

Simple HTTP Client

Problem Description

In this problem, you are required to implement a simple HTTP client using Python. The client will connect to a server, send a GET request, and process the response.

Your task is to implement the following functions:

1. `get_first_length(data: str) -> int`: This function takes a string data as input, which represents the HTTP response. The function should return the length of the header plus the content length (if it exists). If the content length is not a valid integer, the function should return 0.
2. `create_socket() -> socket.socket`: This function should create a socket, connect it to the server at 'localhost' on port 8080, and return the socket.
3. `client() -> None`: This function should create a socket using the `create_socket` function, send a GET request for 'index.html', and process the response. The function should continue receiving data until there is no more data to receive or the length of the received data is less than 1024 bytes.

Input: no input.

Output (without unit test, server is in another problem):

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 188
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Hello world!</title>
</head>
<body>
Hello world!
</body>
</html>
```

Output (with unit test):

```
Testing client ...
HTTP/1.1 200 OK
Content-Length: 5
```

```
12345
connect called with: call(('localhost', 8080))
```

```
send called with: call(b'GET index.html HTTP/1.1\r\nHost:
localhost\r\n\r\n')
recv called with: call(1024)
close called with: call()
Testing create_socket ...
connect called with: call(('localhost', 8080))
Testing get_first_length_no_content_length ...
test attribute passed: 35 is equal to 35
Testing get_first_length_with_content_length ...
test attribute passed: 39 is equal to 39
```