What is difference between list and tuple?

List	Tuple
List is a mutable sequence, after creating list changes can be done	Tuple is immutable sequence, after creating tuple changes cannot done
List occupy more space because to perform mutable operations	Tuple occupy less space because it is immutable
List is created using []	Tuple is created using ()
List cannot used as an item or element in hashable collections(set, dictionary as key)	Tuple can be used as an item or element in hashable collections (set, dictionary key)
In application development list is used to represent mutable sequence	In application development set is used to represent immutable sequence(list)

Example

```
A=(10,20,30,40,50)

#using index
for i in range(len(A)):
    print(A[i],end=' ')

print()
#using slicing
B=A[0:3]
print(B)
C=A[-1:-3:-1]
print(C)

#for loop
for x in A:
    print(x,end=' ')

print()
#Using iterator
x=iter(A)
value1=next(x)
```

```
value2=next(x)
print(value1,value2)

#Using enumerate
e=enumerate(A)
t1=next(e)
t2=next(e)
print(t1,t2)
```

Output

```
10 20 30 40 50
(10, 20, 30)
(50, 40)
10 20 30 40 50
10 20
(0, 10) (1, 20)
```

Example:

```
#Python - Maximum and Minimum K elements in Tuple
'''
Input : test_tup = (3, 7, 1, 18, 9), k = 2
Output : (3, 1, 9, 18)

Input : test_tup = (3, 7, 1), k=1
Output : (1, 7)

'''


test_tup = (3, 7, 1, 18, 9)
k = 2
stuple=tuple(sorted(test_tup))
print(test_tup)
print(stuple)
min_max_tuple=stuple[:k]+stuple[-k:]
print(min_max_tuple)
```

Output

(3, 7, 1, 18, 9)
(1, 3, 7, 9, 18)
(1, 3, 9, 18)

Sorted()
sorted() is a predefined function in python.
This function sort elements of iterable in ascending or descending order, after sorted elements it returns sorted elements in new collection. sorted() perform immutable operations.

Syntax: sorted(iterable,key=function,reverse=False)

In application development sorted is used with immutable collections.

Example:
```
# Create a List of Tuples with Numbers and Their Cubes

'''
For example, if the input is [1, 2, 3],
the output should be [(1, 1), (2, 8), (3, 27)].
'''


A=[1,2,3]
B=[]

for value in A:
    B.append((value,value**3))

print(A)
print(B)
```

Output
[1, 2, 3]
[(1, 1), (2, 8), (3, 27)]

Example:
 # Python - Adding Tuple to List and Vice - Versa

```
A=[1,2,3,4,5]
B=(10,20,30,40,50)

C=A+list(B)
D=B+tuple(A)

print(A,B,C,D,sep="\n")
```

Output

```
[1, 2, 3, 4, 5]
(10, 20, 30, 40, 50)
[1, 2, 3, 4, 5, 10, 20, 30, 40, 50]
(10, 20, 30, 40, 50, 1, 2, 3, 4, 5)
```

Example:

```
# Python - Sum of tuple elements

'''The original tuple is : (7, 8, 9, 1, 10, 7)
The summation of tuple elements are: 42'''

t1=(7, 8, 9, 1, 10, 7)
s=sum(t1)
print(t1,s,sep="\n")

s=0
for value in t1:
    s=s+value
print(s)
```

Output

```
(7, 8, 9, 1, 10, 7)
42
42
```

Example:

```
#Python - Modulo of tuple elements
'''
The original tuple 1 : (10, 4, 5, 6)
The original tuple 2 : (5, 6, 7, 5)
The modulus tuple : (0, 4, 5, 1)
'''

t1=(10, 4, 5, 6)
t2=(5, 6, 7, 5)

A=[]
for i in range(len(t1)):
    r=t1[i]%t2[i]
    A.append(r)

t3=tuple(A)
print(t1,t2,t3,sep="\n")
```

Output

```
(10, 4, 5, 6)
(5, 6, 7, 5)
(0, 4, 5, 1)
```

Example:

```
#Python - Row-wise element Addition in Tuple Matrix
'''

Input : test_list = [[('Gfg', 3)], [('best', 1)]]
cus_eles = [1, 2]
Output : [[('Gfg', 3, 1)], [('best', 1, 2)]]
'''

test_list = [[('Gfg', 3)], [('best', 1)]]
cus_eles = [1, 2]
```

```
for i in range(len(test_list)):
    for j in range(len(test_list[i])):
        t=test_list[i][j]
        x=list(t)
        x.append(cus_eles[i])
        test_list[i][j]=tuple(x)

print(test_list)
```

Output

```
[[('Gfg', 3, 1)], [('best', 1, 2)]]
```

Example:

```
>>> t1=(10,20,30,40,50)
>>> del t1[0]
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    del t1[0]
TypeError: 'tuple' object doesn't support item deletion
>>> t1[0]=100
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    t1[0]=100
TypeError: 'tuple' object does not support item assignment
>>> t1.remove(10)
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    t1.remove(10)
AttributeError: 'tuple' object has no attribute 'remove'
>>> t1.sort()
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    t1.sort()
AttributeError: 'tuple' object has no attribute 'sort'
```

Strings or str data type

String is collection of characters
String is immutable sequence of characters. After creating string changes cannot done.
String is a non numeric data type, we cannot perform arithmetic operations.

How to create string?

1. String is created using single quotes
2. String is created using double quotes
3. String is created using triple single quotes or double quotes
4. String is created using str() function
    a. Str() → empty string
    b. Str(object) → return string object from existing object

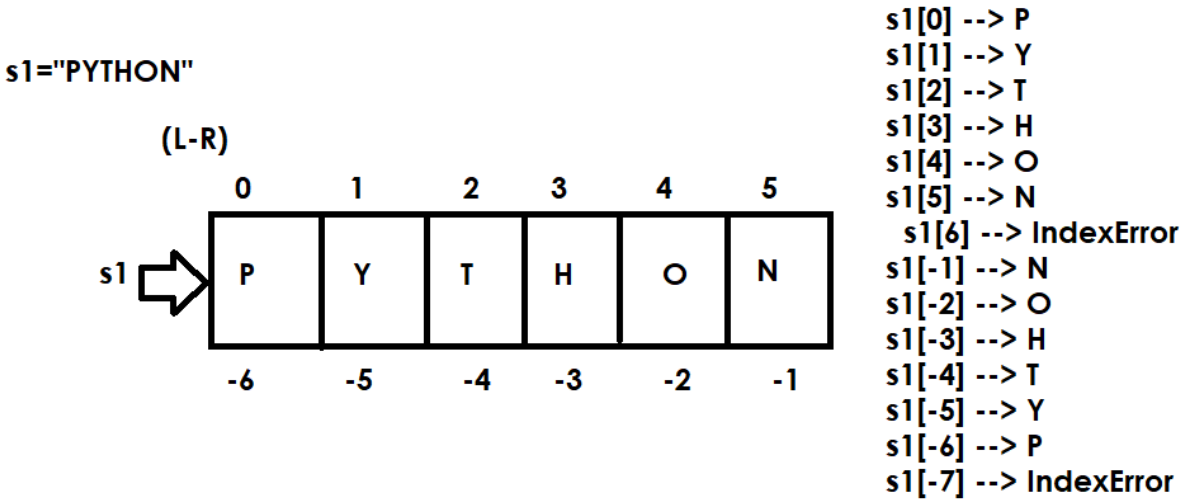
Example:

```
s1='python'
s2='python is "easy" language'
s3="python"
s4="python is 'easy' language"
s5='''python is
high level programming
language'''
s6="""python is
high level programming
language"""
s7=str()
s8=str(20)
s9=str(1.5)
s10=str(1+2j)
s11=str(True)
s12=str("PYTHON")
print(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,sep="\n")
s13=''
print(s13)
```

Output

```
python
python is "easy" language
python
python is 'easy' language
python is
high level programming
language
python is
high level programming
language

20
1.5
(1+2j)
True
PYTHON
```




```
>>> s1="PYTHON"
>>> s1[0]="J"
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    s1[0]="J"
```

```
TypeError: 'str' object does not support item assignment
>>> del s1[0]
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    del s1[0]
TypeError: 'str' object doesn't support item deletion
>>> s1.append("X")
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    s1.append("X")
AttributeError: 'str' object has no attribute 'append'
```

Example:

```
# Check whether the string is Symmetrical or Palindrome

str1=input("Enter any string ")
str2=str1[::-1]

print(str1)
print(str2)

if str1==str2:
    print("Palindrome")
else:
    print("Not Palindrome")


if len(str1)%2==0:
    m=len(str1)//2
    half1=str1[0:m]
    half2=str1[m:]
    if half1==half2:
        print("Symmetrical")
    else:
        print("Not Symmetrical")
else:
    print("Not Symmetrical")
```

Output

```
Enter any string khokho
khokho
ohkohk
Not Palindrome
Symmetrical

Enter any string amaama
amaama
amaama
Palindrome
Symmetrical
```

Example:

```
# Find Length of String in Python
str1=input("Enter any string ")
c=0

for ch in str1:
    c=c+1

print(f'Count or Length is {c}')
```

Output

```
Enter any string nit
Count or Length is 3
```