

(1) 第一次调试时，创建 TCP server socket 的代码段

```
listen_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

出现了错误，报错原因如下：

```
PS F:\SCU\每学期\大2下册\计算机网络\Webserver> python webserver.py
Traceback (most recent call last):
  File "F:\SCU\每学期\大2下册\计算机网络\Webserver\webserver.py", line 12, in <module>
    listen_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
AttributeError: type object 'socket' has no attribute 'socket'
PS F:\SCU\每学期\大2下册\计算机网络\Webserver>
```

后发现代码

```
from socket import *
```

没有引入 socket，添加 import socket 如下，

```
import time
import socket
```

没有报错

关于 from socket import * 和 import socket 的区别，查阅资料后得知：

Why are they so different?

The outputs of `dir(socket)` are different because `socket` is not the same object in the two cases.

```
>>> import socket
>>> type(socket)
<class 'module'>
>>> type(socket.socket)
<class 'type'>
```

VS.

```
>>> from socket import *
>>> type(socket)
<class 'type'>
```

`import socket` does not enter the `socket` module's definitions in the current symbol table; it only enters the module `socket` there. Using the leading module name you can access the definitions, e. g. `socket.socket` (where the trailing `socket` is the socket objects' type).

`from socket import *` enters all the `socket` module's definitions not beginning with an underscore in the current symbol table, so, just `socket` is the socket objects' type.

From socket import * 引入了 socket 的所有定义和形式，而 import socket 会引入 socket module。

(2) 在调试时发现:

```
AttributeError: 'bytes' object has no attribute 'encode'
```

找到对应的代码段:

```
client_connection.send([output_data[0:1000].encode()])
```

在 stackoverflow 上找到一个类似的问题:

Check out this working implementation:

```
import bcrypt

password = "asd123"
hashed_password_encoded = bcrypt.hashpw(password.encode('utf8'), bcrypt.gensalt())
hashed_password = hashed_password_encoded.decode("utf8")

is_valid = bcrypt.checkpw(password.encode('utf8'), hashed_password.encode('utf8'))
print(is_valid)
```

Share Improve this answer Follow

answered Mar 18 '20 at 8:40



Seleme

201 ● 1 ● 8

Friend, im gettin my password from SQLite database. – Andres Hernandez Mar 18 '20 at 18:46

@AndresHernandez If you are (and should) storing your password encrypted and in unicode instead of bytes, then you will need to do `password.encode('utf8')` to convert it to bytes. Whereas, if you are storing the password in bytes, then you do not need to encode it again. – Seleme Mar 19 '20 at 14:05

Add a comment

上面的 `encode('utf-8')` 和我的代码中 `encode()` 一样, 答案给出的解释是, 密码就是以 bytes 方式保存的, 所以不需要将 byte 转换为 byte。所以我将代码中的 `‘.encode()’` 删除, 代码如下:

```
client_connection.send([output_data[0:1000]])
```

后此段代码运行没有问题。

(3) 显示问题，最初的代码段为：

```
25
26 # Server should be up and running and listening to the incoming connections
27 while True:
28     print('Serving HTTP on port {serverPort} ...')
29
30     # Set up a new connection from the client
31     client_connection, client_address = listen_socket.accept()
32
33     # If an exception occurs during the execution of try clause
34     # the rest of the clause is skipped
35     # If the exception type matches the word after except
36     # the except class is executed
37
38     try:
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

PS F:\SCU\每学期\大2下册\计算机网络\Webserver> & C:/Users/48907/AppData/Local/Programs/Python/Python39/python.exe f:/SCU/每学期/大2下册/计算机网络/Webserver/webserver.py
Serving HTTP on port {serverPort} ...

一开始想将 serverPort 的内容输出到 terminal 当中，就用了 python 字典 {serverPort}，后面发现字典语法用错。后面改成：

```
while True:
    print('Serving HTTP on port ' + serverPort + '...')
```

依旧报错，报错如下：

```
File "f:\SCU\每学期\大2下册\计算机网络\Webserver\webserver.py", line 29, in <module>
    print('Serving HTTP on port ' + serverPort + '...')
TypeError: can only concatenate str (not "int") to str
```

serverPort 被赋值为一个 int 类型（8080），报错内容显示 int 类型不能和 str 类型串接，于是将代码做成如下改变：

```
while True:
    print('Serving HTTP on port ' + str(serverPort) + '...')
```

```
27 # Server should be up and running and listening to the incoming connections
28 while True:
29     print('Serving HTTP on port ' + str(serverPort) + '...')
30
31     # Set up a new connection from the client
32     client_connection, client_address = listen_socket.accept()
33
34     # If an exception occurs during the execution of try clause
35     # the rest of the clause is skipped
36     # If the exception type matches the word after except
37     # the except class is executed
38
39     try:
40         # Receives the request message from the client
41         request_data = client_connection.recv(1024).decode()
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

& C:/Users/48907/AppData/Local/Programs/Python/Python39/python.exe f:/SCU/每学期/大2下册/计算机网络/Webserver/webserver.py

Traceback (most recent call last):
File "f:\SCU\每学期\大2下册\计算机网络\Webserver\webserver.py", line 29, in <module>
 print('Serving HTTP on port ' + serverPort + '...')
TypeError: can only concatenate str (not "int") to str
PS F:\SCU\每学期\大2下册\计算机网络\Webserver> python webserver.py
Serving HTTP on port 8080...

serverPort 成功被显示出来。

(4) PC 局域网 IP 地址查询：

一种方法是：左键 wifi，右键连接的 wifi，并点击属性，如下：



然后往下翻，找到 IP 地址：

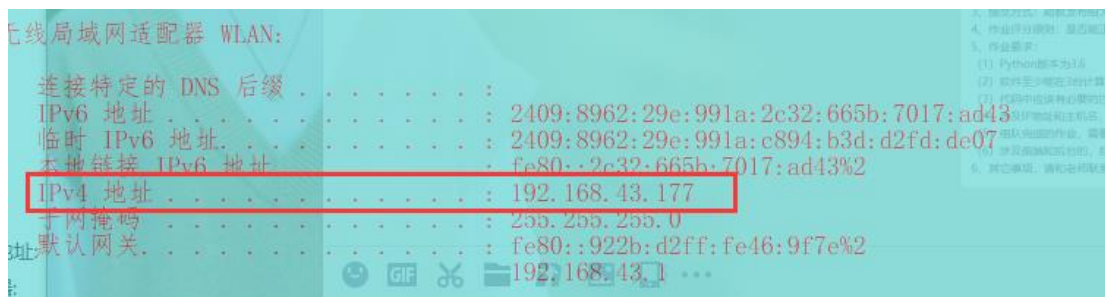
属性

SSID:	你眼睛有春与秋,
协议:	802.11n
安全类型:	WPA2-个人
网络频带:	2.4 GHz
网络通道:	6
IPv6 地址:	2409:8962:29e:991a:2c32:665b:7017:ad43
本地链接 IPv6 地址:	fe80::2c32:665b:7017:ad43%2
IPv6 DNS 服务器:	2409:8962:29e:991a::b4
IPv4 地址:	192.168.43.177
IPv4 DNS 服务器:	192.168.43.1
制造商:	Intel Corporation
描述:	Intel(R) Wireless-AC 9560
驱动程序版本:	21.110.2.1
物理地址(MAC):	58-A0-23-11-E9-DC

复制

另一种方法是:

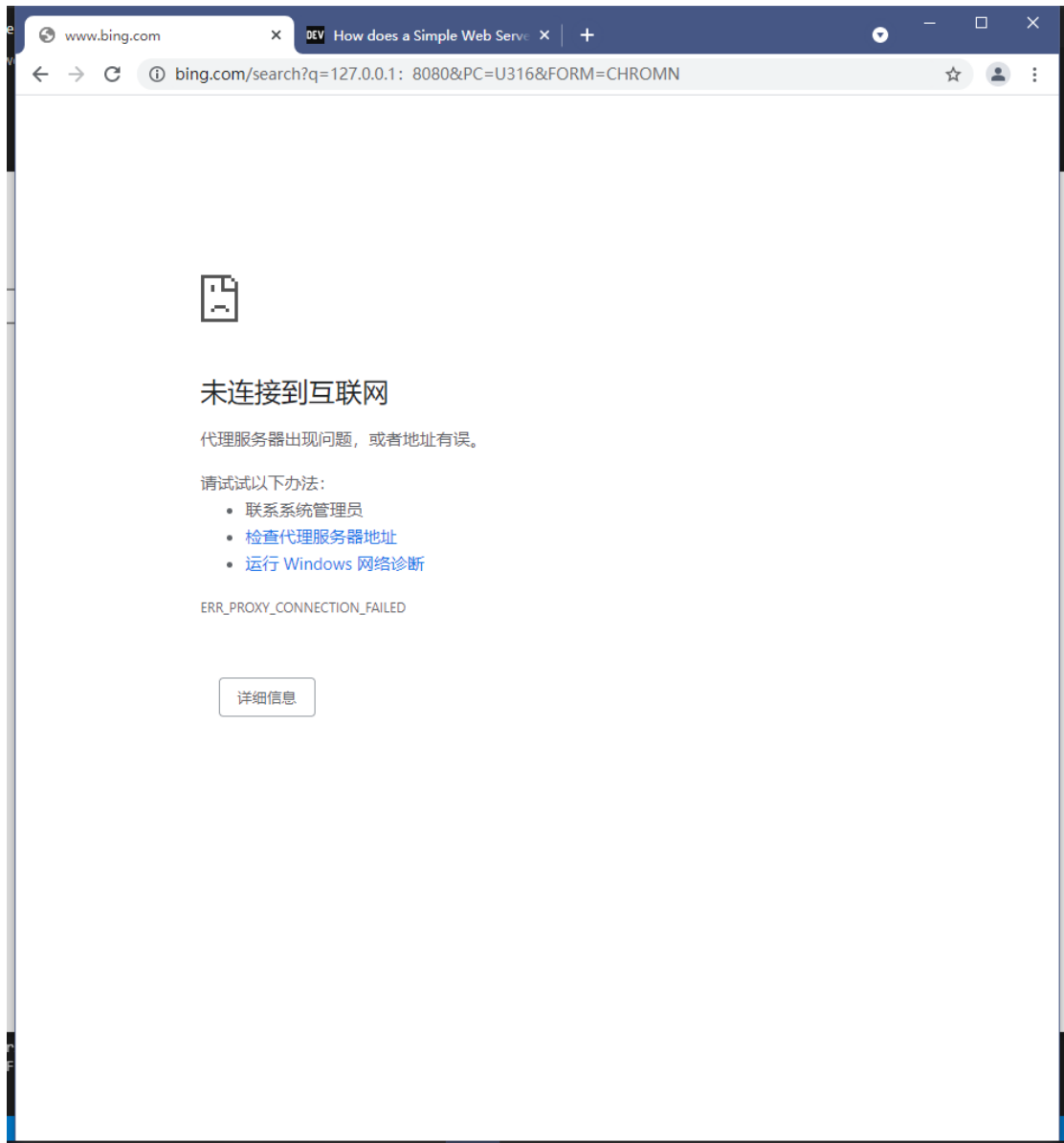
Win+r, 输入 cmd, 然后输入 ipconfig:



因为我是连的 wifi, 所以只需要在无线局域网里面的 subnet mask 上面即可找到 IP 地址。

而我之前用 ipconfig 时一直是用的以太网里的 ip, 后面注意到了 ipconfig 出现的几个网络: 以太网, 局域网才恍然大悟, 最后在代理服务器上 ip 换为 192.168.43.177 也可以显示 webserver 的网页

(5) 访问 127.0.0.1:8080 时，第一次遇见

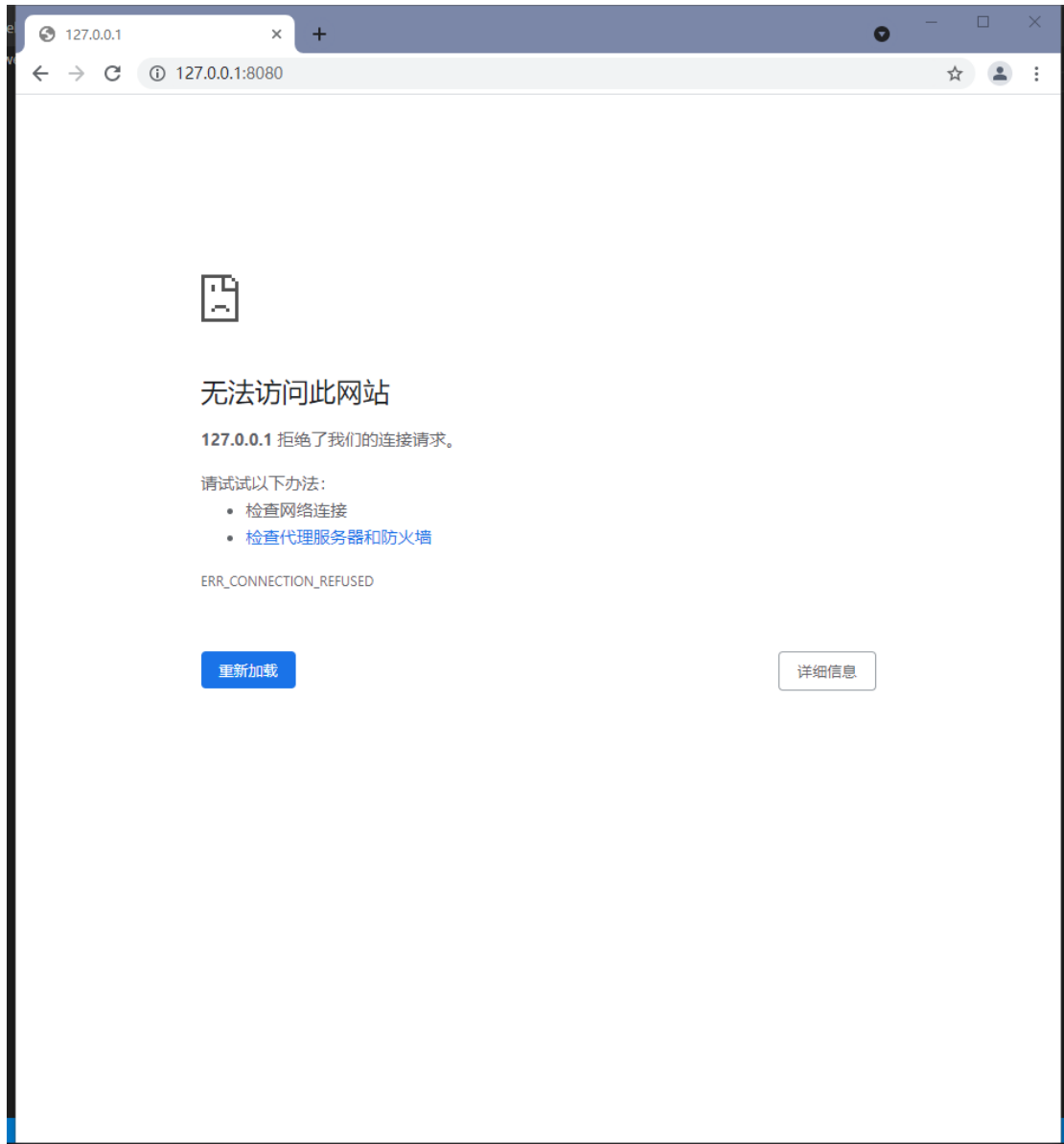


一开始没找出问题所在，后面发现“：”错误，应该改成半角的“:”。

即将 127.0.0.1: 8080

改为 127.0.0.1:8080

改完过后，又出现如下问题：



当时开了两个 webserver.py，后面查阅资料，发现问题可能是 port 冲突导致网页不能显示，当我关闭一个 webserver.py 过后，网页可以正常显示；

出现这种情况还有另一种原因：

当我运行程序时，如果先在 python 上运行 webserver.py，再打开我的代理服务器，网页也会无法访问。当我先打开我事先配置好的代理服务器，然后再运行我的 python 文件，网页也可以正常显示。

正常显示的结果如下：

