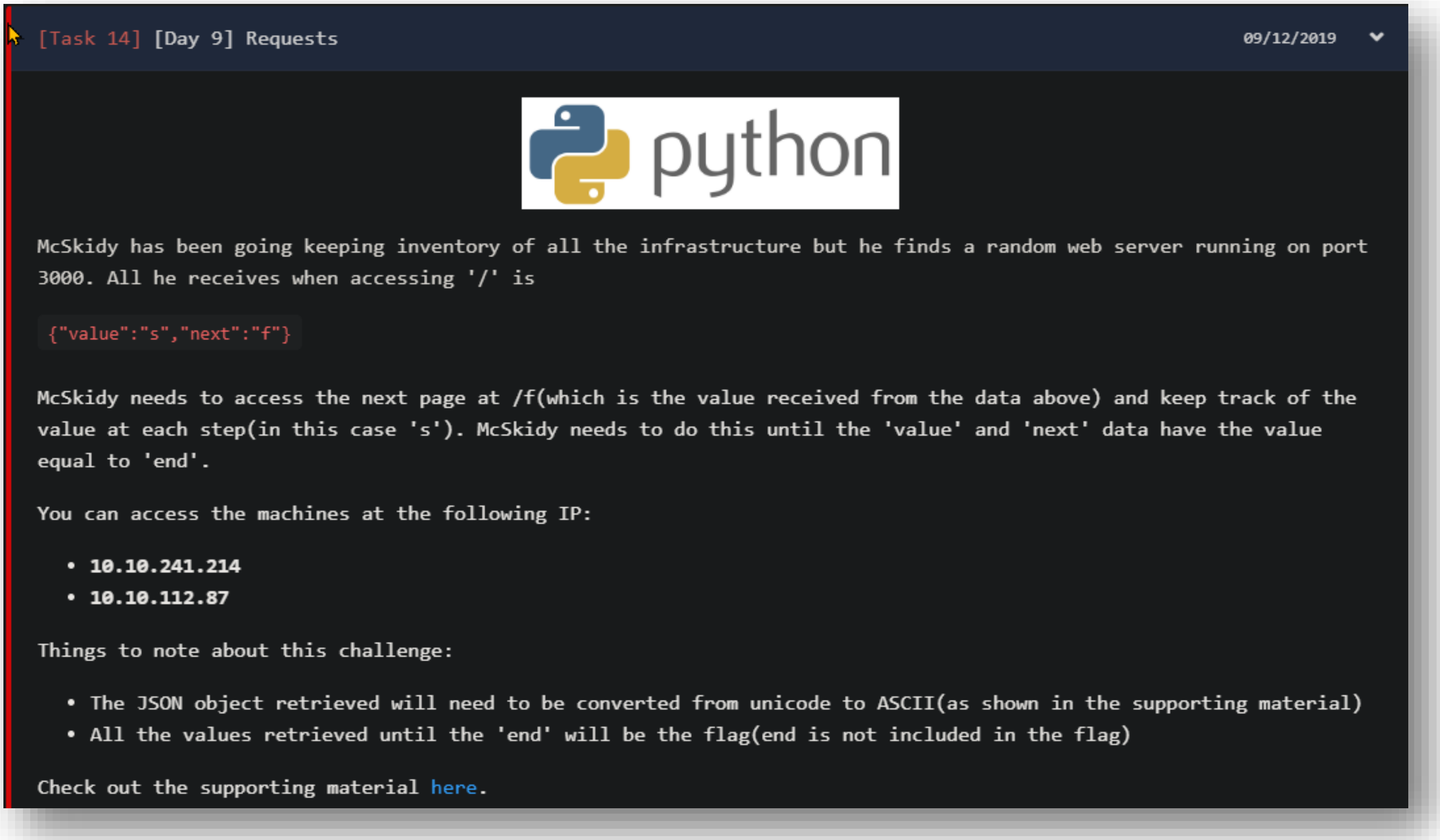
https://docs.google.com/document/d/1FyAnxlQpzh0Cy17cKLsUZYCYqUA3eHu2hm0snilaPL0/edit?usp=sharing



[Task 14] [Day 9] Requests 09/12/2019

McSkidy has been going keeping inventory of all the infrastructure but he finds a random web server running on port 3000. All he receives when accessing '/' is

```
{"value":"s","next":"f"}
```

McSkidy needs to access the next page at /f(which is the value received from the data above) and keep track of the value at each step(in this case 's'). McSkidy needs to do this until the 'value' and 'next' data have the value equal to 'end'.

You can access the machines at the following IP:

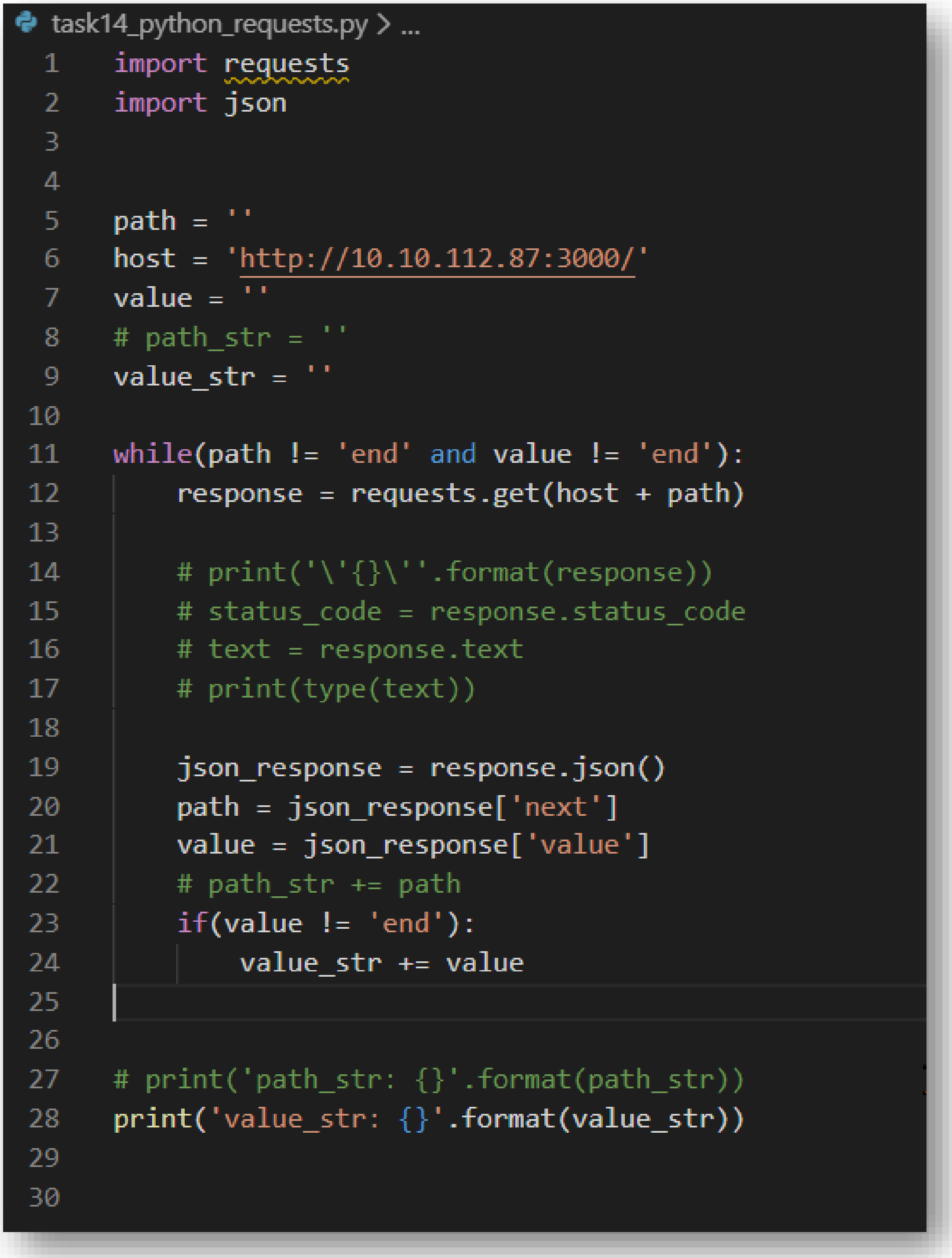
- **10.10.241.214**
- **10.10.112.87**

Things to note about this challenge:

- The JSON object retrieved will need to be converted from unicode to ASCII(as shown in the supporting material)
- All the values retrieved until the 'end' will be the flag(end is not included in the flag)

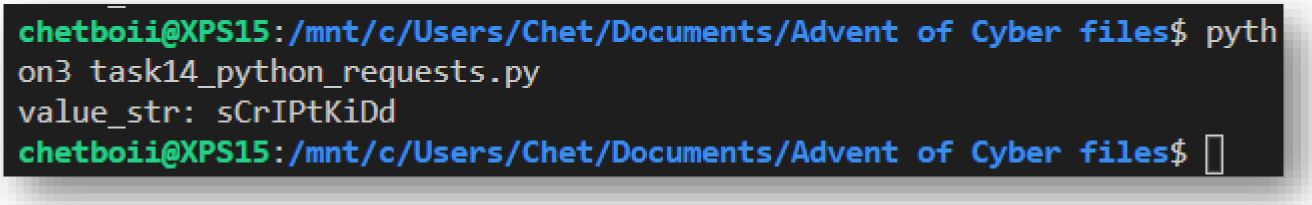
Check out the supporting material here.

1. **What is the value of the flag?**
    - The supporting material goes over the requests library in python3
    - Each request to ‘next’ results in a new value
    - If we keep using the next result as the new path, we can keep track of all the values until path and value result in ‘end’
    - The python script I came up with is:



```python
1   import requests
2   import json
3
4
5   path = ''
6   host = 'http://10.10.112.87:3000/'
7   value = ''
8   # path_str = ''
9   value_str = ''
10
11  while(path != 'end' and value != 'end'):
12      response = requests.get(host + path)
13
14      # print('\'{}\''.format(response))
15      # status_code = response.status_code
16      # text = response.text
17      # print(type(text))
18
19      json_response = response.json()
20      path = json_response['next']
21      value = json_response['value']
22      # path_str += path
23      if(value != 'end'):
24          value_str += value
25
26
27  # print('path_str: {}'.format(path_str))
28  print('value_str: {}'.format(value_str))
29
30
```

- Line 11: A while loop continues executing the block under it until the conditions are met. We keep getting the new value and next values until they hold the string 'end'.
- Lines 14 – 17 are commented out, just used that code to help figure out formatting
- Line 19: json_response is a dictionary object, we can easily grab the values on lines 20 and 21 with [ ]
- Lines 23-24 are building the value/flag string, appending the value to the string unless the value is 'end'



```
chetboii@XPS15:/mnt/c/Users/Chet/Documents/Advent of Cyber files$ pyth
on3 task14_python_requests.py
value_str: sCrIPtKiDd
chetboii@XPS15:/mnt/c/Users/Chet/Documents/Advent of Cyber files$
```

- The flag is: **sCrIPtKiDd**