In [8]:
```
test_str = "gfghsisbjlmknlmkesbestgfgsdlmcnglmfgcsdjnislmdjnllmbestdjskllmgfgc
test_list = ['es','lm','lgf','gfg']

sample=[]

for i in test_list:
    sample.append(test_str.count(i))

print(sample)
maxi=max(sample)

idx=sample.index(maxi)
print(idx)
print(test_list[idx])
```

executed in 10ms, finished 15:16:20 2024-08-16

```
[5, 9, 0, 4]
1
lm
```

2.Lists

- List is also one of the predefined class and treated as list data type.
- The elements of list is written in square braces and seperated by comma.
- On the object of list we can perform indexing and slicing.
- List is mutable, we can update the values inside the list.

In [9]:
```
l1=[]
l2=list()

print(l1,type(l1))
print(l2,type(l2))
```

executed in 8ms, finished 15:20:50 2024-08-16

```
[] <class 'list'>
[] <class 'list'>
```

In [11]:
```
l1=[1,2,3,"hello",True]
print(l1)
```

executed in 5ms, finished 15:22:16 2024-08-16

```
[1, 2, 3, 'hello', True]
```

In [16]:
```
l1=[1,2,3,"hello",True,False,3-6j,10.6]

print(l1[0])
print(l1[-1])

print(l1[0::])
print(l1[::-1])
```

executed in 11ms, finished 15:23:51 2024-08-16

```
1
10.6
[1, 2, 3, 'hello', True, False, (3-6j), 10.6]
[10.6, (3-6j), False, True, 'hello', 3, 2, 1]
```

In [17]:
```
l1[3]="bye"
```

executed in 6ms, finished 15:25:39 2024-08-16

In [19]:
```
print(l1)
```

executed in 7ms, finished 15:26:28 2024-08-16

```
[1, 2, 3, 'bye', True, False, (3-6j), 10.6]
```

inbuilt functions

1.append()

- Used to add the element in the list at the end.

In [21]:
```
l1=[1,2,3,"hello",True,False,3-6j,10.6]

l1.append(10)
l1
```

executed in 12ms, finished 15:27:25 2024-08-16

Out[21]:
```
[1, 2, 3, 'hello', True, False, (3-6j), 10.6, 10]
```

2.insert()

In [23]:
```
l1=[1,2,3,"hello",True,False,3-6j,10.6]

l1.insert(2,12)

print(l1)
```

executed in 7ms, finished 15:29:19 2024-08-16

```
[1, 2, 12, 3, 'hello', True, False, (3-6j), 10.6]
```

3.remove()

In [25]:
```
l1=[1,2,3,"hello",True,False,3-6j,10.6]

l1.remove(2)
```
executed in 6ms, finished 15:31:31 2024-08-16

In [26]:
```
print(l1)
```
executed in 6ms, finished 15:31:33 2024-08-16

```
[1, 3, 'hello', True, False, (3-6j), 10.6]
```

4.pop()

In [31]:
```
l1=[1,2,3,"hello",True,False,3-6j,10.6]

print(l1.pop(3))
print(l1.pop())
```
executed in 7ms, finished 15:33:15 2024-08-16

```
hello
10.6
```

In [32]:
```
l1
```
executed in 10ms, finished 15:33:16 2024-08-16

Out[32]:
```
[1, 2, 3, True, False, (3-6j)]
```

5.count()

In [33]:
```
l1=[1,2,3,1,1,"hello",True,False,1,3-6j,10.6]

print(l1.count(1))
```
executed in 8ms, finished 15:34:39 2024-08-16

```
5
```

6.reverse()

In [34]:
```
l1=[1,2,3,1,1,"hello",True,False,1,3-6j,10.6]

l1.reverse()
print(l1)
```
executed in 8ms, finished 15:35:14 2024-08-16

```
[10.6, (3-6j), 1, False, True, 'hello', 1, 1, 3, 2, 1]
```

7.sort()

In [46]:
```
l2=[10,6,3,22,4,11,7,32,101]

# l2.sort()  # Ascending
l2.sort(reverse=True) # Descending
print(l2)
```

executed in 8ms, finished 15:40:31 2024-08-16

```
[101, 32, 22, 11, 10, 7, 6, 4, 3]
```

8.copy()

In [52]:
```
l2=[10,6,3,22,4,11,7,32,101]

l3=l2.copy()  # Shallow Copy

l3.insert(2,"string")

print(l2)
print(l3)

print(id(l2))
print(id(l3))
```

executed in 9ms, finished 15:43:15 2024-08-16

```
[10, 6, 3, 22, 4, 11, 7, 32, 101]
[10, 6, 'string', 3, 22, 4, 11, 7, 32, 101]
1711150661952
1711150663232
```

In [53]:
```
l2=[10,6,3,22,4,11,7,32,101]

l4=l2  # Deep copy

l4.insert(2,"Dell")
print(l2)
print(l4)

print(id(l2))
print(id(l4))
```

executed in 10ms, finished 15:43:28 2024-08-16

```
[10, 6, 'Dell', 3, 22, 4, 11, 7, 32, 101]
[10, 6, 'Dell', 3, 22, 4, 11, 7, 32, 101]
1711150667392
1711150667392
```

9.Extend()

In [63]:
```
l1=[1,2,3,4,5]
l2=['a','b','c','d','e']
l3=[1.2,2.3,4.5]

l2.extend(l1)

print(l2)
```

executed in 7ms, finished 15:49:28 2024-08-16

```
['a', 'b', 'c', 'd', 'e', 1, 2, 3, 4, 5]
```

In [64]:
```
l3=l1+l2+l3

print(l3)
```

executed in 8ms, finished 15:49:32 2024-08-16

```
[1, 2, 3, 4, 5, 'a', 'b', 'c', 'd', 'e', 1, 2, 3, 4, 5, 1.2, 2.3, 4.5]
```

Nested List:

- List inside a list

In [71]:
```
l1=['dell','hp','acer','macbook']
l2=['redmi','samsung','iphone','realme']
l3=['hyderabad','mumbai','chennai','kolkata']

l4=[l1,l2,l3]

print(l4)
print("="*50)
print(l4[2][2])
```

executed in 10ms, finished 15:53:44 2024-08-16

```
[['dell', 'hp', 'acer', 'macbook'], ['redmi', 'samsung', 'iphone', 'realme'],
['hyderabad', 'mumbai', 'chennai', 'kolkata']]
==================================================
chennai
```

Inputs : l1=[1,2,3,4,[10,20,30,40],["abc","cde","sham"],100,200,[26,10.7]]

Output:

```
[1,2,3,4,10,20,30,40,"abc","cde","sham",100,200,26,10.7]
```

In [76]:
```
l1=[1,2,3,4,[10,20,30,40],["abc","cde","sham"],100,200,[26,10.7]]

l3=[]

for i in l1:
    if type(i)==list:
        for j in i:
            l3.append(j)
    else:
        l3.append(i)
print(l3)
```

executed in 11ms, finished 16:08:30 2024-08-16

```
[1, 2, 3, 4, 10, 20, 30, 40, 'abc', 'cde', 'sham', 100, 200, 26, 10.7]
```

List Comprehension:

list=[var_name for var_name in range/DS]

In [89]:
```
l1=[i for i in range(1,30)]
```

executed in 6ms, finished 16:14:11 2024-08-16

In [90]:
```
print(l1)
```

executed in 7ms, finished 16:14:12 2024-08-16

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 2
2, 23, 24, 25, 26, 27, 28, 29]
```

In [91]:
```
odd=[i for i in l1 if i%2!=0]
```

executed in 5ms, finished 16:14:13 2024-08-16

In [92]:
```
print(odd)
```

executed in 5ms, finished 16:14:13 2024-08-16

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29]
```

In [93]:
```
odd_square=[i*i for i in odd]
```

executed in 6ms, finished 16:21:35 2024-08-16

In [94]:
```
print(odd_square)
```

executed in 7ms, finished 16:21:42 2024-08-16

```
[1, 9, 25, 49, 81, 121, 169, 225, 289, 361, 441, 529, 625, 729, 841]
```

30. Sort a List in Python.(Ascending and Descending)
31. Turn Every item of a list into its square
32. Write a python program to sum all the items in a list

33. Write a python program to multiply all the items in a list
34. Write a Python program to get the Largest number from a list
35. Write a python program to get the least value from a list
36. Write a python program to remove duplicates from a string
37. Reverse a list in python without using inbuilt function and without slicing.

In [103]:
```
string="python full stack"
test="ptfsz"

lis=list(test)

print(lis)
count=0
for i in lis:
    if i in string:
        count+=1

if count==len(lis):
    print("True")
else:
    print("False")
```

executed in 9ms, finished 16:33:42 2024-08-16

```
['p', 't', 'f', 's', 'z']
False
```

In [ ]: