Computer Network Laboratory

Basic Network Programming (I)

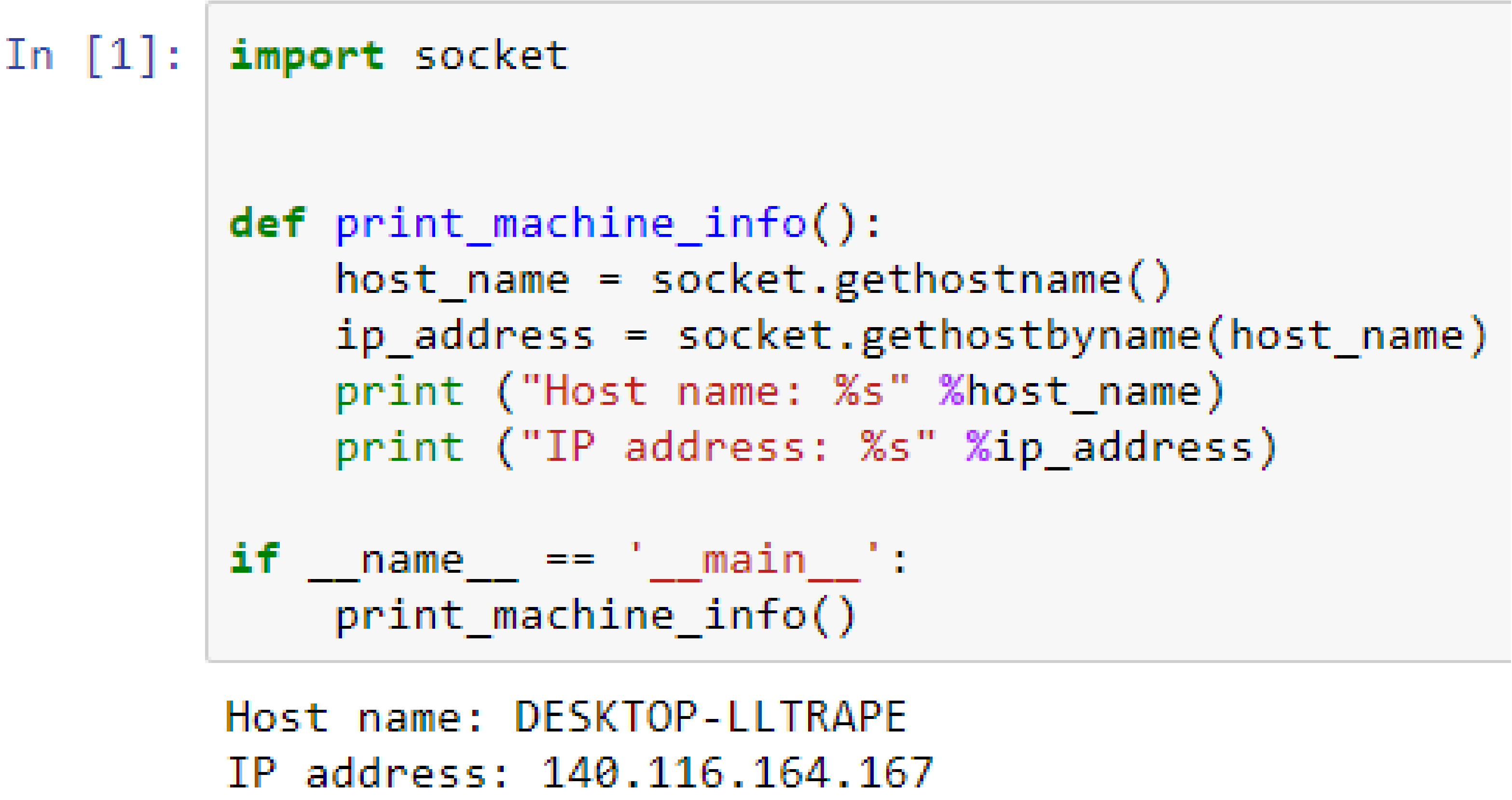
Jiawei Chang

Dept. of Computer Science and Information Engineering
National Taichung University of Science and Technology

Outline

1. local_machine_info
2. remote_machine_info
3. ip4_address_conversion
4. finding_service_name
5. integer_conversion
6. socket_timeout
7. socket_errors

local_machine_info

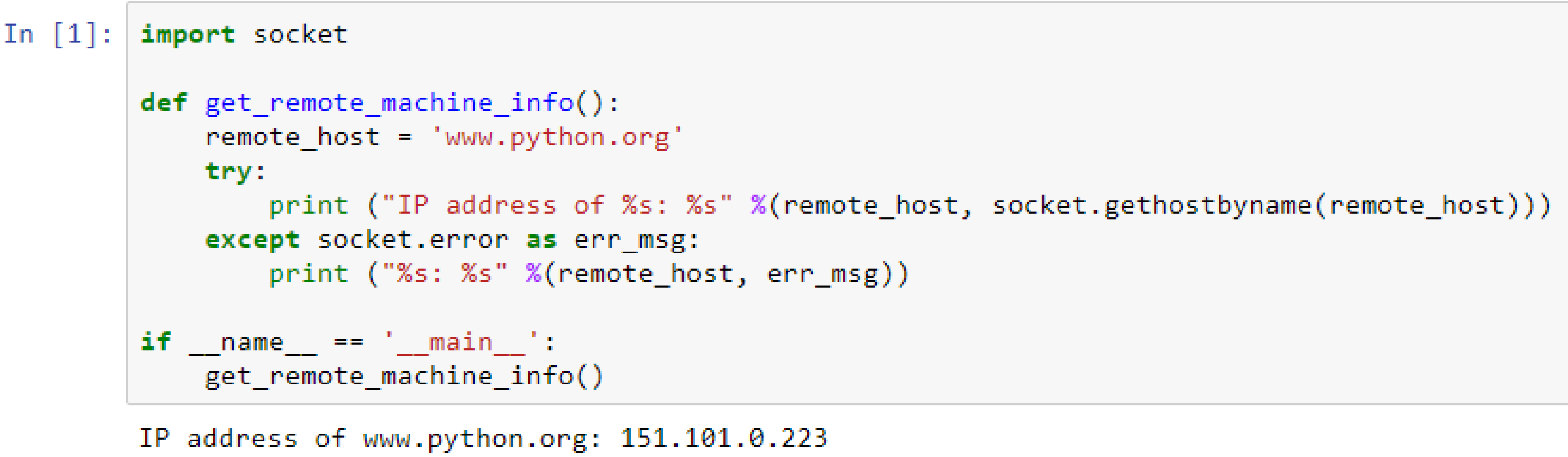


In [1]:

```
import socket


def print_machine_info():
    host_name = socket.gethostname()
    ip_address = socket.gethostbyname(host_name)
    print ("Host name: %s" %host_name)
    print ("IP address: %s" %ip_address)

if __name__ == '__main__':
    print_machine_info()
```

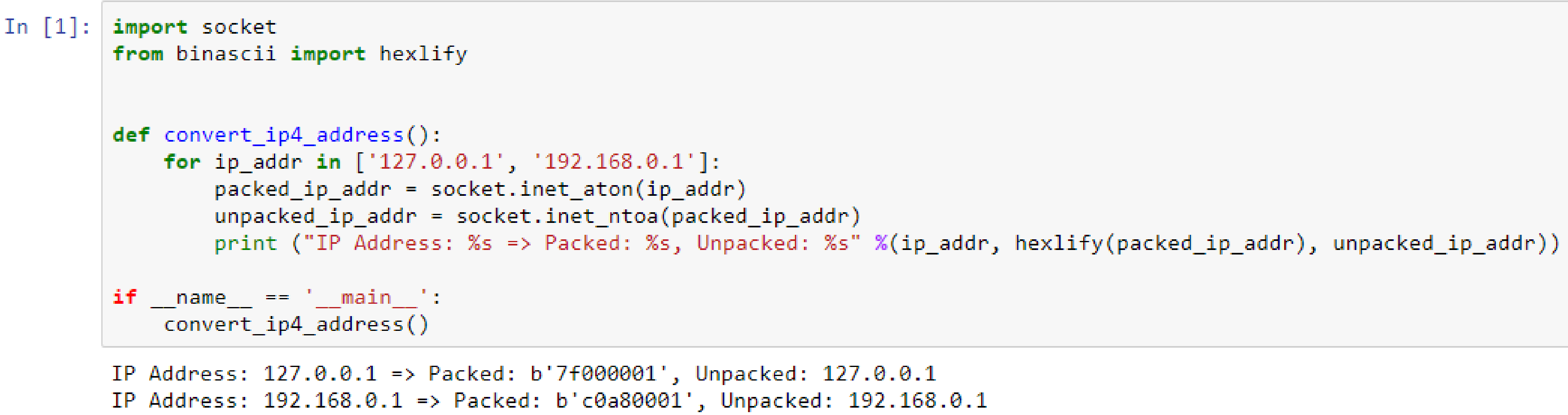
```
Host name: DESKTOP-LLTRAPE
IP address: 140.116.164.167
```

remote_machine_info



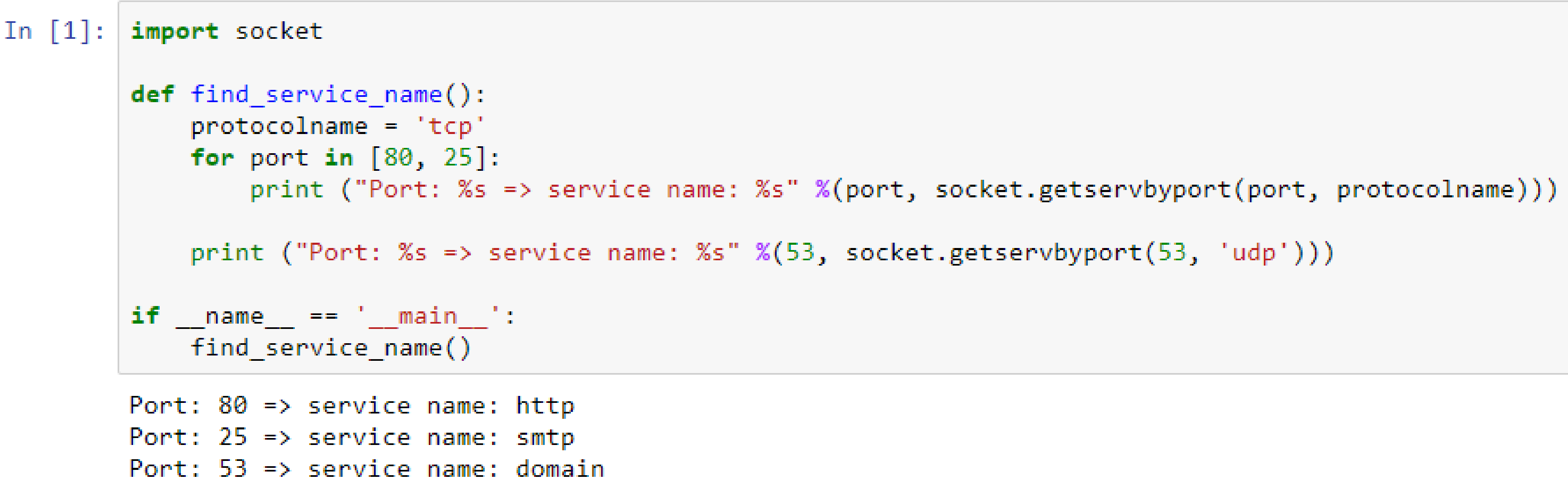
```
In [1]: import socket

        def get_remote_machine_info():
            remote_host = 'www.python.org'
            try:
                print ("IP address of %s: %s" %(remote_host, socket.gethostbyname(remote_host)))
            except socket.error as err_msg:
                print ("%s: %s" %(remote_host, err_msg))

        if __name__ == '__main__':
            get_remote_machine_info()

        IP address of www.python.org: 151.101.0.223
```

ip4_address_conversion



```
In [1]: import socket
        from binascii import hexlify


        def convert_ip4_address():
            for ip_addr in ['127.0.0.1', '192.168.0.1']:
                packed_ip_addr = socket.inet_aton(ip_addr)
                unpacked_ip_addr = socket.inet_ntoa(packed_ip_addr)
                print ("IP Address: %s => Packed: %s, Unpacked: %s" %(ip_addr, hexlify(packed_ip_addr), unpacked_ip_addr))

        if __name__ == '__main__':
            convert_ip4_address()

        IP Address: 127.0.0.1 => Packed: b'7f000001', Unpacked: 127.0.0.1
        IP Address: 192.168.0.1 => Packed: b'c0a80001', Unpacked: 192.168.0.1
```

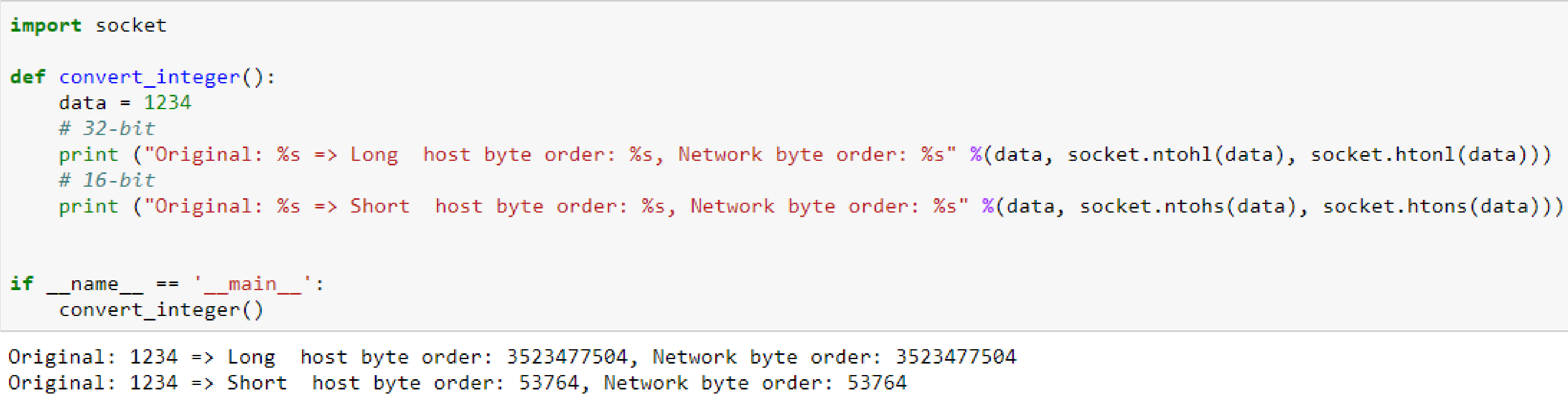
finding_service_name



```
In [1]: import socket

        def find_service_name():
            protocolname = 'tcp'
            for port in [80, 25]:
                print ("Port: %s => service name: %s" %(port, socket.getservbyport(port, protocolname)))

            print ("Port: %s => service name: %s" %(53, socket.getservbyport(53, 'udp')))

        if __name__ == '__main__':
            find_service_name()
```

```
Port: 80 => service name: http
Port: 25 => service name: smtp
Port: 53 => service name: domain
```

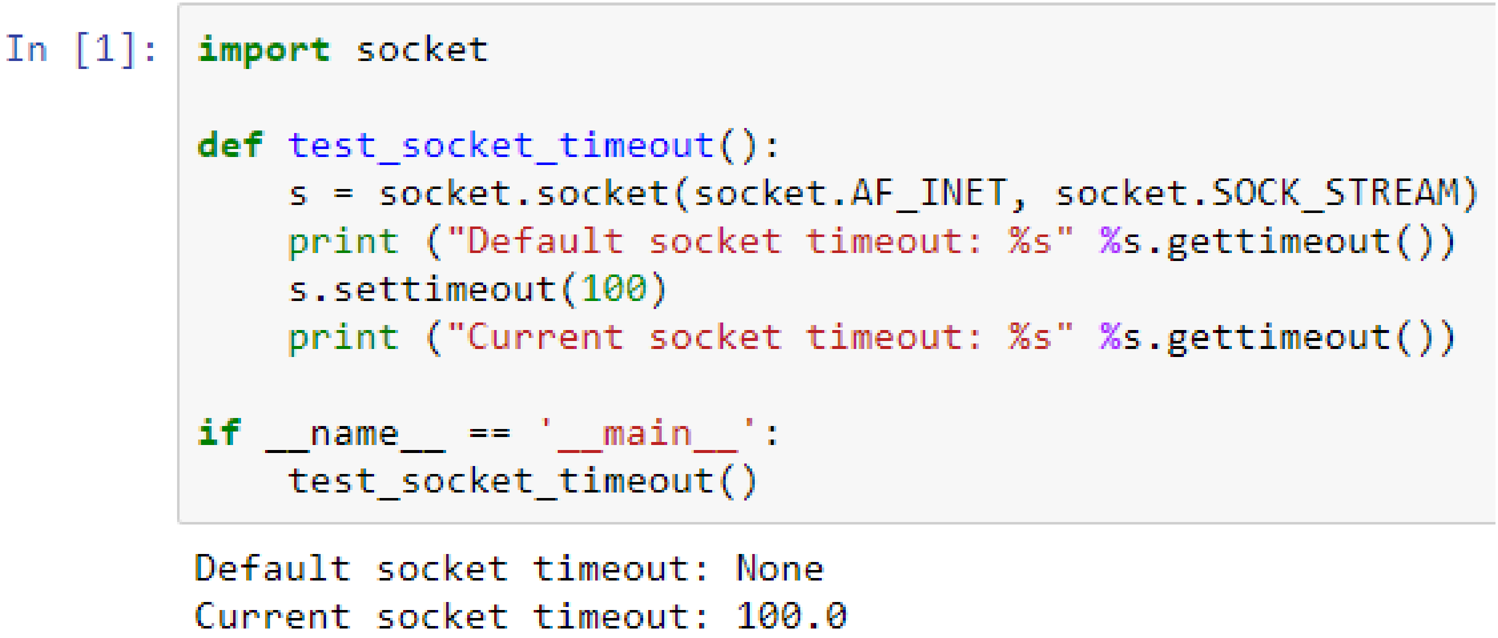
integer_conversion



```python
import socket

def convert_integer():
    data = 1234
    # 32-bit
    print ("Original: %s => Long  host byte order: %s, Network byte order: %s" %(data, socket.ntohl(data), socket.htonl(data)))
    # 16-bit
    print ("Original: %s => Short  host byte order: %s, Network byte order: %s" %(data, socket.ntohs(data), socket.htons(data))


if __name__ == '__main__':
    convert_integer()
```

```
Original: 1234 => Long  host byte order: 3523477504, Network byte order: 3523477504
Original: 1234 => Short  host byte order: 53764, Network byte order: 53764
```

socket_timeout

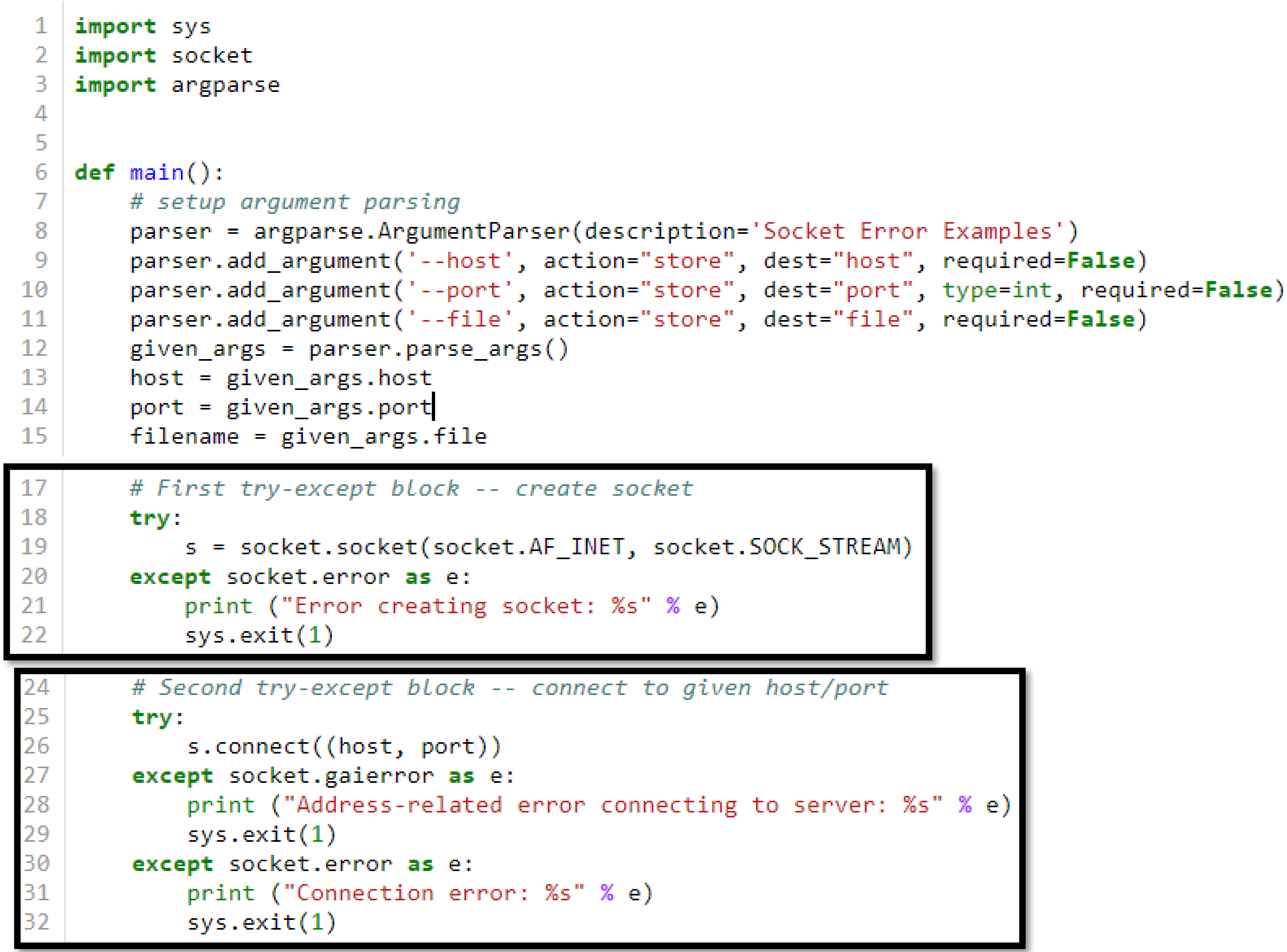


In [1]:

```
import socket

def test_socket_timeout():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    print ("Default socket timeout: %s" %s.gettimeout())
    s.settimeout(100)
    print ("Current socket timeout: %s" %s.gettimeout())

if __name__ == '__main__':
    test_socket_timeout()
```

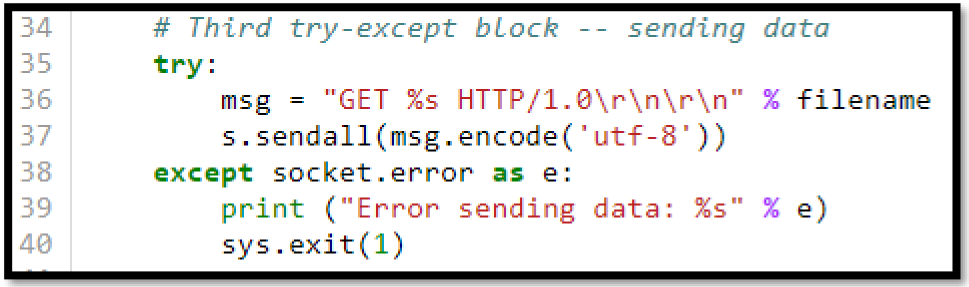
```
Default socket timeout: None
Current socket timeout: 100.0
```

socket_errors (I)

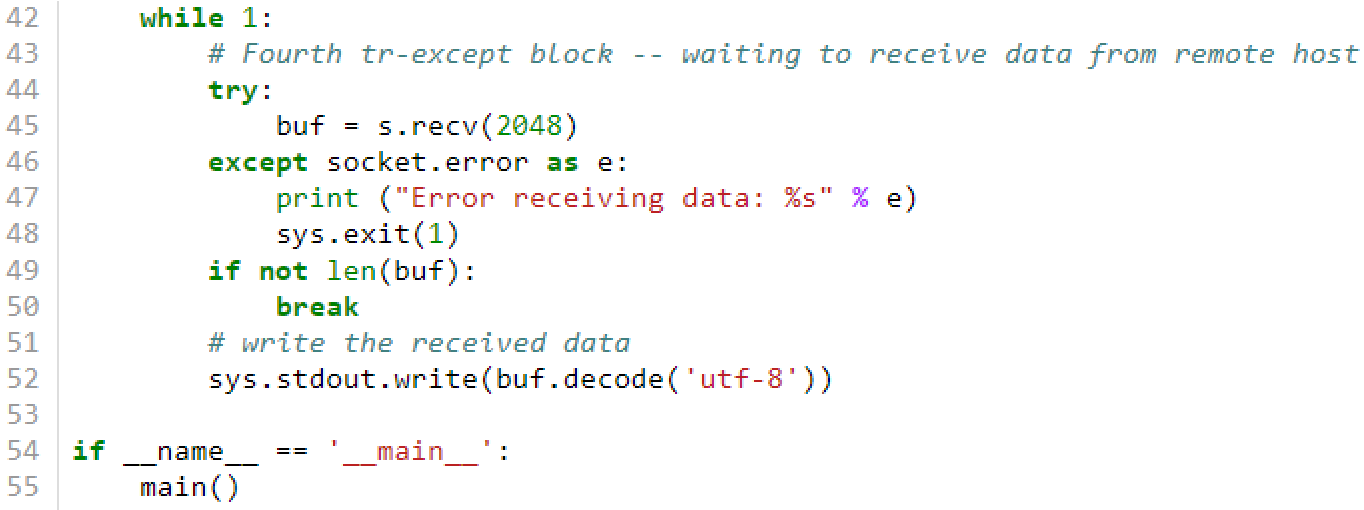


```python
import sys
import socket
import argparse


def main():
    # setup argument parsing
    parser = argparse.ArgumentParser(description='Socket Error Examples')
    parser.add_argument('--host', action="store", dest="host", required=False)
    parser.add_argument('--port', action="store", dest="port", type=int, required=False)
    parser.add_argument('--file', action="store", dest="file", required=False)
    given_args = parser.parse_args()
    host = given_args.host
    port = given_args.port
    filename = given_args.file

    # First try-except block -- create socket
    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    except socket.error as e:
        print ("Error creating socket: %s" % e)
        sys.exit(1)

    # Second try-except block -- connect to given host/port
    try:
        s.connect((host, port))
    except socket.gaierror as e:
        print ("Address-related error connecting to server: %s" % e)
        sys.exit(1)
    except socket.error as e:
        print ("Connection error: %s" % e)
        sys.exit(1)
```

socket_errors (II)



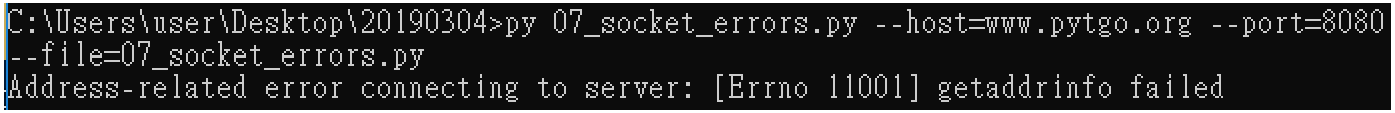
```python
    # Third try-except block -- sending data
    try:
        msg = "GET %s HTTP/1.0\r\n\r\n" % filename
        s.sendall(msg.encode('utf-8'))
    except socket.error as e:
        print ("Error sending data: %s" % e)
        sys.exit(1)
```



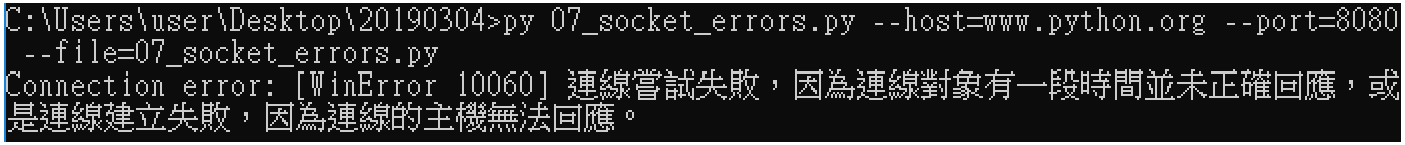
```python
    while 1:
        # Fourth tr-except block -- waiting to receive data from remote host
        try:
            buf = s.recv(2048)
        except socket.error as e:
            print ("Error receiving data: %s" % e)
            sys.exit(1)
        if not len(buf):
            break
        # write the received data
        sys.stdout.write(buf.decode('utf-8'))

if __name__ == '__main__':
    main()
```

socket_errors (III)

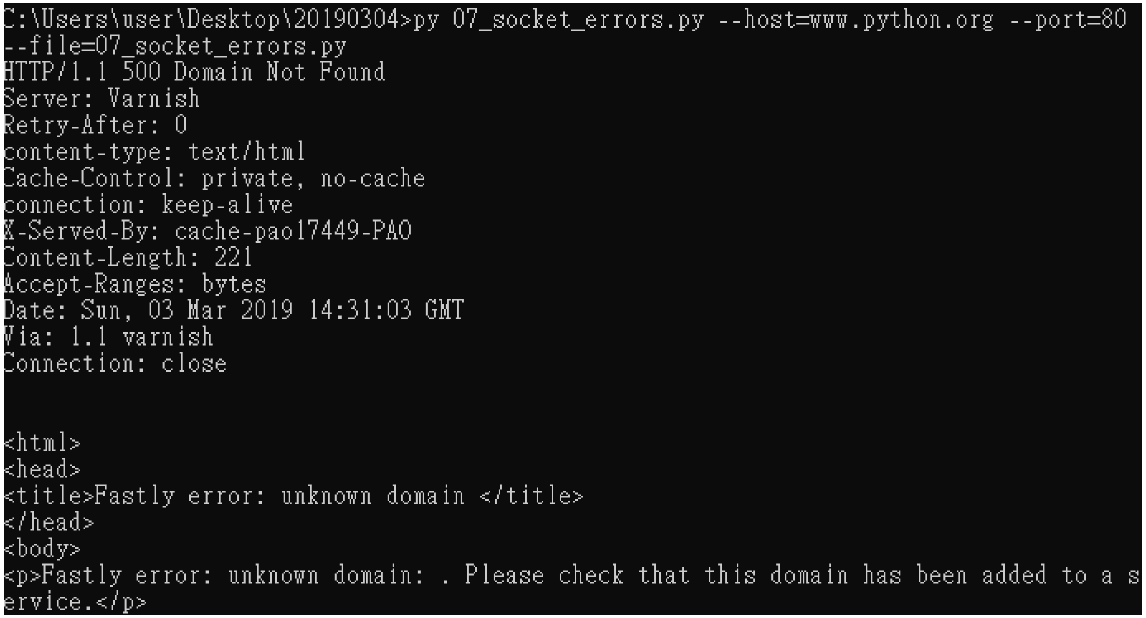
python 07_socket_errors.py --host=<HOST> --port=<PORT> --file=<FILE>



```
C:\Users\user\Desktop\20190304>py 07_socket_errors.py --host=www.pytgo.org --port=8080
--file=07_socket_errors.py
Address-related error connecting to server: [Errno 11001] getaddrinfo failed
```



```
C:\Users\user\Desktop\20190304>py 07_socket_errors.py --host=www.python.org --port=8080
 --file=07_socket_errors.py
Connection error: [WinError 10060] 連線嘗試失敗，因為連線對象有一段時間並未正確回應，或
是連線建立失敗，因為連線的主機無法回應。
```



```
C:\Users\user\Desktop\20190304>py 07_socket_errors.py --host=www.python.org --port=80
--file=07_socket_errors.py
HTTP/1.1 500 Domain Not Found
Server: Varnish
Retry-After: 0
content-type: text/html
Cache-Control: private, no-cache
connection: keep-alive
X-Served-By: cache-pao17449-PAO
Content-Length: 221
Accept-Ranges: bytes
Date: Sun, 03 Mar 2019 14:31:03 GMT
Via: 1.1 varnish
Connection: close


<html>
<head>
<title>Fastly error: unknown domain </title>
</head>
<body>
<p>Fastly error: unknown domain: . Please check that this domain has been added to a s
ervice.</p>
```

延伸閱讀

- Socket Programming in Python (Guide)
  - https://realpython.com/python-sockets/#socket-api-overview

- Python 网络编程
  - http://www.runoob.com/python/python-socket.html

Resource is available by
https://jiaweichang.github.io/biography/

THANKS