STRINGS

In [7]:

```
name="Harshith Varma"
```

In [8]:

```
#Capitalize - makes first character of String Capital and rest all lower
name.capitalize()
```

Out[8]:

```
'Harshith varma'
```

In [9]:

```
#Reversed - reverses a string
reversed(name)
```

Out[9]:

```
<reversed at 0x225df2d1a30>
```

In [10]:

```
# use .join to display reversed string
" ".join(reversed(name))
```

Out[10]:

```
'a m r a V   h t i h s r a H'
```

In [11]:

```
list(reversed(name))
```

Out[11]:

```
['a', 'm', 'r', 'a', 'V', ' ', 'h', 't', 'i', 'h', 's', 'r', 'a', 'H']
```

In [12]:

```
" KLM ".join(name)
```

Out[12]:

```
'H KLM a KLM r KLM s KLM h KLM i KLM t KLM h KLM   KLM V KLM a KLM r KLM m KLM
a'
```

In [13]:

```
# Remove the characters from the end of string
name=" pw skills "
name.strip()
```

Out[13]:

```
'pw skills'
```

In [14]:

```
name.lstrip()
```

Out[14]:

```
'pw skills '
```

In [15]:

```
name.rstrip()
```

Out[15]:

```
' pw skills'
```

In [17]:

```
# Replace a string
name="Good Moooooorniiiiiiing Hydrebad"
name.replace("Hydrebad","Mumbai")
```

Out[17]:

```
'Good Moooooorniiiiiiing Mumbai'
```

In [18]:

```
name.replace("o",'k')
```

Out[18]:

```
'Gkkd Mkkkkkkrniiiiiiing Hydrebad'
```

In [19]:

```
name.replace('o',"")
```

Out[19]:

```
'Gd Mrniiiiiiing Hydrebad'
```

In [20]:

```
"Hello World"
```

Out[20]:

```
'Hello World'
```

In [21]:

```
"Hello \tWorld"
```

Out[21]:

```
'Hello \tWorld'
```

In [22]:

```
"Hello \tWorld".expandtabs()
```

Out[22]:

```
'Hello   World'
```

Practice

In [25]:

```
name="welcome to pw skills. dat cience masters"
name.replace("dat","date").replace("cience","science")
```

Out[25]:

```
'welcome to pw skills. date science masters'
```

In [27]:

```
# check whether string is upper or not
name.isupper()
```

Out[27]:

```
False
```

In [28]:

```
name.islower()
```

Out[28]:

```
True
```

In [29]:

```
name.isspace()
```

Out[29]:

```
False
```

In [30]:

```
"".isspace()
```

Out[30]:

```
False
```

In [31]:

```
" ".isspace()
```

Out[31]:

```
True
```

In [32]:

```
name.endswith("s")
```

Out[32]:

```
True
```

In [34]:

```
name.endswith("K")
```

Out[34]:

```
False
```

In [35]:

```
name.startswith("H")
```

Out[35]:

```
False
```

In [37]:

```
#AlphaNumeric
a="1j373gh"
a.isalnum()
```

Out[37]:

```
True
```

In [39]:

```
# Check str is digit or not
a.isdigit()
```

Out[39]:

```
False
```

In [41]:

```
#check string is alpha or not
a.isalpha()
```

Out[41]:

```
False
```

In [43]:

```
a="1234"
a.isdigit()
```

Out[43]:

```
True
```

In [44]:

```
a="hshdhshsn"
a.isalpha()
```

Out[44]:

```
True
```

Reverse string using loops

In [51]:

```
n="harshith"
l=len(n)-1
while l!=-1:
    print(n[l],end="")
    l=l-1
```

```
htihsrah
```

Print all vowles is given string

In [52]:

```
name="harshith"
vowels="AEIOUaeiou"
for i in name:
    if i in vowels:
        print(i,end=" ")
```

```
a i
```

List

In [53]:

```
type([])
```

Out[53]:

```
list
```

In [54]:

```
list(name)
```

Out[54]:

```
['h', 'a', 'r', 's', 'h', 'i', 't', 'h']
```

In [57]:

```
name="welcome to pw skills. dat cience masters"
name=name.split()
```

In [58]:

```
name[1]
```

Out[58]:

```
'to'
```

In [59]:

```
name[1]="too"
name
```

Out[59]:

```
['welcome', 'too', 'pw', 'skills.', 'dat', 'cience', 'masters']
```

In [60]:

```
name[::-1]
```

Out[60]:

```
['masters', 'cience', 'dat', 'skills.', 'pw', 'too', 'welcome']
```

In [61]:

```
name[::-2]
```

Out[61]:

```
['masters', 'dat', 'pw', 'welcome']
```

In [62]:

```
name[4:0:-1]
```

Out[62]:

```
['dat', 'skills.', 'pw', 'too']
```

In [69]:

```
name1=name+[["nsbns",141]]
```

In [70]:

```
name1
```

Out[70]:

```
['welcome', 'too', 'pw', 'skills.', 'dat', 'cience', 'masters', ['nsbns', 14
1]]
```

In [71]:

```
name1[-1][1]
```

Out[71]:

```
141
```

In [72]:

```
name*2
```

Out[72]:

```
['welcome',
 'too',
 'pw',
 'skills.',
 'dat',
 'cience',
 'masters',
 'welcome',
 'too',
 'pw',
 'skills.',
 'dat',
 'cience',
 'masters']
```

Check too in list or not

In [76]:

```
for i in name:
    if i=='too':
        print(i,"is found")
        break
```

```
too is found
```

In [77]:

```
if "too" in name:
    print("too is found")
```

```
too is found
```

In [78]:

```
"too" in name
```

Out[78]:

```
True
```

In [80]:

```
lst1=["america","japan","china","india"]
lst2=[23,5,5,33,64,35,3,23,5,35,55]
max(lst1),max(lst2)
```

Out[80]:

```
('japan', 64)
```

In [81]:

```
min(lst1),min(lst2)
```

Out[81]:

```
('america', 3)
```

In [83]:

```
#append
lst1.append("British")
lst1
```

Out[83]:

```
['america', 'japan', 'china', 'india', 'British']
```

In [84]:

```
lst1.pop()
```

Out[84]:

```
'British'
```

In [85]:

```
lst1
```

Out[85]:

```
['america', 'japan', 'china', 'india']
```

In [86]:

```
lst1.pop(1)
```

Out[86]:

```
'japan'
```

In [87]:

```
lst1
```

Out[87]:

```
['america', 'china', 'india']
```

In [88]:

```
lstrem=lst1.pop(1)
lstrem
```

Out[88]:

```
'china'
```

In [89]:

```
lst1
```

Out[89]:

```
['america', 'india']
```

In [90]:

```
lst1.append(lstrem)
```

In [91]:

```
lst1
```

Out[91]:

```
['america', 'india', 'china']
```

In [92]:

```
# Sorting and Reverse in string
```

In [97]:

```
alp=["d","u","f","U"]
alp.reverse()
alp
```

Out[97]:

```
['U', 'f', 'u', 'd']
```

In [98]:

```
alp.sort()
```

In [99]:

```
alp
```

Out[99]:

```
['U', 'd', 'f', 'u']
```

In [101]:

```
alp.sort(reverse=True)
alp
```

Out[101]:

```
['u', 'f', 'd', 'U']
```

In [102]:

```
alp1=alp
alp.append(["s",'j'])
alp
```

Out[102]:

```
['u', 'f', 'd', 'U', ['s', 'j']]
```

In [105]:

```
alp1=[1,2,3,4,5]
```

In [106]:

```
alp1.extend([22,44])
alp1
```

Out[106]:

```
[1, 2, 3, 4, 5, 22, 44]
```

In [107]:

```
l1=[1,2,3]
l2=[4,5,6]
l3=[7,8,9]
matrix=[l1,l2,l3]
matrix
```

Out[107]:

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

In [108]:

```
matrix[2][1:]
```

Out[108]:

```
[8, 9]
```

List Comprehension

In [114]:

```
[i for i in range(20)]
```

Out[114]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [116]:

```
[i if i%2==0 else "" for i in range(20)]
```

Out[116]:

```
[0, '', 2, '', 4, '', 6, '', 8, '', 10, '', 12, '', 14, '', 16, '', 18, '']
```

In [128]:

```
lst=[1,2,3,4,5,6,7,8]
sum([i for i in lst if i%2==0 ])
```

Out[128]:

```
20
```

In [129]:

```
sum([i for i in lst if i%2!=0 ])
```

Out[129]:

```
16
```

In [130]:

```
n=[1,2,3,-2,-4,2]
[i for i in n if i>0]
```

Out[130]:

```
[1, 2, 3, 2]
```

In [131]:

```
lst1=["Apple","Banana","Candy"]
[i[0] for i in lst1]
```

Out[131]:

```
['A', 'B', 'C']
```

In [132]:

```
lst=[[1,2,3],[4,5,6],[7,8,9]]
[j for i in lst for j in i]
```

Out[132]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Find the prime numbers in List

In [156]:

```
num=[1,2,3,4,5,6,7,8,9,10]
for i in num:
    p=0
    for j in range(2,i):
        if i%j==0:
            p=1
            break
    if p==0:
        print(i)
```

```
1
2
3
5
7
```

Print all posible duo's from list

In [141]:

```
n=[1,2,3,4,5,6]
for i in range(0,len(n)):
    for j in range(i,len(n)):
        print("("+str(n[i])+","+str(n[j])+")")
```

```
(1,1)
(1,2)
(1,3)
(1,4)
(1,5)
(1,6)
(2,2)
(2,3)
(2,4)
(2,5)
(2,6)
(3,3)
(3,4)
(3,5)
(3,6)
(4,4)
(4,5)
(4,6)
(5,5)
(5,6)
(6,6)
```

In [144]:

```
n=[1,2,3,4,5,6]
["("+str(n[i])+","+str(n[j])+")" for i in range(0,len(n)) for j in range(i,len(n)) if n[i]!=
```

Out[144]:

```
['(1,2)',
 '(1,3)',
 '(1,4)',
 '(1,5)',
 '(1,6)',
 '(2,3)',
 '(2,4)',
 '(2,5)',
 '(2,6)',
 '(3,4)',
 '(3,5)',
 '(3,6)',
 '(4,5)',
 '(4,6)',
 '(5,6)']
```

In [ ]: