Regular Expressions

CODE WITH HARRY YT CHANNEL

what is regular expression?
actually, regular expression helps us to find and return various pattern from a string.

In [1]:

```
import re
mystr = '''Tata Limited
Dr. David Landsman, executive director
18, Grosvenor Place
London SW1X 7HSc
Phone: +44 (20) 7235 8281
Fax: +44 (20) 7235 8727
Email: tata@tata.co.uk
Website: www.europe.tata.com
Directions: View map

Tata Sons, North America
1700 North Moore St, Suite 1520
Arlington, VA 22209-1911
USA
Phone: +1 (703) 243 9787
Fax: +1 (703) 243 9791
66-66
455-4545
Email: northamerica@tata.com
Website: www.northamerica.tata.com
Directions: View map fass
harry bhai lekin
bahut hi badia aadmi haiaiinaiiiiiiiiiiii'''
```

In [2]:

```
# findall, search, split, sub, finditer
```

RAW STRING

In re we always use raw string to find anything in a string.

In [3]:

```
st = ('\n')
print(st)
```

In [4]:

```
st = (r'\n')
print(st)
```

```
\n
```

In [5]:

```
patt = re.compile(r'fass') # by compile we take out a pattern from a string & store it in
to a variable.

matches = patt.finditer(mystr)
```

```
for match in matches:
    print(match)
                    # here for loop return us a match object.
```

```
<re.Match object; span=(448, 452), match='fass'>
```

In [6]:

```
print(mystr[448:452])
```

```
fass
```

__English Edureka Channel__

In [7]:

```
import re

NameAge = '''
Janice is 22 and Theon is 33
Gabriel is 44 and Jeoy is 21
'''

ages = re.findall(r'\d{1,3}', NameAge)
names = re.findall(r'[A-Z][a-z]*', NameAge)
```

In [8]:

```
ages
```

Out[8]:

```
['22', '33', '44', '21']
```

In [9]:

```
names
```

Out[9]:

```
['Janice', 'Theon', 'Gabriel', 'Jeoy']
```

In [10]:

```
ageDict = {}
x = 0
for eachname in names:
    ageDict[eachname] = ages[x]
    x = x+1

print(ageDict)
```

```
{'Janice': '22', 'Theon': '33', 'Gabriel': '44', 'Jeoy': '21'}
```

In [11]:

```
# Full Code:-

import re

NameAge = '''
Janice is 22 and Theon is 33
Gabriel is 44 and Jeoy is 21
'''

ages = re.findall(r'\d{1,3}', NameAge)
names = re.findall(r'[A-Z][a-z]*', NameAge)

ageDict = {}
x = 0
for eachname in names:
```

```
    ageDict[eachname] = ages[x]
    x = x+1

print(ageDict)
```

```
{'Janice': '22', 'Theon': '33', 'Gabriel': '44', 'Jeoy': '21'}
```

re.search()

In [12]:

```
# Searching Inform in string.

import re

if re.search("inform","we need to inform him with the latest information"):
    print("there is inform")
```

```
there is inform
```

In [13]:

```
import re

randstr = "here is \\drogba"

print(re.search(r"\\drogba",randstr))
```

```
<re.Match object; span=(8, 15), match='\\drogba'>
```

In [14]:

```
sentence = 'Start a sentence and bring it to an end'

pattern = re.compile(r'start',re.IGNORECASE)      # IGNORECASE can also be written as "I"
.

matches = pattern.search(sentence)

print(matches)
```

```
<re.Match object; span=(0, 5), match='Start'>
```

re.finditer()`

It is used to generate iterator,
by this we get startring and the ending index for the word we are searching for.

In [15]:

```
import re

str = "we need to inform him with the latest information"

for i in re.finditer("inform", str):            # iter gives us index no.

    loctup = i.span()      # To get our answer in a tuple.
    print(loctup)

# re.MatchObject.span() method returns a tuple containing starting
# and ending index of the matched string.
# If group did not contribute to the match it returns(-1,-1) '''

# Syntax: re.MatchObject.span()
```

```
(11, 17)
(38, 44)
```

re.findall()

It will return list of matches

In [16]:

```
import re

allinform = re.findall("inform","we need to inform him with the latest information")

for i in allinform:
    print(i)
```

```
inform
inform
```

Pattern

In [17]:

```
import re

str = "Sat,hat,mat,pat"

allstr = re.findall("[Shmp]at",str)

for i in allstr:
    print(i)
```

```
Sat
hat
mat
pat
```

In [18]:

```
import re

str = "Sat,hat,mat,pat"

allstr = re.findall("[h-m]at",str)

for i in allstr:
    print(i)
```

```
hat
mat
```

In [19]:

```
import re

str = "Sat,hat,mat,pat"

allstr = re.findall("[h-z]at",str)

for i in allstr:
    print(i)
```

```
hat
mat
pat
```

In [20]:

```
import re

str = "sat,hat,mat,pat"
```

```
allstr = re.findall("[h-z]at",str)

for i in allstr:
    print(i)
```

```
sat
hat
mat
pat
```

In [21]:

```
# carrot symbol(^)

import re

str = "Sat,hat,mat,pat"

allstr = re.findall("[^h-m]at",str)     # exclude characters between (h-m)

for i in allstr:
    print(i)
```

```
Sat
pat
```

Compile & Sub

The re.compile() method
re.compile(pattern, repl, string):

We can combine a regular expression pattern into pattern objects, which can be used for pattern matching. It also helps to search a pattern again without rewriting it.

In [22]:

```
import re

food = "Sat,hat,rat,pat"

regex = re.compile("[r]at")     # regular expression pattern to pattern object.

food = regex.sub("food",food)
print(food)
```

```
Sat,hat,food,pat
```

SPACES

In [23]:

```
# remove new line and add replace it with space.
import re

randstrd = '''
keep the blue flag
flying high
Chelsea'''

print(randstrd)

regex = re.compile("\n")

randstrd = regex.sub(" ",randstrd)

print(randstrd)
```

```
keep the blue flag
```

```
keep the blue flag
flying high
Chelsea
 keep the blue flag  flying high Chelsea
```

- **\b: backspace**
  \f: formfeed
  \r: carriage return
  \t: Tab
  \v: Vertical tab

\d and \D

In [24]:

```
import re

randstr = "12345"
print("Matches:",len(re.findall("\d", randstr)))    # match every digits
```

```
Matches: 5
```

In [25]:

```
import re

randstr = "12345"
print("Matches:",len(re.findall("\D", randstr)))    # match everything excepts digits.
```

```
Matches: 0
```

In [26]:

```
import re

randstr = "12345 Aadil Mansoori"
print("Matches:",len(re.findall("\D", randstr)))
```

```
Matches: 15
```

In [27]:

```
# Matching a perticular int.
import re

randstr = "12345"
print("Matches:",len(re.findall("\d{5}", randstr)))
```

```
Matches: 1
```

In [28]:

```
# matching numbers to a range.
import re

randstr = "12345 1234 123456 1234567"
print("Matches:",len(re.findall("\d{5,7}", randstr)))
```

```
Matches: 3
```

APPLICATIONS OF REGULAR EXPRESSIONS

\w = [a-z,A-Z,0-9,_] # it matches all the things present into the boxes.

\W = [^a-z,A-Z,0-9] # it matches all the thing except present in the boxes

In [29]:

```
#  matching a phone number:-

phn = "412-555-1212"

if re.search("\w{3}-\w{3}-\w{4}",phn):
    print("it is a phone number")
```

```
it is a phone number
```

In [30]:

```
phn = "412-555-1212"

if re.search("\d{3}-\d{3}-\d{4}",phn):
    print("it is a phone number")
```

```
it is a phone number
```

In [31]:

```
phn = "412-5551-1212"
if re.search("\d{3}-\d{3}-\d{4}",phn):
    print("it is a phone number")
else:
    print("it is not a phone number")

phn = "412-5515-1212"
if re.search("\w{3}-\w{3}-\w{4}",phn):
    print("it is a phone number")
else:
    print("it is not a phone number")

phn = "412-55a-1212"
if re.search("\w{3}-\w{3}-\w{4}",phn):
    print("it is a phone number")
else:
    print("it is not a phone number")
```

```
it is not a phone number
it is not a phone number
it is a phone number
```

\s = [\f,\n,\r,\t,\v]
\S = [^\f,\n\r,\t,\v]

In [32]:

```
# Validating Full Name
import re

if re.search("\w{2,20}\s\w{2,20}","Aadil Mansoori"):        # s is a space.
    print("full name is Valid")
```

```
full name is Valid
```

In [33]:

```
# Varifying the e-mail address
import re

email = "skit@zaoaol.com md@.com @seo.com dc@.com aadil@gmail.com"
```

In [34]:

```
print("EmailMatches:", len(re.findall('[\w._%+-]{1,20}@[\w.-]{2-20}.[A-Za-z]{2,3}', email
)))
```

```
EmailMatches: 0
```

In [35]:

```
email = 'rahul@incognitor.com md@com @seo.com dc@com'
```

In [36]:

```
print("Email Matches:", len(re.findall('[\w._%+-]{1,20}@[\w.-]{2,20}.[A-Za-z]{2,3}',emai
l)))

                                    # [\w._%+-] = iska mtlb  w ke saath(_%+-) bhi chalega.
                                    # {1,20}  =  value should be b\w 1 to 19.
```

```
Email Matches: 1
```

In [37]:

```
email = 'rahul@incognitor.com  md@com  @seo.com  dc@com  rahul@incognitor.com'

print("Email Matches:", len(re.findall('[\w._%+-]{1,20}@[\w.-]{2,20}.[A-Za-z]{2,3}',emai
l)))

                               # space b/w emails is more than 1.  becoz 1-space prblm de rh
a h.
```

```
Email Matches: 2
```

WebScrap

In [38]:

```
# Code of Web Scrapping:-
import urllib.request
from re import findall

url = "https://www.summet.com/dmsi/html/codesamples/addresses.html"

response = urllib.request.urlopen(url)

html = response.read()

htmlStr = html.decode()
```

In [39]:

```
pddata = findall("\(\d{3}\) \d{3}-\d{4}", htmlStr)    # (r'\d\d\d-\d\d\d-\d\d\d\d) == d{3
}-d{3}-d{4}

for i in pddata:
    print(i)
```

```
(257) 563-7401
(372) 587-2335
(786) 713-8616
(793) 151-6230
(492) 709-6392
(654) 393-5734
(404) 960-3807
(314) 244-6306
(947) 278-5929
(684) 579-1879
(389) 737-2852
(660) 663-4518
(608) 265-2215
(959) 119-8364
(468) 353-2641
(248) 675-4007
(939) 353-1107
(570) 873-7090
(302) 259-2375
```

(717) 450-4729
(453) 391-4650
(559) 104-5475
(387) 142-9434
(516) 745-4496
(326) 677-3419
(746) 679-2470
(455) 430-0989
(490) 936-4694
(985) 834-8285
(662) 661-1446
(802) 668-8240
(477) 768-9247
(791) 239-9057
(832) 109-0213
(837) 196-3274
(268) 442-2428
(850) 676-5117
(861) 546-5032
(176) 805-4108
(715) 912-6931
(993) 554-0563
(357) 616-5411
(121) 347-0086
(304) 506-6314
(425) 288-2332
(145) 987-4962
(187) 582-9707
(750) 558-3965
(492) 467-3131
(774) 914-2510
(888) 106-8550
(539) 567-3573
(693) 337-2849
(545) 604-9386
(221) 156-5026
(414) 876-0865
(932) 726-8645
(726) 710-9826
(622) 594-1662
(948) 600-8503
(605) 900-7508
(716) 977-5775
(368) 239-8275
(725) 342-0650
(711) 993-5187
(882) 399-5084
(287) 755-9948
(659) 551-3389
(275) 730-6868
(725) 757-4047
(314) 882-1496
(639) 360-7590
(168) 222-1592
(896) 303-1164
(203) 982-6130
(906) 217-1470
(614) 514-1269
(763) 409-5446
(836) 292-5324
(926) 709-3295
(963) 356-9268
(736) 522-8584
(410) 483-0352
(252) 204-1434
(874) 886-4174
(581) 379-7573
(983) 632-8597
(295) 983-3476
(873) 392-8802
(360) 669-3923
(840) 987-9449

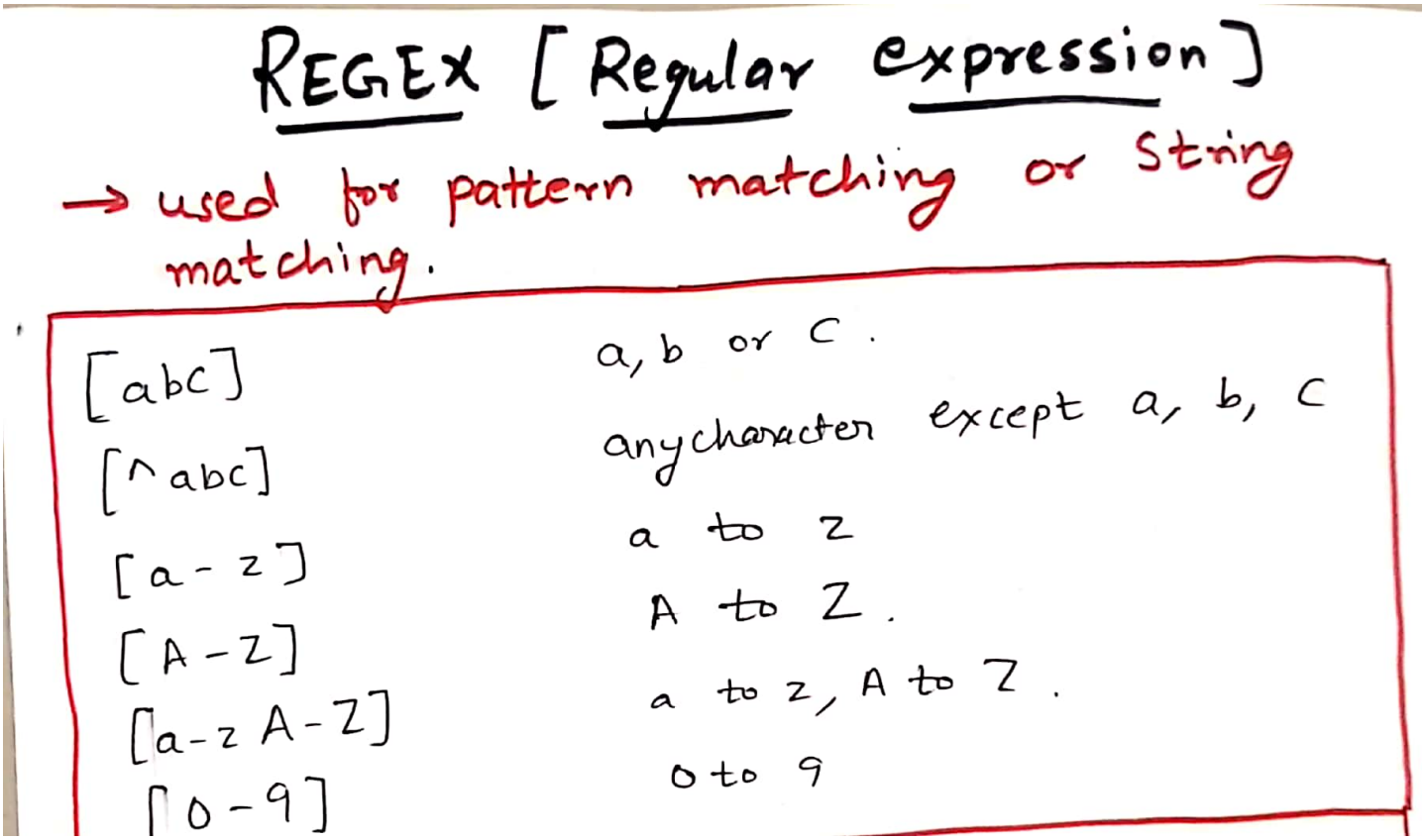
```
(422) 517-6053
(126) 940-2753
(427) 930-5255
(689) 721-5145
(676) 334-2174
(437) 994-5270
(564) 908-6970
(577) 333-6244
(655) 840-6139
```

YT CHANNEL Crack Concept

In [40]:

```
from IPython.display import Image
Image(filename='My Docs_1.jpg')
```

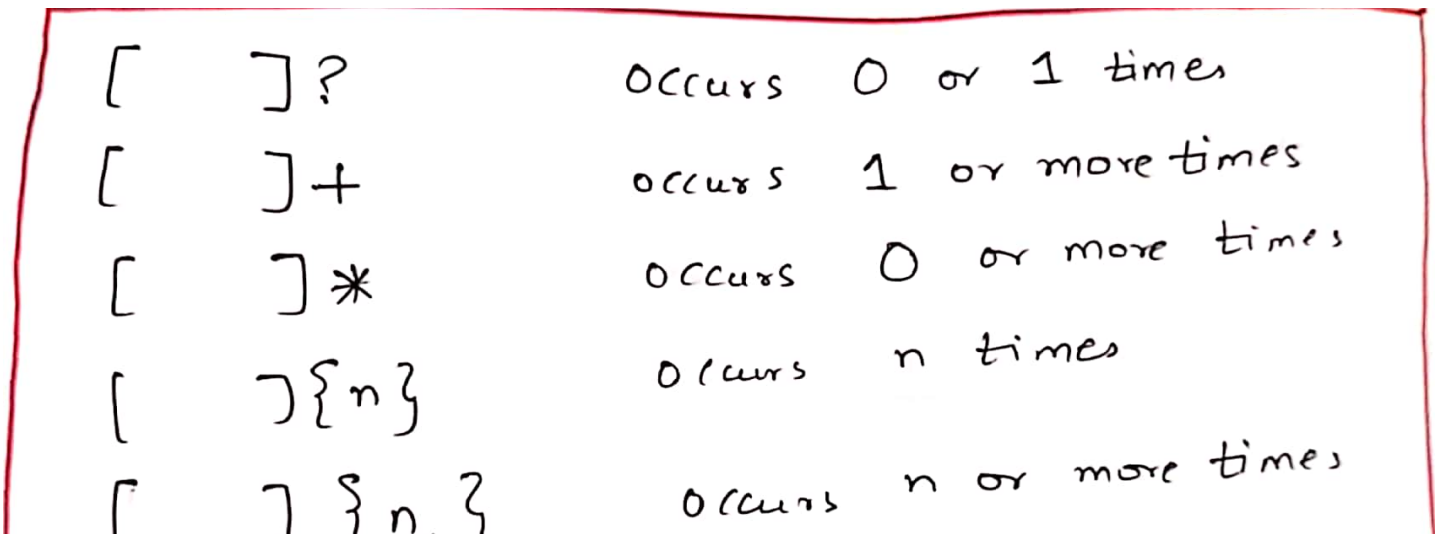
Out[40]:

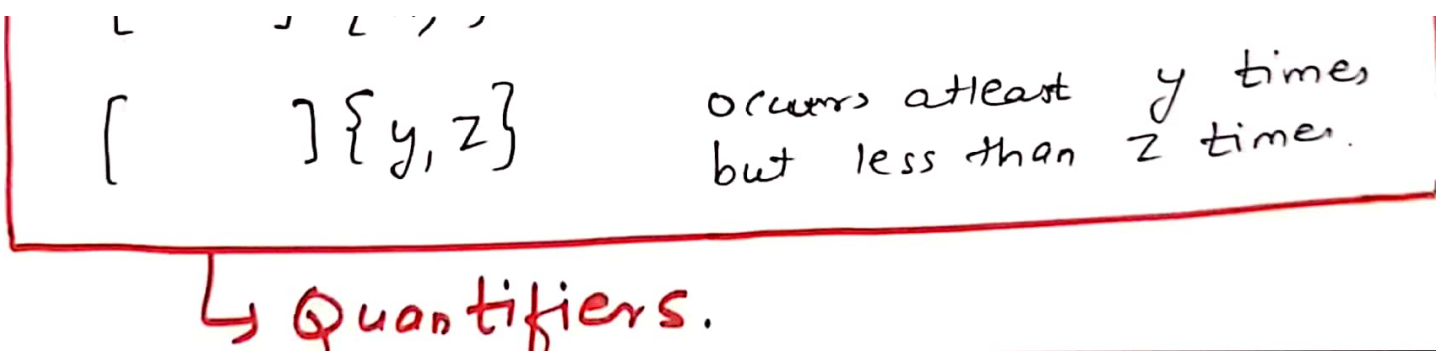


In [41]:

```
from IPython.display import Image
Image(filename='My Docs_2.jpg')
```

Out[41]:

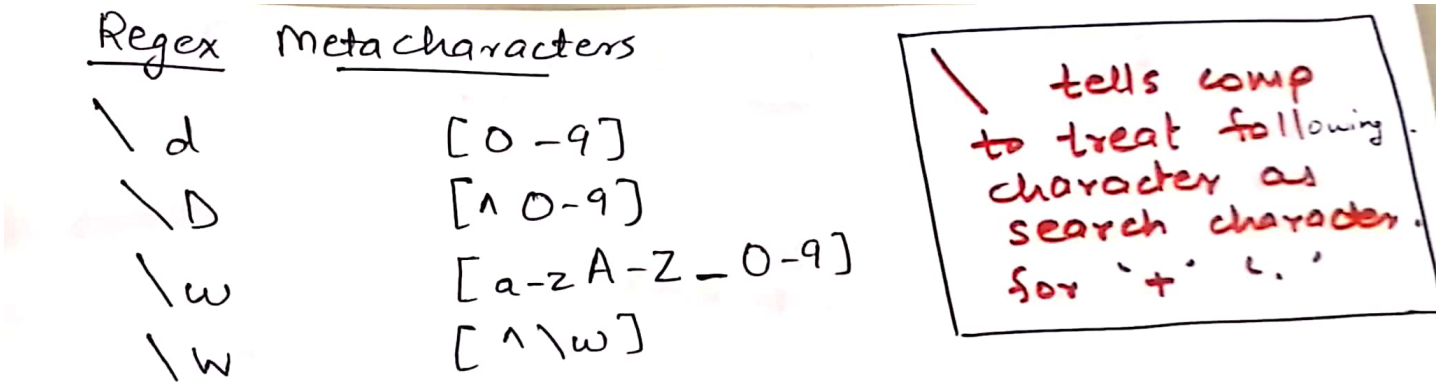




In [42]:

```
from IPython.display import Image
Image(filename='My Docs_3.jpg')
```

Out[42]:



Examples:-

1) Mobile No. --> start with 8 or 9 and total digit should be 10
Ans:- [89][0-9]{9}

```
first digit shoul be 8 or 9
then remaining 9 digit can be from 0-9,
and this should be repeat for 9 times
```

2) First Character should be upper case, contains lower case alphabates, only one digit allowed in between.
Ans:- [A-Z][a-z]*[0-9][a-z]*

```
                    * = this box can be repeated for 0 or n no. of times.
[a-z]*[0-9][a-z]* = this means that only 1 digit is allowed between smaller case.(a
-z)
             [a-z]* = [a-z] can be repeated for 0 or n no. of times.
```

3) Email ID :- sadia123@gmail.com

First part of email can have [0-9][A-Z][a-z][-,_,.] etc.
In second part we want @
In Third Part we want [a-z]
In Forth part we need .
In last part we need [a-z]

Ans:- [a-zA-Z0-9-_\,]+[@][a-z]+[.][a-z]{2,4}

```
      {2,4}= basically means that it can allow 2 to 3 characters. i.e.:- .in and .co
m
```

Searching special characters

Searching special characters.

To search a special characters we have to write that character after backslash \. \- _ etc.

In [43]:

```
email = "aadil_coreyms.com"

pattern = re.compile(r'coreyms\.com')

matches = pattern.finditer(email)

for match in matches:
    print(match)
```

```
<re.Match object; span=(6, 17), match='coreyms.com'>
```

word Boundry \b

\b = matlab esa word find kro jiski boundry space ya newline(\n) se milti ho.
\B = [^b]

In [44]:

```
email = '''
aadil_coreyms.com

ha haha
'''

pattern = re.compile(r'\bha')

matches = pattern.finditer(email)

for match in matches:
    print(match)

    # basically yeh  first and second ha de rha hai, becoz one has new line and second ha
s space.
```

```
<re.Match object; span=(20, 22), match='ha'>
<re.Match object; span=(23, 25), match='ha'>
```

In [45]:

```
email = '''
aadil_coreyms.com

ha haha
'''

pattern = re.compile(r'\Bha')

matches = pattern.finditer(email)

for match in matches:
    print(match)
                # third 'ha' hai yeh which does not touch with any space and new line.
```

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

In [46]:

```
### start & end matching (^,$)
```

In [47]:

```
email = "start a sentence and end"
```

```
pattern = re.compile(r'^start')
matches = pattern.finditer(email)
for match in matches:
    print(match)

pattern = re.compile(r'end$')
matches = pattern.finditer(email)
for match in matches:
    print(match)
```

```
<re.Match object; span=(0, 5), match='start'>
<re.Match object; span=(21, 24), match='end'>
```

In [48]:

```
# how to match Names havig different parameters or Period.
str = '''
Mr. Schafer
Mr Smith
Ms Davis
Mrs. Robinson
Mr. T
'''
                                                    # | = Or or Either.
pattern = re.compile(r'Mr\.?\s[A-Z]\w*')
matches = pattern.finditer(str)

for match in matches:
    print(match)
```

```
<re.Match object; span=(1, 12), match='Mr. Schafer'>
<re.Match object; span=(13, 21), match='Mr Smith'>
<re.Match object; span=(45, 50), match='Mr. T'>
```

In [49]:

```
pattern = re.compile(r'M(r|s|rs)\.?\s[A-Z]\w*')
matches = pattern.finditer(str)

for match in matches:
    print(match)
```

```
<re.Match object; span=(1, 12), match='Mr. Schafer'>
<re.Match object; span=(13, 21), match='Mr Smith'>
<re.Match object; span=(22, 30), match='Ms Davis'>
<re.Match object; span=(31, 44), match='Mrs. Robinson'>
<re.Match object; span=(45, 50), match='Mr. T'>
```

In [50]:

```
pattern = re.compile(r'(Mr|Ms|Mrs)\.?\s[A-Z]\w*')
matches = pattern.finditer(str)

for match in matches:
    print(match)
```

```
<re.Match object; span=(1, 12), match='Mr. Schafer'>
<re.Match object; span=(13, 21), match='Mr Smith'>
<re.Match object; span=(22, 30), match='Ms Davis'>
<re.Match object; span=(31, 44), match='Mrs. Robinson'>
<re.Match object; span=(45, 50), match='Mr. T'>
```

----------==========++++++++++++----------==========----

Capturting matches from match object by group method

In [51]:

```
import re

urls = '''
https://www.google.com
http://coreyms.com
https://youtube.com
https://www.nasa.gov
'''

pattern = re.compile(r'https?://(www\.)?\w+\.\w+')
                        # s is optional here in https?
matches = pattern.finditer(urls)

for match in matches:
    print(match)
```

```
<re.Match object; span=(1, 23), match='https://www.google.com'>
<re.Match object; span=(24, 42), match='http://coreyms.com'>
<re.Match object; span=(43, 62), match='https://youtube.com'>
<re.Match object; span=(63, 83), match='https://www.nasa.gov'>
```

Grouping

In [52]:

```
pattern = re.compile(r'https?://(www\.)?(\w+)(\.\w+)')
                                    # we made the groups of www. ,domain name, .com by
parenthesis.
matches = pattern.finditer(urls)

for match in matches:
    print(match.group(0))    # match has a group method, for group zero it gives all the
matches found.
```

```
https://www.google.com
http://coreyms.com
https://youtube.com
https://www.nasa.gov
```

Groups Capturing

In [53]:

```
pattern = re.compile(r'https?://(www\.)?(\w+)(\.\w+)')

matches = pattern.finditer(urls)

for match in matches:
    print(match.group(1))
```

```
www.
None
None
www.
```

In [54]:

```
pattern = re.compile(r'https?://(www\.)?(\w+)(\.\w+)')

matches = pattern.finditer(urls)

for match in matches:
    print(match.group(2))
```

```
google
coreyms
youtube
nasa
```

In [55]:

```
pattern = re.compile(r'https?://(www\.)?(\w+)(\.\w+)')

matches = pattern.finditer(urls)

for match in matches:
    print(match.group(3))
```

```
.com
.com
.com
.gov
```

Substituting the Values Of Groups

In [56]:

```
pattern = re.compile(r'https?://(www\.)?(\w+)(\.\w+)')

matches = pattern.finditer(urls)

subbed_urls = pattern.sub(r'\2\3', urls)

print(subbed_urls)

                      # yaha humne kya kiya:-
                    # humne new variable create kiya and usme urls ka group2 and group3
ko store kiya
                    # and phr new url ko print kiya.
```

```
google.com
coreyms.com
youtube.com
nasa.gov
```

In [57]:

```
### Reading phone no.,from file        WRONG CODE                    :-)),:-)),:-))
```

In [58]:

```
import re
pattern = re.compile(r'[50]4[-.]d\d\d\d[-.]d\d\d\d')
with open('conatacts.txt', 'r') as f:
    contents = f.read()


    matches = pattern.finditer(contents)
```

In [59]:

```
print(matches)
```

```
<callable_iterator object at 0x0000000005588448>
```

In [60]:

```
times = []
for result in matches:
    times.append(result.group('time'))
```