- Strings
- Lists
- Dictionary
- tuples
- sets
- Lambda function
- File handling

Strings

- strings can intialize in three ways
    - single quotes
        - very common way to present
    - double quotes
        - Double quotes also very common way to present
    - triple quotes
        - Triple quoutes means **doc string**

```
In [1]: str1='python'
        str1
```

```
Out[1]: 'python'
```

```
In [2]: type(str1)
```

```
Out[2]: str
```

```
In [3]: str2='PYTHON'
        str3='10.5'
        str4='10'
        str5='True'
        str6="&$"
```

triple quotes

- triple quotes is used to convey the information
- it is also called as **Doc string**
- it is also called as **multiline**

In [5]:

```
str7="""Hello how are you
      Im good"""
print(str7)
```

```
Hello how are you
      Im good
```

In [6]:

```
str8="I like 'python'"
str8
```

Out[6]:

```
"I like 'python'"
```

- type
- print
- len
- max
- min
- reversed
- sorted
- in
- index
- slice

type

- type is a keyword , it will give the trype of value

In [8]:

```
str1='python'
type(str1)
```

Out[8]:

```
str
```

** what is the difference between keywords and package methods**

- keyword is reserved word
- different methods are avaliable on different packages
- we have limited keywords are there : 35 keywords are there these are fixed
- whenever any keyword : keyword(value)
- before keyword there is another statement
- there is no dot before keyword

    - eval()
    - input()
    - type
    - range
    - print
- if you want use methods from package
    - you need to import the package
    - or packagename.methodname

In [ ]:

```
eval()
input()
type()
print()

random.randint
time.sleep
```

len

In [10]:

```
str1='python'
len(str1), len('naresh it')
```

Out[10]:

```
(6, 9)
```

In [13]:

```
len('  ')
```

Out[13]:

```
2
```

max

- it will retrive the maximum charcter based on ascii value

In [14]:

```
s1='python'
max(s1)
```

Out[14]:

```
'y'
```

In [15]:

```
ord('p'),ord('y'),ord('t'),ord('h'),ord('o'),ord('n')
```

Out[15]:

```
(112, 121, 116, 104, 111, 110)
```

min

- it will retrive the maximum charcter based on ascii value

In [17]:

```
s1='python'
min(s1)
```

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

sum

- It will return the sum of the values
- It will not works for strings
- It will works for numbers

```
In [18]:  sum('123'), sum('python')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 sum('123'), sum('python')

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

Reversed

```
In [21]:  str1=reversed('python')
          for i in str1:
              print(i)
```

```
n
o
h
t
y
p
```

Note

- whenever you see the output between inside greater than and less than symbol
- Means the output is stored inside that
- It is nothing but a **iterator**
- Iterator means the output we can see using loop only

sorted

- Will give the output ascending order or descending order based on ascii values
- Sorted has one argument called reverse
- reverse is a default arguement with value False
- False means it retruns in ascending order
- acesnding means small to big
- Now give reverse=True and check the order

- It should be descending order
- descending meang bigs to small

```
In [23]: ord('p'),ord('y'),ord('t'),ord('h'),ord('o'),ord('n')
```

```
Out[23]: (112, 121, 116, 104, 111, 110)
```

```
In [22]: sorted('python')

         # [104,110,111,112,116,121]
```

```
Out[22]: ['h', 'n', 'o', 'p', 't', 'y']
```

```
In [24]: sorted('python',reverse=True)
```

```
Out[24]: ['y', 't', 'p', 'o', 'n', 'h']
```

```
In [25]: sorted(iterable='python',reverse=True)
         # If you see any /
         # do not assign any arguments before the slash
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[25], line 1
----> 1 sorted(iterable='python',reverse=True)

TypeError: sorted expected 1 argument, got 0
```

```
In [29]: complex() # it is written
         complex(2,3) # w
         complex(real=2,imag=3) # w
         complex(real=2,3) # F
         complex(2,imag=3) # w

         # why failed for complex(real=2,3)
```

```
Out[29]: (2+3j)
```

```
In [27]: import random
         random.randint()  # F
         random.randint(10,20) # w
         random.randint(a=10,b=20) # w
         random.randint(c=20,d=50) # F
         random.randint(a=10,20)
         random.randint(10,b=20)
```

```
In [ ]: 0+0j
        0j
        0
        NT
```

in

```
In [32]: ## WAP ask the user get the number of 'a' in a given string
         str1='hello how are you, im anand'
         # count=0
# iterate through each letter
# if that letter equal to 'a'
# then count=count+1

count=0
for i in str1:
    if i=='a':
        print(i)
        count=count+1

print(count)

# step-1: i='a'  if 'a'=='a'   True
```

```
a
a
a
3
```

- intialisation of strings
- type
- print
- len
- min
- max
- sorted
- reversed
- in

index

In [1]:

```
str1='hello python'
# In python index start with zero
str1[0]
str1[1]
str1[2]
str1[3]

str1[i]
```

Out[1]:

```
'h'
```

In [ ]:

```
h  e   l   l   o   p   y   t   h   o   n
0  1   2   3   4  5   6   7   8   9  10   11
```

In [4]:

```
len(str1)
```

Out[4]:

```
12
```

In [3]:
```
for i in range(12):
    print(i,str1[i])
```

```
0 h
1 e
2 l
3 l
4 o
5  
6 p
7 y
8 t
9 h
10 o
11 n
```

In [5]:
```
str1='naresh it'
n=len(str1)
for i in range(n):
    print(i,str1[i])
```

```
0 n
1 a
2 r
3 e
4 s
5 h
6  
7 i
8 t
```

In [ ]:
```
-12  -11  -10  -9  -8  -7  -6  -5  -4   -3  -2  -1
 h    e    l   l    o       p   y   t   h   o   n
 0    1    2   3    4  5    6   7   8   9  10   11
```

In [8]:
```
str1='hello python'
str1[-12], str1[0]
```

Out[8]:
```
('h', 'h')
```

In [10]:
```
str1[-12]
str1[-11]
str1[-10]

# hello python
# -12  -1
```

Out[10]:
```
'l'
```

In [12]:
```
for i in range(-12,0):
    print(i,str1[i],end=' ' )
```

```
-12 h -11 e -10 l -9 l -8 o -7   -6 p -5 y -4 t -3 h -2 o -1 n
```

In [13]:
```
str1='hello python'
n=len(str1)
for i in range(-n,0):
    print(i,str1[i] )
```

```
-12 h
-11 e
-10 l
-9 l
-8 o
-7  
-6 p
-5 y
-4 t
-3 h
-2 o
-1 n
```

postive index

In [ ]:

```
str1='naresh it'
n=len(str1)
for i in range(n):
    print(i,str1[i])
```

negative index

In [ ]:

```
str1='hello python'
n=len(str1)
for i in range(-n,0):
    print(i,str1[i] )
```

In [ ]:

```
#Q2) WAP to print the each letter index postive way
#    the postive index of h is  0
#    the postive index of e is 1

#Q3) WAP to print the each letter index negative way
#    the negative index of h is  -12
#    the negative index of e is -11

#Q4) WAP to print the each letter index postive and negative way(single for loop
#    the postive index of h is  0 and negative index of h is -12
#    the postive index of e is 1  and negative index of e is -11
```

In [ ]:

```
#Q2) WAP to print the each letter index postive way
#    the postive index of h is  0
#    the postive index of e is 1
str1='hello python'
n=len(str1)
for i in range(n):
    print(i,str1[i])
```

In [24]:

```
# for i in range(len(str1)):
#     print(i,end=' ')

# for i in range(-len(str1),0):
#     print(i)

for i in range(len(str1)):
    print(i,i-len(str1))

# -12  -11  -10  -9  -8  -7  -6  -5  -4   -3  -2  -1
#  h    e    l   l    o       p   y   t   h   o   n
#  0    1    2   3    4  5    6   7   8   9  10   11
```

```
0 -12
1 -11
2 -10
3 -9
4 -8
5 -7
6 -6
7 -5
8 -4
9 -3
10 -2
11 -1
```

In [25]:
```
for i in range(len(str1)):
    print(f"the positive index of {str1[i]} is {i}")
```

```
the positive index of H is 0
the positive index of e is 1
the positive index of l is 2
the positive index of l is 3
the positive index of o is 4
the positive index of   is 5
the positive index of P is 6
the positive index of y is 7
the positive index of t is 8
the positive index of h is 9
the positive index of o is 10
the positive index of n is 11
```

In [26]:
```
for i in range(len(str1)):
    print(f"the negative index of {str1[i]} is {i-len(str1)}")
```

```
the negative index of H is -12
the negative index of e is -11
the negative index of l is -10
the negative index of l is -9
the negative index of o is -8
the negative index of   is -7
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 [28]:
```
for i in range(len(str1)):
    print(f"the positive index of {str1[i]} is {i} and the negative index of {st
the positive index of H is 0 and the negative index of H is -12
the positive index of e is 1 and the negative index of e is -11
the positive index of l is 2 and the negative index of l is -10
the positive index of l is 3 and the negative index of l is -9
the positive index of o is 4 and the negative index of o is -8
the positive index of   is 5 and the negative index of   is -7
the positive index of P is 6 and the negative index of P is -6
the positive index of y is 7 and the negative index of y is -5
the positive index of t is 8 and the negative index of t is -4
the positive index of h is 9 and the negative index of h is -3
the positive index of o is 10 and the negative index of o is -2
the positive index of n is 11 and the negative index of n is -1
```

In [ ]:

```
for i in range(len(str1)):
    print(f"the positive index of {str1[i]} is {i}")

for i in range(len(str1)):
    print(f"the negative index of {str1[i]} is {i-len(str1)}")

for i in range(len(str1)):
    print(f"the positive index of {str1[i]} is {i} and the negative index of {st
```

In [ ]:

```
# Q5) WAP ask the user get the indexes of occurences of 'a' in a given string
# str1='avaialable'

# Iterate through str1
# apply the if condition if charcter == a
# then print that i
```

when to use in and when to use range operator

In [29]:

```
# In this case  i ===== > represents  letter
str1= 'available'
for i in str1:
    print(i)
```

```
a
v
a
i
l
a
b
l
e
```

In [31]:

```
# In this case  i ==== > represents a number i.e. index
str1= 'available'
for i in range(len(str1)):
    print(str1[i])
```

```
a
v
a
i
l
a
b
l
e
```

In [32]:

```
str1= 'available'
for i in range(len(str1)):
    if str1[i]=='a':
        print(i)

# i=0  str1[0]='a'  a=='a'   True  print(o)
# i=1  str1[1]=='a'  'v'=='a'  False
# i=2  str1[2]=='a'  'a'=='a'   True  print(2)
```

```
0
2
5
```

In [33]:

```
#Q6)Print  the  1 occurence of 'a' index is 0
#       the 2 occurence of 'a' index  is 2
#       the 3 occurence of 'a' index   is 5

count=0
str1= 'available'
for i in range(len(str1)):
    if str1[i]=='a':
        count=count+1
        print(f"the {count} occurence of 'a' is {i}")
```

```
the 1 occurence of 'a' is 0
the 2 occurence of 'a' is 2
the 3 occurence of 'a' is 5
```

In [34]:

```
# Q7)  Get the sum of all the indexes
#      Get the count of all occurences :3
str1 = "available"
count = 0
summ = 0
for i in range(len(str1)):
    if str1[i]=='a':
        count = count+1
        summ = summ+i
print(f"The count of indexes of all occurence of 'a' is {count}")
print(f"The sum of indexes of all occurence of 'a' is {summ}")
```

```
The count of indexes of all occurence of 'a' is 3
The sum of indexes of all occurence of 'a' is 7
```

In [ ]:

```
#Q8)
# A)Find the vowels in the given string
# str1: hai how are you
# B)Find the indexes of vowels
# C)Find the count of the vowels
# D)Find the sum of before index of vowels
```

In [35]:

```
str='hi how are you'
for i in range(len(str)):
    if str[i] in 'aeiou':
        print(str[i], end=' ')

# 'h' in 'aeiou'   False
# 'i'  in 'aeiou'  True
```

```
i o a e o u
```

In [ ]: