

# 实验一：用python实现web服务器

|  |  |
| --- | --- |
| **学号：** | 20231199 |
| **姓名：** | **尚文韬** |
| **专业：** | **保密技术** |
| **学院：** | **计算机与信息技术学院** |

目录

**[一、 准备实验](#_Toc1409880734_WPSOffice_Level1)** **[4](#_Toc1409880734_WPSOffice_Level1)**

[1.实验目的](#_Toc533553845_WPSOffice_Level2) [4](#_Toc533553845_WPSOffice_Level2)

[2. 实验环境](#_Toc1695246690_WPSOffice_Level2) [4](#_Toc1695246690_WPSOffice_Level2)

[3.实验内容](#_Toc1345574081_WPSOffice_Level2) [4](#_Toc1345574081_WPSOffice_Level2)

[4.实验步骤](#_Toc2060776457_WPSOffice_Level2) [4](#_Toc2060776457_WPSOffice_Level2)

**[二、 实验内容](#_Toc533553845_WPSOffice_Level1)** **[5](#_Toc533553845_WPSOffice_Level1)**

[1.代码实现](#_Toc853653983_WPSOffice_Level2) [5](#_Toc853653983_WPSOffice_Level2)

[2.实验结果展示](#_Toc24246674_WPSOffice_Level2) [7](#_Toc24246674_WPSOffice_Level2)

**[三、 实验总结](#_Toc1695246690_WPSOffice_Level1)** **[9](#_Toc1695246690_WPSOffice_Level1)**

[1.问题总结](#_Toc1015346931_WPSOffice_Level2) [9](#_Toc1015346931_WPSOffice_Level2)

[2. 改进](#_Toc1030810255_WPSOffice_Level2) [10](#_Toc1030810255_WPSOffice_Level2)

[3.其他实现方法](#_Toc1077375436_WPSOffice_Level2) [10](#_Toc1077375436_WPSOffice_Level2)

# 准备实验

## 1.实验目的

（1）处理一个http请求

（2）接收并解析请求

（3）从服务器文件系统中获得被请求的文件

（4）创建一个包括被请求的文件的http响应信息

（5）直接发送该信息到客户端

本次实验选择使用python语言，是基于课本第二章的课后套接字编程作业，详情参考《计算机网络自顶向下方法》（第七版）第二章应用层P119套接字编程作业1

## 实验环境

1. 操作系统：macOS Montery 12.1
2. 使用软件：Visual Studio Code
3. 编程语言：python 3.9.1
4. 硬件环境：MacBook Pro M1芯片

注意：现在提供的工具存在版本与环境的问题，需要选择合适系统完成

## 3.实验内容

（1）实现基于http协议的web服务器

（2）web页面应包含图片，文字，链接以及文件

（3）服务器应对不存在的文件的请求予以正确的回应

## 4.实验步骤

（1）首先根据书上所给出的框架代码建立的基本的tcp通信功能，可以向客户端发送报文

（2）通过分析wireshark抓包的使用的浏览器的请求报文的格式，设计了更便于分析的报文返回格式。

（3）使用socket包实现server的操作，设置本地ip地址：127.0.0.1，端口号：12000（区别于TCP协议中默认的端口号80）

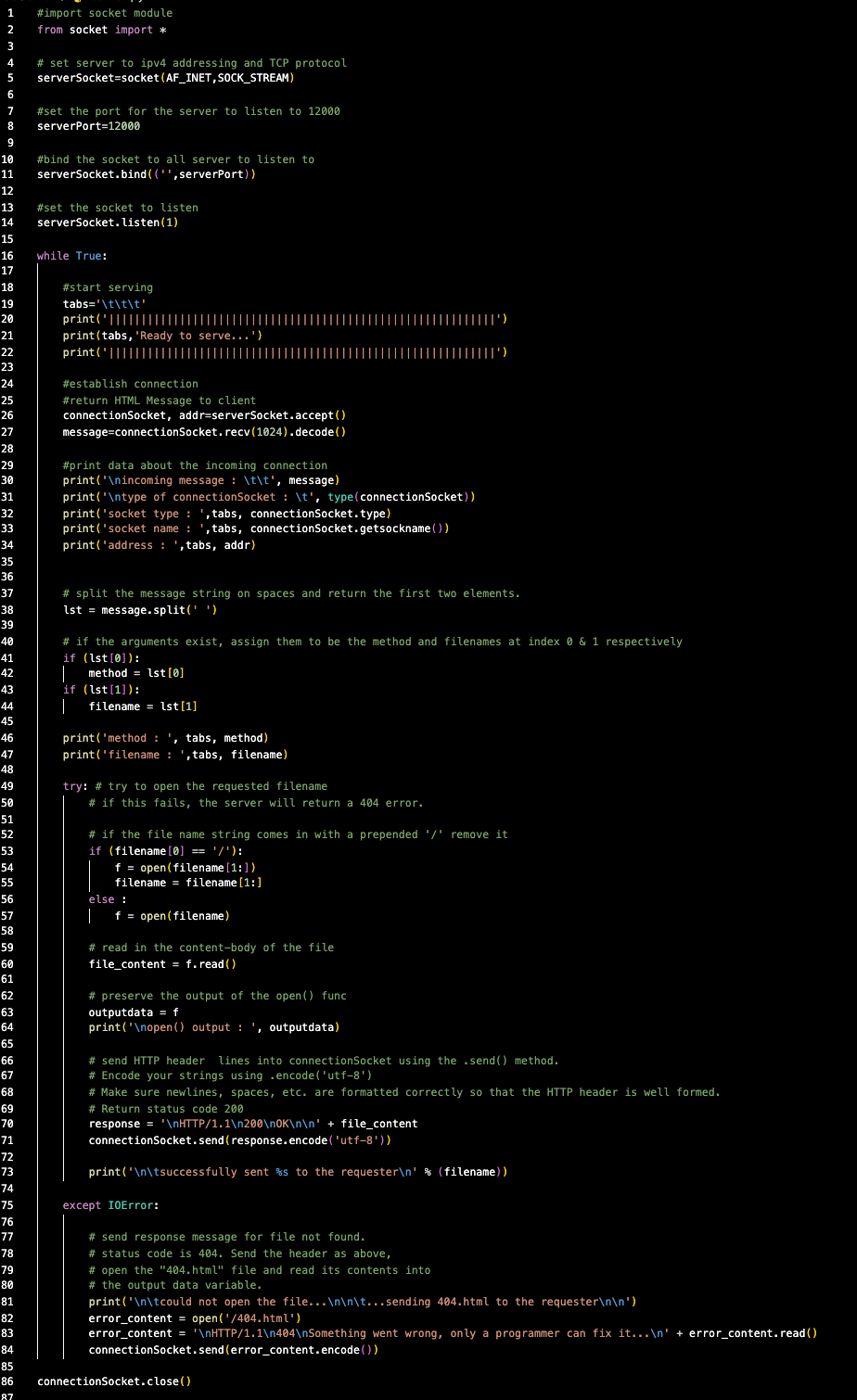
（4）使用html实现网页，并以http形式打开

（5）按照实验指南操作，打开网站。

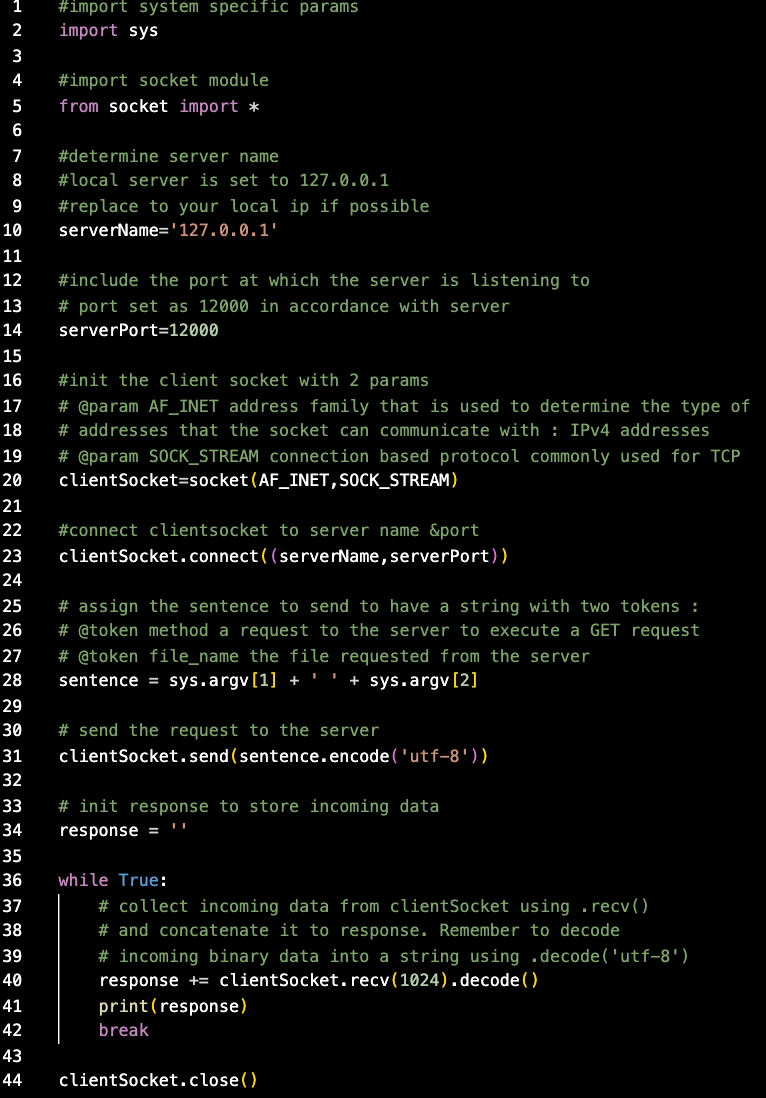
# 实验内容

## 1.代码实现

server.py源代码：



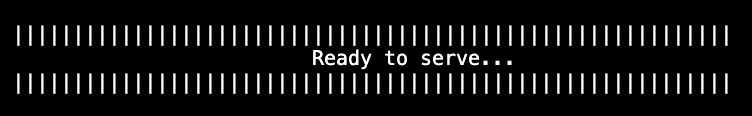
client.py源代码：



## 2.实验结果展示

参考procedure.txt过程：

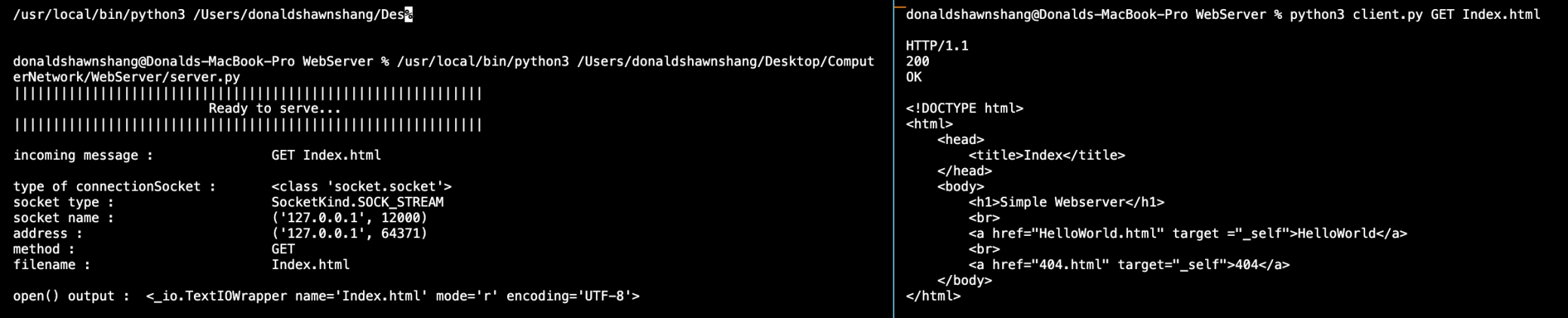
1. 运行server.py



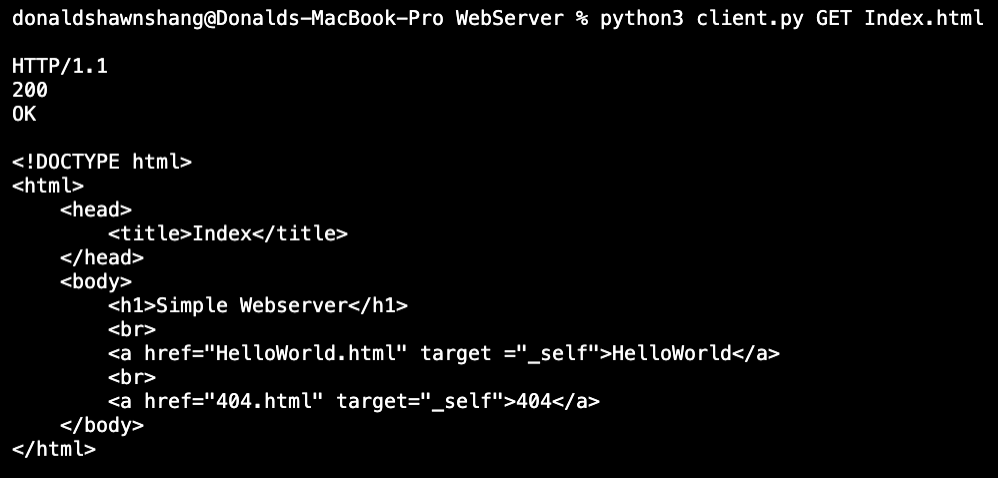
图一：等待接受客户端的请求

1. 在另一平行的终端，输入：python3 client.py GET Index.html

服务器端：

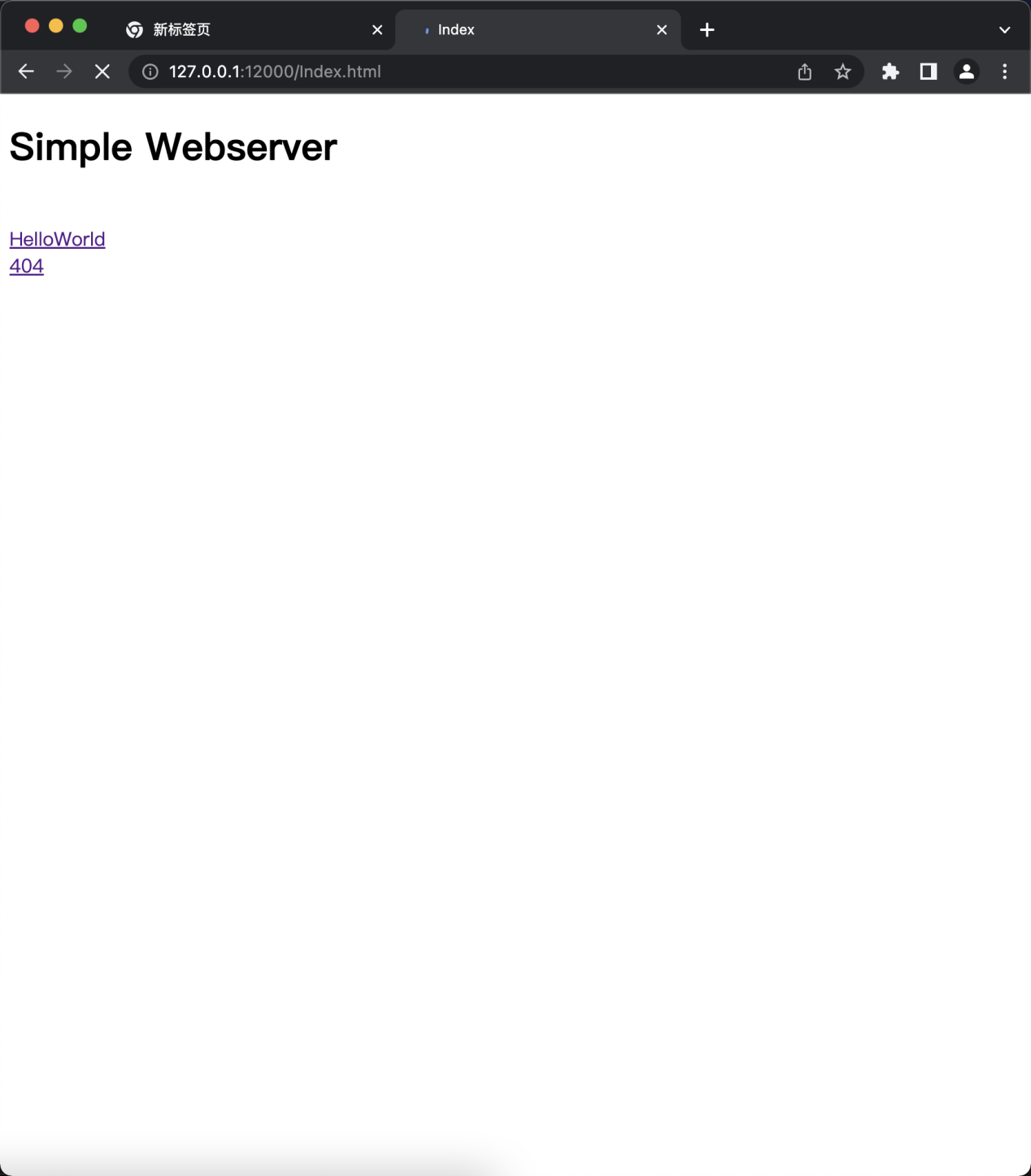


图二：收到客户端请求

客户端：

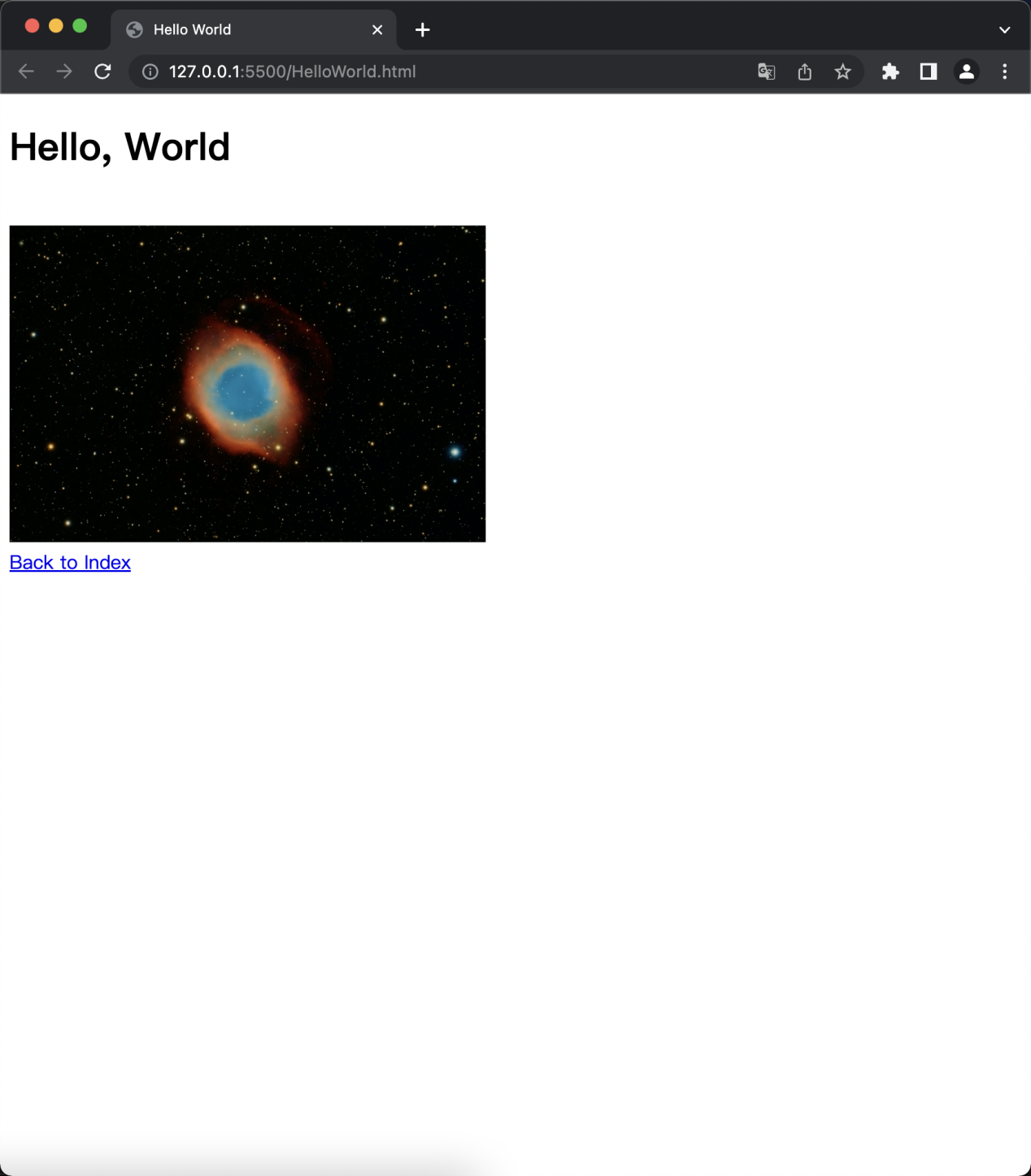
图三：客户端报文回复

1. 打开http://127.0.0.1:12000/Index.html



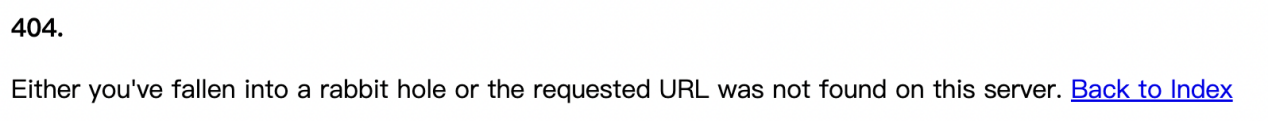
图四：打开Index

1. 点击网页超链接HelloWorld

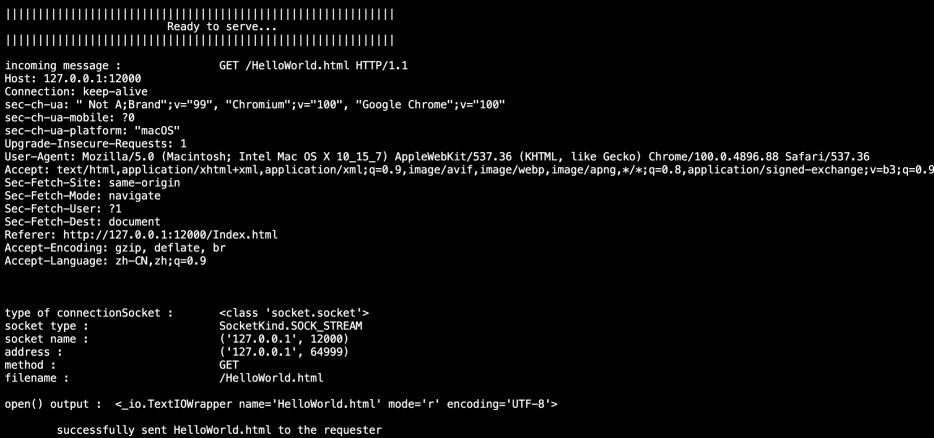


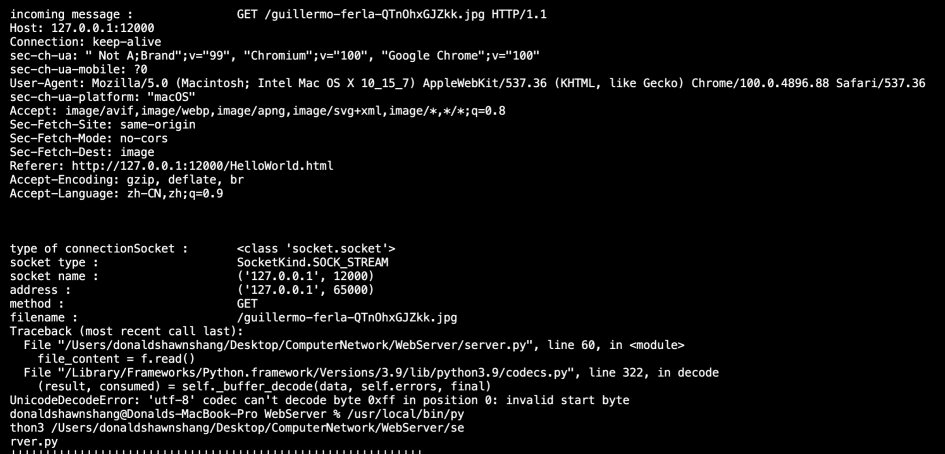
图五：打开HelloWorld

1. 在客户端终端输入：python3 client.py GET NonExistentFile.html



1. 查看服务器报文信息





图六、七：分别为两次打开网页所返回的报文信息

# 实验总结

## 1.问题总结

1. 注意python的版本问题，python3内部基本可以兼容，但python2的很多接口都发生了变化，注意差别
2. 在socket会话结束之后一定要关闭连接，否则下次不能再使用（操作系统会提示端口被占用）
3. 使用macos终端，接近于linux，使用时也可以利用linux来实现
4. 此次试验中，仅实现了局域网上访问服务器，如果想要在公网上的话，则需要另行配置服务器（例如：阿里云ECS服务器）
5. 初期我在字符串拼接的时候首先对文本文件进行解码之后在拼接，这样是没有必要的，因为python支持字节流的直接拼接，无论何种文件均采用相同的方式发送即可
6. unicode必须使用utf-8,而且若实现中文网页的话，还另需配置unicode。在utf-8下，会乱码

注意：注释中已经详细阐述了每一个包的作用以及每个功能的实现过程，如需更多可参考procedure.txt

## 改进

由于实验要求所必需的状态码只有200以及404，可以实现更多的状态码来使得服务器更完整。

Html页面的设计可以更丰富

## 3.其他实现方法

可以使用java的socket库或者c++相应的库实现，本次由于想要结合课本中的理解，以及Prof. Kurose的开源资料，作为打开计算机网络学习的优质窗口。