

Assignment 2: A (very) simple web browser

The goal of this assignment is to implement a basic, text-based web browser using HTTP requests. The web browser will accept one string as a command line parameter (the URL to open) and use the sample code provided in the file `http/connection_helper.py` (contained in the examples Git repo) to perform the following tasks:

- Send a properly constructed GET request to the server.
- Parse the HTTP response headers and show them to the user as key-value pairs.
- Parse the HTTP response body (the HTML response) and list the links available on the web page as a numbered list with the following format:
 - Number of the link, starting with 1 for the first link, 2 for the second, etc.
 - Text of the link.
 - The character sequence "->"
 - The target address of the link.
- The user will be able to select one of the links of the page by entering the number displayed with that link, or 0 to exit the browser.

For testing and development, the web server provided in the file `http/webserver.py` can be used. This file starts a web server on port 80 of the local machine serves an example web page with two links on the URL `localhost/example`.

Example input:

```
python webbrowser.py localhost/example
```

Example output:

```
Server: BaseHTTP/0.6 Python/3.9.5
Date: Sun, 01 May 2022 19:51:47 GMT
Content-type: text/html
[1] Link 1 -> link1.html
[2] Link 2 -> link2.html
Press 0 to exit
```

If the user enters 1, a new GET request for `localhost/link1.html` will be submitted to the server.

Deadline: 21/05/2024, 23:59 CET.

Notes

- It is **not allowed** to use the Python requests package. The HTTP requests have to be handled explicitly.
- Every text line that is sent using HTTP has to end with the carriage return sequence `\r\n`. Use an additional carriage return sequence to end an HTTP message.
- All submitted files shall be enclosed in a single zip file.
- Every function and relevant code blocks have to be documented using comments, including purpose, meaning of the parameters/variables and return value, when applicable. Similarly, the code itself has to provide comments where necessary to help understand it.



Assessment

Total: 30 points.

- The web browser shows all headers returned from the server: 5 points.
- The HTTP requests are formed correctly: 10 points.
- The web browser shows correctly all links contained in the HTML body: 10 points.
- The code is clean and well documented: 5 points.