navigating_the_parse_tree_solution

March 18, 2023

1 TODO: Access The Meta Tag

In the cell below, access the `<meta>` tag contained within the `<head>` tag. Start by importing BeautifulSoup, then open the `sample.html` file and pass the open filehandle to the BeautifulSoup constructor and use the `lxml` parser. Save the BeautifulSoup object returned by the constructor in a variable called `page_content`. Then access the `<meta>` tag contained within the `<head>` tag from the `page_content` object. Save the Tag object in a variable called `page_meta` and then print it.

In [1]:
```
# Import BeautifulSoup
from bs4 import BeautifulSoup

# Open the HTML file and create a BeautifulSoup Object
with open('./sample.html') as f:
    page_content = BeautifulSoup(f, 'lxml')

# Access the head tag
page_meta = page_content.head.meta

# Print the Tag Object
print(page_meta)
```

```
<meta charset="utf-8"/>
```

2 TODO: Access The `<h1>` Tag

In the cell below, access the `<h1>` tag contained within the `<body>` tag. Start by importing BeautifulSoup, then open the `sample.html` file and pass the open filehandle to the BeautifulSoup constructor and use the `lxml` parser. Save the BeautifulSoup object returned by the constructor in a variable called `page_content`. Then access the `<h1>` tag contained within the `<body>` tag from the `page_content` object. Save the Tag object in a variable called `page_h1` and then print it.

In [2]:
```
# Import BeautifulSoup
from bs4 import BeautifulSoup

# Open the HTML file and create a BeautifulSoup Object
with open('./sample.html') as f:
```

```
            page_content = BeautifulSoup(f, 'lxml')

        # Access the h1 tag
        page_h1 = page_content.body.h1

        # Print the Tag Object
        print(page_h1)
```

```
<h1 id="intro">Get Help From Peers and Mentors</h1>
```

3 TODO: Remove HTML Tags

In the cell below, use the `.get_text()` method to remove all the HTML tags from the `sample.html` document. In other words, just print the entire text in the document, with no HTML tags.

HINT: Use the `.get_text()` method on the `page_content` object.

In [3]:
```
# Import BeautifulSoup
from bs4 import BeautifulSoup

# Open the HTML file and create a BeautifulSoup Object
with open('./sample.html') as f:
    page_content = BeautifulSoup(f, 'lxml')

# Print only the text in the whole document
print(page_content.get_text())
```

```


AI For Trading


.h2style {background-color: tomato;color: white;padding: 10px;}


Get Help From Peers and Mentors

Student Hub
Student Hub is our real time collaboration platform where you can work with peers and mentors. Y



Knowledge
Search or ask questions in Knowledge


Good Luck
Good luck and we hope you enjoy the course
```

4 TODO: Get The Website Address

In this exercise, you will get the website address in a hyperlink tag. Hyperlinks are defined by the `<a>` tag. In our `sample.html` document we only have one hyperlink:

```
<a href="https://knowledge.udacity.com/">Knowledge</a>
```

Hyperlinks are used to link webpages together. The `href` attribute in the `<a>` tag, indicates the link's destination, *i.e.* a website address.

In the cell below, open the `sample.html` file and pass the open filehandle to the BeautifulSoup constructor and use the `lxml` parser. Save the BeautifulSoup object returned by the constructor in a variable called `page_content`. Then access the `<a>` tag from the `page_content` object. Save the Tag object in a variable called `page_hyperlink`. Then get the `href` attribute from the `page_hyperlink` object and save it into a variable called `href_attr`. Finally, print the `href_attr` variable.

In [4]:
```
# Import BeautifulSoup
from bs4 import BeautifulSoup

# Open the HTML file and create a BeautifulSoup Object
with open('./sample.html') as f:
    page_content = BeautifulSoup(f, 'lxml')

# Access the a tag
page_hyperlink = page_content.a

# Get the href attribute from the a tag
href_attr = page_hyperlink['href']

# Print the href attribute
print(href_attr)
```

```
https://knowledge.udacity.com/
```

In [ ]: