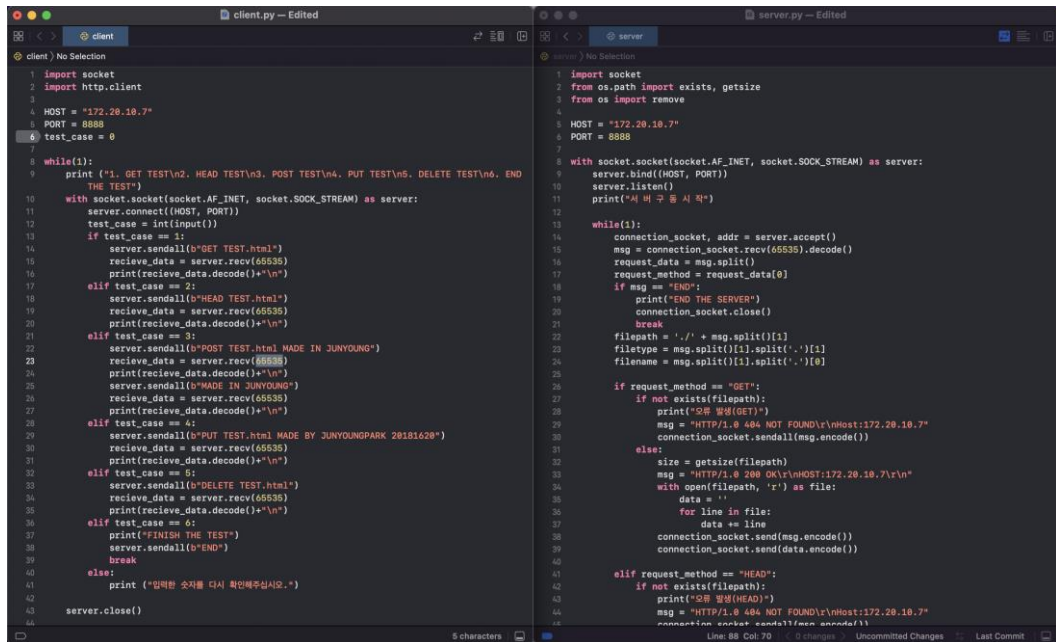


소스파일에는 client.py 파일과 server.py 파일이 있으며



The image shows two side-by-side code editors. The left editor, titled 'client.py - Edited', contains the following Python code:

```
1 import socket
2 import http.client
3
4 HOST = "172.20.10.7"
5 PORT = 8888
6 test_case = 0
7
8 while(1):
9     print ("1. GET TEST\n2. HEAD TEST\n3. POST TEST\n4. PUT TEST\n5. DELETE TEST\n6. END\nTHE TEST")
10    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server:
11        server.connect((HOST, PORT))
12        test_case = int(input())
13        if test_case == 1:
14            server.sendall(b"GET TEST.html")
15            recieve_data = server.recv(65535)
16            print(recieve_data.decode()+"\n")
17        elif test_case == 2:
18            server.sendall(b"HEAD TEST.html")
19            recieve_data = server.recv(65535)
20            print(recieve_data.decode()+"\n")
21        elif test_case == 3:
22            server.sendall(b"POST TEST.html MADE IN JUNYOUNG")
23            recieve_data = server.recv(65535)
24            print(recieve_data.decode()+"\n")
25            server.sendall(b"MADE IN JUNYOUNG")
26            recieve_data = server.recv(65535)
27            print(recieve_data.decode()+"\n")
28        elif test_case == 4:
29            server.sendall(b"PUT TEST.html MADE BY JUNYOUNGPARK 20181628")
30            recieve_data = server.recv(65535)
31            print(recieve_data.decode()+"\n")
32        elif test_case == 5:
33            server.sendall(b"DELETE TEST.html")
34            recieve_data = server.recv(65535)
35            print(recieve_data.decode()+"\n")
36        elif test_case == 6:
37            print("FINISH THE TEST")
38            server.sendall(b"END")
39            break
40        else:
41            print ("입력한 숫자를 다시 확인해주세요.")
42    server.close()
```

The right editor, titled 'server.py - Edited', contains the following Python code:

```
1 import socket
2 from os.path import exists, getsize
3 from os import remove
4
5 HOST = "172.20.10.7"
6 PORT = 8888
7
8 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server:
9     server.bind((HOST, PORT))
10    server.listen()
11    print("서버 구동 시작")
12
13    while(1):
14        connection_socket, addr = server.accept()
15        msg = connection_socket.recv(65535).decode()
16        request_data = msg.split()
17        request_method = request_data[0]
18        if msg == "END":
19            print("END THE SERVER")
20            connection_socket.close()
21            break
22        filepath = './' + msg.split()[1]
23        filetype = msg.split()[1].split('.')[-1]
24        filename = msg.split()[1].split('.')[0]
25
26        if request_method == "GET":
27            if not exists(filepath):
28                print("오류 발생(GET)")
29                msg = "HTTP/1.0 404 NOT FOUND\r\nHost:172.20.10.7"
30                connection_socket.sendall(msg.encode())
31            else:
32                size = getsize(filepath)
33                msg = "HTTP/1.0 200 OK\r\nHost:172.20.10.7\r\n"
34                with open(filepath, 'r') as file:
35                    data = ''
36                    for line in file:
37                        data += line
38                connection_socket.send(msg.encode())
39                connection_socket.send(data.encode())
40
41        elif request_method == "HEAD":
42            if not exists(filepath):
43                print("오류 발생(HEAD)")
44                msg = "HTTP/1.0 404 NOT FOUND\r\nHost:172.20.10.7"
45                connection_socket.sendall(msg.encode())
```

동작 환경의 실행컴퓨터는 MacBook Air M1, 버전은 macOS Monterey 12.3.0 이고 터미널에서 실행했으며

파이썬의 버전은 3.10.4 버전으로 진행하였습니다

```
juniyoung@bagjun-yeong-ui-MacBookAir ~ % ipconfig getifaddr en0
172.20.10.7
```

<pre>import socket import http.client  HOST = "172.20.10.7" PORT = 8888 test_case = 0</pre>	<pre>import socket from os.path import exists, getsize from os import remove  HOST = "172.20.10.7" PORT = 8888</pre>
client.py	server.py

client.py 파일과 server.py 파일에서 HOST는 터미널에서 명령어를 통해 ip를 찾아 입력하였고  
PORT는 8888을 사용하였습니다.

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server:
    server.bind((HOST, PORT))
    server.listen()
    print("START THE SERVER")
```

server.py

bind함수를 통해 입력한 PORT와 HOST로 서버를 엽니다.

```

while(1):
    print ("1. GET TEST\n2. HEAD TEST\n3. POST TEST\n4. PUT TEST\n5. DELETE TEST\n6. END THE TEST")
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server:
        server.connect((HOST, PORT))
        test_case = int(input())

```

client.py

client에서도 connect 함수를 통해 입력한 PORT와 HOST로 서버에 연결합니다

---

```

junyoung@bagjun-yeong-ui-MacBookAir network % python3 server.py
START THE SERVER

```

server.py 실행화면

```

junyoung@bagjun-yeong-ui-MacBookAir network % python3 client.py
1. GET TEST
2. HEAD TEST
3. POST TEST
4. PUT TEST
5. DELETE TEST
6. END THE TEST

```

client.py 실행화면

---

```

while(1):
    connection_socket, addr = server.accept()
    msg = connection_socket.recv(65535).decode()
    request_data = msg.split()
    request_method = request_data[0]
    if msg == "END":
        print("END THE SERVER")
        connection_socket.close()
        break
    filepath = './' + msg.split()[1]
    filetype = msg.split()[1].split('.')[1]
    filename = msg.split()[1].split('.')[0]

```

server.py

while문을 통해 client에게 msg를 끝내기 전까지는 반복적으로 입력 받는 내용입니다.

```

if request_method == "GET":
    if not exists(filepath):
        print("오류 발생(GET)")
        msg = "HTTP/1.0 404 NOT FOUND\r\nHost:172.20.10.7"
        connection_socket.sendall(msg.encode())
    else:
        size = getsize(filepath)
        msg = "HTTP/1.0 200 OK\r\nHOST:172.20.10.7\r\n"
        with open(filepath, 'r') as file:
            data = ''
            for line in file:
                data += line
            connection_socket.send(msg.encode())
            connection_socket.send(data.encode())

```

server.py

Client에게 받은 msg가 GET 일때의 함수 부분입니다.

파일 경로에 파일이 없다면 404 에러를 보내주고, 존재한다면 파일을 읽기전용으로 읽은 후 200 성공과 함께 내용을 보내줍니다.

```

if test_case == 1:
    server.sendall(b"GET TEST.html")
    recieve_data = server.recv(65535)
    print(recieve_data.decode()+"\n")

```

Client.py

Test\_case를 1이라 입력함으로서 서버에게 GET TEST.html을 보내고 데이터를 받아 터미널에 출력해줍니다.

---

```

junyoung@bagjun-yeong-ui-MacBookAir network % python3 server.py
START THE SERVER
오류 발생 (GET)

```

Server.py 실행화면

```
1
HTTP/1.0 404 NOT FOUND
Host:172.20.10.7
```

1. GET TEST
2. HEAD TEST
3. POST TEST
4. PUT TEST
5. DELETE TEST
6. END THE TEST



Client.py 실행화면

client가 server에게 GET을 요청했지만 아직 만들어진 TEST.html이 없기 때문에 404 에러를 출력합니다

-----

```
elif request_method == "HEAD":
    if not exists(filepath):
        print("오류 발생(HEAD)")
        msg = "HTTP/1.0 404 NOT FOUND\r\nHost:172.20.10.7"
        connection_socket.sendall(msg.encode())
    else:
        size = getsize(filepath)
        print(msg)
        msg = "HTTP/1.0 200 OK\r\nHOST:172.20.10.7"
        connection_socket.send(msg.encode())
```

Server.py

Client에게 받은 msg가 HEAD 일때의 함수 부분입니다.

파일 경로에 파일이 없다면 404 에러를 보내주고, 존재한다면 200 성공 메시지를 보내줍니다.

```
elif test_case == 2:
    server.sendall(b"HEAD TEST.html")
    recieve_data = server.recv(65535)
    print(recieve_data.decode()+"\n")
```

Client.py

Test\_case를 2를 입력함으로서 서버에게 HEAD TEST.html을 보내고 데이터를 받아 터미널에 출력해줍니다.

---

```
junyoun@bagjun-yeong-ui-MacBookAir network % python3 server.py
START THE SERVER
오류 발생 (GET)
오류 발생 (HEAD)
█
```

Server.py 실행화면

```
2
HTTP/1.0 404 NOT FOUND
Host:172.20.10.7

1. GET TEST
2. HEAD TEST
3. POST TEST
4. PUT TEST
5. DELETE TEST
6. END THE TEST
█
```

Client.py 실행화면

client가 server에게 HEAD를 요청했지만 아직 만들어진 TEST.html이 없기 때문에 404 에러를 출력합니다

---

```
elif request_method == "POST":
    print(msg)
    f = open(filename+'.'+filetype,'w')
    msg = "HTTP/1.0 100 OK\r\nHOST:172.20.10.7"
    connection_socket.send(msg.encode())
    msg = connection_socket.recv(65535).decode()
    for i in msg:
        f.write(i)
    f.close()
    msg = "HTTP/1.0 201 OK\r\nHOST:172.20.10.7"
    connection_socket.send(msg.encode())
```

Server.py

Client에게 받은 msg가 POST 일때의 함수 부분입니다.

메시지를 입력받으면 파일을 생성한 후 100 성공을 출력한 후 파일의 내용을 다시 입력받고

파일을 작성한 후 201 성공 메시지를 client에게 전송해줍니다.

```
elif test_case == 3:
    server.sendall(b"POST TEST.html")
    recieve_data = server.recv(65535)
    print(recieve_data.decode()+"\n")
    server.sendall(b"MADE IN JUNYOUNG")
    recieve_data = server.recv(65535)
    print(recieve_data.decode()+"\n")
```

Client.py

Test\_case를 3를 입력함으로서 서버에게 POST TEST.html을 보내고 데이터를 받고  
파일의 내용을 추가로 전송한 후 데이터를 받아 터미널에 출력해줍니다.

```
-----
junyoung@bagjun-yeong-ui-MacBookAir network % python3 server.py
START THE SERVER
오류 발생 (GET)
오류 발생 (HEAD)
POST TEST.html
█
```

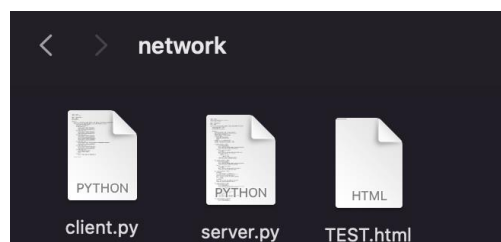
Server.py 실행화면

```
3
HTTP/1.0 100 OK
HOST:172.20.10.7

HTTP/1.0 201 OK
HOST:172.20.10.7

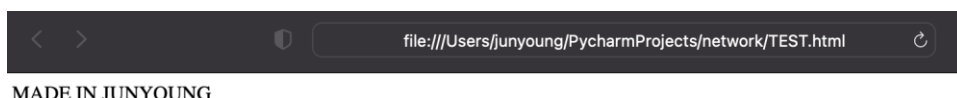
1. GET TEST
2. HEAD TEST
3. POST TEST
4. PUT TEST
5. DELETE TEST
6. END THE TEST
█
```

Client.py 실행화면



파일탐색기

Server 화면에는 입력받은 내용 POST TEST.html이 출력되고 Client 화면에는 100 성공과 201 성공을 받은 것을 볼 수 있습니다. 또한 파일 탐색기에 파일이 생성된 것을 볼 수 있습니다.



TEST.html 실행화면

TEST.html에도 입력한 내용이 들어있는 것을 볼 수 있습니다.

---

```
elif request_method == "PUT":
    if not exists(filepath):
        print("오류 발생(PUT)")
        msg = "HTTP/1.0 404 NOT FOUND\r\nHost:172.20.10.7"
        connection_socket.sendall(msg.encode())
        connection_socket.close()
    else:
        print(msg)
        f = open(filename+'.'+filetype, 'w')
        for i in msg.split()[2:]:
            f.write(i+" ")
        f.close()
        msg = "HTTP/1.0 201 OK\r\nHOST:172.20.10.7"
        connection_socket.send(msg.encode())
```

Server.py

Client에게 받은 msg가 PUT 일때의 함수 부분입니다.

메시지를 입력받으면 파일의 유무를 확인후에 없다면 404에러를 전송하고 파일이 존재한다면 입력받은 내용을 기반으로 파일을 작성한 후 201 성공 메시지를 client에게 전송해줍니다.

```
elif test_case == 4:
    server.sendall(b"PUT TEST.html MADE BY JUNYOUNGPARK 20181620")
    recieve_data = server.recv(65535)
    print(recieve_data.decode()+"\n")
```

Client.py

Test\_case를 4를 입력함으로서 서버에게 PUT TEST.html을 보내고 데이터를 받고 파일의 수정할 내용을 함께 전송한 후 데이터를 받아 터미널에 출력해줍니다.

---

```
junyoung@bagjun-yeong-ui-MacBookAir network % python3 server.py
START THE SERVER
오류 발생 (GET)
오류 발생 (HEAD)
POST TEST.html
PUT TEST.html MADE BY JUNYOUNGPARK 20181620
█
```

Server.py 실행화면

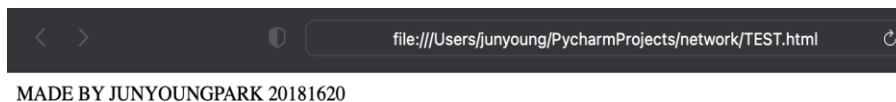


```
4
HTTP/1.0 201 OK
HOST:172.20.10.7
```

```
1. GET TEST
2. HEAD TEST
3. POST TEST
4. PUT TEST
5. DELETE TEST
6. END THE TEST
```

Client.py 실행화면

Server 화면에는 입력받은 내용 (PUT TEST.html MADE ~~~)이 출력되고 Client 화면에는 201 성공을 받은 것을 볼 수 있습니다.



TEST.html 실행화면

TEST.html을 실행했을때에 내용이 성공적으로 바뀐 것을 볼 수 있습니다.

POST와 PUT을 사용해서 파일이 생성되었으므로 GET, HEAD를 호출하면

```
1
HTTP/1.0 200 OK
HOST:172.20.10.7
MADE BY JUNYOUNGPARK 20181620

1. GET TEST
2. HEAD TEST
3. POST TEST
4. PUT TEST
5. DELETE TEST
6. END THE TEST
```

```
2
HTTP/1.0 200 OK
HOST:172.20.10.7

1. GET TEST
2. HEAD TEST
3. POST TEST
4. PUT TEST
5. DELETE TEST
6. END THE TEST
```

GET을 호출했을때와 HEAD를 출력했을때 정상적으로 출력되는 것도 또한 확인할 수 있었습니다.

---



```

elif request_method == "DELETE":
    if not exists(filepath):
        print("오류 발생(REMOVE)")
        msg = "HTTP/1.0 404 NOT FOUND\r\nHost:172.20.10.7"
        connection_socket.sendall(msg.encode())
        connection_socket.close()
    else:
        print(msg)
        remove(filepath)
        msg = "HTTP/1.0 204 OK\r\nHOST:172.20.10.7"
        connection_socket.send(msg.encode())

```

Server.py

```

elif test_case == 5:
    server.sendall(b"DELETE TEST.html")
    recieve_data = server.recv(65535)
    print(recieve_data.decode()+"\n")

```

Client.py

소스코드를 작성하다보니 DELETE를 만들어 파일을 지우는 기능도 만들어 보고 싶어서 응용해보았습니다

Client에게 받은 msg가 DELETE 일때의 함수 부분입니다.

메시지를 입력받으면 파일의 유무를 확인후에 없다면 404에러를 전송하고 파일이 존재한다면 파일의 경로를 찾아가 파일을 삭제하는 기능을 수행합니다.

```

junyoung@bagjun-yeong-ui-MacBookAir network % python3 server.py
START THE SERVER
GET TEST.html
HEAD TEST.html
POST TEST.html
PUT TEST.html MADE BY JUNYOUNGPARK 20181620
GET TEST.html
HEAD TEST.html
DELETE TEST.html
□

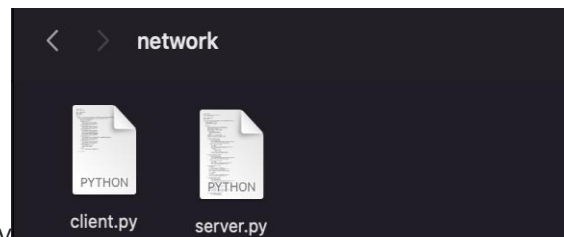
```

Server.py

```
5
HTTP/1.0 204 OK
HOST:172.20.10.7
```

```
1. GET TEST
2. HEAD TEST
3. POST TEST
4. PUT TEST
5. DELETE TEST
6. END THE TEST
```

Client.py



파일탐색기

Server 화면에는 입력받은 내용 (DELETE TEST.html)이 출력되고 Client 화면에는 204 성공을 받은 것을 볼 수 있습니다.

또한 파일 탐색기에도 사라진 것을 볼 수 있습니다.

```
junyoung@bagjun-yeong-ui-MacBookAir network % python3 server.py
START THE SERVER
GET TEST.html
HEAD TEST.html
POST TEST.html
PUT TEST.html MADE BY JUNYOUNGPARK 20181620
GET TEST.html
HEAD TEST.html
DELETE TEST.html
오류 발생 (GET)
[]
```

Server.py 실행화면

server실행 화면에서도 파일이 사라져 GET을 실행할 때 오류가 발생하는 것을 볼 수 있습니다.

-----

```
else:
    msg = "HTTP/1.0 400 WRONG REQUEST\r\nHOST:172.20.10.7"
    connection_socket.send(msg.encode())
```

Server.py

정해진 메소드 말고 다른 메소드를 사용하거나 오타가 발생 했을때 400에러를 발생시켜주는 함수입니다.

```

elif test_case == 6:
    print("FINISH THE TEST")
    server.sendall(b"END")
    break
else:
    print ("입력한 숫자를 다시 확인해주시요.")

server.close()

```

Client.py

클라이언트에서 6을 입력하여 END를 전송하면

```

if msg == "END":
    print("END THE SERVER")
    connection_socket.close()
    break

```

Server.py

Server 소스파일의 윗부분의 if를 통해 break로 while문을 끝내고 서버를 종료합니다.

```

junyoung@bagjun-yeong-ui-MacBookAir network % python3 server.py
START THE SERVER
GET TEST.html
HEAD TEST.html
POST TEST.html
PUT TEST.html MADE BY JUNYOUNGPARK 20181620
GET TEST.html
HEAD TEST.html
DELETE TEST.html
오류 발생 (GET)
END THE SERVER

```

Server.py 실행화면

**6**  
**FINISH THE TEST**

Client.py 실행화면

Server 실행화면에서는 END THE SERVER로 서버가 종료되고

Client 실행화면에서도 FINISH THE TEST로 연결이 종료됩니다.