Data_Preprocessing_NLP_Removing_HTML_tags

Removing HTML Tags

In [1]:
```
def remove_html_tags(text):
    """Remove html tags from a string"""
    import re
    clean = re.compile('<.*?>')
    return re.sub(clean, '', text)

text = "<p class = 'importan'>This is an Important Text</p>"
remove_html_tags(text)
```

Out[1]:
```
'This is an Important Text'
```

In [2]:
```
text = "<p class = 'importan'>This is an Important Text</p>"
import re
clean = re.compile('<.*?>')
text = re.sub(clean,"",text)
text
```

Out[2]:
```
'This is an Important Text'
```

In [3]:
```
tags = re.compile(r'<[^>]+>')
def remove_tags(text1):
    return tags.sub('', text1)
text1 = "<p>The CSS <code>background-color</code> property defines the background color of an element.</p>"
remove_tags(text1)
```

Out[3]:
```
'The CSS background-color property defines the background color of an element.'
```

Remove HTML tags from string in python Using the lxml Module

Instead of using regular expressions, we can also use the lxml module to remove HTML tags from string in python. For this, we will first parse the original string using the fromstring() method.

The fromstring() method takes the original string as an input and returns a parser. After getting the parser, we can extract the text using the text_content() method, leaving behind the HTML tags. The text_content() method returns an object of lxml.etree._ElementUnicodeResult data type. Therefore, we need to convert the output to string using the str() function

In [4]:
```
from lxml.html import fromstring
pattern = '<[^<]+?>'
myString = """<!DOCTYPE html>
<html>
<head>
<title>venkateshmungi</title>
</head>
<body>
<h1>I am a sentence inside an HTML string.</h1>
<p>I am just another sentence written by Mungi.</p>
</body>
</html>"""
print("The HTML String is:")
print(myString)
parserObj = fromstring(myString)
outputString = str(parserObj.text_content())
print("The output String is:")
print(outputString)
```

```
The HTML String is:
<!DOCTYPE html>
<html>
<head>
<title>venkateshmungi</title>
</head>
<body>
<h1>I am a sentence inside an HTML string.</h1>
<p>I am just another sentence written by Mungi.</p>
</body>
</html>
The output String is:


venkateshmungi


I am a sentence inside an HTML string.
I am just another sentence written by Mungi.
```

Remove HTML tags from string in python Using the Beautifulsoup Module

Like the lxml module, the BeautifulSoup module also provides us with various functions to process text data. To remove HTML tags from a string using the BeautifulSoup module, we can use the BeautifulSoup() method and the get_text() method. In this approach, we will first create a parser to parse the string that contains HTML tags using the BeautifulSoup() method. The BeautifulSoup() method takes the original string as its first input argument and the type of parser to be created as its second input argument, which is optional. After execution, it returns the parser. We can invoke the get_text() method on the parser to get the output string.

In [5]:
```
import bs4
pattern = '<[^<]+?>'
myString = """<!DOCTYPE html>
<html>
<head>
<title>Mungi</title>
</head>
<body>
<h1>I am a sentence inside an HTML string.</h1>
<p>I am just another sentence written by Venkatesh.</p>
</body>
</html>"""
print("The HTML String is:")
print(myString)
parserObj = bs4.BeautifulSoup(myString)
outputString = parserObj.get_text()
print("The output String is:")
print(outputString)
```

```
The HTML String is:
<!DOCTYPE html>
<html>
<head>
<title>Mungi</title>
</head>
<body>
<h1>I am a sentence inside an HTML string.</h1>
<p>I am just another sentence written by Venkatesh.</p>
</body>
</html>
The output String is:


Mungi


I am a sentence inside an HTML string.
I am just another sentence written by Venkatesh.
```

Remove HTML tags from string in python Using Regular Expressions

Regular expressions are one of the best ways to process text data. We can also remove HTML tags from string in python using regular expressions. For this, we can use the sub() method defined in the regex module.

The sub() method takes the pattern of the sub-string that needs to be replaced as its first argument, the string that will be substituted at the place of the replaced sub-string as the second input argument, and the original string as the third input argument.

After execution, it returns the modified string by replacing all the occurrences of the substring given as the first input argument with the substring given as the second input argument in the original string.

To remove HTML tags from string in python using the sub() method, we will first define a pattern that represents all the HTML tags. For this, we will create a pattern that reads all the characters inside an HTML tag <>. The pattern is as follows.

After creating the pattern, we will substitute each substring having the defined pattern with an empty string "" using the sub() method. In this way, we can remove the HTML tags from any given string in Python.

In [6]:
```
import re
pattern = '<[^<]+?>'
myString = """<!DOCTYPE html>
<html>
<head>
<title>Venkatesh</title>
</head>
<body>
<h1>I am a sentence inside an HTML string.</h1>
<p>I am just another sentence written by Mungi.</p>
</body>
</html>"""
print("The HTML String is:")
print(myString)
outputString = re.sub(pattern, "", myString)
print("The output String is:")
print(outputString)
```

```
The HTML String is:
<!DOCTYPE html>
<html>
<head>
<title>Venkatesh</title>
</head>
<body>
<h1>I am a sentence inside an HTML string.</h1>
<p>I am just another sentence written by Mungi.</p>
</body>
</html>
The output String is:



Venkatesh


I am a sentence inside an HTML string.
I am just another sentence written by Mungi.
```

In [7]:
```
def remove_html(string):
    tags = False
    quote = False
    output = ""
    for ch in string:
            if ch == '<' and not quote:
                tag = True
            elif ch == '>' and not quote:
                tag = False
            elif (ch == '"' or ch == "'") and tag:
                quote = not quote
            elif not tag:
                output = output + ch
    return output
text=input("Enter String:")
new_text=remove_html(text)
print(f"Text without html tags: {new_text}")
```

```
Enter String:<div class="header"> Welcome to my website </div>
Text without html tags:  Welcome to my website
```

In [8]:
```
import xml.etree.ElementTree
def remove_html(string):
    return ''.join(xml.etree.ElementTree.fromstring(string).itertext())
text=input("Enter String:")
new_text=remove_html(text)
print(f"Text without html tags: {new_text}")
```

```
Enter String:<div class="header"> Welcome to my website </div>
Text without html tags:  Welcome to my website
```

In [ ]: