

In [1]:

```
1 print(dir(str),end=' ')
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',  
 '__eq__', '__format__', '__ge__', '__getattr__', '__getitem__',  
 '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__',  
 '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__',  
 '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__',  
 '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize',  
 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find',  
 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal',  
 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace',  
 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans',  
 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit',  
 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title',  
 'translate', 'upper', 'zfill']
```

In [2]:

```
1 ## strip(),lstrip(),rstrip()  
2 h = "          work          "  
3 print(h.strip())
```

work

In [3]:

```
1 # lstrip()  
2 j = "      python"  
3 j.lstrip()
```

Out[3]:

'python'

In [4]:

```
1 # rstrip()  
2 j1 = "pen          "  
3 j1.rstrip()
```

Out[4]:

'pen'

In [8]:

```
1 # join()  
2 h = "apssdc"  
3 k = "#".join(h)  
4 print(k)
```

a#p#s#s#d#c

In [13]:

```
1 d1 = "we#l@come to p@yt#h@on wor@ks#hop"
2 d1.split("#")
```

Out[13]:

```
['we', 'l@come to p@yt', 'h@on wor@ks', 'hop']
```

In [14]:

```
1 a = "hello"
2 b = "world"
3 a+b
```

Out[14]:

```
'helloworld'
```

In [15]:

```
1 g = input('Enter any string: ')
2 g
```

Enter any string: jhgjh24343ggjgyg

Out[15]:

```
'jhgh24343ggjgyg'
```

In [18]:

```
1 h = "program"
2 for i in h:
3     print(i,end=' ')
```

```
p r o g r a m
```

In [19]:

```
1 h = "work"
2 h[::-1]
```

Out[19]:

```
'krow'
```

In [22]:

```
1 # mom,dad,level,adam.....
2 st = input()
3 if(st==st[::-1]):
4     print("palindrom")
5 else:
6     print("not palindrom")
```

```
we
not palindrom
```

In [30]:

```

1 # i/p: internship
2 # o/p: i e i
3 j = input()
4 vowels = "aeiouAEIOU"
5 for i in j:
6     if i in vowels:
7         print(i,end=' ')

```

INterNShip

I e i

i/p: ApssDc@123#&

o/p: Uppercase letters are : AD

lowecase letters are : pssc

Digits are : 123

Special characters are : @#&

In [31]:

```

1 st = input()
2 up=lw=dg=sp=""
3 for i in st:
4     if(i.isupper()):
5         up = up+i
6     elif(i.islower()):
7         lw=lw+i
8     elif(i.isdigit()):
9         dg=dg+i
10    else:
11        sp=sp+i
12 print("Uppercase Lettes are: ",up)
13 print("Lowercase Lettes are : ",lw)
14 print("Digits are: ",dg)
15 print("Special Characters are: ",sp)
16

```

...

In []:

```

1 a = 67,68,2,78,23,89

```

In []:

```

1 # Data Structures
2 - list
3 - tuple
4 - dict
5 - set

```

List

- It is used to store multiple items in a single variable
- It is mutable
- We can store heterogeneous data
- It can allow the duplicates
- Represented by []

In [33]:

```
1 # empty list
2 li = []
3 print(li,type(li))
```

[] <class 'list'>

In [34]:

```
1 li1 = [8,5,88,"a","b","c",6.8,43.87]
2 print(li1)
```

[8, 5, 88, 'a', 'b', 'c', 6.8, 43.87]

In [35]:

```
1 li1[::-1]
```

Out[35]:

[43.87, 6.8, 'c', 'b', 'a', 88, 5, 8]

In [40]:

```
1 # min(),max(),len(),sorted(),sum()
2 li = [6,4,8,22,9.5,4.9,77]
3 print(min(li),max(li))
4 print(len(li),sorted(li))
5 print(sum(li))
```

4 77

7 [4, 4.9, 6, 8, 9.5, 22, 77]

131.4

In [41]:

```
1 g = [8,8,5,3,6,8,9,9,92]
2 g
```

Out[41]:

[8, 8, 5, 3, 6, 8, 9, 9, 92]

In [43]:

```
1 print(dir(list),end=' ')
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__  
dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__',  
 '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__',  
 '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',  
 '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reverse  
d__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',  
 '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index',  
 'insert', 'pop', 'remove', 'reverse', 'sort']
```

In [44]:

```
1 # append()  
2 k = [78,4,88,23,90]  
3 k.append(11)  
4 k
```

Out[44]:

```
[78, 4, 88, 23, 90, 11]
```

In []:

```
1
```

In []:

```
1
```