CSE3231 Assignment 4 - TCP Programming

Grant Butler | e. gbutler2020@my.fit.edu

Part I:

The code in `get_html(host, port, page)` retrieves the html from the server:

```
def get_html(host, port, page):
    request = f'GET {page} HTTP/1.1\r\n{host}\r\n\r\n'

    # open the socket as TCP → SOCK_STREAM since TCP connects a data stream
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
        sock.connect((host, port))  # connect to the server
        sock.send(request.encode())  # send the request to the server
        response = sock.recv(4096)  # get response from server
        html = repr(response)  # get html doc text from response data

    return html
```

Then, in `main` the function is called and the html is printed out:

```
def main():
    host = '0.cloud.chals.io'
    port = 23456

    main_page = '/'
    main_html = get_html(host, port, main_page)
    print(host + main_page)
    print(main_html, end='\n\n')  # part 1 done
```

This is what gets printed out:

```
0.cloud.chals.io/
b'HTTP/1.0 200 OK\r\nContent-Type: text/html; charset=utf-8\r\nContent-Length: 266\r\nServer: Werkzeug/0.11.15 Python/3.6.1\r\nDate: Tue, 12 Apr 2022 07:09:00 GMT\r\n\r\n<HTML><TITLE>CSE 3231 - Assignment # 4</TITLE><BODY><H3>CSE 3231 Assignment # 4 - first page</H3>This page contains a link to another page.  Extract that link and follow it to retrieve that page.<P><A HREF="./a4-page-2.html">This is the second page</A></BODY></HTML>'
```

... not very pretty. Oh well, such is life.

Part II:

For this part, we can use something super cool about html to our advantage!

```
<A HREF="./a4-page-2.html">This is the second page</A>
```

The part `HREF` always is followed by a link! And since there is only one link, and I don't have access to something like `lxml` or `bs4`, I will be using regular expressions. So, I made this `HREF="(?:[^"]|"")*"`. A bit of substring magic and boom! Link.

Then, I simply call this:

```
# now to parse the other link
link_regex = re.compile(r'HREF="(?:[^"]|"")*"')  # regex for the href
href_str = link_regex.search(main_html).group()  # getting the link

# substring magic
sub_page = f'{href_str[7:-1]}'
sub_html = get_html(host, port, sub_page)
print(host + sub_page)
print(sub_html, end='\n\n')  # part 2 done
```

This is the resultant response:

```
0.cloud.chals.io/a4-page-2.html
b'HTTP/1.0 200 OK\r\nContent-Type: text/html; charset=utf-8\r\nContent-Length: 159\r\nServer: Werkzeug/0.11.15 Py
thon/3.6.1\r\nDate: Tue, 12 Apr 2022 07:09:01 GMT\r\n\r\n<HTML><TITLE>CSE 3231 - Assignment # 4</TITLE><BODY><H3>
CSE 3231 Assignment # 4 - second page</H3>You have successfully accessed the second page.</BODY></HTML>'
```

And I believe that is it! Thank you for reading, I appreciate you. The following is my code and the total execution of all parts at once:

program:

```python
import socket
import re


def get_html(host, port, page):
    request = f'GET {page} HTTP/1.1\r\n{host}\r\n\r\n'

    # open the socket as TCP → SOCK_STREAM since TCP connects a data stream
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
        sock.connect((host, port))  # connect to the server
        sock.send(request.encode())  # send the request to the server
        response = sock.recv(4096)  # get response from server
        html = repr(response)  # get html doc text from response data

    return html


def main():
    host = '0.cloud.chals.io'
    port = 23456

    main_page = '/'
    main_html = get_html(host, port, main_page)
    print(host + main_page)
    print(main_html, end='\n\n')  # part 1 done

    # now to parse the other link
    link_regex = re.compile(r'HREF="(?:[^"]|"")*"')  # regex for the href
    href_str = link_regex.search(main_html).group()  # getting the link

    # substring magic
    sub_page = f'{href_str[7:-1]}'
    sub_html = get_html(host, port, sub_page)
    print(host + sub_page)
    print(sub_html, end='\n\n')  # part 2 done


main()

```

output:

```
cse3231assignments/assignment4_tcp_programming/src on  trunk via  v3.10.4 (cse3231assignment4)
❯ python tcp_programming.py
0.cloud.chals.io/
b'HTTP/1.0 200 OK\r\nContent-Type: text/html; charset=utf-8\r\nContent-Length: 266\r\nServer: Werkz

0.cloud.chals.io/a4-page-2.html
b'HTTP/1.0 200 OK\r\nContent-Type: text/html; charset=utf-8\r\nContent-Length: 159\r\nServer: Werkz
```