In [4]:
```
# Regx
# in unix shell script ->grep ; sed ;awk +Regx
# in perl =~ !~

# in python -> module(re) + conditional + loop + fileHandling
# ----------
# search
# substitue
#  +
# input validation (or) type validation
# ex: n=input("Enter any digits:")
#    ---
# ex: function(str)

import re

# re.search() ->search the pattern from input
#                            ------------------
#                             if pattern is matched -><Ack>
#                             if pattern is NOT matched ------->None
# re.findall() -> search the pattern from input
#                             ---------
#                                |__[result]/ []
#

# re.search("PatternString","inputString") -><ack>/None
#
# re.findall("PatternString","inputString") ->[Result]/[]
#
var="root:x:bin:bash:/bin:/bin/bash:/root:bash:"

obj=re.search("bash",var)
print(obj.group())

print(re.findall("bash",var))
```

```
bash
['bash', 'bash', 'bash']
```

In [9]:
```
var="101,ram,sales,pune,1000"

print(re.search("Sales",var))
print(re.search("Sales",var,re.I))
bool(re.search("Sales",var,re.I))
```

```
None
<re.Match object; span=(8, 13), match='sales'>
```

Out[9]:
```
False
```

In [13]:
```
if(re.search("Sales",var,re.I)):
    print("Yes - pattern is matched")
    print(var)
else:
    print("No -pattern is not matched")
```

```
Yes - pattern is matched
101,ram,sales,pune,1000
```

In [14]:
```
# findstr/grep commands
# ----------------------
# STEP 1: read the data from input file - linebyline
# STEP 2: search the pattern from each inputline
# STEP 3: print matched pattern only
#
# C:\>findstr sales emp.csv  (or) root@host~]# grep sales emp.csv
#
for var in open("D:\\emp.csv"):
    if(re.search("sales",var)):
        print(var.strip())
'''
>>> for var in open("emp.csv"):
...     if(re.search("sales",var)):
...             print(var.strip())
...
ram,sales,pune,1000
xerox,sales,chennai,45900
theeb,sales,hyd,5678
>>>
'''
```

```
ram,sales,pune,1000
xerox,sales,chennai,45900
theeb,sales,hyd,5678
```

In [ ]:
```

```

In [ ]:

```
'''
^  --> ^pattern
$  --> pattern$
^pattern$ ->pattern only - like condition
.(dot) -> match any single character
                ---
                A-Za-z0-9 space specialchar
*  - zero or more
.* - list of all
character class - []
--------------------
matching any single
[Aa]run
---------
  |--> Arun  arun
[Aavb]run
-----------
 |--> Arun arun vrun brun
 [Aa][BC]45
 ===========
  AB45
  AC45
  aB45
  aC45

[A-Z] - match any single uppercase char
[a-z] - match any single lowercase char
[A-Za-z] - match any alpha
[0-9] - match any single digit -->\d
[A-Za-z0-9] - match any alpha and number -->\w
\s - matching space

[aT5] -> search 'a' 'T' '5'
^[aT5] ->line starts 'a' 'T' '5'
[aT5]$ ->line ends with 'a' 'T' '5'

[^aT5] - NOT matching 'a' 'T' '5'
^$ - empty line

ERE
====
 |  ()  +  {}

re.search("Pattern1|Pattern2|Pattern3|Pattern4")
any one pattern is matched ; any order ;any where ->True

re.search("(Patter1)(Pattern2)","input")
Both pattern should match ; same order - like logical and operator

<Pattern>+  -> 1 or more
a+ ->a aaaaaaaaaaa //matched
ab+ -->ab abbbbbbbbbbbbb //matched
ab+c ==>abc abbbbbbbbbbbbbbbbbbbbbbbbc //matched

ab+c ==> abbbbbbbbbbbbbbbbbb,bbbbbbbbbbbbbbbbbc //not-matched
{}
```

```
<Pattern>{n} ==> ntimes
ab{2}c ==> abbc //matched ;  abc abbbc abbbbbbbbc //not-matched

^[A-Z][0-9][0-9][0-9][a-z][a-z]$ -->^[A-Z][0-9]{3}[a-z]{2}$ ->^[A-Z]\d{2}[a-z]$

<Pattern>{n,} ==> minimum 'n' times - maximum nolimit

ab{2,}c ==> abc - not matched
--------
 |____ abbc abbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbc //matched

re.search("^\s+\d{3,}")

ab+c ----same as --> ab{1,}c
 ---                  -----

<Pattern>{n,m} ->minimum 'n' times - maximum 'm' times
ab{2,4}c
=========
 |__abbc abbbc abbbbc //matched

 abc abbbbbc //not-matched
'''
```

In [19]:
```
n=input("Enter any two digits:")
if not(re.search("^[0-9][0-9]$",n)):# ^\d\d$,n
    print("Invalid format")
```

```
Enter any two digits:4
Invalid format
```

In [29]:
```
var='Afg#$$#35434sfdsf 5'
re.findall("[^A-Za-z0-9\s]",var) # (or)re.findall("[^\w\s]",var)
#        not matching alpha number and space ->match specialchars
# ^[A-Za-z0-9\s] -> line starts with A-Za-z0-9<space>
#  [^A-Za-z0-9\s] -> NOT matching any A-Za-z0-9<space>
```

Out[29]:
```
['#', '$', '$', '#']
```

In [34]:
```
re.search("python|java|sql","programming perl and java")
```

Out[34]:
```
<re.Match object; span=(21, 25), match='java'>
```

In [36]:
```
re.search("^[A-Za-z].*\d$|^\s|^\d.*\d$"," sample data")
```

Out[36]:
```
<re.Match object; span=(0, 1), match=' '>
```

In [42]:
```
re.search("(python).*(java)","programming python and java")
```

Out[42]:
```
<re.Match object; span=(12, 27), match='python and java'>
```

In [43]:
```
re.search("(java).*(python)","programming python and java")
```

In [46]:
```
# valid URL -> https ... org  https .... com
#              http      org  http       com
#
re.search("(^https|^http).*(com$|org$)","https://www.abc.org")
```

Out[46]:
```
<re.Match object; span=(0, 19), match='https://www.abc.org'>
```

In [48]:
```
var="SAMPLE 5 Data Reports on 13th sep 2020 code 43322 TIME:11:43:53 AM IST"
re.findall("[0-9]+",var)
```

Out[48]:
```
['5', '13', '2020', '43322', '11', '43', '53']
```

In [50]:
```
re.findall("[0-9]+[^\s\w]",var)
```

Out[50]:
```
['11:', '43:']
```

In [52]:
```
re.findall("\d+[a-z]+",var)
```

Out[52]:
```
['13th']
```

In [ ]:
```
re.sub("OldPattern","replaced_String","input_string") ->String
    #   (Regx)                                          |___replaced
    #                                                  |__inputstring(if old pa
```

In [58]:
```
var="101,ram,sales,pune,1000,sales,sales,SALES"
re.sub("sales","QA",var,0,re.I)
```

Out[58]:
```
'101,ram,QA,pune,1000,QA,QA,QA'
```

In [59]:
```
var="SAMPLE 5 Data Reports on 13th sep 2020 code 43322 TIME:11:43:53 AM IST"

re.sub("\d+","-",var)
```

Out[59]:
```
'SAMPLE - Data Reports on -th sep - code - TIME:-:-:- AM IST'
```

In [62]:
```
re.sub("\d+","",var) # deleting all the digits

re.sub("sales.","","ram,sales,pune,1000") # delete a sales word
```

Out[62]:
```
'ram,pune,1000'
```

In [63]:
```
with open("D:\\emp.csv") as FH:
    for var in FH:
        s=re.sub("sales","ADMIN",var)
        print(s.strip())
```

```
ram,ADMIN,pune,1000
ashi,prod,bglore,2345
xerox,ADMIN,chennai,45900
yahoo,prod,pune,32450
anu,HR,hyd,4560
biju,prod,bglore,4567
vijay,hr,chennai,3453
theeb,ADMIN,hyd,5678
nithin,prod,pune,1236
```

In [64]:
```
with open("D:\\emp.csv") as FH:
    for var in FH:
        if(re.search("sales",var)):
            s=re.sub("sales","ADMIN",var)
            print(s.strip())
```

```
ram,ADMIN,pune,1000
xerox,ADMIN,chennai,45900
theeb,ADMIN,hyd,5678
```

In [65]:
```
with open("D:\\emp.csv") as FH:
    for var in FH:
        s=re.sub("chennai|hyd","MUMBAI",var,0,re.I)
        print(s.strip())
```

```
ram,sales,pune,1000
ashi,prod,bglore,2345
xerox,sales,MUMBAI,45900
yahoo,prod,pune,32450
anu,HR,MUMBAI,4560
biju,prod,bglore,4567
vijay,hr,MUMBAI,3453
theeb,sales,MUMBAI,5678
nithin,prod,pune,1236
```

In [70]:
```
V="17:23:22 up  9:37,  4 users,  load average: 0.00, 0.01, 0.05"
# how to delete all the digits and specialchars from given string

print(re.sub("\d+|[^\w\s]+","",V))
```

```
 up     users  load average
```

In [76]:
```
L=['154G','343M','344G','120K']
# how to calculate sum of the list?

t=0
for var in L:
    r=re.sub(".$","",var)
    print(r,type(r))
```

```
154 <class 'str'>
343 <class 'str'>
344 <class 'str'>
120 <class 'str'>
```

In [77]:
```
t=0
for var in L:
    r=re.sub(".$","",var)
    t=t+int(r)
else:
    print(t)
```

```
961
```

In [84]:
```
s="root:x:bin:bash"
print(s.split(":"))
print(re.split(":",s))
```

```
['root', 'x', 'bin', 'bash']
['root', 'x', 'bin', 'bash']
```

In [83]:
```
s="root:x-bin~bash"
# re.split("Pattern","inputstring") ->[list output]
re.split("[^\w\s]",s)
```

Out[83]:
```
['root', 'x', 'bin', 'bash']
```

In [88]:
```
var="SAMPLE 5 Data Reports on 13th sep 2020 code 43322 TIME:11:43:53 AM IST"
re.split("\d+|[^\w\s]+",var)
```

Out[88]:
```
['SAMPLE ',
 ' Data Reports on ',
 'th sep ',
 ' code ',
 ' TIME',
 '',
 '',
 '',
 '',
 '',
 ' AM IST']
```

In [89]:
```
re.findall("\d+|[^\w\s]+",var)
```

Out[89]:
```
['5', '13', '2020', '43322', ':', '11', ':', '43', ':', '53']
```

In [92]:
```
#re.match("PAttern","Inputstring") -->re.search("^Pattern","Inputstring")
print(re.match("sales","arun,sales")) # re.search("^sales","arun,sales")
re.match("sales","sales asfsdfsd")
```

```
None
```

Out[92]:
```
<re.Match object; span=(0, 5), match='sales'>
```

In [ ]:
```
>>> for var in os.popen("ps -f"):
...     print(var.strip())
...
UID        PID  PPID  C STIME TTY          TIME CMD
apelix    4097  4090  0 10:14 pts/0    00:00:00 bash
apelix    4188  4097  0 10:17 pts/0    00:00:00 python
apelix    4455  4188  0 10:51 pts/0    00:00:00 [ps] <defunct>
apelix    4458  4188  0 10:52 pts/0    00:00:00 [grep] <defunct>
apelix    5560  4188  0 14:38 pts/0    00:00:00 [sh] <defunct>
apelix    5566  4188  0 14:38 pts/0    00:00:00 sh -c ps -f
apelix    5567  5566  0 14:38 pts/0    00:00:00 ps -f
>>>
>>> for var in open("/home/apelix/emp.csv"):
...     print(var.strip())
...
ram,sales,pune,1000
ashi,prod,bglore,2345
xerox,sales,chennai,45900
yahoo,prod,pune,32450
anu,HR,hyd,4560
biju,prod,bglore,4567
vijay,hr,chennai,3453
theeb,sales,hyd,5678
nithin,prod,pune,1236


>>>
>>> for v in os.popen("ps -e"):
...     if(re.search("[seh]$",v)):
...             print(v.strip())
...
9 ?        00:00:00 netns
10 ?        00:00:00 sync_supers
1894 ?        00:00:00 console-kit-dae
2298 ?        00:00:00 dbus-launch
2347 ?        00:00:03 unity-2d-launch
2351 ?        00:00:00 dconf-service
2355 ?        00:00:00 bluetooth-apple
2366 ?        00:00:01 nautilus
2382 ?        00:00:00 gvfs-gdu-volume
2456 ?        00:00:00 gvfs-afc-volume
2466 ?        00:00:00 gvfsd-trash
2525 ?        00:00:00 zeitgeist-datah
2775 ?        00:00:01 unity-2d-places
2794 ?        00:00:00 unity-music-dae
2795 ?        00:00:00 unity-files-dae
4096 ?        00:00:00 gnome-pty-helpe
4097 pts/0    00:00:00 bash
4302 pts/1    00:00:00 bash
4463 pts/2    00:00:00 bash
5642 pts/3    00:00:00 bash
6203 pts/3    00:00:00 sh
6204 pts/3    00:00:00 ps
>>>
>>>
>>>
```

```
>>> for v in os.popen("ps -e"):
...     if(re.search("\d$",v)):
...             print(v.strip())
...
3 ?        00:00:00 ksoftirqd/0
5 ?        00:00:01 kworker/u:0
6 ?        00:00:00 migration/0
18 ?        00:00:06 kworker/0:1
21 ?        00:00:00 kswapd0
36 ?        00:00:00 scsi_eh_0
37 ?        00:00:01 scsi_eh_1
38 ?        00:00:00 kworker/u:3
198 ?        00:00:00 mpt_poll_0
199 ?        00:00:00 mpt/0
274 ?        00:00:00 scsi_eh_2
298 ?        00:00:00 jbd2/sda1-8
317 ?        00:00:00 flush-8:0
864 ?        00:00:00 hci0
1058 ?        00:00:01 apache2
1062 ?        00:00:00 apache2
1063 ?        00:00:00 apache2
1064 ?        00:00:00 apache2
2330 ?        00:00:00 gconfd-2
6183 ?        00:00:00 kworker/0:0
6216 ?        00:00:00 kworker/0:2
>>>


>>> for v in os.popen("ps -e"):
...     if(re.search("[a-z]\d$",v)):
...             print(v.strip())
...
21 ?        00:00:00 kswapd0
864 ?        00:00:00 hci0
1058 ?        00:00:01 apache2
1062 ?        00:00:00 apache2
1063 ?        00:00:00 apache2
1064 ?        00:00:00 apache2
>>> with open("/home/apelix/process.log") as FH:
...     for var in FH:
...             if(re.search("^$",var)):
...                     continue
...             else:
...                     print(var.strip())
...
PID TTY          TIME CMD
4463 pts/2    00:00:00 bash
6290 pts/2    00:00:00 ps
>>>
>>>
>>> for var in os.popen("ps"):
...     re.split("\s+",var)
...
['', 'PID', 'TTY', 'TIME', 'CMD', '']
['', '5642', 'pts/3', '00:00:00', 'bash', '']
['', '5951', 'pts/3', '00:00:00', 'python', '']
['', '6557', 'pts/3', '00:00:00', 'sh', '']
```

```
['', '6558', 'pts/3', '00:00:00', 'ps', '']
>>>
>>> for var in os.popen("ps"):
...     L=re.split("\s+",var)
...     print(L[0],L[-1])
...
('', '')
('', '')
('', '')
('', '')
('', '')
>>> for var in os.popen("ps"):
...     print(L[0],L[-1])
... for var in os.popen("ps"):
  File "<stdin>", line 3
    for var in os.popen("ps"):
      ^
SyntaxError: invalid syntax
>>>
>>> for var in os.popen("ps"):
...     L=re.split("\s+",var)
...     print(L[1],L[-2])
...
('PID', 'CMD')
('5642', 'bash')
('5951', 'python')
('6568', 'sh')
('6569', 'ps')
>>>
>>> for var in os.popen("ps"):
...     L=re.split("\s+",var)
...     print("{}\t{}".format(L[1],L[-2]))
...
PID CMD
5642    bash
5951    python
6572    sh
6573    ps
>>>
>>> for var in os.popen("ps"):
...     L=re.split("\s+",var)
...     print("{}-{}".format(L[1],L[-2]))
...
PID-CMD
5642-bash
5951-python
6576-sh
6577-ps
>>>
>>> for var in os.popen("ps -e"):
...     if(re.search("bash|apache2|java|python",var)):
...             L=re.split("\s+",var)
...             print("{}--{}".format(L[1],L[-2]))
...
1058--apache2
1062--apache2
1063--apache2
```

```
1064--apache2
4097--bash
4188--python
4302--bash
4463--bash
5642--bash
5951--python
>>>
```