STRING

In [ ]:
```
# We will create a string using single quotes(''),double quotes(""),and triple quotes(""" """).
```

In [5]:
```
name='python'
print(name)
```

```
python
```

In [3]:
```
name="python"
print(name)
```

```
python
```

In [4]:
```
name="""python"""
print(name)
```

```
python
```

type

In [6]:
```
name='python'
type(name)
```

Out[6]:
```
str
```

In [ ]:
```
# Triple quotes use for doc string and multiple lines
# It is an information about the code
```

In [7]:
```
string1="""hi how are you
  im good
  im learning python"""
```

In [8]:
```
string1
```

Out[8]:
```
'hi how are you\n  im good\n  im learning python'
```

In [ ]:
```
# entire string will be in double quotes, the highlited string in single quotes
# entire string will be in single quotes, the highlited string in double quotes
```

In [11]:
```
print("hello 'python'")
```

```
hello 'python'
```

In [12]:
```
print('hello "python"')
```

```
hello "python"
```

type

In [13]:
```
string1='python'
type(string1)
```

Out[13]:
```
str
```

len

In [14]:
```
len(string1)
```

Out[14]:
```
6
```

max

In [16]:
```
max(string1)
```

Out[16]:
```
'y'
```

In [17]:
```
min(string1)
```

Out[17]:
```
'h'
```

In [ ]:
```
# by the using of ascii value we know that what is maax or min value of a string.
```

in

In [ ]:
```
# iterate a loop on string1 print each letter get the ascii value

```

In [19]:
```
for i in 'python':
    print(i,ord(i))
```

```
p 112
y 121
t 116
h 104
o 111
n 110
```

Concatination

In [ ]:
```
# adding the two string
```

In [25]:
```
string1='Hello'
string2='Python'
```

In [24]:
```
string1+string2
```

Out[24]:
```
'HelloPython'
```

In [26]:
```
string1-string2
# i can't do substraction bewtween two strings
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[26], line 1
----> 1 string1-string2

TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

In [27]:
```
string1*string2
# i can't do multiply between two string
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[27], line 1
----> 1 string1*string2

TypeError: can't multiply sequence by non-int of type 'str'
```

In [28]:
```
string1/string2
# i can't do division between two string
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[28], line 1
----> 1 string1/string2

TypeError: unsupported operand type(s) for /: 'str' and 'str'
```

In [29]:
```
2*string2
```

Out[29]:
```
'PythonPython'
```

indexing

In [32]:
```
name="H e l l o"
    # 0 1 2 3 4   python index start with zero
```

In [33]:
```
name[0]
```

Out[33]:
```
'H'
```

In [34]:
```
name[3]
```

Out[34]:
```
'l'
```

In [37]:
```
# print any word/sentence by using range method
# if we used range function on strings use index method
# if you use in operator on string do direct

for i in range(5):
    print(name[i])
```

```
H
e
l
l
o
```

In [38]:
```
name="Hello"
for i in range(6):
    print(name[i])
```

```
H
e
l
l
o
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[38], line 3
      1 name="Hello"
      2 for i in range(6):
----> 3     print(name[i])

IndexError: string index out of range
```

In [40]:
```
name1='hello welcom to python'
print(len(name1))
for i in range(len(name1)):
    print(name1[i],end=' ')


# spaces also consider one character
```

```
22
h e l l o   w e l c o m   t o   p y t h o n
```

In [46]:
```
#using for loop to print the index number of word
#positive index

name="hello"
for i in range(len(name)):
    print("The index of {} is: {}".format(name[i],i))
```

```
The index of h is: 0
The index of e is: 1
The index of l is: 2
The index of l is: 3
The index of o is: 4
```

In [48]:
```
# also do using a while loop
i=0
name='hello'
while i<len(name):
    print("The index of {} is: {}".format(name[i],i))
    i+=1
```

```
The index of h is: 0
The index of e is: 1
The index of l is: 2
The index of l is: 3
The index of o is: 4
```

In [77]:
```
name='hello'
name[-2]
```

Out[77]:
```
'l'
```

In [83]:
```
# negative index
name="hello"
for i in range(-len(name),0):
     print("the negative index of {} is {}".format(name[i],i))
```

```
the negative index of h is -5
the negative index of e is -4
the negative index of l is -3
the negative index of l is -2
the negative index of o is -1
```

In [78]:
```
name="hello"
for i in range(len(name)):
     print("the negative index of {} is".format(name[i]),i-len(name))
```

```
the negative index of h is -5
the negative index of e is -4
the negative index of l is -3
the negative index of l is -2
the negative index of o is -1
```

In [65]:
```
i=0
name4='hello'
while i>-len(name4):
    print('the negative index of {} is : {}'.format(name4[i],-len(name4)-i))
    i-=1
```

```
the -ve index of p is : -6
the -ve index of n is : -5
the -ve index of o is : -4
the -ve index of h is : -3
the -ve index of t is : -2
the -ve index of y is : -1
```

In [67]:
```
#other method

name='python'
i=-len(name)
while i<0:
    print("The negative Index of {} is {}".format(name[i],i))
    i=i+1
```

```
The negative Index of p is -6
The negative Index of y is -5
The negative Index of t is -4
The negative Index of h is -3
The negative Index of o is -2
The negative Index of n is -1
```

In [88]:
```
name='python'
for i in range(len(name)):
    print('the positivr index is: {} the negetive index is: {} for {} '.format(i,i-len(name),name[
```

```
the positivr index is: 0 the negetive index is: -6 for p
the positivr index is: 1 the negetive index is: -5 for y
the positivr index is: 2 the negetive index is: -4 for t
the positivr index is: 3 the negetive index is: -3 for h
the positivr index is: 4 the negetive index is: -2 for o
the positivr index is: 5 the negetive index is: -1 for n
```

count

In [ ]:
```
name="hello how are you"
# how many 'h' are there
#print index of 'h'
#print the no.of vowel
#print the no. of unique vowel
```

In [8]:
```
name="hello how are you"
count=0
for i in range(len(name)):
    if name[i]=="h":
        count=count+1
print("No of repeted h are:",count)
```

```
No of repeted h are 2
```

In [10]:
```
# print index of repated no.
name="hello how are you"
count=0
for i in range(len(name)):
    if name[i]=="h":
        count=count+1
        print(i)
print("No of repeted h are:",count)
```

```
0
6
No of repeted h are: 2
```

In [12]:
```
name="hello how are you"
count=0
for i in range(len(name)):
    if name[i] in "aeiou":
        count=count+1
print("No of repeted vowel are:",count)
```

```
No of repeted vowel are: 7
```

In [ ]:
```
# check unique vowels
```

Mutability-Immutability

In [14]:
```
string1="hello python"
# if you change the value by using index operation: mutable
# if you could not change the value by using index operation: immutable
# I want to replace 'h' with 'H'
# based on index operation
```

In [15]:
```
string1[0]="H"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 string1[0]="H"

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

In [ ]:
```
# string are immutable
```

Slicing

In [ ]:
```
#syntax [start:stop:step]
```

In [22]:
```
string1="hello how are you"
h e l l o   h o w   a  r  e     y  o  u
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  16
```

In [ ]:
```
#nothing mentioned at start postition: simply satrting of letter
#nothing mentioned at stop postion: simply last letter
#nothing mentioned at step size: it is postive direction with step value +1
```

In [17]:
```
string1[3:10]
# direction: positive
#start=3
#stop=10-1=9
```

Out[17]:
```
'lo how '
```

In [19]:
```
string1[1:15:2]
#start=1
#stop=15-1=14
#step=2 post
```

Out[19]:
```
'el o r '
```

In [21]:
```
string1[1:15:-2]
# start=1
# stop=16
# dire=-ve
```

Out[21]:
```
''
```

In [23]:
```
string1[:] # sting1[start:stop]
# notging is mentioned means
# postive direction
# start=0
# stop=till last charcter
```

Out[23]: 'hello how are you'

In [ ]:
```
string1[::] # sting1[start:stop]
# notging is mentioned means
# postive direction
# start=0
# stop=till last charcter
```

In [24]:
```
print(string1[0:])
print(string1[:len(string1)])
print(string1[:])
print(string1[::])
```

```
hello how are you
hello how are you
hello how are you
hello how are you
```

In [25]:
```
string1[-3:-14]
# start=-3
#stop=-14+1=-13
# step value : negative
# not possible
```

Out[25]: ''

In [26]:
```
string1[-14:-3:-1]
# start=-3
# stop=-3+1=2
# step value : negative
# not possible
```

Out[26]: ''

In [27]:
```
string1[-2:-15:-1]
# start=-2
# stop=-15+1=14
# step value : negative
# not possible
```

Out[27]: 'oy era woh ol'

In [28]:
```
string1[-15:8:3]
#start=-15
#stop=8-1=7
#step=postive
```

Out[28]: 'l '

In [29]:
```
string1[::1]
```

Out[29]: 'hello how are you'

In [30]:
```
# reverse string
string1[::-1]
```

Out[30]: 'uoy era woh olleh'

$StringMethod$

In [31]:
```
dir('name')
```

Out[31]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__getstate__',
 '__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',
 'removeprefix',
 'removesuffix',
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
```

```
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

capitalize

In [ ]:
```
# make the first character have upper case and the rest lower case.
```

In [35]:
```
name="hello python"
```

In [32]:
```
help(name.capitalize)
```

```
Help on built-in function capitalize:

capitalize() method of builtins.str instance
    Return a capitalized version of the string.

    More specifically, make the first character have upper case and the rest lower
    case.
```

In [36]:
```
name.capitalize()
```

Out[36]:
```
'Hello python'
```

Upper

In [ ]:
```
# return a copy of the string converted to uppercase
```

In [42]:
```
name="hello python"
name.upper()
```

Out[42]:
```
'HELLO PYTHON'
```

lower

In [ ]:
```
# Return a copy of the string converted to lowercase.
```

In [41]:
```
name1="HELLO PYTHON"
name1.lower()
```

Out[41]:
```
'hello python'
```

casefold

In [ ]:
```
#Return a version of the string suitable for caseless(lower) comparisons.
```

In [44]:
```
name2="HeLLo PythOn"
```

In [45]:
```
help(name2.casefold)
```

```
Help on built-in function casefold:

casefold() method of builtins.str instance
    Return a version of the string suitable for caseless comparisons.
```

In [46]:
```
name2.casefold()
```

Out[46]:
```
'hello python'
```

In [50]:
```
#string1="hello"
#without using string method print "Hello"
#hint:using of slicing and concatination

string1='hello'
string2='H'
string3=string1[1:]
string2+string3
```

Out[50]: 'Hello'

start&endtime

In [63]:
```
string1="welcome to python"
import time
start=time.time()
count=0
for i in string1:
    if i=='o':
        count+=1
print(count)
end=time.time()
print(end-start)
```

```
3
0.00099945068359375
```

In [ ]:
```
# other method
```

In [92]:
```
string1="welcome to python"
string1.count('o')
```

Out[92]: 3

In [93]:
```
string1="hai Hai hai Hai hai"
string1.count('h')
```

Out[93]: 3

In [94]:
```
# count no. of 'h' by using string method
string1="hai Hai hai Hai hai"
string2=string1.lower()
string2.count("h")
```

Out[94]: 5

In [96]:
```
#other method
string1="hai Hai hai Hai hai"
string1.lower().count("h")
```

Out[96]: 5

In [67]:
```
#count no. of 'h' by using without string method
string1="hai hai Hai Hai"
import time
start=time.time()
count=0
for i in string1:
    if i=='h' or i=='H':  # both condition cheaking at a time and or means any condition is true i
        count+=1
print(count)
end=time.time()
print(end-start)
```

```
4
0.0009996891021728516
```

In [70]:
```
string1="Bye bye Bye bye Bye"
count=0
for i in string1.lower():
    if i=="b":
        count=count+1
print(count)
```

```
5
```

In [72]:
```
string1="Bye bye Bye bye Bye"
count=0
for i in string1:
    if i.lower()=="b":
        count=count+1
print(count)
```

```
5
```

In [99]:
```
# in count not use only single latter we can use word and sentence also
string2='ola ola ola ola'
string2.count('ola')
```

Out[99]: 4

In [74]:
```
string1="bye bye bye bye bye"
print(string1.count('bye'))
print(string1.count('ye'))
print(string1.count('by'))
print(string1.count('be'))
```

```
5
5
5
0
```

In [101]:
```
# I want to know how many 'a' are there after 4th index
string2='ola ola ola ola'
string2.count('a',4)
# here 4 means we are counting 'a' from 4th index
```

Out[101]: 3

In [107]:
```
# using for if
string2='ola ola ola ola'
count=0
for i in range(len(string2)-2):
    if i>=4:
        if string2[i]+string2[i+1]+string2[i+2]=='ola':
            count=count+1

print(count)
```

```
3
```

In [136]:
```
# how many 'a' are there between 4 to 8th index
string3='ola ola ola ola'
count=0
for i in range(len(string3)):
    if  i>=4 & i<=8:
        if string3[i]=="a":
            count=count+1

print(count)

#confusion
```

```
4
```

In [130]:
```
string2.count('a',4,8)
```

Out[130]:
```
1
```

Replace

In [139]:
```
string='python'
# i want replace 'H' with 'h'
string[2]='H'
# error because strings are immutable
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[139], line 3
      1 string='python'
      2 # i want replace 'H' with 'h'
----> 3 string[2]='H'

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

In [143]:
```
# normal
string[:3]+'H'+string[4:]
```

Out[143]:
```
'pytHon'
```

In [152]:
```
 # using replace method
string.replace('h','H')
```

Out[152]:
```
'pytHon'
```

In [153]:
```
# print $ in the place of r
string='restart'
print(string.replace('r','$'))
```

```
$esta$t
```

In [154]:
```
string='restart'
print(string.replace("r","$",1))
```

```
$estart
```

In [158]:
```
string='restart'
str1=string[1:]
str2=string1[:1]
str3=str1.replace('r','$',1)
print(str2+str3)


print(string[:1]+string[1:].replace('r','$',1))
```

```
resta$t
resta$t
```

index

In [160]:
```
string1='hello'
string1.index('l')
```

Out[160]: 2

In [161]:
```
string1='restart'
# r e s t a r t
# 0 1 2 3 4 5 6
print(string1.index('t'))
f_o=string1.index('t') # 3
# it will provide only first occurence of charcter index
print(string1.index('t',f_o+1))
```

```
3
6
```

In [174]:
```
#wts the index of last apperence of 't'
string1='restart how to t'
f_o=(string1.index('t'))
s_o=string1.index('t',f_o+1)
t_o=string1.index('t',s_o+1)
fo_o=string1.index('t',t_o+1)
print(f_o,s_o,t_o,fo_o)
```

```
3 6 12 15
```

In [178]:
```
print(string1.count('T')) # it gives 0
print(string1.index('T')) # it gives error
```

```
0

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[178], line 2
      1 print(string1.count('T')) # it gives 0
----> 2 print(string1.index('T'))

ValueError: substring not found
```

In [177]:
```
'Python'.replace('z','ZZZZZ')
```

Out[177]: 'Python'

Find

In [179]:
```
string1='restart how to'
string1.find('t')
string1.index()
```

Out[179]: 3

In [ ]:
```
#what are the difference between index and find

#in find-----Return -1 on failure.
#in index------Raises ValueError when the substring is not found.
```

In [181]:
```
print(string1.count('T')) # 0
#print(string1.index('T')) # sub string error
print(string1.replace('TTTT','&'))# NO error
print(string1.find('TTTT')) # -1
```

```
0
restart how to
-1
```

Split

In [182]:
```
string1.split() # nothing mention means it will take the space
```

Out[182]:
```
['restart', 'how', 'to']
```

In [187]:
```
string2="hai how,are you"
```

In [193]:
```
string2="hai how,are you"
string2.split(',')
```

Out[193]:
```
['hai how', 'are you']
```

In [194]:
```
string2="hai how,are you"
string2.split('a')
```

Out[194]:
```
['h', 'i how,', 're you']
```

In [195]:
```
sent1='rashaad.omkar@gmail.com'
sent1.split('@')
```

Out[195]:
```
['rashaad.omkar', 'gmail.com']
```

In [ ]:
```

```