## 指定端口运行服务器

**笔记本**: HTTP协议

**创建时间:** 2018/5/16 15:39 **更新时间:** 2018/5/16 17:24

**作者:** ly

标签: sys.argv

#### 命令行参数:

```
import sys print(sys.argv)

# 运行python3 命令行参数.py a ly c

# 打印了以列表方式返回['命令行参数.py', 'a', 'ly', 'b', 'c']

# py后面的数当做前面的的行数的参数传入
```

E: \Python\HTTP协议\HTTP编程>python3 命令行参数.py a ly b c d e f ['命令行参数.py', 'a', 'ly', 'b', 'c', 'd', 'e', 'f']

### 简单判断:参数是否合格

```
def main():
    print(sys.argv)

# 判断后面跟的参数是否为两个
    if len(sys.argv) != 2:
        print("error, usage: xxx port")
        return

port = int(sys.argv[1])
    print = ("port = ", port)

if __name__ == "__mian__":
    main()
```

```
E: \Python \HTTP协议 \HTTP编程>python3 命令行参数.py 1123
['命令行参数.py', '1123']
port = 1123

E: \Python \HTTP协议 \HTTP编程>python3 命令行参数.py 123 456
['命令行参数.py', '123', '456']
error, usage: xxx port
```

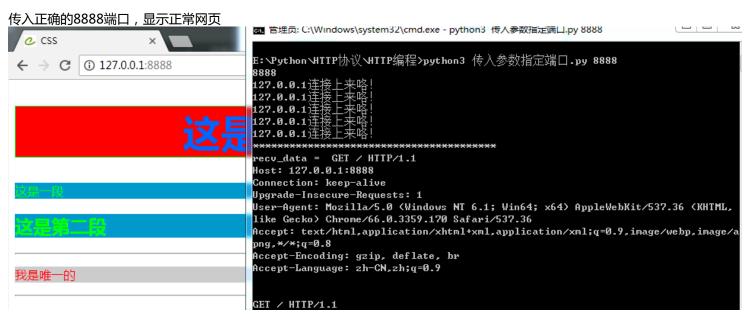
# 指定端口运行服务器 手动指定端口服务

```
import socket
import re
import gevent
import sys

from gevent import monkey
monkey.patch_all()
```

```
def handle_client(new_socket):
  """接收 并返回固定数据"""
  #接收数据
  recv_data = new_socket.recv(1024).decode()
  print("*" * 40)
  print("recv_data = ", recv_data)
  #接收到浏览器后面的用户请求内容: http://127.0.0.1:8888/index.html
  #实际:已经经过浏览器包装HTTP格式的请求报文 下面是解析这些报文
  # 以行为单位切割
  recv_list = recv_data.splitlines()
  # GET /index.html HTTP/I.1 ====> recv_list[0]
  # [^/]+(/[^]+).*
  #请求头部
  #空行
  #请求包体
  # 通过正则匹配请求的index.html
  print(str(recv_list[0]))
  firstLine = recv_list[0]
  fileName = re.match(r"[^/]+(/[^]*)", firstLine).group(1)
  print(fileName)
  # 支持默认路径
  if fileName == "/":
    fileName = "/index.html"
  # 组成路径 ./ 当前路径下查找html文件夹
  #本页代码的路径有一个html文件夹,要访问的页面在html页面下面
  filePath = "./html" + fileName
  print(filePath)
 try:
    with open(filePath, "rb") as f:
      content = f.read()
  except Exception as ret:
    # 没有这个请求的资源 404
    send_data = "HTTP/1.1 404 NOT FOUND\r\n"
    send_data += "\r\n"
    content = "file not found".encode()
  else:
    # 有这个请求对应的资源 200
    # 构造适合网页格式的响应体
    send_data = "HTTP/1.1 200 OK\r\n"
    send_data += "\r\n"
    # send_data += "ok"
    # 发送数据
  new_socket.send(send_data.encode())
  new_socket.send(content)
 new_socket.close()
def main():
  """返回固定数据"""
  # 创建套接字
 tcp = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  #端口复用
  tcp.setsockopt(socket.SOL SOCKET, socket.SO REUSEADDR, 1)
```

```
# 判断是否是只传入了一个绑定端口
  if len(sys.argv) != 2:
    print("参数传入错误")
    return
  #端口绑定
  port = int(sys.argv[1]) #第一个参数为本身.py文件,第二个为绑定端口 角标为1
  print(port)
  tcp.bind(("", port))
  # 监听
 tcp.listen(128)
 while True:
    # 取出客户 返回服务套接字
    new_socket, cli_addr = tcp.accept()
    print("%s连接上来咯!"% cli_addr[0])
    # handle_client(new_socket)
    gevent.spawn(handle_client, new_socket)
    # 关闭监听套接字
  tcp.close()
if __name__ == '__main__':
  main()
```



#### 传入参数错误 直接return 无法绑定端口 也无发显示网页

E:\Python\HTTP协议\HTTP编程>python3 传入参数指定端口.py 8888 444 参数传入错误

