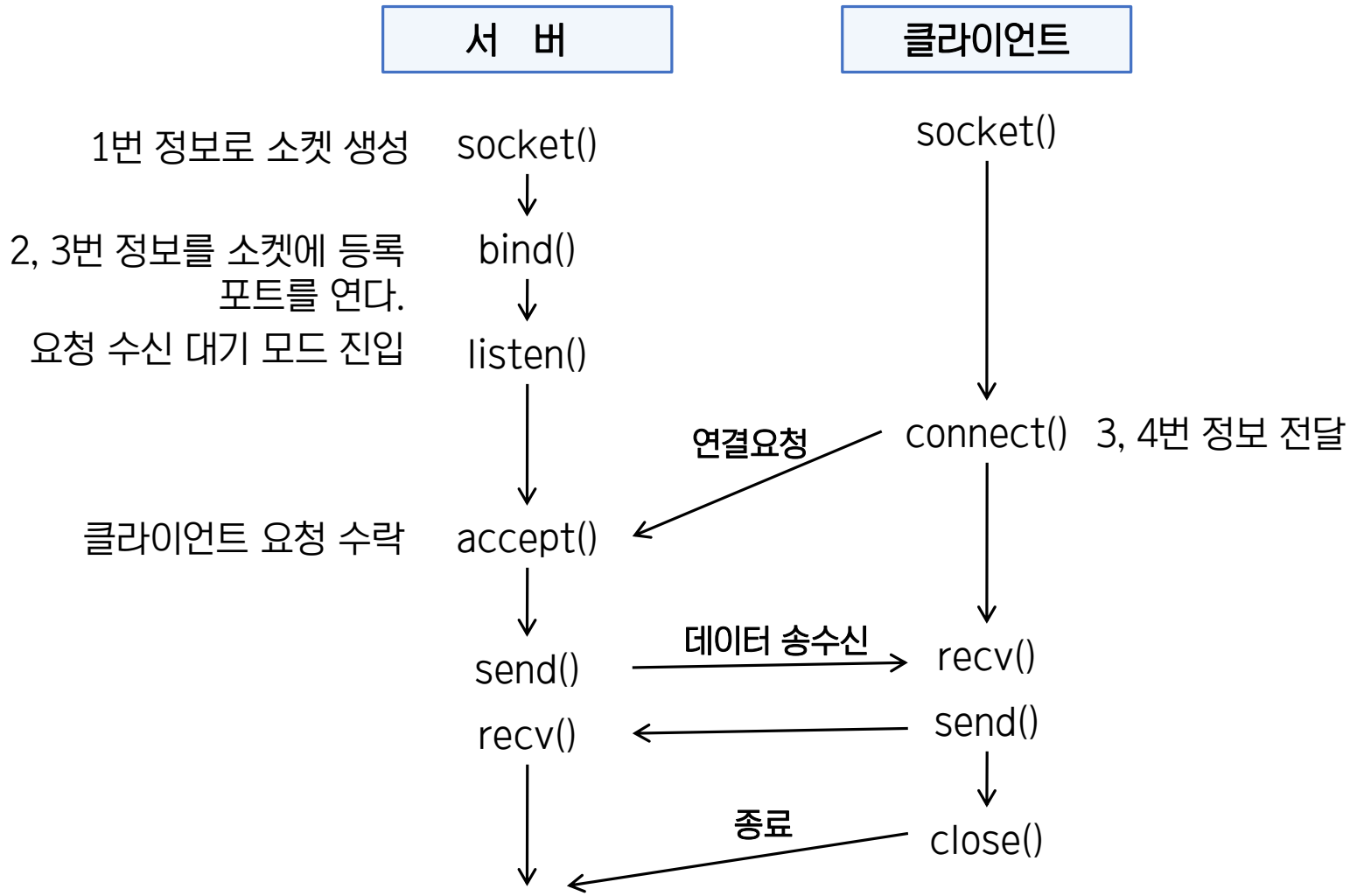


IP : 호스트를 식별

Port : 호스트 내 응용을 식별

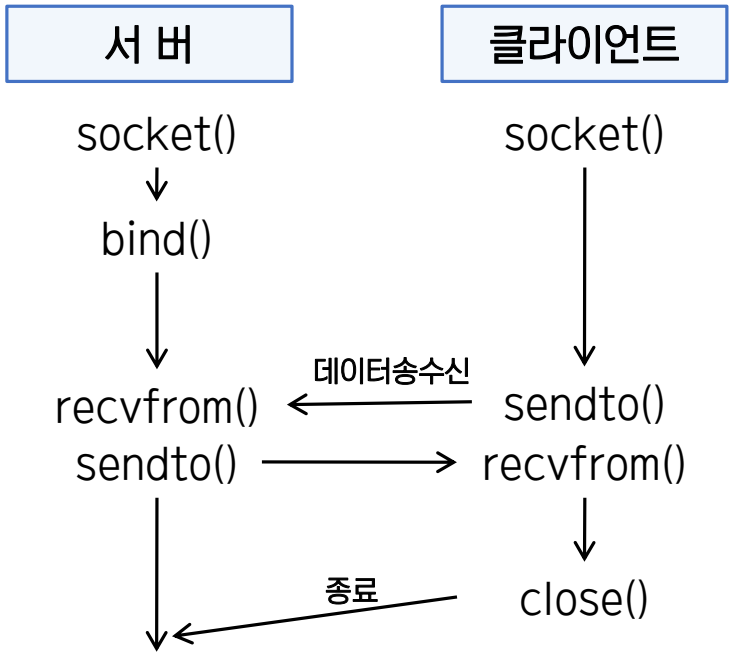
Socket 번호 : 응용 내에서 논리적 연결(5-Tuple) 식별

TCP Socket API 호출 순서



UDP Socket API 호출 순서

- ❖ TCP와 달리 1대1 통신에만 사용되지는 않는다
- ❖ socket 생성 후 연결절차 없이 바로 통신가능



2. Network Socket

2.7 TCP Socket Example(Server)

```python
## server.py

import socket
import argparse


def run_server(port=4000):
    host = '' ## 127.0.0.1 Loopback

    with socket.socket(family=socket.AF_INET, type=socket.SOCK_STREAM) as s:
        s.bind((host, port))
        s.listen(1) ## max 1 client

        conn, addr = s.accept()
        msg = conn.recv(1024)
        print(msg.decode()) ## msg is a binary data, so we need to decode it

        conn.sendall(msg)
        conn.close()

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description="Echo server -p port")
    parser.add_argument('-p', help="port_number", required=True)

    args = parser.parse_args()
    run_server(port=int(args.p))
```

2. Network Socket

2.7 TCP Socket Example(Client)

```python
## client.py

import socket
import argparse

def run(host, port):
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((host, port))
        line = input(':')
        s.sendall(line.encode())
        resp = s.recv(1024)
        print(resp.decode())

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description="Echo client -p port -i host")
    parser.add_argument('-p', help="port_number", required=True)
    parser.add_argument('-i', help="host_name", required=True)

    args = parser.parse_args()
    run(host=args.i, port=int(args.p))
```

2. Network Socket

2.7 TCP Socket Example

```
root@ubuntu:/home/famous/Desktop/network/socket/assignment/assignment_2# python3 server.py -p 1112
Hello world!!
root@ubuntu:/home/famous/Desktop/network/socket/assignment/assignment_2#
```

```
root@ubuntu:/home/famous/Desktop/network/socket/assignment/assignment_2# python3 client.py -i 127.0.0.1 -p 1112
:Hello world!!
Hello world!!
root@ubuntu:/home/famous/Desktop/network/socket/assignment/assignment_2#
```

Assignment #2

- 클라이언트가 보낸 문자열을 거꾸로 전송해주는 서버 구현
  - python client.py –i 127.0.0.1 –p 8888 -s
    - -i : 서버 아이피, -p : 포트번호
    - -s 보낼 문자열
  - python server.py –p 8888
    - -p 포트번호
  - 팀 대표가 barcel@naver.com으로 제출 (3.19일까지)
  - Title : [컴퓨터네트워크][학번][이름][과제_N]
  - Content : github repo url

    팀명 : 길동이네

    팀원 : 홍길동(학번), 고길동(학번)

Q & A