



Practice > Python > Regex and Parsing > HTML Parser - Part 2

HTML Parser - Part 2 ☆

63/115 challenges solved

Rank: 6324 | Points: 1115



Your HTML Parser - Part 2 submission got 30.00 points.

Share

Tweet

[Try the next challenge](#) | [Try a Random Challenge](#)

Problem

Submissions

Leaderboard

Discussions

Editorial

* This section assumes that you understand the basics discussed in [HTML Parser - Part 1](#)

[.handle_comment\(data\)](#)

This method is called when a comment is encountered (e.g. `<!--comment-->`).The *data* argument is the content inside the comment tag:

```
from HTMLParser import HTMLParser

class MyHTMLParser(HTMLParser):
    def handle_comment(self, data):
        print "Comment :", data
```

[.handle_data\(data\)](#)

This method is called to process arbitrary data (e.g. text nodes and the content of `<script>...</script>` and `<style>...</style>`).The *data* argument is the text content of HTML.

```
from HTMLParser import HTMLParser

class MyHTMLParser(HTMLParser):
    def handle_data(self, data):
        print "Data :", data
```

Task

You are given an *HTML* code snippet of *N* lines.Your task is to print the *single-line comments*, *multi-line comments* and the *data*.

Print the result in the following format:

```
>>> Single-line Comment
Comment
>>> Data
My Data
>>> Multi-line Comment
Comment_multiline[0]
Comment_multiline[1]
>>> Data
```

Author

DOSHI

Difficulty

Easy

Max Score

30

Submitted By

3578

NEED HELP?

[View discussions](#)[View editorial](#)[View top submissions](#)

RATE THIS CHALLENGE



MORE DETAILS

[Download problem statement](#)[Download sample test cases](#)[Suggest Edits](#)

```
My Data
>>> Single-line Comment:
```

Note: Do not print *data* if `data == '\n'`.

Input Format

The first line contains integer N , the number of lines in the *HTML* code snippet.

The next N lines contains *HTML* code.

Constraints

$$0 < N < 100$$

Output Format

Print the *single-line comments*, *multi-line comments* and the *data* in order of their occurrence from top to bottom in the snippet.

Format the answers as explained in the problem statement.

Sample Input

```
4
<!--[if IE 9]>IE9-specific content
<![endif]-->
<div> Welcome to HackerRank</div>
<!--[if IE 9]>IE9-specific content<![endif]-->
```

Sample Output

```
>>> Multi-line Comment
[if IE 9]>IE9-specific content
<![endif]
>>> Data
Welcome to HackerRank
>>> Single-line Comment
[if IE 9]>IE9-specific content<![endif]
```

Current Buffer (saved locally, editable)



Python 3



```
1 from html.parser import HTMLParser
2
3 class MyHTMLParser(HTMLParser):
4     def handle_comment(self,comment):
5         if('\n' in comment):
6             print('>>> Multi-line Comment')
7         else:
8             print('>>> Single-line Comment')
9             print(comment)
10    def handle_data(self,data):
11        if(data=='\n'):
12            return
13        print('>>> Data')
14        print(data)
15    html = ""
16    for i in range(int(input())):
17        html += input().rstrip()
18        html += '\n'
19
```

```
20 parser = MyHTMLParser()
21 parser.feed(html)
22 parser.close()
```

Line: 14 Col: 16

[Upload Code as File](#) ☐ Test against custom input

Run Code

Submit Code



You have earned 30.00 points!

63/115 challenges solved.

55%

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next
Challenge

✔ Testcase 0

✔ Testcase 1

✔ Testcase 2

✔ Testcase 3

6 Testcases

Input (stdin)

[Download](#)

```
4
<!--[if IE 9]>IE9-specific
content
</endif-->
```

Expected Output

[Download](#)

```
>>> Multi-line Comment
[if IE 9]>IE9-specific cont
ent
</endif-->
```

Compiler Message

Success