ROHAN NYATI

500075940

R177219148

BATCH – 5 (Ai & Ml)

Experiment -2

Q) Explore the (regular expressions) library in detail. What kind of analysis you can in it. Explain with proper codes.

A Regular Expressions (RegEx) is a special sequence of characters that uses a search pattern to find a string or set of strings. It can detect the presence or absence of a text by matching with a particular pattern, and also can split a pattern into one or more sub-patterns. Python provides a “re” module that supports the use of regex in Python. Its primary function is to offer a search, where it takes a regular expression and a string. Here, it either returns the first match or else none.

Metacharacters in RegEX :-

1) \	Used to drop the special meaning of character following it
2) []	Represent a character class
3) ^	Matches the beginning
4) $	Matches the end
5) .	Matches any character except newline
6) |	Means OR (Matches with any of the characters separated by it.
7) ?	Matches zero or one occurrence
8) *	Any number of occurrences (including 0 occurrences)
9) +	One or more occurrences
10) {}	Indicate the number of occurrences of a preceding regex to match.
11) (<regex>)	Enclose a group of Regex

<u>Special Sequences in Regular Expressions</u> :-

1) \b

=> \b returns a match where the specified pattern is at the beginning or at the end of a word.

2) \d

=> \d returns a match where the string contains digits (numbers from 0-9).

3) \D

=> \D returns a match where the string does not contain any digit. It is basically the opposite of \d.

4) \w

=> \w helps in extraction of alphanumeric characters only (characters from a to Z, digits from 0-9, and the underscore _ character)

5) \W

=> \W returns match at every non-alphanumeric character. Basically opposite of \w.

6) \s

=> \s Matches any whitespace character.

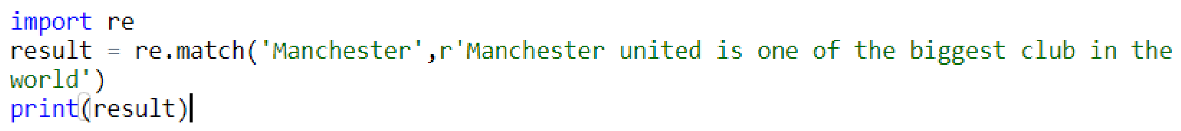
7) \S

=> Matches any non-whitespace character.

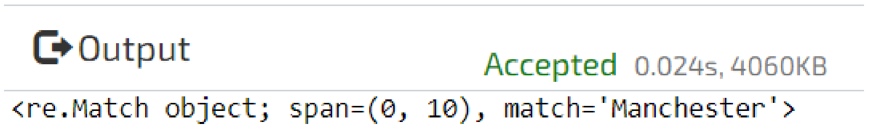
<u>Built-in RegEx modules in python</u> :-

1) re.match(pattern, string):-

The re.match function returns a match object on success and none on failure.



```
import re
result = re.match('Manchester',r'Manchester united is one of the biggest club in the world')
print(result)
```

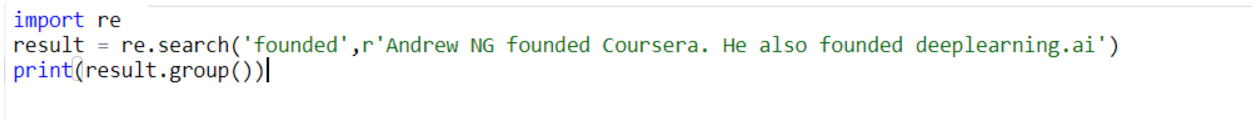


Output Accepted 0.024s, 4060KB

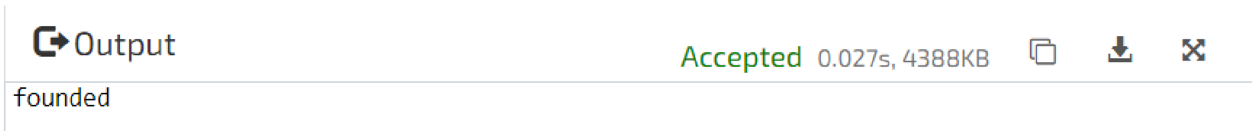
```
<re.Match object; span=(0, 10), match='Manchester'>
```

2) re.search(pattern, string):-

Matches the first occurrence of a pattern in the entire string(and not just at the beginning).



```
import re
result = re.search('founded',r'Andrew NG founded Coursera. He also founded deeplearning.ai')
print(result.group())
```

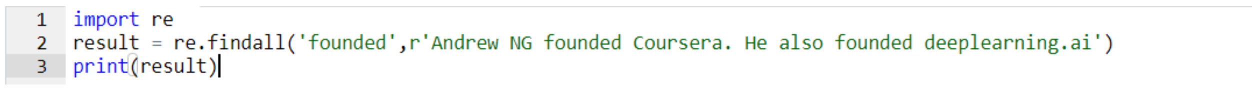


Output Accepted 0.027s, 4388KB

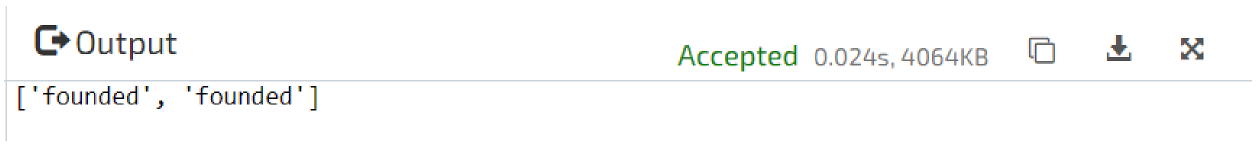
```
founded
```

3) re.findall(pattern, string) :-

It will return all the occurrences of the pattern from the string.



```
import re
result = re.findall('founded',r'Andrew NG founded Coursera. He also founded deeplearning.ai')
print(result)
```



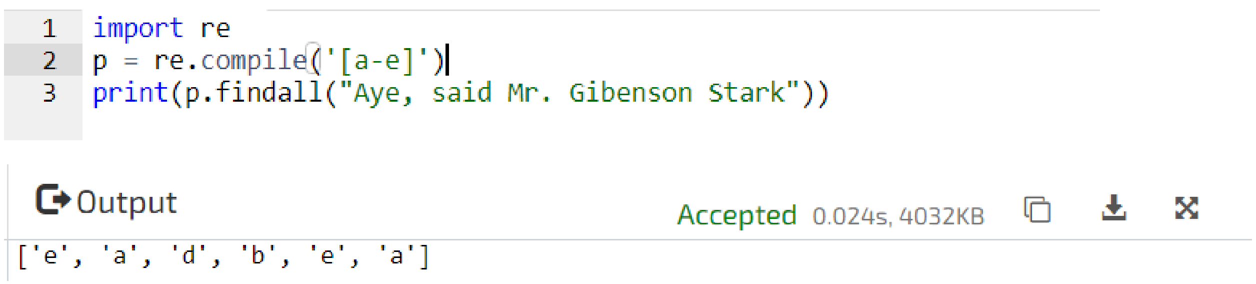
Output Accepted 0.024s, 4064KB

```
['founded', 'founded']
```

4) re.compile() :-

Regular expressions are compiled into pattern objects, which have methods for various operations such as searching for pattern matches or performing string substitutions.

In this ex - compile() creates regular expression character class [a-e], which is equivalent to [abcde]. class [abcde] will match with string with – ('a', 'b', 'c', 'd', 'e'). Upon finding it will output the present char on the basis of their order in string.



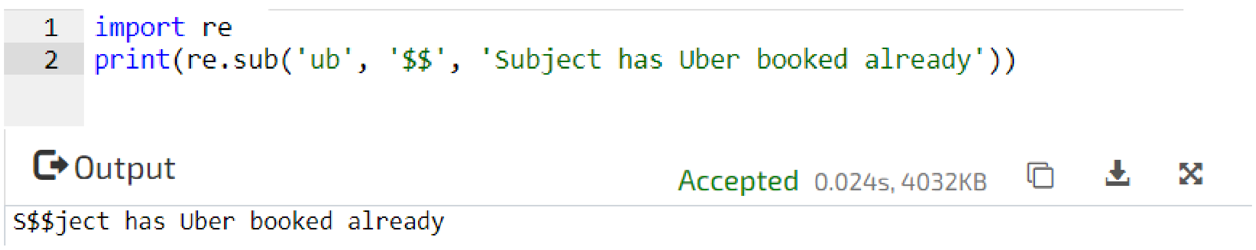
```python
import re
p = re.compile('[a-e]')
print(p.findall("Aye, said Mr. Gibenson Stark"))
```

Output   Accepted 0.024s, 4032KB

```
['e', 'a', 'd', 'b', 'e', 'a']
```

5) re.sub() :-

The ‘sub’ in the function stands for SubString, a certain regular expression pattern is searched in the given string(3rd parameter), and upon finding the substring pattern is replaced by 2nd parameter, count checks and maintains the number of times this occurs.

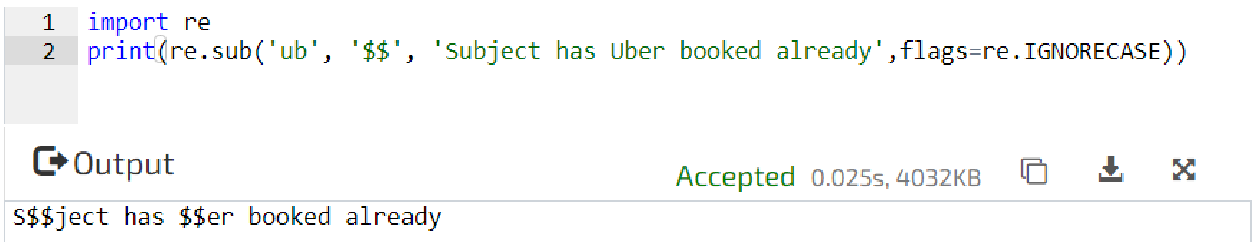


```python
import re
print(re.sub('ub', '$$', 'Subject has Uber booked already'))
```

Output   Accepted 0.024s, 4032KB

```
S$$ject has Uber booked already
```

Without case-sensitive



```python
import re
print(re.sub('ub', '$$', 'Subject has Uber booked already',flags=re.IGNORECASE))
```

Output   Accepted 0.025s, 4032KB

```
S$$ject has $$er booked already
```