Roll No:18 Rebecca Dias BECMPNA

Pid:182027

Experiment 3

Aim : To implement the various regular expressions in Python

Theory : Regular expressions can be used to specify strings extracted from a document. Regular Expressions play an important role to a set of tasks collectively called text normalization. Normalizing text means converting it to a more convenient, standard form. The simplest kind of regular expression is a sequence of simple characters. To search for woodchuck, we type woodchuck.Regular expressions are case sensitive.With the use of the square braces [and],the string of characters inside the braces specifies a disjunction of characters to match. Kleene * means “zero or more occurrences of the immediately previous character or regular expression”. So /a*/ means “any string of zero or more as”. The special character period (/./) is a wildcard expression that matches any single character (except a carriage return). Anchors are special characters that anchor regular expressions to particular places in a string. The most common anchors are the caret ˆ and the dollar sign $. Disjunction Operator which is also called pipe symbol (|) is used to select either of the strings. E.g /cat|dog/ implies cat or dog.

Predefined Function

```
import re

def check_pattern(pattern,  text):
  if re.search(pattern,  text):
          return 'Found a match!'
  else:
          return 'Not matched!'
```

Write a Python program that matches a string that has an a followed by zero or more b

```
check_pattern("ab*?","ac")
```

```
     'Found a match!'
```

Write a Python program that matches a string that has an a followed by one or more b

```
check_pattern('ab+?',"ac")
```

```
    'Not matched!'
```

Write a Python program to find sequences of lowercase letters joined with a underscore

```
check_pattern('^[a-z]+_[a-z]+$',"aaa_ccc")
```

```
    'Found a match!'
```

Write a Python program to find the sequences of one upper case letter followed by lower case letters

```
check_pattern('^[A-Z][a-z]+$',"Aaaa")
```

```
    'Found a match!'
```

Write a Python program that matches a word containing 'z'

```
check_pattern('\w*z\w*',"word wordz")
```

```
    'Found a match!'
```

Write a Python program that matches a word containing 'z', not at the start or end of the word

```
check_pattern('\Bz\B',"wordz")
```

```
    'Not matched!'
```

Write a Python program to match a string that contains only upper and

```
check_pattern('^[a-zA-Z0-9_]*$',"aaAA11__")

     'Found a match!'
```

Write a Python program to search the numbers (0-9) of length between 1 to 3 in a given string

```
check_pattern(r"([0-9]{1,3})","abc 123")

     'Found a match!'
```

Write a Python program to search some literals strings in a string. Go to the editor Sample text : 'The quick brown fox jumps over the lazy dog.' Searched words : 'fox', 'dog', 'horse'

```
patterns = [ 'fox', 'dog', 'horse' ]
text = 'The quick brown fox jumps over the lazy dog.'
for pattern in patterns:
    print('Searching for "%s" in "%s" : ' % (pattern, text),end="")
    if re.search(pattern,  text):
        print('Matched!')
    else:
        print('Not Matched!')

     Searching for "fox" in "The quick brown fox jumps over the lazy dog." : Matched!
     Searching for "dog" in "The quick brown fox jumps over the lazy dog." : Matched!
     Searching for "horse" in "The quick brown fox jumps over the lazy dog." : Not Matched!
```

Write a Python program to replace whitespaces with an underscore and vice versa

```
text = 'Hello World'
text = text.replace (" ", "_")
print(text)
text =text.replace ("_", " ")
print(text)

     Hello_World
     Hello World
```

Write a Python program to separate and print the numbers of a given string.

```
text = "Ten 10, Twenty 20"
result = re.split("\D+", text)
for element in result:
    print(element)
```

```

10
20
```

. Write a Python program to find all words starting with a or e in a given string.

```
text = "oo eooo asdas cer oo"
re.findall("[ae]\w+", text)
```

```
['eooo', 'asdas', 'er']
```

Write a Python program to abbreviate 'Road' as 'Rd.' in a given string.

```
street = 'Hello Road'
re.sub('Road$', 'Rd.', street)
```

```
'Hello Rd.'
```

Write a Python program to remove multiple spaces in a string.

```
text = 'Hello      World'
print("Original string:",text)
print("Without extra spaces:",re.sub(' +',' ',text))
```

```
Original string: Hello      World
Without extra spaces: Hello World
```

Write a Python program to remove everything except alphanumeric characters from a string.

```
text = '**###&&&***Hello Wolrd ----== 12. '
pattern = re.compile('[\W_]+')
pattern.sub('', text)
```

```
'HelloWolrd12'
```

Conclusion

In this experiment we learned about regular expressions and implemented the programming exercises using python programming language

✓ 0s completed at 2:16 PM