Name Validation

- M Srilalitha
- Srilalitha M
- M. Srilalitha
- M Sri Lalitha
- Sri Lalitha Mulpuru
- Mulpuru Srilalitha

```
In [2]:   1  import re
          2  pattern = "[A-Z][A-Za-z]{0,10}[.]{0,1}[ ][A-Z][A-za-z]{2,10}[ ]{0,1}[A-Z]{0,
          3  n = input()
          4  re.match(pattern,n)
```

M Srilalitha

Out[2]: <re.Match object; span=(0, 12), match='M Srilalitha'>

Phone number pattern

- 6745321986
- 06745321986
- 916745321986
- +916745321986

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

917865455439

Out[3]: <re.Match object; span=(0, 12), match='917865455439'>

Email Validator

- srilalitha.m@apssdc.in (mailto:srilalitha.m@apssdc.in)
- all letters must be lowercase alphabets ===> 6 max 14
- contains some numbers (optional) ==> 1 to 6
- it may contain special charecters(optional) ==>[&.-_]
- after special charecter must contain numbers or digits
- must contain @
- must contains some alphabets ===> length 4-8
- must contain .
- must contains some alphabets ===> length 2-4

Comprehensions

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

Even numbers: [2, 4, 6, 8]

```
In [8]:   1  ### List Comprehension
          2  s = [i for i in l if i%2==0]
          3  print("Even numbers:",s)
```

Even numbers: [2, 4, 6, 8]

```
In [9]:   1  ["Even" if i%2==0 else "odd" for i in l]
```

Out[9]: ['odd', 'Even', 'odd', 'Even', 'odd', 'Even', 'odd', 'Even', 'odd']

```
In [11]:  1  l = [1,2,3,4,"a","b"]
          2  [i for i in l if str(i).isalpha()]  #1,2,3 # "1","2"
          3
```

Out[11]: ['a', 'b']

```
In [ ]:   1
```