In [1]:
```
#Multi-threading
```

In [2]:
```
##program without threads
```

In [3]:
```
import threading
import time
def printmsg(msg, stime):
  print ('Thread started')
  for i in range(10):
    print(msg, ' - ', (i+1))
    time.sleep(stime)

print(time.ctime())
printmsg('Good Morning', 1)
printmsg('Good Morning', 1)
print(time.ctime())
```

```
Tue May 23 13:52:48 2023
Thread started
Good Morning  -  1
Good Morning  -  2
Good Morning  -  3
Good Morning  -  4
Good Morning  -  5
Good Morning  -  6
Good Morning  -  7
Good Morning  -  8
Good Morning  -  9
Good Morning  -  10
Thread started
Good Morning  -  1
Good Morning  -  2
Good Morning  -  3
Good Morning  -  4
Good Morning  -  5
Good Morning  -  6
Good Morning  -  7
Good Morning  -  8
Good Morning  -  9
Good Morning  -  10
Tue May 23 13:53:08 2023
```

In [4]:
```
##program with threads
```

In [5]:
```
import threading
import time
def printmsg(msg, stime):
  print ('Thread started')
  for i in range(10):
    print(msg, ' - ', (i+1))
    time.sleep(stime)

t1 = threading.Thread(target=printmsg, args=('Good Morning',1))
t2 = threading.Thread(target=printmsg, args=('Good Afternoon',1))
print(time.ctime())
t1.start()
t2.start()
t1.join()
```

```
t2.join()
print(time.ctime())
```

```
Tue May 23 13:53:38 2023
Thread started
Thread started
Good Afternoon  -  1
Good Morning  -  1
Good AfternoonGood Morning  -  2
  -  2
Good AfternoonGood Morning  -  3
  -  3
Good MorningGood Afternoon   -   -  4
4
Good AfternoonGood Morning   -   -  5
5
Good AfternoonGood Morning   -   -  6
6
Good AfternoonGood Morning  -  7
  -  7
Good Morning  -  8
Good Afternoon  -  8
Good Morning  -  9
Good Afternoon  -  9
Good Morning  -  10
Good Afternoon  -  10
Tue May 23 13:53:49 2023
```

**The .join() method delays a program's flow of execution until the target thread has been completely read.**

In [7]:
```
# time Same as normal without use of threads
import threading
import time
def printmsg(msg, stime):
  print ('Thread started')
  for i in range(10):
    print(msg, ' - ', (i+1))
    time.sleep(stime)

t1 = threading.Thread(target=printmsg, args=('Good Morning',1))
t2 = threading.Thread(target=printmsg, args=('Good Afternoon',1))
print(time.ctime())
t1.start()
t1.join()
t2.start()
t2.join()
print(time.ctime())
```

```
Tue May 23 13:56:12 2023
Thread started
Good Morning  -  1
Good Morning  -  2
Good Morning  -  3
Good Morning  -  4
Good Morning  -  5
Good Morning  -  6
Good Morning  -  7
Good Morning  -  8
Good Morning  -  9
Good Morning  -  10
Thread started
Good Afternoon  -  1
Good Afternoon  -  2
Good Afternoon  -  3
Good Afternoon  -  4
Good Afternoon  -  5
Good Afternoon  -  6
Good Afternoon  -  7
Good Afternoon  -  8
Good Afternoon  -  9
Good Afternoon  -  10
Tue May 23 13:56:32 2023
```

**t2 cannot start before t1 completes as t1.join() is present right after t1.start()**

**#Python Regular Expressions**

A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern.

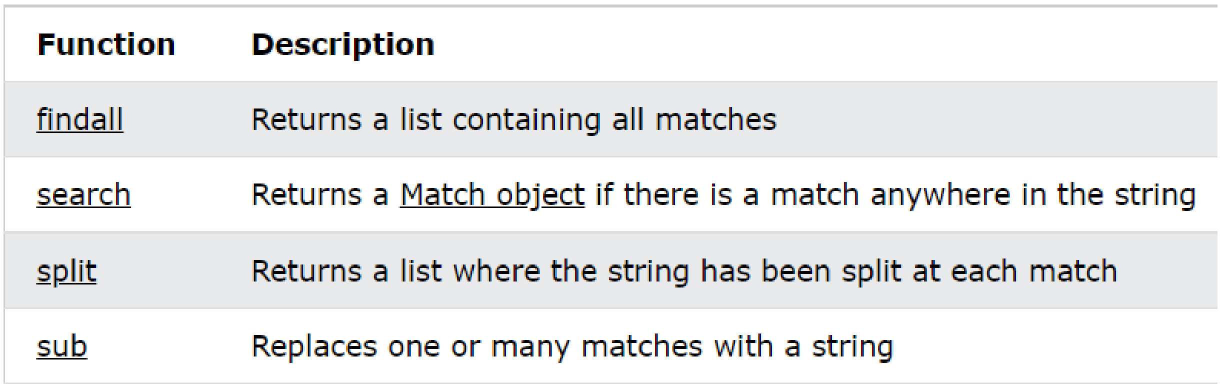
RegEx can be used to check if a string contains the specified search pattern.

Python has a built-in package called re, which can be used to work with Regular Expressions.

Import the re module:

In [8]:
```
import re
```

The re module offers a set of functions that allows us to search a string for a match:



| Function | Description |
|---|---|
| findall | Returns a list containing all matches |
| search | Returns a Match object if there is a match anywhere in the string |
| split | Returns a list where the string has been split at each match |
| sub | Replaces one or many matches with a string |

Metacharacters are characters with a special meaning:

| Character | Description | Example |
|---|---|---|
| [] | A set of characters | "[a-m]" |
| \ | Signals a special sequence (can also be used to escape special characters) | "\d" |
| . | Any character (except newline character) | "he..o" |
| ^ | Starts with | "^hello" |
| $ | Ends with | "planet$" |
| * | Zero or more occurrences | "he.*o" |
| + | One or more occurrences | "he.+o" |
| ? | Zero or one occurrences | "he.?o" |
| {} | Exactly the specified number of occurrences | "he.{2}o" |
| \| | Either or | "falls\|stays" |
| () | Capture and group | |

A special sequence is a \ followed by one of the characters in the list below, and has a special meaning:

| | | |
|---|---|---|
| \d | Returns a match where the string contains digits (numbers from 0-9) | "\d" |
| \D | Returns a match where the string DOES NOT contain digits | "\D" |
| \s | Returns a match where the string contains a white space character | "\s" |
| \S | Returns a match where the string DOES NOT contain a white space character | "\S" |
| \w | Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character) | "\w" |
| \W | Returns a match where the string DOES NOT contain any word characters | "\W" |

A set is a set of characters inside a pair of square brackets [] with a special meaning:

| Set | Description |
|---|---|
| [arn] | Returns a match where one of the specified characters (`a`, `r`, or `n`) is present |
| [a-n] | Returns a match for any lower case character, alphabetically between `a` and `n` |
| [^arn] | Returns a match for any character EXCEPT `a`, `r`, and `n` |
| [0123] | Returns a match where any of the specified digits (`0`, `1`, `2`, or `3`) are present |
| [0-9] | Returns a match for any digit between `0` and `9` |
| [0-5][0-9] | Returns a match for any two-digit numbers from `00` and `59` |
| [a-zA-Z] | Returns a match for any character alphabetically between `a` and `z`, lower case OR upper case |
| [+] | In sets, `+`, `*`, `.`, `\|`, `()`, `$`,`{}` has no special meaning, so `[+]` means: return a match for any `+` character in the string |

**The findall() function**

The findall() function returns a list containing all matches.

In [11]:
```
import re

txt = "The rain in Spain"
x = re.findall("ai", txt)
print(x)
```

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

In [12]:
```
import re

txt = "The rain in Spain"
x = re.findall("[arn]", txt)
print(x)
```

```
['r', 'a', 'n', 'n', 'a', 'n']
```

The list contains the matches in the order they are found.

If no matches are found, an empty list is returned:

In [13]:
```
import re

txt = "The rain in Spain"
x = re.findall("Portugal", txt)
print(x)
```

```
[]
```

In [14]:
```
import re
txt="The rain is Spain"
x=re.findall('[^arn]',txt)
print(x)
```

```
['T', 'h', 'e', ' ', 'i', ' ', 'i', 's', ' ', 'S', 'p', 'i']
```

In [15]:
```
#write a python program to extract year month and date from url using re
import re
url1= "https://www.washingtonpost.com/news/football-insider/wp/2016/09/02/odell-beckham
x=re.findall('\d{4}[/]\d{2}[/]\d{2}', url1)
print(x)
print(x[0])
```

```
['2016/09/02']
2016/09/02
```

In [21]:
```
import re
txt = "The rain in Spain"
#Find all lower case characters alphabetically between "a" and "m":
x = re.findall("[a-m]", txt)
print(x)
```

```
['h', 'e', 'a', 'i', 'i', 'a', 'i']
```

In [22]:
```
import re
txt = "That will be 59 dollars"
#Find all digit characters:
x = re.findall("\d", txt)
print(x)
```

```
['5', '9']
```

In [23]:
```
import re
txt = "That will be 59 dollars"
x = re.findall("\d+", txt)
print(x)
```
```
['59']
```

In [24]:
```
import re
txt = "hello planet"
#Search for a sequence that starts with "he", followed by two (any) characters, and an
x = re.findall("he..o", txt)
print(x)
```
```
['hello']
```

In [25]:
```
import re
txt = "hello planet"
#Check if the string starts with 'hello':
x = re.findall("^h.+\s", txt)
print(x)
```
```
['hello ']
```

In [26]:
```
import re
txt = "hello planet"
#Check if the string ends with 'planet':
x = re.findall("planet$", txt)
print(x)
```
```
['planet']
```

In [27]:
```
import re
txt = "hello"
#Search for a sequence that starts with "he", followed by 0 or more  (any) characters,
x = re.findall("hell.*o", txt)
print(x)
```
```
['hello']
```

In [29]:
```
import re
txt = "hello"
#Search for a sequence that starts with "he", followed by 1 or more  (any) characters,
x = re.findall("hell.+o", txt)
print(x)
```
```
[]
```

In [32]:
```
import re
txt = "hello planet"
#Search for a sequence that starts with "he", followed by 0 or 1  (any) character, and
x = re.findall("he.?o", txt)
print(x)
#This time we got no match, because there were not zero, not one, but two characters be
```
```
[]
```

In [34]:
```
import re
txt = "helo planet"
#Search for a sequence that starts with "he", followed by 0 or 1  (any) character, and
x = re.findall("he.?o", txt)
print(x)
```

```
['helo']
```

In [36]:
```python
import re
txt = "hello helo planet"
#Search for a sequence that starts with "he", followed excactly 2 (any) characters, and
x = re.findall("he.{2}o", txt) #Exact two characters after he and last o
print(x)
```

```
['hello']
```

In [37]:
```python
import re
txt = "The rain in Spain falls mainly in the plain!"
#Check if the string contains either "falls" or "stays":
x = re.findall("Spain|plain", txt)
print(x)
```

```
['Spain', 'plain']
```

In [39]:
```python
import re
txt = "The rain in Spain123"
#Return a match at every no-digit character:
x = re.findall("\D", txt)
print(x)
```

```
['T', 'h', 'e', ' ', 'r', 'a', 'i', 'n', ' ', 'i', 'n', ' ', 'S', 'p', 'a', 'i', 'n']
```

In [40]:
```python
import re
txt = "The rain in Spain"
#Return a match at every white-space character:
x = re.findall("\s", txt)
print(x)
```

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

In [41]:
```python
import re
txt = "The rain in Spain"
#Return a match at every NON white-space character:
x = re.findall("\S", txt)
print(x)
```

```
['T', 'h', 'e', 'r', 'a', 'i', 'n', 'i', 'n', 'S', 'p', 'a', 'i', 'n']
```

In [42]:
```python
import re
txt = "The rain in Spain since_1990"
#Return a match at every word character (characters from a to Z, digits from 0-9, and t
x = re.findall("\w", txt)
print(x)
```

```
['T', 'h', 'e', 'r', 'a', 'i', 'n', 'i', 'n', 'S', 'p', 'a', 'i', 'n', 's', 'i', 'n',
'c', 'e', '_', '1', '9', '9', '0']
```

In [43]:
```python
import re
txt = "8 times before 11:45 AM"
#Check if the string has any digits:
x = re.findall("[0-9]", txt)
print(x)
```

```
['8', '1', '1', '4', '5']
```

In [45]:
```python
import re
txt = "8 times before 11:45 AM"
#Check if the string has any two-digit numbers, from 00 to 59:
```

```
x = re.findall("[0-5][0-9]", txt)
print(x)
```

```
['11', '45']
```

In [72]:
```
# Program to extract numbers from a string
import re

string = 'hello 12 hi 89. Howdy 34'
pattern = '\d'

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

```
['1', '2', '8', '9', '3', '4']
```

In [74]:
```
# Program to extract numbers from a string
import re
string = 'hello 12 hi 89. Howdy 34'
pattern = '\d+'

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

```
['12', '89', '34']
```

### The search() function

The search() function searches the string for a match, and returns a Match object if there is a match.

If there is more than one match, only the first occurrence of the match will be returned:

In [46]:
```
import re

txt = "The rain in Spain"
x = re.search("\s", txt)

print("The first white-space character is located in position:", x.start())
```

```
The first white-space character is located in position: 3
```

If no matches are found, the value None is returned:

In [47]:
```
import re

txt = "The rain in Spain"
x = re.search("Portugal", txt)
print(x)
```

```
None
```

In [48]:
```
#search
import re
txt="The rain is Spain"
x=re.search("rain",txt)
print(x) # span is first occurence index
```

```
<re.Match object; span=(4, 8), match='rain'>
```

In [56]:
```
import re
txt="The rain is Spain"
x=re.search("\s",txt)
```

```
print(x.start()) # start index of space
print(x.end()) # end index of space
```

```
3
4
```

In [57]:

```
import re
txt="The rain is Spain"
x=re.search("rain",txt)
print(x.start())
print(x.end())
print(x.span())
```

```
4
8
(4, 8)
```

In [63]:

```
import re
txt="no 7756spain"
x=re.search("\d",txt)
print(x.end())
```

```
4
```

In [64]:

```
import re
txt="no 7756spain"
x=re.search("\d+",txt)
print(x.end())
```

```
7
```

In [71]:

```
import re
string = "Python is fun"
# check if 'Python' is at the beginning
match = re.search('^Python', string)
print(match.span())
print(match.start())
print(match.end())
```

```
(0, 6)
0
6
```

### The split() function

The split() function returns a list where the string has been split at each match:

In [61]:

```
#Split at each white-space character:

import re
txt = "The rain in Spain"
x = re.split("\s", txt)
print(x)
```

```
['The', 'rain', 'in', 'Spain']
```

You can control the number of occurrences by specifying the maxsplit parameter:

Split the string only at the first occurrence:

In [62]:

```
import re

txt = "The rain in Spain"
```

```
x = re.split("\s", txt, 1)
print(x)
```

```
['The', 'rain in Spain']
```

In [65]:
```
import re
txt='The_quick_brown@fox*jumps#over$the^lazy&dog'
pattern='[a-zA-Z]+'
x=re.split(pattern,txt)
print(x)
```

```
['', '_', '_', '@', '*', '#', '$', '^', '&', '']
```

In [66]:
```
import re
txt='The quick brown fox jumps over the lazy dog'
pattern=r'\s+\w+\s'
x=re.split(pattern,txt)
print(x)
```

```
['The', 'brown', 'jumps', 'the', 'dog']
```

In [67]:
```
import re
txt='The quick brown fox jumps over the lazy dog'
pattern=r'\s[a-z]+\s'
x=re.split(pattern,txt)
print(x)
```

```
['The', 'brown', 'jumps', 'the', 'dog']
```

In [75]:
```
import re

string = 'Twelve:8 Eighty nine:9.'
pattern = '\d'

result = re.split(pattern, string)
print(result)
```

```
['Twelve:', ' Eighty nine:', '.']
```

In [76]:
```
import re

string = 'Twelve:12 Eighty nine:89.'
pattern = '\d'

result = re.split(pattern, string)
print(result)
```

```
['Twelve:', '', ' Eighty nine:', '', '.']
```

In [77]:
```
import re

string = 'Twelve:12 Eighty nine:89.'
pattern = '\d+'

result = re.split(pattern, string)
print(result)
```

```
['Twelve:', ' Eighty nine:', '.']
```

In [81]:
```
# write a python program to write all words starting with a and e using re
import re

txt="The Rain in ahmedabad earth"
```

```
y=re.split(" ",txt)
for i in y:
    if i[0]=='a' or i[0]=='e':
        print(i)
```

```
ahmedabad
earth
```

**The sub() function**

The sub() function replaces the matches with the text of your choice:

Replace every white-space character with the number 9:

In [82]:
```
import re

txt = "The rain in Spain"
x = re.sub("\s", "9", txt)
print(x)
```

```
The9rain9in9Spain
```

You can control the number of replacements by specifying the count parameter:

Replace the first 2 occurrences:

In [83]:
```
import re

txt = "The rain in Spain"
x = re.sub("\s", "9", txt, 2)
print(x)
```

```
The9rain9in Spain
```

In [85]:
```
import re
txt="The Rain in Spain"
x=re.sub('\s','9',txt,1)
print(x)
```

```
The9Rain in Spain
```

In [86]:
```
##########################################################################################
```

In [91]:
```
#Program to find mobile number make regular exp

import re
z=[]
txt= "9687000000 8502502520 12456287465 822222222422"
y=re.split(" ",txt)
# print(y)
for i in y:
    if (len(i)==10):
            x=re.findall('[6-9][0-9]{9}',i)
            z.append(x)
print(z)
```

```
[['9687000000'], ['8502502520']]
```

In [90]:
```
#write a python program to remove multiple spaces and make single space.

import re
txt="The Rain in     Ahmedabad"
```

```
x=re.sub("\s+"," ",txt)
print(x)
```

```
The Rain in Ahmedabad
```

In [92]:

```
# write a python program to write all words starting with a and e using re

import re

txt="The Rain in ahmedabad earth"
y=re.split(" ",txt)

for i in y:
    if i[0]=='a' or i[0]=='e':
        print(i)
```

```
ahmedabad
earth
```

In [95]:

```
#write a python program to extract year month and date from url using re
import re
url1= "https://www.washingtonpost.com/news/football-insider/wp/2016/09/02/odell-beckham
x=re.findall('\d{4}[/]\d{2}[/]\d{2}', url1)
print(x)
print(x[0])
```

```
['2016/09/02']
2016/09/02
```

In [96]:

```
# Program to remove all whitespaces

import re

string = 'abc 12de 23 \n f45 6'

# matches all whitespace characters
pattern = '\s+'

# empty string
replace = ''

new_string = re.sub(pattern, replace, string)
print(new_string)
```

```
abc12de23f456
```

In [98]:

```
import re

# multiline string
string = 'abc 12de 23 \n f45 6'

# matches all whitespace characters
pattern = '\s+'
replace = ''

new_string = re.sub('\s+', replace, string,1)
print(new_string)
```

```
abc12de 23 
 f45 6
```

In [99]:

```
#Write a python program to find email ids
import re
txt=" my email id is abc.def@gmail.com"
x=re.findall('\w+[.a-zA-Z]+@+[a-zA-Z.]+',txt)
print(x)
```

```
['abc.def@gmail.com']
```