

PYTHON SOCKET MODULE

Moon Yong Joon

SOCKET

기본



Socket

Socket 이란

Socket이란 양방향 통신채널(endpoint)이고,
Sockets은 프로세스간, 머신들간 등의 통신을 지원

Term	Description
domain	Transport 메커니즘에 사용하는 Protocol Family AF_INET, PF_INET, PF_UNIX, PF_X25 등
type	2개의 endpoint 사이의 커뮤니케이션 타입. - SOCK_STREAM : connection-oriented protocols(TCP) - SOCK_DGRAM : connectionless protocols.(UDP)
protocol	a domain and type 내의 다양한 protocol를 의미. 기본값 0
hostname	실제 서버 네임 및 주소(DNS, IP)
port	각 서버가 서비스를 처리하기 위한 주소

Socket 종류

- SOCKET STREAM : SOCK_STREAM

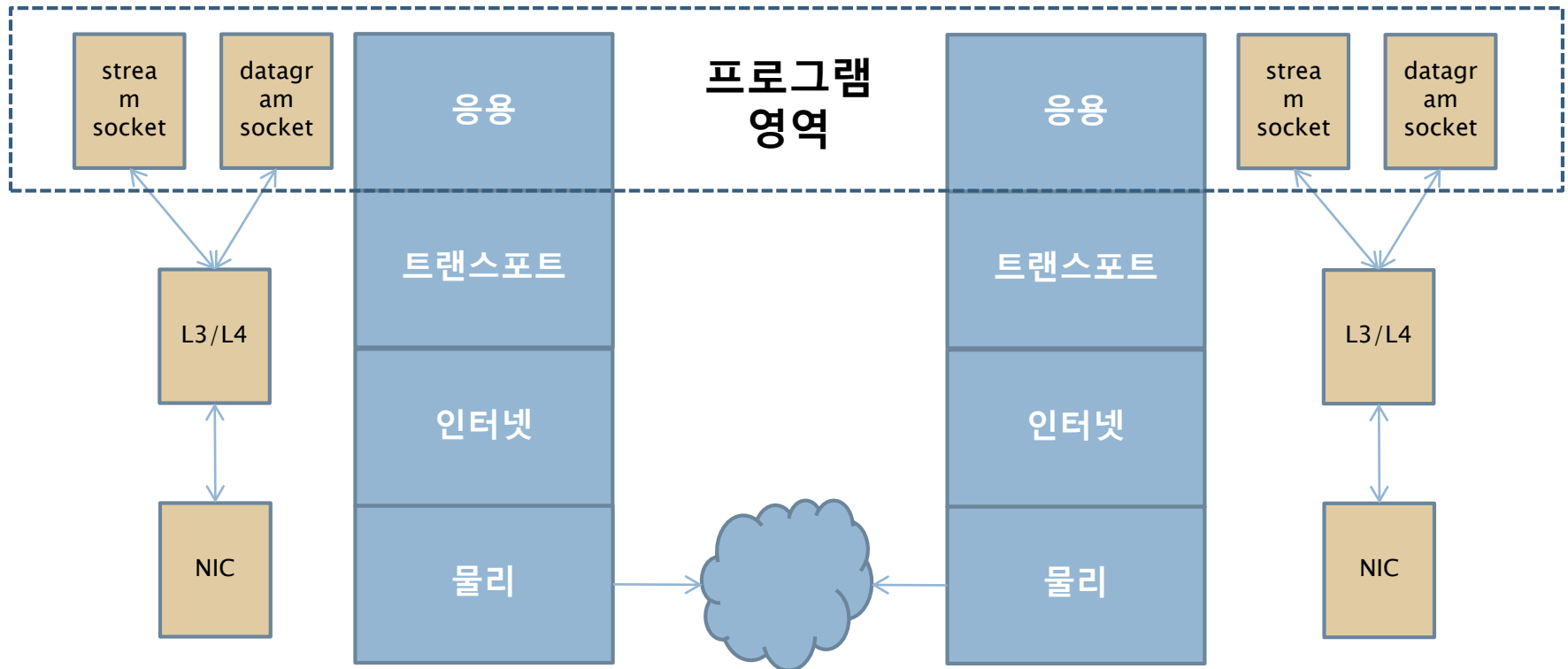
TCP 전송 계층 프로토콜을 사용하여 통신하는 소켓 연결-지향 형태를 지원

- SOCKET DGRAM : SOCK_DGRAM

UDP 전송 계층 프로토콜을 사용하여 통신하는 소켓 신뢰적이지 못한 데이터그램 형태를 지원

Socket 구조

Socket 프로그램 구조



Socket 생성

Socket 객체를 생성하기 위해서는 도메인, 타입, 프로토콜을 파라미터로 받아서 생성

```
s = socket.socket (socket_family, socket_type, protocol=0)
```

domain → socket_family: AF_UNIX or AF_INET

type → socket_type: SOCK_STREAM or SOCK_DGRAM.

protocol: default는 0.

AF_INET : IP version 4 or IPv4

SOCK_STREAM : TCP protocol

Client Socket 연결

클라이언트에서 서버 연결

```
s.connect((TCP_IP, TCP_PORT))
```

Method	Description
s.connect()	TCP server 연결하기 위해 파라미터로 서버의 IP 주소와 Port를 넘김

Server Socket 연결

서버 내의 socket 활성화 및 클라이언트 연결

Method	Description
s.bind()	TCP 서버 연결 s.bind((TCP_IP, TCP_PORT))
s.listen()	클라이언트에서 이벤트 요청을 위한 TCP listener 시작 s.listen(1)
s.accept()	TCP client 연결, 클라이언트 연결이 올 때까지 대기(blocking). conn, addr = s.accept()

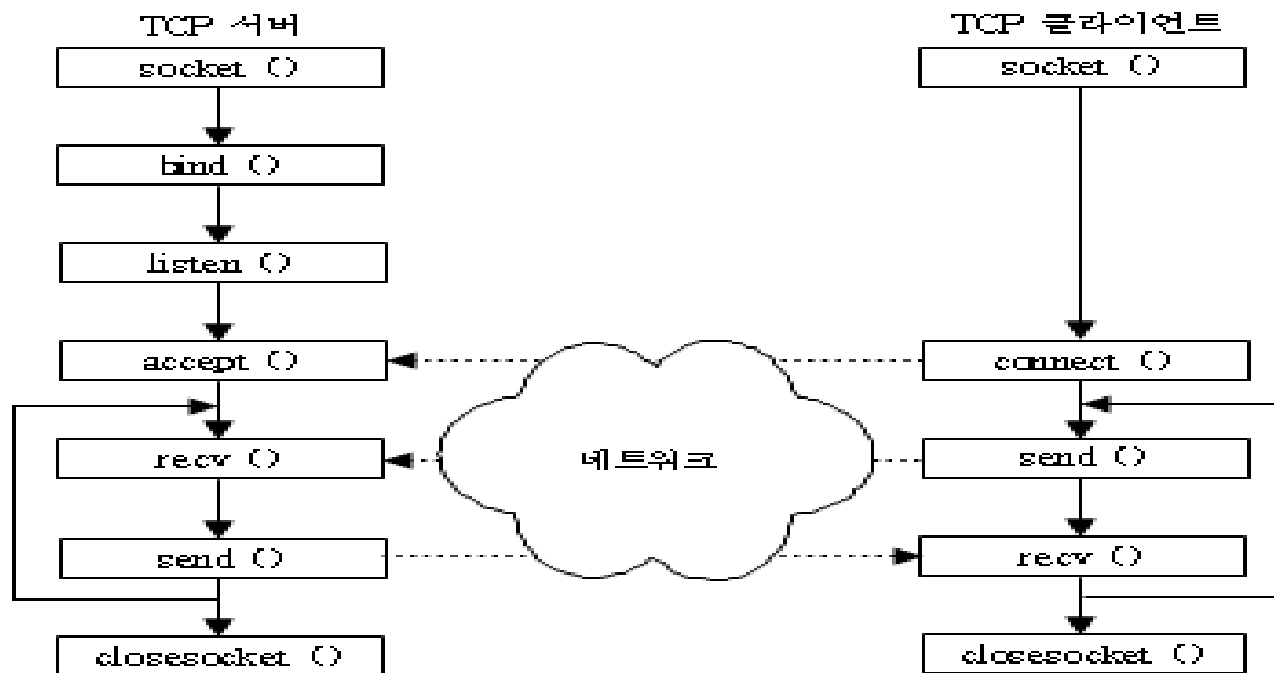
Socket간 메시지 송수신

클라이언트와 서버간 TCP/UDP 메시지 송수신

Method	Description
s.recv()	수신 TCP message : data = s.recv(BUFFER_SIZE)
s.send()	송신 TCP message : s.send(MESSAGE)
s.sendall()	송신 TCP message : s.sendall(MESSAGE)
s.recvfrom()	수신 UDP message
s.sendto()	송신 UDP message
s.close()	socket 클로징

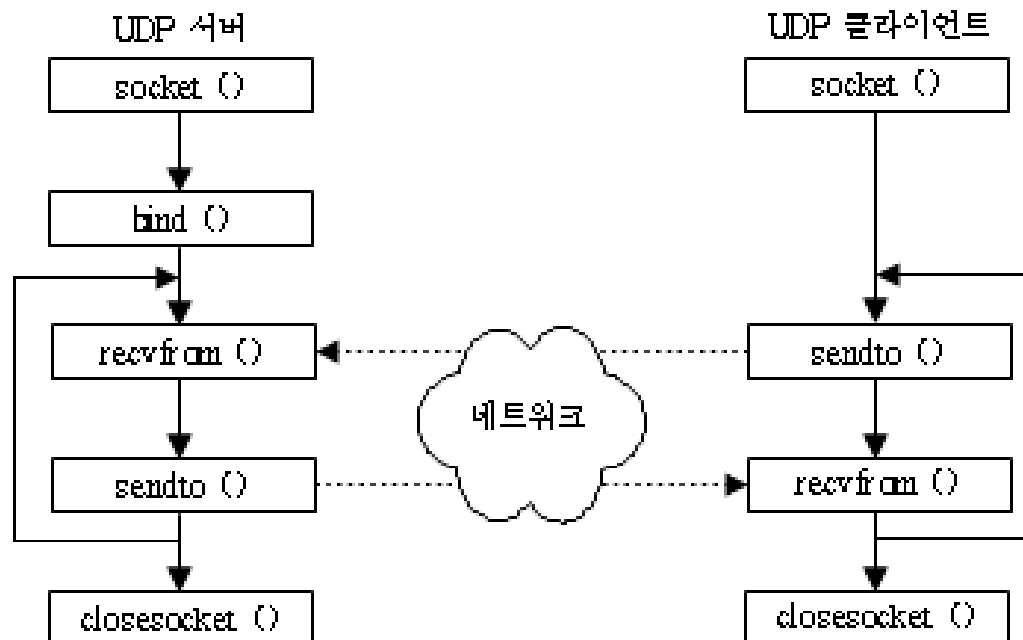
TCP : SOCK_STREAM

TCP 즉 연결-지향 (Connection-oriented)에 대한 클라이언트와 서버간 메시지 송수신



UDP : SOCK_DGRAM

UDP 즉 비연결(Connectionless)에 대한 클라이언트와 서버간 메시지 송수신



Blocking & Non-Blocking

Socket 처리는 기본 Blocking 처리

- 어떤 일이 일어나기를 기다리면서 멍하니 있는 상태
- 기본값 : `socket.setblocking(1)` ➔ `socket.settimeout(None)`

Non-blocking 처리 : flag 인자가 0

`socket.setblocking(0)` ➔ `socket.settimeout(0.0)`

Blocking 처리 : flag 인자가 1

`socket.setblocking(1)` ➔ `socket.settimeout(100)`



Socket 서버 정보 검색

Socket 함수

서버에 대한 host 이름이나 ip 주소 검색

Method	Description
<code>socket.gethostbyname(obj)</code>	DNS로 IP 주소 가져오기
<code>socket.gethostname()</code>	내부 및 외부 서버 내의 DNS 나 서버 네임 가져오기
<code>socket.getservbyport(obj,'tcp')</code>	특정 port가 처리하는 서비스 네임 가져오기

Hostname/ipaddress 검색(1)

파이썬 함수는 인자와 결과값에 대한 타입정보를 3.5버전 이상부터 hint로 추가되어 결과값에 대한 확인을 별도의 함수로 작성하여 확인

```
# return_check.py
type_str =
['str','int','float','list','dict','function','object']

def return_type(obj) :

    type_check = obj.__class__.__name__
    if type_check in type_str :
        return True, type_check
    else :
        return False, type_check
```


Hostname/ipaddress 검색(2)

Hostname을 가지고 ip address 검색하는 함수 정의

```
# socket_test.py
import socket
import return_check as ret

def get_ipaddress(obj) :
    """
    get ip address
    """
    print " host name :", obj

    ip_addr = socket.gethostbyname(obj)

    print ret.return_type(ip_addr)
    print " ip address :", ip_addr
```

Hostname/ipaddress 검색(3)

자신의 서버 및 remote 검색하기

```
# socket_test.py
```

```
# 자신의 PC hostname 가져오기  
obj = socket.gethostname()  
print ret.return_type(obj)  
get_ipaddress(obj)
```

```
# localhost  
obj = 'localhost'  
get_ipaddress(obj)
```

```
# python org  
obj = 'www.python.org'  
get_ipaddress(obj)
```

```
(True, 'str')  
host name : Moon  
(True, 'str')  
ip address : xxx.xxx.xxx.xxx  
host name : localhost  
(True, 'str')  
ip address : 127.0.0.1  
host name : www.python.org  
(True, 'str')  
ip address : 103.245.222.223
```

외부ip 호출하여 client연결

구글을 검색해서 클라이언트 서버 생성

```
import socket # for socket
import sys

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    print "Socket successfully created"
except socket.error as err:
    print "socket creation failed with error %s" %(err)

# default port for socket
port = 80

try:
    host_ip = socket.gethostbyname('www.google.com')
except socket.gaierror:
    # this means could not resolve the host
    print "there was an error resolving the host"
    sys.exit()

# connecting to the server
s.connect((host_ip,port))

print "the socket has successfully connected to google \
on port == %s" %(host_ip)
```



Port Protocol 정보 조회

세부 서비스 프로토콜

어플리케이션 프로토콜 및 파이선 모듈

Protocol	Common function	Port No	Python module
HTTP	Web pages	80	httplib, urllib, xmlrpclib
NNTP	Usenet news	119	nntplib
FTP	File transfers	20	ftplib, urllib
SMTP	Sending email	25	smtplib
POP3	Fetching email	110	poplib
IMAP4	Fetching email	143	imaplib
Telnet	Command lines	23	telnetlib
Gopher	Document transfers	70	gopherlib, urllib

Port별 서비스 검색

TCP 내의 port별 프로토콜 서비스를 검색

```
# socket_test.py
def get_service(obj) :
    service = socket.getservbyport(obj,'tcp')
    print " port : " + str(obj) + " service name : " + service

print ' get port '
get_service(80)
get_service(53)
get_service(25)
```

```
#결과
get port
port : 80 service name : http
port : 53 service name : domain
port : 25 service name : smtp
```

SOCKET

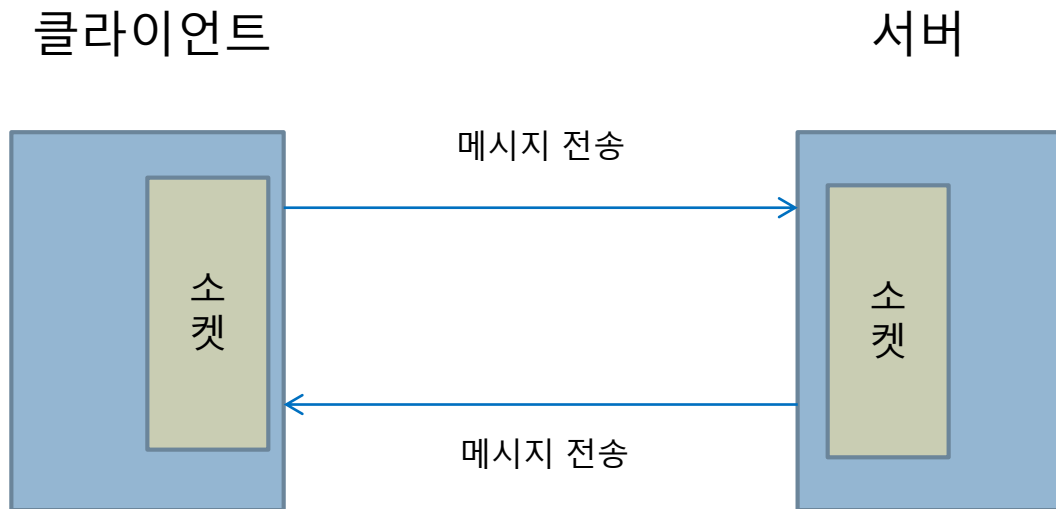
생성



Socket 생성 기초

Echo 통신처리 흐름

클라이언트에서 서버로 전송하면 그대로 전달하는 통신을 처리



서버 생성

내부의 서버를 가지고 소켓서버 생성하여 처리

```
import socket

HOST = 'localhost'          # Symbolic name meaning all available
                              interfaces
PORT = 50007                # Arbitrary non-privileged port
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(1)
while 1:                    # 서버 순환
    conn, addr = s.accept()  # 클라이언트 연결
    print 'Connected by', addr

    while 1 :               # 클라이언트 순환
        data = conn.recv(1024)
        if not data: break   # 클라이언트로 부터 수신
        conn.send(data)      # 클라이언트에 송신
    conn.close()
    break
s.close()
```

클라이언트 생성

내부의 서버를 가지고 소켓서버 생성하여 연결 처리

```
# Echo client program
import socket

HOST = 'localhost'          # The remote host
PORT = 50007                # The same port as used by the server
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))

while 1 :
    data = raw_input('> ')   #데이터를 입력창에서 받음
    if not data: break
    s.send(data)             # 서버에 데이터 전송
    data = s.recv(1024)      # 서버로부터 데이터 수신
    if not data: break
    print 'Received', repr(data)
s.close()
```



Socket Exception

Socket exception 처리

Socket 모듈 내의 error에 exception 사용

```
import socket #for sockets
import sys #for exit

try:
    #create an AF_INET, STREAM socket (TCP)
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
except socket.error, msg:
    print 'Failed to create socket. Error code: ' + str(msg[0]) + ' , Error message : ' + msg[1]
    sys.exit();

print 'Socket Created'
```

Remote 서버 검색

참조문서 : <http://www.binarytides.com/python-socket-programming-tutorial/>

Client socket 생성

내 서버에 socket 생성

```
#client_remote_test.py

import socket #for sockets
import sys #for exit

#Send some data to remote server
message = "GET / HTTP/1.1\r\n\r\n"

try :
    #Set the whole string
    s.sendall(message)
except socket.error:
    #Send failed
    print 'Send failed'
    sys.exit()

print 'Message send successfully'
```

Client에서 외부 서버 연결

Google 서버에 연결

```
#client_remote_test.py

host = 'www.google.com'
port = 80

try:
    remote_ip = socket.gethostbyname( host )
except socket.gaierror:
    #could not resolve
    print 'Hostname could not be resolved. Exiting'
    sys.exit()

print 'Ip address of ' + host + ' is ' + remote_ip

#Connect to remote server
s.connect((remote_ip , port))

print 'Socket Connected to ' + host + ' on ip ' + remote_ip
```


Remote 연결 실행 결과

클라이언트 socket을 생성하고 remote 서버인 www.google.com 으로 접속

```
C:\myPython\socket_test>python client_remote_test.py
Socket Created
Ip address of www.google.com is 216.58.197.100
Socket Connected to www.google.com on ip 216.58.197.100
```

Client : 외부서버 메시지 전송

Google 서버로 Http 메시지 전송

```
#client_remote_test.py

#Send some data to remote server
message = "GET / HTTP/1.1\r\n\r\n"

try :
    #Set the whole string
    print 'send message ', message
    s.sendall(message)
except socket.error:
    #Send failed
    print 'Send failed'
    sys.exit()

print 'Message send successfully'
```

Client : 외부서버 메시지 수신

Google 서버에서 http 메시지 수신

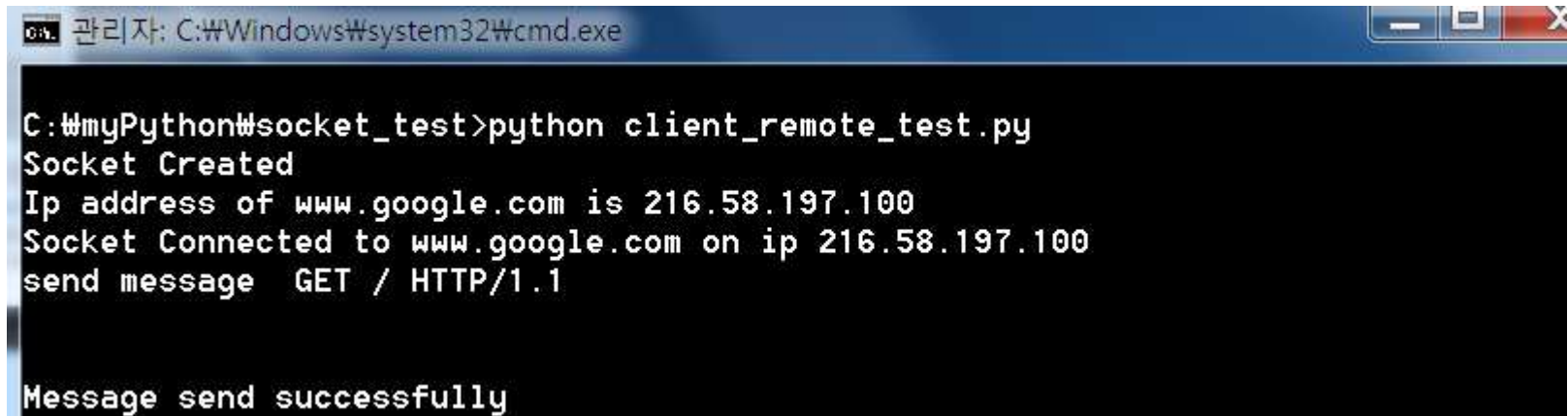
```
#client_remote_test.py
```

```
#Now receive data  
reply = s.recv(4096)
```

```
print 'receive message '  
print reply
```

http 메시지 실행 결과 - 송신

www.google.com 으로 접속하여 get method
로 메시지 전송 및 수신



```
관리자: C:\Windows\system32\cmd.exe

C:\myPython\socket_test>python client_remote_test.py
Socket Created
Ip address of www.google.com is 216.58.197.100
Socket Connected to www.google.com on ip 216.58.197.100
send message  GET / HTTP/1.1

Message send successfully
```

http 메시지 실행 결과 - 수신

www.google.com 으로 접속하여 get method
로 메시지 전송 및 수신

관리자: C:\windows\system32\cmd.exe

```
receive message
HTTP/1.1 302 Found
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Location: http://www.google.co.kr/?gfe_rd=cr&ei=LCCrUv-vEs_C8gfSzoyAAQ
Content-Length: 261
Date: Fri, 29 Jan 2016 08:17:48 GMT
Server: GFE/2.0

<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.co.kr/?gfe_rd=cr&ei=LCCrUv-vEs_C8gfSzoyAAQ">here<
/A>.
</BODY></HTML>
```



Socket Exception

Socket exception

Socket Exception 속성

속성	Description
socket.error	소켓관련 에러 처리
socket.herror	주소관련 에러 gethostbyname_ex() and gethostbyaddr()
socket.gaierror	주소관련 에러 getaddrinfo() and getnameinfo()
socket.timeout	타임아웃 발생에러 settimeout()

Socket 오류: 기본

Socket 생성에 대한 예러 처리

```
# GetRemortIP.py
# Socket 생성

import socket # for socket
import sys

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    print "Socket successfully created"
except socket.error as err:
    print "socket creation failed with error %s" %(err)

# default port for socket
port = 80
```


Socket 오류: gaierror

Google 서버에서 접속시 에러 처리

```
# GetRemortIP.py
# google 주소 검색

try:
    host_ip = socket.gethostbyname('www.googlx.co')
except socket.gaierror, e:
    # this means could not resolve the host
    print "there was an error resolving the host",e
    sys.exit()

# connecting to the server
s.connect((host_ip,port))

print "the socket has successfully connected to google \
on port == %s" %(host_ip)
```

DSN 이름을 잘
못 입력해서 오
류 발생

Socket 오류: timeout(1)

Socket 생성후에 타임아웃 정의

```
import socket
import sys

TCP_IP = '127.0.0.1'
TCP_PORT = 51874
BUFFER_SIZE = 1024

def test_socket_modes() :
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setblocking(1)
    s.settimeout(0.5)
    s.bind((TCP_IP,TCP_PORT))
    s.listen(1)
    while 1 :
        client, address = s.accept()
        while 1 :
            data = client.recv(1024)
            if not data: break
            client.send(data)
        client.close()
        break
    s.close()
```

타임아웃 세팅해서
타임아웃초과시 에
러 처리

Socket 오류: timeout(2)

Socket 실행시 타임아웃 처리

```
if __name__ == "__main__":  
    try:  
        test_socket_modes()  
    except socket.timeout, e:  
        print "timeout :", e  
        sys.exit()
```

```
C:\myPython\socket_test> python socket_timeout.py  
socket : ('127.0.0.1', 51874)  
timeout : timed out  
C:\myPython\socket_test>
```

타임아웃 세팅해서
타임아웃초과시 예
러 처리