task 45 Rejex balu

July 30, 2022

[1]:

```
f='python \n programming'
print(f)
```

```
python
 programming
```

[2]:

```
# raw string
r=R'python \n programming'
print(r)
```

```
python \n programming
```

[3]:

```
import re
```

[4]:

```
# extract the numbers
```

[5]:

```
address = '8-28  89 main,4th cross, 123,road,marathalli,567 bangalore,517520'
```

[6]:

```
add_nos=re.findall(R'\d+',address)
print(f'sorting the only nums from address {add_nos}')
```

```
sorting the only nums from address ['8', '28', '89', '4', '123', '567',
'517520']
```

[7]:

```
add_dig=re.findall(R'\d{1}',address)
print(f'sorting the only nums from address {add_dig}')
```

```
sorting the only nums from address ['8', '2', '8', '8', '9', '4', '1', '2', '3',
'5', '6', '7', '5', '1', '7', '5', '2', '0']
```

[8]:

```
add_dig=re.findall(R'\d{2}',address)
print(f'sorting the only nums from address {add_dig}')
```

```
sorting the only nums from address ['28', '89', '12', '56', '51', '75', '20']
```

[9]:

```
add_dig=re.findall(R'\d{1,6}',address)
print(f'sorting the only nums from address {add_dig}')
```

```
sorting the only nums from address ['8', '28', '89', '4', '123', '567',
'517520']
```

[10]:

```
s = '''
<html>
<head>
<title>Current IP Address Allocations
</title>
</head>
<body>
IP Address are 172.45.78.109
LoopBack Address: 127.0.0.1
Computer 1: 10.67.89.101
Computer 2: 11.67.98.102
Computer 3: 12.68.98.102
</body>
</html>
'''
```

[11]:

```
ip_s=re.findall(r'\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3}',s)

print(f'ip address are -: {ip_s}')
```

```
ip address are -: ['172.45.78.109', '127.0.0.1', '10.67.89.101', '11.67.98.102',
'12.68.98.102']
```

[12]:

```
ip_s1=re.findall(r"1[0-1]\.\d{1,3}\.\d{1,3}\.\d{1,3}", s)
print(f'ip address are -: {ip_s1}')
```

```
ip address are -: ['10.67.89.101', '11.67.98.102']
```

[13]:

```
ip_s1=re.findall(r"1[0|1]\.\d{1,3}\.\d{1,3}\.\d{1,3}", s)
print(f'ip address are -: {ip_s1}')
```

```
ip address are -: ['10.67.89.101', '11.67.98.102']
```

[14]:

```
ip_s1=re.findall(r"1[01]\.\d{1,3}\.\d{1,3}\.\d{1,3}", s)
print(f'ip address are -: {ip_s1}')
```

```
ip address are -: ['10.67.89.101', '11.67.98.102']
```

[15]:

```
ip_s0=re.findall(r"10\.\d{1,3}\.\d{1,3}\.\d{1,3}", s)
print(f'ip address are -: {ip_s0}')
```

```
ip address are -: ['10.67.89.101']
```

[16]:

```
print("Find all matches for format Month day")

matches = re.findall(r"[A-Z][a-z]+\s\d{1,2}","These are the match dates June␣
  ↪24, August 9, Dec 12")
print(f'gives Month Date format - {matches}')
```

```
matches = re.findall(r"[A-Z][a-z]+\s(\d{1,2})", "These are the match dates June␣
  ↪24, August 9, Dec 12")
print(f'gives Date format - {matches}')

matches = re.findall(r"([A-Z][a-z]+)\s(\d{1,2})", "These are the match dates␣
  ↪June 24, August 9, Dec 12")
print(f'gives tuple of Month & Date format - {matches}')
```

```
Find all matches for format Month day
gives Month Date format - ['June 24', 'August 9', 'Dec 12']
gives Date format - ['24', '9', '12']
gives tuple of Month & Date format - [('June', '24'), ('August', '9'), ('Dec',
'12')]
```

[17]:
```
s = "purple alice@google.com abcde helloab@abc.com ---@gmail.com 23@gmail.com␣
  ↪my23@gmail.com _@gmail.com"

emails = re.findall(r"\w+@\w+\.\w+", s)
print(emails)
```

```
['alice@google.com', 'helloab@abc.com', '23@gmail.com', 'my23@gmail.com',
'_@gmail.com']
```

[18]:
```
s = "purple alice@google.com abcde helloab@abc.com ---@gmail.com 23@gmail.com␣
  ↪my23@gmail.com _@gmail.com"

emails = re.findall(r"[A-Za-z]+@\w+\.\w+", s)
print(f'starts with alphabets only {emails}')
```

```
starts with alphabets only ['alice@google.com', 'helloab@abc.com']
```

[19]:
```
s = "purple alice@google.com abcde helloab@abc.com ---@gmail.com 23@gmail.com␣
  ↪my23@gmail.com _@gmail.com"

emails = re.findall(r"[\d\w]+@\w+\.\w+", s)
print(f'starts with alphabets only {emails}')
```

```
starts with alphabets only ['alice@google.com', 'helloab@abc.com',
'23@gmail.com', 'my23@gmail.com', '_@gmail.com']
```

[20]:
```
new_st2 = 'Friend in need is 23 friend in 453214 deed'
nr6 = re.findall('\d+',new_st2)
print(nr6)
```

```
['23', '453214']
```

[21]:
```
new_st2 = 'Friend in need is 23 friend in 453214 deed'
nr6 = re.findall('\D',new_st2)
print(nr6)
```

```
['F', 'r', 'i', 'e', 'n', 'd', ' ', 'i', 'n', ' ', 'n', 'e', 'e', 'd', ' ', 'i',
's', ' ', ' ', 'f', 'r', 'i', 'e', 'n', 'd', ' ', 'i', 'n', ' ', ' ', 'd', 'e',
'e', 'd']
```

[22]:

```
new_st2 = 'Friend in need is 23 friend in 453214 deed'
nr6 = re.findall('\w',new_st2)
print(nr6)
```

```
['F', 'r', 'i', 'e', 'n', 'd', 'i', 'n', 'n', 'e', 'e', 'd', 'i', 's', '2', '3',
'f', 'r', 'i', 'e', 'n', 'd', 'i', 'n', '4', '5', '3', '2', '1', '4', 'd', 'e',
'e', 'd']
```

[23]:

```
new_st2 = 'Friend in need is 23 friend in 453214 deed'
nr6 = re.findall('\w+',new_st2)
print(nr6)
```

```
['Friend', 'in', 'need', 'is', '23', 'friend', 'in', '453214', 'deed']
```

[24]:

```
new_st2 = 'Friend in need is 23 friend in 453214 deed'
nr6 = re.findall('[23]',new_st2) # 2,3
print(nr6)
```

```
['2', '3', '3', '2']
```

[25]:

```
# search
```

[26]:

```
target_string = "Emma is a Python developer \n Emma also knows ML and AI"

# caret (^) matches at the beginning of a string
result = re.search(r"^\w{4}", target_string)

print(result.group())
```

```
Emma
```

[27]:

```
str1 = "Emma is a Python developer \nEmma also knows ML and AI"
# dollar sign($) to match at the end of the string
result = re.search(r"\w{2}$", str1)
print(result.group())
```

```
AI
```

[28]:

```
st = 'In a world where you can be anything, be kind'

match_object = re.search('In',st)
print(f'type is object {match_object}')
```

```
type is object <re.Match object; span=(0, 2), match='In'>
```

[29]:

```
match_object1 = re.search('be',st)
print(f'type is object {match_object1}')
```

```
type is object <re.Match object; span=(25, 27), match='be'>
```

[30]:

```
match_object.start()
```

[30]:

```
0
```

[31]:

```
match_object1.start()
```

[31]:

```
25
```

[32]:

```
match_object.span()
```

[32]:

```
(0, 2)
```

[33]:

```
match_object1.span()
```

[33]:

```
(25, 27)
```

[34]:

```
source_str  = 'we need to inform him with the latest information'

info = re.search('inform', source_str)
info
```

[34]:

```
<re.Match object; span=(11, 17), match='inform'>
```

[35]:

```
if re.search('inform', source_str):
    print('inform is there')
```

```
inform is there
```

[36]:

```
randomstr = 'here is \\kane'

print(randomstr)

re.search(r'\\kane',randomstr)
```

```
here is \kane
```

[36]:

```
<re.Match object; span=(8, 13), match='\\kane'>
```

[37]:

```
# split
```

[38]:

```
r=re.split(' ',st)
r
```

[38]:

```
['In', 'a', 'world', 'where', 'you', 'can', 'be', 'anything,', 'be', 'kind']
```

[39]:

```
r1=re.split('e',st)
r1
```

```
[39]: ['In a world wh', 'r', ' you can b', ' anything, b', ' kind']
```

```
[40]: r12=re.split('e',st,2)
      r12
```

```
[40]: ['In a world wh', 'r', ' you can be anything, be kind']
```

```
[41]: s='welcome to regex programming using python'
      print(f'the value of s    :{s}')

      lstval=re.split(r'\s',s)
      print(f'regix split value of s    :{lstval}')

      lstval2=re.split(r'\s+',s)
      print(f'regix split value of s    :{lstval2}')
```

```
the value of s    :welcome to regex programming using python
regix split value of s    :['welcome', 'to', 'regex', 'programming', 'using',
'python']
regix split value of s    :['welcome', 'to', 'regex', 'programming', 'using',
'python']
```

```
[ ]:
```

```
[42]: # sub-subtitute
```

```
[43]: sb = re.sub('e','E',st)
      sb
```

```
[43]: 'In a world whErE you can bE anything, bE kind'
```

```
[44]: sb1 = re.sub('e','E',st,1)
      sb1
```

```
[44]: 'In a world whEre you can be anything, be kind'
```

```
[45]: # compile
```

```
[46]: a = 'hat mat rat pat '

      reg = re.compile('[r]at')
      reg
```

```
[46]: re.compile(r'[r]at', re.UNICODE)
```

```
[47]: rplce = reg.sub('FOOD',a)
      rplce
```

```
[47]: 'hat mat FOOD pat '
```

[48]:

```
rplc = re.sub('rat','FOOD',a)
rplc
```

[48]:

```
'hat mat FOOD pat '
```

[ ]:

```

```

[49]:

```
chelsea = '''keep the blue flag
flying high
chelsa
'''
chelsea
```

[49]:

```
'keep the blue flag\nflying high\nchelsa\n'
```

[50]:

```
new_str = re.sub('\n',' ',chelsea)
new_str
```

[50]:

```
'keep the blue flag flying high chelsa '
```

[51]:

```
comp = re.compile('\n')

new =comp.sub(' ',chelsea)
new
```

[51]:

```
'keep the blue flag flying high chelsa '
```

[52]:

```
phone_no = '''
444-122-1234
123-122-78999
111-123-23
67-7890-2019
'''
# 3 digit @ start & middle, end -4 digit

reg = re.findall(r'\d{3}\-\d{3}\-\b\d{4}\b',phone_no)
reg
```

[52]:

```
['444-122-1234']
```

[53]:

```
phone_no = '''
444-122-1234
123-122-78999
111-123-23
67-7890-2019
'''
# 3 digit @ start & middle, end -4 digit
```

```
reg = re.findall(r'\d{3}\-\d{3}\-\b\d{4}',phone_no)
reg
```

[53]:

```
['444-122-1234', '123-122-7899']
```

[ ]:

```

```

[54]:

```
# match
```

[55]:

```
import re

target_string = "Jess loves Python and pandas fives"
# Match six-letter word
pattern = r"\b\w{5}\b"

# # match() method
result = re.match(pattern, target_string)
print(result)


# search() method
result = re.search(pattern, target_string)
print(result.group())


# findall() method
result = re.findall(pattern, target_string)
print(result)
```

```
None
loves
['loves', 'fives']
```

[56]:

```
target_string = "Jess loves Python and pandas fives"
# Match six-letter word
pattern = r"\b\w{6}\b"

# # match() method
result = re.match(pattern, target_string)
print(result)


# search() method
result = re.search(pattern, target_string)
print(result.group())


# findall() method
```

```
result = re.findall(pattern, target_string)
print(result)
```

```
None
Python
['Python', 'pandas']
```

[58]:

```
target_string = "Jess loves Python and pandas fives"
# Match six-letter word
pattern = r"\b\w{3}\b"

# # match() method
result = re.match(pattern, target_string)
print(result)


# search() method
result = re.search(pattern, target_string)
print(result.group())


# findall() method
result = re.findall(pattern, target_string)
print(result)
```

```
None
and
['and']
```

[60]:

```
# what is diff match search findout
```

[ ]:

[ ]: