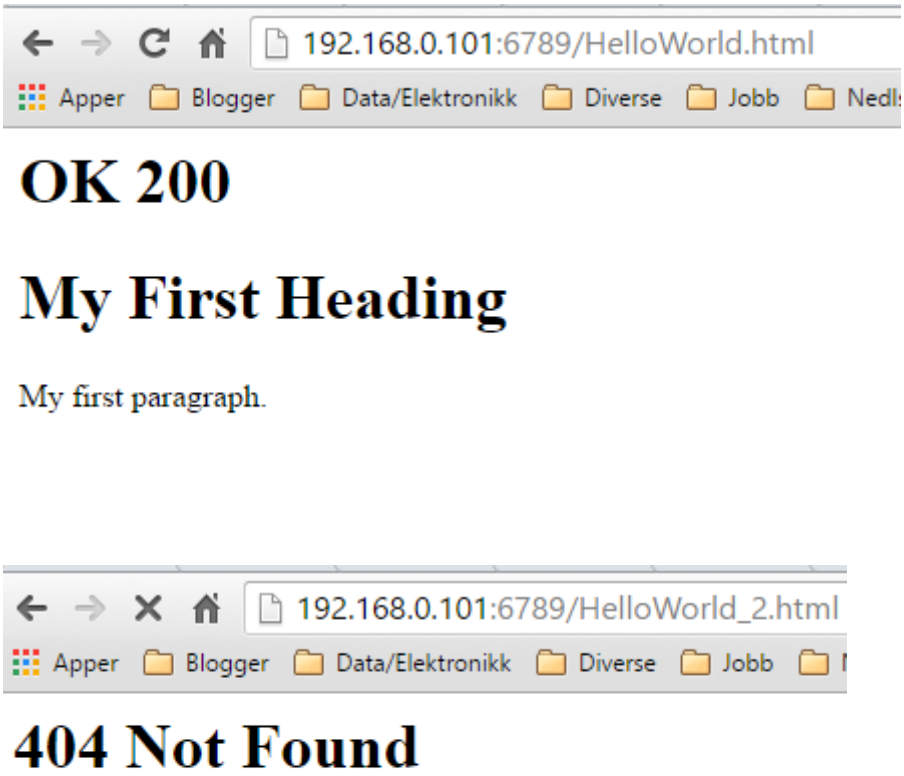
Programming lab 1



```
# Import socket module
from socket import *

# Create a TCP server socket
# (AF_INET is used for IPv4 protocols)
#(SOCK_STREAM is used for TCP)
serverSocket = socket(AF_INET, SOCK_STREAM)

# Prepare a server socket
# FILL IN START
serverName = 'hostname'

# Assign a port number
serverPort = 6789

# Bind the socket to server address and server port
serverSocket.bind(('', serverPort))

# Listen to at most 1 connection at a time
serverSocket.listen(1)

# FILL IN END

# Server should be up and running and listening to the incoming connections
while True:
    print 'Ready to serve...'

    # Set up a new connection from the client
    connectionSocket, addr = serverSocket.accept() # FILL IN START      # FILL IN END

    # If an exception occurs during the execution of try clause
    # the rest of the clause is skipped
    # If the exception type matches the word after except
    # the except clause is executed
    try:
        # Receives the request message from the client
        message =  connectionSocket.recv(1024) # FILL IN START      # FILL IN END

        # Extract the path of the requested object from the message
        # The path is the second part of HTTP header, identified by [1]
        filepath = message.split()[1]

        # Because the extracted path of the HTTP request includes
        # a character '\', we read the path from the second character
        f = open(filepath[1:])

        # Read the file "f" and store the entire content of the requested file in a temporary buffer
        outputdata = f.readlines()# FILL IN START       # FILL IN END
```

```
        # Send the HTTP response header line to the connection socket
        # Format: "HTTP/1.1 *code-for-successful-request*\r\n\r\n"
        # FILL IN START
        connectionSocket.send("<html><head></head><body><h1>OK 200</h1></body></html>\r\n")
        # FILL IN END

        # Send the content of the requested file to the connection socket
        for i in range(0, len(outputdata)):
            connectionSocket.send(outputdata[i])
        connectionSocket.send("\r\n")

        # Close the client connection socket
        connectionSocket.close()

    except IOError:
        # Send HTTP response message for file not found
        # Same format as above, but with code for "Not Found"
        # FILL IN START

        # FILL IN END
        connectionSocket.send("<html><head></head><body><h1>404 Not Found</h1></body></html>\r\n")

        # Close the client connection socket
        # FILL IN START

        # FILL IN END

serverSocket.close()
```