

NAME VALIDATION

- P Likhitha
- Likhitha P
- P.Likhitha
- P Likhitha Reddy
- Likhitha Reddy Putluru
- Putluru Likhitha

```
import re
pattern="[A-Z][A-Za-z]{0,10}[.]{0,1}[.][A-Z][A-Za-z]{0,10}[.]{0,1}[A-Z]{0,1}[A-Za-z]{0,8}"
n=input()
re.match(pattern,n)
```

PHONE NUMBER PATTERN

8523016216
 918523016216
 08523016216
 +918523016216

```
In [3]: pattern="[+]{0,1}[9][1][6-9][0-9]{9}|[0]{0,1}[6-9][0-9]{9}"
n=input()
re.match(pattern,n)
```

9573714909

```
Out[3]: <re.Match object; span=(0, 10), match='9573714909'>
```

Email Validator

- srilalitha11234.m@apssdc.in
- all letters must be lowercase alphabets ==> 6 max 14
- contains some numbers (optional) ==> 1 to 6
- it may contain special charactes(optional) ==> [&,@_....]
- after special character must contain numbers or digits
- must contain @
- contains some alphabets ==> length 4-8
- must contain { . }
- must contain some alphabets ==> length 2-4

COMPREHENSIONS

COMREHENSIONS ARE USED TO REDUCE THE NUMBER OF LINES OF CODE

TYPES

- list comprehension
- set comprehension
- dictionary comprehension

LIST COMPREHENSION

syntax:

- If only if condition is there==>[output loop condition]
- if contains both if and else==>[output(if) condition loop]

```
In [9]: l=[1,2,3,4,5,6,7,8,9] # output : even numbers
l1=[]
for i in l:
    if i%2==0:
        l1.append(i)
print(l1)
```

[2, 4, 6, 8]

```
In [10]: ###List comprehension
s=[i for i in l if i%2==0]
print('even number:',s)
```

even number: [2, 4, 6, 8]

```
In [11]: ['even' if i%2==0 else 'odd' for i in l]
```

Out[11]: ['odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd']

```
In [14]: l=[1,2,3,4,'a','b']
[i for i in l if str(i).isalpha()]
```

Out[14]: ['a', 'b']

In []: