In [ ]:
```
#Regular Expressions
if we want to represent a group of strings , according to a particular pattern then
we should use Regular Expression.
```

In [ ]:
```
Example of RE --> fixed pattern is there for our mobile numbers
Example of RE --> Fixed pattern is given to our mail id
Example of RE --> fixed pattern is for vechile or mobile number
Example of RE --:> PAN CARD
Example of RE --> Strong password, weak password , and medium password
```

In [ ]:
```
Mobile Numbers --> 10 --> Starting digit --> 7,8,9,6
In India There is one fixed pattern for the mobile Number:
    1.only 10 digits are  allowed.
    2.No two Mobile numbers are same
    3. Starting digit of a mobile Number is either 6,7,8,9
If someone enters --> 99197843526
```

In [ ]:
```
Email Account --> username@domain_name
domain_name --> yahoo, hotmail, gmail
Can we tell that after @ we will get the domain name
```

In [ ]:
```
Where we need to use Regular Expression?
1.Password
2.Validation of Data
3.Email
4.Translator --> TOC , COMPILER DESIGN
5. Digital Circuit
6. Protocols like TCP/IP
```

In [ ]:
```
How we can create pattern in Python
# re module is use for regular expression
```

In [ ]:
```

```

In [ ]:
```
Finditer()--> return the iterator object which yields match object for every match
```

In [ ]:
```
start() --> return the start index of the match
end () --> return end+1 index of the match
group()--> return the matched string
```

In [14]:
```
#ababababa
#if you want to perform anything on the object for that it is necessary that both the
#operand are objects.
import re

matcher = re.finditer("abc","abaababa")
print(matcher)
for match in matcher:
   # print(match.start())
    print(match.end())
    print(match.group())
```

```
<callable_iterator object at 0x000001A8799F8070>
```

In [22]:
```
import re
x=input("Enter your Mail Id")
matcher = re.finditer("@",x)
print(matcher)
for match in matcher:
    y = match.start()
if  x[y+1:] == "gmail.com":
    print("It is a google account")
elif  x[y+1:] == "yahoo.com":
    print("It is a yahoo account")
elif  x[y+1:] == "hotmail.com":
    print("It is a hotmail account")
else:
    print("Please enter valid mail id")
```

```
Enter your Mail Idshubham@gmail.com
<callable_iterator object at 0x000001A879A7E340>
It is a google account
```

In [ ]:
```
Character Classes :
    [abc] --> either a or b or c
    [^abc] --> except a , b,c
    [a-z] --> any lower alphabet symbol
    [A-Z] --> Any Upper case alphabet symbol
    [a-zA-z] --> any alphabet either lower case or upper
    [0-9] -->any digit from 0-9
    [a-zA-Z0-9]--> any alphabnumeric symbol
    [^a-zA-Z0-9] -->Special Characters
```

In [24]:
```
import re
matcher=re.finditer("[abc]","a7b@k9x")
for match in matcher:
    print(match.start())     #0    #2
    print(match.end())       #1    #3
    print(match.group())     #a    #b
```

```
0
1
a
2
3
b
```

In [27]:
```
import re
matcher=re.finditer("[^abc]","a7b@k9x")
for match in matcher:
    print(str(match.start())+"-----" +str(match.end())+"------"+str(match.group()))
```

```
1-----2------7
3-----4-----@
4-----5-----k
5-----6-----9
6-----7-----x
```

In [30]:
```
import re
matcher=re.finditer("[a-z]","a7b@k9x")
for match in matcher:
    print(str(match.start())+"-----" +str(match.end())+"------"+str(match.group()))
```

```
0-----1-----a
2-----3-----b
4-----5-----k
6-----7-----x
```

In [33]:
```
import re
matcher=re.finditer("[A-Z]","a7bQk9x")
for match in matcher:
    print(str(match.start())+"-----" +str(match.end())+"------"+str(match.group()))
```

```
3-----4-----Q
```

In [34]:
```
import re
matcher=re.finditer("[a-zA-Z]","a7bQk9x")
for match in matcher:
    print(str(match.start())+"-----" +str(match.end())+"------"+str(match.group()))
```

```
0-----1-----a
2-----3-----b
3-----4-----Q
4-----5-----k
6-----7-----x
```

In [35]:
```
import re
matcher=re.finditer("[0-9]","a7bQk9x")
for match in matcher:
    print(str(match.start())+"-----" +str(match.end())+"------"+str(match.group()))
```

```
1-----2-----7
5-----6-----9
```

In [37]:
```
import re
matcher=re.finditer("[a-zA-z0-9]","a7bQ@#k9x")
for match in matcher:
    print(str(match.start())+"-----" +str(match.end())+"------"+str(match.group()))
```

```
0-----1-----a
1-----2-----7
2-----3-----b
3-----4-----Q
6-----7-----k
7-----8-----9
8-----9-----x
```

In [38]:
```
import re
matcher=re.finditer("[^a-zA-z0-9]","a7bQ@#k9x")
for match in matcher:
    print(str(match.start())+"-----" +str(match.end())+"------"+str(match.group()))
```

```
4-----5-----@
5-----6-----#
```

In [ ]:
```
Important Functions related to RE Module
```

In [ ]:
```
1.Match()-->  check the string at the begining of the string
```

In [46]:
```
import re
s=input("Enter the String")
m=re.match(s,"abcabdefg")
if m!=None:
    print("Match is found")
else:
    print("Match is not Found")
```

```
Enter the Stringabd
Match is not Found
```

In [ ]:
```
2.fullMatch -->match the whole string
```

In [48]:
```
import re
s=input("Enter the String")
m=re.fullmatch(s,"ababab")
if m!=None:
    print("Match is found")
else:
    print("Match is not Found")
```

```
Enter the Stringab
Match is not Found
```

In [ ]:
```
3.search()-->search the given pattern
```

In [49]:
```
import re
s=input("Enter the String")
m=re.search(s,"abaaaba")
if m!=None:
    print("Match is found")
    print(m.start())
else:
    print("Match is not Found")
```

```
Enter the Stringaaa
Match is found
2
```

In [ ]:
```
4.findall()->find all the occurances of the match
```

In [51]:
```
import re
l=re.findall("[0-9]","a7b@342reqfv")
print(l)
```

```
['7', '3', '4', '2']
```

In [ ]:
```
sub()--> sub means substitution , or replacement
synatx -->
re.sub(characterclass ,symbol , string)
```

In [52]:
```
import re
s=re.sub("[a-z]","#","a7b@342reqfv")
s
```

Out[52]:
```
'#7#@342#####'
```

In [ ]:
```
^ symbol --> check weather the given target string starts with our provided pattern or not
```

In [56]:
```
import re
s="learning python is easy"
res=re.search("^learning ",s)
if res!=None:
    print("Target start with our matching pattern")
else:
    print("Invalid pattern")
```

```
Target start with our matching pattern
```

In [ ]:
```
$  symbol -->check weather the given target string end with our provided pattern or not
```

In [67]:
```
import re
s="learning python is Easy"
res=re.search("Easy$",s)
if res!=None:
    print("Target end with our matching pattern")
else:
    print("Invalid pattern")
```

```
Target end with our matching pattern
```

In [ ]:
```

```

In [ ]:
```

```