In [32]:
```
a="sud'h"
```

In [33]:
```
b='sud"h'
```

In [34]:
```
a
```

Out[34]:
```
"sud'h"
```

In [35]:
```
b
```

Out[35]:
```
'sud"h'
```

In [36]:
```
c ="this is my first class  for full stack of data science that's why I have  joined this class
```

In [37]:
```
c
```

Out[37]:
```
"this is my first class  for full stack of data science that's why I have  joined this class "
```

In [38]:
```
s = "this is full stack class"
```

In [39]:
```
s[0]
```

Out[39]:
```
't'
```

In [40]:
```
s[-1]
```

Out[40]:
```
's'
```

In [41]:
```
len(s)
```

Out[41]:
```
24
```

In [42]:
```
s[4:10]
```

Out[42]:
```
' is fu'
```

In [43]:
```
s[-1:-10:1]  #output will be blanked because step size is +ve and start & end are -ve
```

Out[43]:
```
''
```

In [44]:
```
s
```

Out[44]:
```
'this is full stack class'
```

In [45]:
```
for i in s:
    print(i)
```

```
t
h
i
s

i
s

f
u
l
l

s
t
a
c
k

c
l
a
s
s
```

In [46]:
```
for i in range(len(s)):
    print(s[i])
```

```
t
h
i
s

i
s

f
u
l
l

s
t
a
c
k

c
l
a
s
s
```

In [47]:
```
s[::-1]  #reverse the string
```

Out[47]:
```
'ssalc kcats lluf si siht'
```

In [48]:
```
s[::1]
```

Out[48]:
```
'this is full stack class'
```

In [49]:
```
s[::-2]
```

Out[49]:
```
'sackaslu ish'
```

In [50]:
```
s[0]
```

Out[50]:
```
't'
```

In [51]:
```
s[0]="x"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-51-291ad313d499> in <module>
----> 1 s[0]="x"

TypeError: 'str' object does not support item assignment
```

In [52]:
```
# NOTE : string is immutable but list are mutable
```

In [53]:
```
"sudh" +" kumar"
```

Out[53]:
```
'sudh kumar'
```

In [54]:
```
"sudh" + 3
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-54-e4ecc15d4dc9> in <module>
----> 1 "sudh" + 3

TypeError: can only concatenate str (not "int") to str
```

In [55]:
```
"sudh" + "3" #typecasting
```

Out[55]:
```
'sudh3'
```

In [56]:
```
"sudh" + str(3) #typecasting
```

Out[56]:
```
'sudh3'
```

In [57]:
```
"sudh" + ["Kumar", 3,4,5,6]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-57-07d2fed71067> in <module>
----> 1 "sudh" + ["Kumar", 3,4,5,6]

TypeError: can only concatenate str (not "list") to str
```

In [58]:
```
"sudh " * 6
```

Out[58]:
```
'sudh sudh sudh sudh sudh sudh '
```

In [59]:
```
s1= "my name is "
```

In [60]:
```
s1
```

Out[60]: 'my name is '

In [61]:
```
s1.find("m")
```

Out[61]: 0

In [62]:
```
s1.find("txm")
```

Out[62]: -1

In [63]:
```
s1.find("name")
```

Out[63]: 3

In [64]:
```
s2= "my name is sudhanshu  kumar, name of org is ineuron "
```

In [65]:
```
s2
```

Out[65]: 'my name is sudhanshu kumar, name of org is ineuron '

In [66]:
```
 # finding the index of first "name" in whole string s2
a= s2.find("name")
for i in range(len("name")):
    print(a+i)
```

```
3
4
5
6
```

In [67]:
```
s2
```

Out[67]: 'my name is sudhanshu kumar, name of org is ineuron '

In [68]:
```
s2.count("my")
```

Out[68]: 1

In [69]:
```
s2.count("a")
```

Out[69]: 4

In [70]:
```
s2.count("name")
```

Out[70]: 2

In [71]:
```
s2.count("Name") #count is case sensitive
```

Out[71]: 0

In [72]:
```
s2.split()
```

Out[72]:
```
['my',
 'name',
 'is',
 'sudhanshu',
 'kumar,',
 'name',
 'of',
 'org',
 'is',
 'ineuron']
```

In [73]:
```
s2.split(',')
```

Out[73]:
```
['my name is sudhanshu  kumar', ' name of org is ineuron ']
```

In [74]:
```
s2.split('n')
```

Out[74]:
```
['my ', 'ame is sudha', 'shu  kumar, ', 'ame of org is i', 'euro', ' ']
```

In [75]:
```
s2.split('na')
```

Out[75]:
```
['my ', 'me is sudhanshu  kumar, ', 'me of org is ineuron ']
```

In [76]:
```
type(s2.split('na'))
```

Out[76]:
```
list
```

In [77]:
```
s2.upper()
```

Out[77]:
```
'MY NAME IS SUDHANSHU  KUMAR, NAME OF ORG IS INEURON '
```

In [78]:
```
s2.lower()
```

Out[78]:
```
'my name is sudhanshu  kumar, name of org is ineuron '
```

In [79]:
```
s3= "My Name is sudhanSHU  kumar, name of org is ineuron "
```

In [80]:
```
s3
```

Out[80]:
```
'My Name is sudhanSHU  kumar, name of org is ineuron '
```

In [81]:
```
s3.swapcase() #wherever we have lower letter it will convert into upper and vice-verca
```

Out[81]:
```
'mY nAME IS SUDHANshu  KUMAR, NAME OF ORG IS INEURON '
```

In [82]:
```
s2
```

Out[82]:
```
'my name is sudhanshu  kumar, name of org is ineuron '
```

In [83]:
```
" ". join(s2)
```

Out[83]:
```
'm y   n a m e   i s   s u d h a n s h u    k u m a r ,   n a m e   o f   o r g   i s   i n e u r o n  '
```

In [84]:
```
"s". join(s2)
```

Out[84]:
```
'msys snsasmses sisss sssusdshsasnsssshsus s sksusmsasrs,s snsasmses sosfs sosrsgs sisss sisnses usrsosns '
```

In [85]:
```
reversed(s2) #output in the format of object
```

Out[85]:
```
<reversed at 0x1acbb0adf10>
```

In [86]:
```
s2[::-1]
```

Out[86]:
```
' norueni si gro fo eman ,ramuk  uhsnahdus si eman ym'
```

In [87]:
```
list(reversed(s2))
```

Out[87]:
```
[' ',
 'n',
 'o',
 'r',
 'u',
 'e',
 'n',
 'i',
 ' ',
 's',
 'i',
 ' ',
 'g',
 'r',
 'o',
 ' ',
 'f',
 'o',
 ' ',
 'e',
 'm',
 'a',
 'n',
 ' ',
 ',',
 'r',
 'a',
 'm',
 'u',
 'k',
 ' ',
 ' ',
 'u',
 'h',
 's',
 'n',
 'a',
 'h',
 'd',
 'u',
 's',
 ' ',
 's',
```

```
 'i',
 ' ',
 'e',
 'm',
 'a',
 'n',
 ' ',
 'y',
 'm']
```

In [88]:
```
for i in reversed(s2):
    print(i)
```

```
n
o
r
u
e
n
i

s
i

g
r
o

f
o

e
m
a
n

,
r
a
m
u
k


u
h
s
n
a
h
d
u
s

s
i

e
m
a
n

y
m
```

In [89]:
```
s4 = "  sudh"
```

In [90]:
```
s4
```

Out[90]:
```
'  sudh'
```

In [91]:
```
s4.strip() #by defaqult it will  remove the space from string but ONLY after and before spaces
```

Out[91]:
```
'sudh'
```

In [92]:
```
s5 = "  su d h"
```

In [93]:
```
s5.strip() # ONLY after and before spaces can be removed by this function
```

Out[93]:
```
'su d h'
```

In [94]:
```
s5.lstrip() # removed space from left side
```

Out[94]:
```
'su d h'
```

In [95]:
```
s5.rstrip() # removed space from right side
```

Out[95]:
```
'  su d h'
```

In [96]:
```
s6="greeting from ineuron"
```

In [97]:
```
s6
```

Out[97]:
```
'greeting from ineuron'
```

In [98]:
```
s6.replace("g", 's')
```

Out[98]:
```
'sreetins from ineuron'
```

In [99]:
```
s6.replace("gr", 'm')
```

Out[99]:
```
'meeting from ineuron'
```

In [100…
```
s6 #original is not affected because string is immutable above replace will be just for the out
```

Out[100…
```
'greeting from ineuron'
```

In [101…
```
s7 = "sudh"
```

In [102…
```
s7.replace("u", "kumar")
```

Out[102…
```
'skumardh'
```

In [103…
```
s[1] = "kumar" #'str' object does not support item assignment
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-103-863cfa42584e> in <module>
----> 1 s[1] = "kumar" #'str' object does not support item assignment

TypeError: 'str' object does not support item assignment
```

In [104…
```
s7.center(20, 'b')  #total length of output will be 20 and in center string is placed sorouned
```

Out[104…
```
'bbbbbbbbsudhbbbbbbbb'
```

In [105…
```
s7.center(20, ' ')
```

Out[105…
```
'        sudh        '
```

In [106…
```
s7.center(20, 'K')
```

Out[106…
```
'KKKKKKKKsudhKKKKKKKK'
```

In [107…
```
s7.center(20, '@')
```

Out[107…
```
'@@@@@@@@sudh@@@@@@@@'
```

In [108…
```
s7.center(20, 'oK')   #TypeError: The fill character must be exactly one character long
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-108-1e1eb2dc7d32> in <module>
----> 1 s7.center(20, 'oK')   #TypeError: The fill character must be exactly one character long

TypeError: The fill character must be exactly one character long
```

In [109…
```
s8="sudh\tku\tmar"
```

In [110…
```
s8
```

Out[110…
```
'sudh\tku\tmar'
```

In [111…
```
s8.expandtabs()
```

Out[111…
```
'sudh    ku      mar'
```

In [112…
```
a1="we all are a part of Full Stack " # practice task
```

In [113…
```
a1.lower()
```

Out[113…
```
'we all are a part of full stack '
```

In [114…
```
a1.count("a")
```

Out[114… 5

In [115…
```
print([i for i in range(len(a1)) if a1[i]=='a'])
```

[3, 7, 11, 14, 28]

In [116…
```
a1.replace("a", "ineuron")
```

Out[116… 'we ineuronll ineuronre ineuron pineuronrt of Full Stineuronck '

In [117…
```
a1.split()
```

Out[117… ['we', 'all', 'are', 'a', 'part', 'of', 'Full', 'Stack']

In [118…
```
a3= "Sudh" # will return true or fasle
```

In [119…
```
a3.isupper()
```

Out[119… False

In [120…
```
a3.islower()
```

Out[120… False

In [121…
```
a3.isspace()
```

Out[121… False

In [122…
```
a4 = "    "
```

In [123…
```
a4.isspace()
```

Out[123… True

In [124…
```
a3.isdigit()
```

Out[124… False

In [125…
```
a3.endswith("h")
```

Out[125… True

In [126…
```
a3.startswith("s") #case sensitive
```

Out[126… False

In [127…
```
a3.isalnum() #either numeric or alphabets  then TRUE
```

Out[127…

```
True
```

In [128…

```
a3.isalpha()
```

Out[128…

```
True
```

In [129…

```
a3.istitle() #check the fisrt letter is uppercase or not
```

Out[129…

```
True
```

In [130…

```
a4= "this is a full stack batch"
```

In [131…

```
count = 0  # program to identifying length of string without using function len()
for i in a4:
    count=count+1
print(count)
```

```
26
```

In [132…

```
for i in range(1, len(a4)):   # reverse the string
    print(s[-i])
```

```
s
s
a
l
c

k
c
a
t
s

l
l
u
f

s
i

s
i
h
t
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-132-9929057e3e4e> in <module>
      1 for i in range(1, len(a4)):   # reverse the string
----> 2     print(s[-i])

IndexError: string index out of range
```

In [133…

```
a5 = "sudh"
ch=len(a5)-1
while ch >=0:
    print(a5[ch])
    ch=ch -1
```

```
h
d
u
s
```

In [134…
```
name = "ineuron"
vowels = "AaEeIiOoUu"
for i in name:
    if i in vowels:
        print("{} is a vowel".format(i))
    else:
        print("{} is not a vowel".format(i))
```

```
i is a vowel
n is not a vowel
e is a vowel
u is a vowel
r is not a vowel
o is a vowel
n is not a vowel
```

In [135…
```
"{} name is sudh".format("my")
```

Out[135… `'my name is sudh'`

In [136…
```
"{} name {} sudh".format("my", "is")
```

Out[136… `'my name is sudh'`

In [137…
```
a= input()
b= input()
```

```
my
is
```

In [138…
```
"{} name {} sudh ".format(a,b)
```

Out[138… `'my name is sudh '`

In [139…
```
"{} name {} sudh {}".format(a,b,"Fsjk")
```

Out[139… `'my name is sudh Fsjk'`

In [140…
```
C= input("Enter a data to check palindrome or not:")
C1=C[::-1]
if C==C1:
    print("it is same ")
else:
    print("it is not a same")
```

```
Enter a data to check palindrome or not:oyo
it is same
```

In [141…
```
l=["sudh",4,35,[34,35]]
```

In [142…
```
l[2]
```

Out[142… 35

In [143…
```
l[0:3]
```

Out[143… ['sudh', 4, 35]

In [144…
```
l[0:3:-1]
```

Out[144… []

In [145…
```
l[3:0:-1]
```

Out[145… [[34, 35], 35, 4]

In [146…
```
l[::-1]
```

Out[146… [[34, 35], 35, 4, 'sudh']

In [147…
```
s="sudh"
```

In [148…
```
l+s
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-148-c62842885d3f> in <module>
----> 1 l+s

TypeError: can only concatenate list (not "str") to list
```

In [151…
```
s="sudh"
s=list(s)
```

In [152…
```
l+s
```

Out[152… ['sudh', 4, 35, [34, 35], 's', 'u', 'd', 'h']

In [153…
```
l*3
```

Out[153… ['sudh', 4, 35, [34, 35], 'sudh', 4, 35, [34, 35], 'sudh', 4, 35, [34, 35]]

In [154…
```
len(l)
```

Out[154… 4

In [156…
```
l
```

Out[156… ['sudh', 4, 35, [34, 35]]

In [155…
```
"sudh" in l
```

Out[155… True

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: