Packages ¶

- It is a collection of modules.
- It is also a directory of modules containing aadditional file as "**init**.py".
- It is a third party software repository for python.
- Some python managers,including PIP,use PyPias the default source for packages and their dependencies.

Modules

- It is a collection of python prigramming files.
- It allows you to logically organize your python code.
- Grouping makes code easy to understand and use.
- It can define functions,classes,variables.
- It can also include runnable code.

Regular expression

- In computing a regular expression also referred as "regex","regexp".
- It provides a concise and flexible means for matching strings of text such as particular characters,words,patterns of characters.
- A regular expression is written in a formal language that can be interpreted by a regular expression processor.
- Really clever "Wild card" expressions for matching and parsing string.

RE methods

- The common methods used in RE are;
- 1 match()
- 2 search()
- 3 findall()

In [1]:

```
import re
s="NsritStudents"
if re.match('N',s):
    print("True")
else:
    print("False")
```

In [2]:

```
import re
re.search('u',s)
```

Out[2]:

```
<re.Match object; span=(7, 8), match='u'>
```

In [3]:

```
import re
if re.search('S',s):
    print("True")
else:
    print("False")
```

```
True
```

In [4]:

```
import re
re.findall('t',s)
```

Out[4]:

```
['t', 't', 't']
```

Methods for RE

- "^" is for checking the first letter of the given string.
- "$" is for checking the last letter of the given string.
- "." is for checking the character .
- "\s" is for checking whitespaces and prints if present.
- "\s" is for checking nonwhite spaces and ingores if whitespaces are present.
- "*" repeats a character for zero or more times.
- "+" repeats a character for one or more times.
- [a-c][0-9] matches a single characters in the list.
- [^abc] matches a single character in the list.
- {9} range.
- {0,5} between range.

In [5]:

```
import re
s="Uday Kumar "
print(re.match('^U',s))
```

```
<re.Match object; span=(0, 1), match='U'>
```

In [6]:

```
import re
if re.match('^u',s):
    print("True")
else:
    print("False")
```

```
False
```

In [7]:

```
import re
print(re.search('r$',s))
```

```
None
```

In [8]:

```
import re
if re.search('a$',s):
    print("True")
else:
    print("False")
```

```
False
```

In [9]:

```
import re
print(re.search('a.',s))
```

```
<re.Match object; span=(2, 4), match='ay'>
```

In [10]:

```
import re
if re.search('r.',s):
    print("True")
else:
    print("False")
```

```
True
```

In [11]:

```
s
```

Out[11]:

```
'Uday Kumar '
```

In [12]:

```
import re
print(re.findall('\s',s))
```

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

In [13]:

```
import re
if re.findall('\s',s):
    print("True")
else:
    print("False")
```

```
True
```

In [14]:

```
import re
print(re.findall('\S',s))
```

```
['U', 'd', 'a', 'y', 'K', 'u', 'm', 'a', 'r']
```

In [15]:

```
import re
if re.findall("\S",s):
    print("True")
else:
    print("False")
```

```
True
```

In [16]:

```
import re
print(re.findall("$",s))
```

```
['']
```

In [17]:

```
import re
s=input("Enter a mob num:")
if re.match("^[6-9][0-9]{9}$|[0][6-9][0-9]{9}$|[+][9][1][6-9][0-9]{9}$",s):
    print("True")
else:
    print("False")
```

```
Enter a mob num:8726387345986
False
```

In [18]:

```
import re
def PNV(number):
    pattern='^[6-9][0-9]{9}$|^[0][6-9][0-9]{9}$|^[+][9][1][6-9][0-9]{9}$'
    if re.match(pattern,str(number)):
        return True
    return False
PNV(input())
```

```
658376458076
```

Out[18]:

```
False
```

In [19]:

```
import re
id=input("Enter a mail id:")
if re.match("^[a-z0-9][a-z0-9_.]{3,16}[@][a-z0-9]{3,7}[.][a-z]{2,5}",id):
    print("Your mail is valid")
else:
    print("Your mail is invalid")
```

```
Enter a mail id:jhsgckjhdsf@gmail.com
Your mail is valid
```

In [20]:

```
import re
def mail(email):
    if re.match("^[a-z0-9][a-z0-9_.]{3,16}[@][a-z0-9]{3,7}[.][a-z]{2,5}",id):
        print("Your mail is valid")
    else:
        print("Your mail is invalid")
mail(input())
```

```
iuhgjsoidhg@gmail.com
Your mail is valid
```

In [21]:

```
contacts={"name1":[8989767654,"name1@gmail.com"],"name2":[9878786578,"name2@gmail.com"]}
def addcontacts(name,number,email):
    if name in contacts:
        print(name,"already exists")
    else:
        if not PNV(number):
            print("Invalid phone number")
            return
        if not mail(email):
            print("Invalid email address")
            return
        newcontact=[]
        newcontact.append(phone)
        newcontact.append(email)
        contacts[name]=newcontact
        print(name,"added successfully")
    return
addcontacts("name4",9876543219,"name3@gmail.com")
contacts
```

```
Your mail is valid
Invalid email address
```

Out[21]:

```
{'name1': [8989767654, 'name1@gmail.com'],
 'name2': [9878786578, 'name2@gmail.com']}
```

File

- It is a collection of information or data.
- In file handling concept,python have two types of files:

    i.Text file
    ii.Binary file

- Modes

    i.Write mode(w).
    ii.Read mode(r).
    iii.Create and update mode(a).
    iv.Create a file(x).

In [22]:

```
f=open("566.txt","w")
f.write("Create and write data in it.")
f.close()
```

In [23]:

```
f=open("566.txt","r")
print(f.read())
f.close()
```

```
Create and write data in it.
```

In [24]:

```
f=open("566.txt","w")
f.write("update the data")
f.close()
```

In [25]:

```
f=open("566.txt","r")
print(f.read())
f.close()
```

```
update the data
```

In [30]:

```
f=open("566.txt","r")
print(f.readlines(3))
f.close()
```

```
['line1\n']
```

In [40]:

```
#Task-1:
#Sorting the data present in files
l1=[]
with open("566.txt","r") as f:
    for i in f.read():
        if i.isdigit():
            l1.append(i)
            l1.sort()
for i in l1:
    with open("5661.txt","a") as f:
        i=str(i)
        f.write(i+'\n')
```

In [43]:

```
with open("cs.txt","r+") as f:
    print(f.tell())
    print(f.read())
    f.write(",g,h,i,j,k")
    print(f.tell())
```

```
0
a,b,c,d,e,g,h,i,j,k
29
```

In [9]:

```
import re
s=str(input())
def pw(pattern,s):
    pattern="^[A-Z]{1}$|^[a-z]{5}$|^[!@#$%^]{1}$|^[0-9]{1}"
    if s in re.match(pattern,s):
        print(s)
    else:
        print("False")
pw('kjhgdL@1',s)
```

```
kieuhL@1

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-9-156500ee889d> in <module>
      7     else:
      8         print("False")
----> 9 pw('kjhgdL@1',s)

<ipython-input-9-156500ee889d> in pw(pattern, s)
      3 def pw(pattern,s):
      4     pattern="^[A-Z]{1}$|^[a-z]{5}$|^[!@#$%^]{1}$|^[0-9]{1}"
----> 5     if s in re.match(pattern,s):
      6         print(s)
      7     else:

TypeError: argument of type 'NoneType' is not iterable
```

In [ ]:

```

```