- List is collection of homogeneous and heterogeneous element separated by comma.
- Boundary :[…]
- Element in list are ordered.
- It is a mutable datatype.[we can modify the original list]
- It allows duplicate elements.
- To find length of the list = len(list).
- Default value of the list is [](empty list). And len = 0
- Inside a list it can have one or more list ,and it is called nested list.

  Syntax : var_name = [item1, item2, item3,…..,item n]

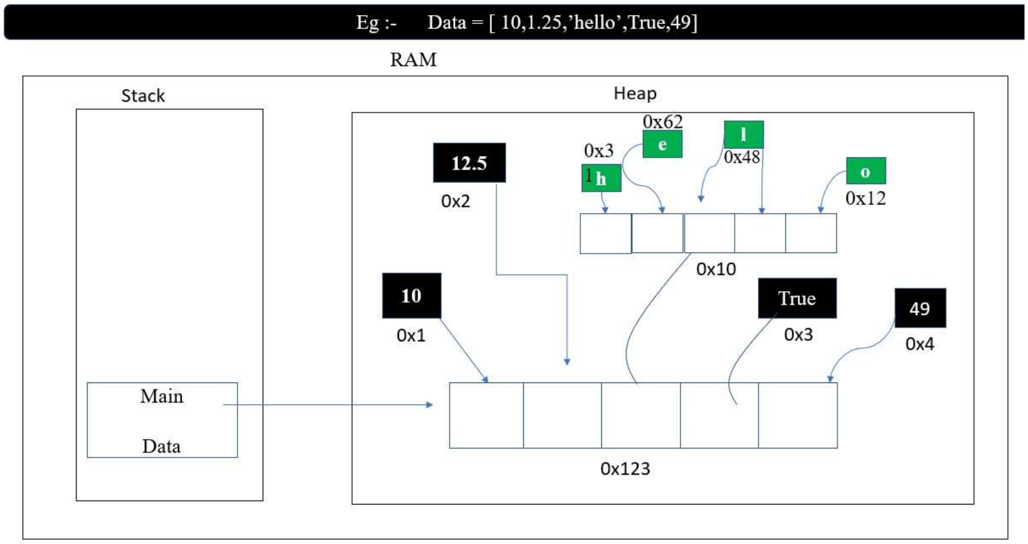
  Ex :  lst = [3,4,2,6,8,9,1]

| | |
|---|---|
| type(lst) | <class 'list'> |
| Len(lst) | 7 |

l = ['apple', 12, 3.4, 4+6j, True]

| | |
|---|---|
| type(l) | <class 'list'> |
| Len(l) | 5 |
| lst = ['hello', 'hai', ['87', 'hi' ,98],65] | # Nested list |

## Memory allocation in list :



**Indexing in list :** it is process of extracting individual element from a list is called indexing

- +ve indexing is start from o and ends with len(lst)-1, it traverses from left to right.
- -ve indexing is start from -1 ,it traverses from right to left.

Ex : names = ['apple' , 'google' , 'yahoo' , 'gmail' , 'amazon' , 'flipchart' ]

| | |
|---|---|
| names[1] | 'google' |
| names[3] | 'gmail' |
| names[-1] | 'flipchart' |
| (names[1 + 3]) | # Prints 4th item of the list |
| (names[1 - 3]) | # Prints 5th item of the list |

names = ['apple' , 'google' , 'yahoo' ,[ 'gmail' , 'amazon'] , 'flipchart' ]

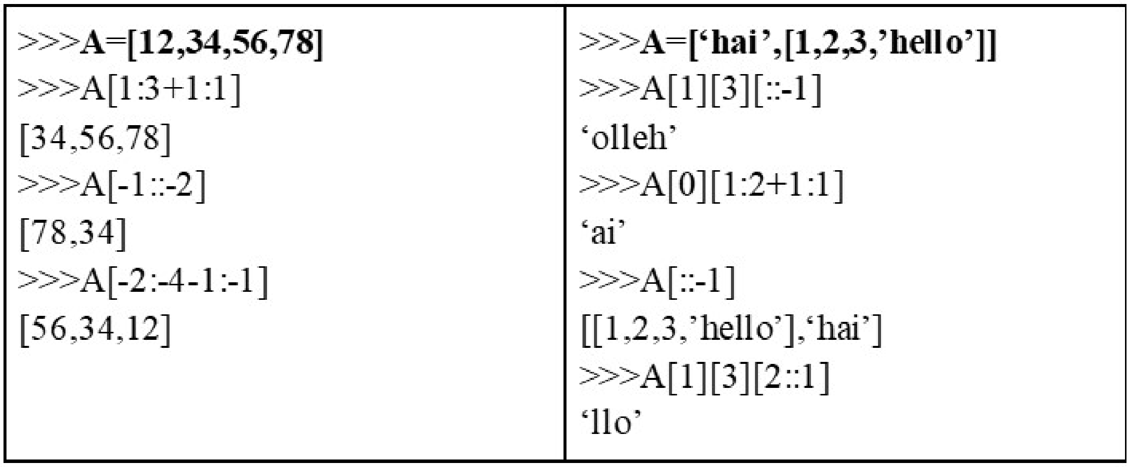
| | |
|---|---|
| names[3][0] | 'gmail' |
| names[3][-1] | 'amazon' |
| names[3[1]] | Type error |

l = [1, 2, 3, ['hello' , 34, 4.5, ['python', 'world']],3+4j, 'hello']

| | |
|---|---|
| l[3][3][0] | 'python' |
| l[3][-1][1] | 'world' |
| l[4] | 3+4j |

**Slicing :** the process of extracting multiple element character simultaneously is called slicing.

Ex :



| | |
|---|---|
| >>>**A=[12,34,56,78]**<br>>>>A[1:3+1:1]<br>[34,56,78]<br>>>>A[-1::-2]<br>[78,34]<br>>>>A[-2:-4-1:-1]<br>[56,34,12] | >>>**A=[‘hai’,[1,2,3,’hello’]]**<br>>>>A[1][3][::-1]<br>‘olleh’<br>>>>A[0][1:2+1:1]<br>‘ai’<br>>>>A[::-1]<br>[[1,2,3,’hello’],‘hai’]<br>>>>A[1][3][2::1]<br>‘llo’ |

names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft']

| | |
|---|---|
| print(names[2:5]) | # Prints all the items from 2nd index upto but not including 5th index. |
| print(names[:4]) | # Prints all items from 0th index and upto 4th index, but not including 4th index. |
| print(names[2:]) | # Prints all items from 2nd index till the end of the List. |

# Slicing using negative indexing

| | |
|---|---|
| print(names[-4:-2]) | # Prints ['amazon', 'facebook'] |
| print(names[-6:5]) | # prints ['google', 'yahoo', 'amazon', 'facebook', 'instagram'] |
| print(names[1:-1]) | # prints ['google', 'yahoo', 'amazon', 'facebook', 'instagram'] |
| print(names[:-1]) | # Prints ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram'] |
| print(names[:]) | # Prints the entire list |
| print(names[::2]) | # Prints alternate items in the list |
| print(names[::-1]) | # Prints the items in the list in reverse order |
| print(names[::2]) | # Prints alternate items in the list |
| print(names[2:7:2]) | |
| print(names[-1:2:-1]) | |
| print(names[::-1]) | # Prints the list in Reverse order |

# Methods in list

**append():**

Adds an element at the end of the list(both individual and collections)
Syntax: list.append(element)
A Element can be of any data type.

| | |
|---|---|
| Eg: [1,2].append([3,4]) | o/p: [1,2,[3,4]] |

names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft']

| | |
|---|---|
| names.append('gmail') | ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft', 'gmail'] |

**extend():**
Extends the existing list with the items of the given sequence.
Syntax: list.extend(iterable)

| | |
|---|---|
| Eg: [1,2].extend([3,4]) | o/p: [1,2,3,4] |
| names.extend(['netflix', 'walmart', 'kroger']) | ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft', 'gmail','netflix', 'walmart', 'kroger'] |

**insert():**
Adds an element at the specified position (both individual and collections)
Syntax: list.insert(pos, elmnt)
Ex : names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft']

| | |
|---|---|
| names.insert(3, 'watsapp') | ['apple', 'google', 'yahoo','watsapp', 'amazon', 'facebook', 'instagram', 'microsoft'] |

**pop():** Removes the element at the specified position
Syntax: list.pop([pos])
By default pop() removes and returns the last element in the list
If the index specified is not present - -> IndexError
Ex :

| | |
|---|---|
| names.pop() | # By default this will remove the last item in the List |
| names.pop(3) | # Removes the item in the 3rd index of the List |

**remove():** Removes the first occurrence of the element specified.
Syntax: list.remove(element)
If the value is not present ValueError
Ex : names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft', 'gmail','netflix', 'walmart', 'kroger']

| | |
|---|---|
| names.remove('kroger') | # Removes the item 'kroger' from the List |

**clear():** It is used to clear the entire list without deleting the list.
It returns None

syntax: list_.clear()

**sort():**
Sorts the list.
A In order to sort, a list should be homogeneous.
Modifies the original list itself→ returns None
Syntax : list.sort([key=function], [reverse=True])
Eg: names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft’]

| | |
|---|---|
| names.sort() | # Sorts the List based on ASCII values # sort method modifies the list inplace. |
| names.sort(reverse=True) | # Sorts the List in Descending Order |
| names.sort(key=len) | # sorts the list based on length of the elements |

**index() :**
Returns the index of the first element with the specified value
Syntax : list.index(element)
Eg: names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'instagram', 'microsoft']

| | |
|---|---|
| names.index("facebook") | 4 |
| names.index("flipkart") | ValueError |

**count() :**
A Returns the number of occurrences of the specified value
Syntax : list.count(element)
Eg: names = ['apple', 'google', 'yahoo', 'amazon', 'facebook', 'amazon', 'microsoft’]

| | |
|---|---|
| names.count("facebook") | 1 |
| names.count("flipkart") | 0 |
| names.count(“amazon”) | 2 |

'''

class examples

names = ['Apple','Google','Yahoo','Microsoft','TestYantra','Amazon']

Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)] on win32

```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Trainer\OneDrive\Desktop\M-24 batch
Tr\M24BATCHTRAINING.py =
>>>
= RESTART: C:\Users\Trainer\OneDrive\Desktop\M-24 batch
Tr\M24BATCHTRAINING.py =
>>> age=26
>>> height=5.9
>>> name='steve'
>>> c=2+5j
>>> b=True
>>> #list :- collection of homegeneous or heterogeneous data elements inside
the pairs of []
>>> #syntax:- var_name = [val1,val2,val3.........valn]
>>> lst=[1,2,3,4,5,6]
>>> type(lst)
<class 'list'>
>>> names=['apple','google','yahoo']
>>> data=['steve',26,5.9]
>>> data
['steve', 26, 5.9]
>>> type(data)
<class 'list'>
>>> lst=[1,2,3,4,5,6,1,3,4,5,6]
>>> lst
[1, 2, 3, 4, 5, 6, 1, 3, 4, 5, 6]
```

```
>>> number=[1,3,5,7]
>>> number
[1, 3, 5, 7]
>>> number=[1,3,5,7]
>>> len(number)
4
>>> number=[1,3,5,7,1,3,7,5]
>>> len(number)
8
>>> numbers=[1,3.4,'steve',3+5j,True,[1,2],23]
>>> len(numbers)
7
>>> int()
0
>>> float()
0.0
>>> str()
''
>>> list()
[]
>>> l=list([2,5,7])
>>> l
[2, 5, 7]
>>> converter = list('hello')
>>> converter
['h', 'e', 'l', 'l', 'o']
```

```
>>>
============= RESTART: C:\Users\Trainer\OneDrive\Desktop\M-24 batch Tr\M24BATCHTRAINING.py =============
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> type(names)
<class 'list'>
>>> len(names)
6
>>> id(names)
1876802738560
>>> name='steve'
>>> name[0]
's'
>>> names[0]
'Apple'
>>> names[1]
'Google'
>>> names[2]
'Yahoo'
>>> names[-1]
'Amazon'
>>> names[-2]
'TestYantra'
>>> 'TestYantra'
'TestYantra'
>>> names
```

```
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names[0]
'Apple'
>>> names[-6]
'Apple'
>>> names[4]
'TestYantra'
>>> names[4][0]
'T'
>>> names[4][1]
'e'
>>> names[0]
'Apple'
>>> names[0][-1]
'e'
>>> names[0:2:1]
['Apple', 'Google']
>>> names[:2:]
['Apple', 'Google']
>>> names[0:3:1]
['Apple', 'Google', 'Yahoo']
>>> names[:3:]
['Apple', 'Google', 'Yahoo']
>>> names[:4:]
['Apple', 'Google', 'Yahoo', 'Microsoft']
>>> names[4::]
```

```
['TestYantra', 'Amazon']
>>> names[3::]
['Microsoft', 'TestYantra', 'Amazon']
>>> names[-1]
'Amazon'
>>> names[-2::1]
['TestYantra', 'Amazon']
>>> names[-2::]
['TestYantra', 'Amazon']
>>> names[-3::]
['Microsoft', 'TestYantra', 'Amazon']
>>> names[-4::]
['Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names[0:6:1]
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names[0:6:2]
['Apple', 'Yahoo', 'TestYantra']
>>> names[1:6:2]
['Google', 'Microsoft', 'Amazon']
>>>
= RESTART: C:\Users\Trainer\OneDrive\Desktop\M-24 batch Tr\M24BATCHTRAINING.py
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names[3]
'Microsoft'
>>> names[3][0:5:]
```

```
'Micro'
>>> names[0]
'Apple'
>>> names[0][0:2:]
'Ap'
>>> names[0]
'Apple'
>>> names[::-1]
['Amazon', 'TestYantra', 'Microsoft', 'Yahoo', 'Google', 'Apple']
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names[::-1]
['Amazon', 'TestYantra', 'Microsoft', 'Yahoo', 'Google', 'Apple']
>>> names[2:5:1]
['Yahoo', 'Microsoft', 'TestYantra']
>>> names[4:1:-1]
['TestYantra', 'Microsoft', 'Yahoo']
>>> lst=[1,2,3]
>>> lst2=[4,5,6]
>>> lst+lst2
[1, 2, 3, 4, 5, 6]
>>> name=['instagram','Apple']
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> name
['instagram', 'Apple']
```

```
>>> names+name

['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon', 'instagram',
'Apple']

>>> l1=[2,4,6]

>>> l2=[8,10,12,14]

>>> l1+l2

[2, 4, 6, 8, 10, 12, 14]

>>> [*l1,*l2]

[2, 4, 6, 8, 10, 12, 14]

>>> [*names,*name]

['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon', 'instagram',
'Apple']

>>> names

['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']

>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']

>>> names

['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']

>>> dir(str)

['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
'__getnewargs__', '__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']

>>> dir(list)

['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__',
'__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',
'__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert',
'pop', 'remove', 'reverse', 'sort']

>>> names

['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']

>>> dir(names)

['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__',
'__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',
'__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert',
'pop', 'remove', 'reverse', 'sort']

>>> names

['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']

>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']

>>> names

['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']

>>> names.append('Gmail')

>>> names

['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon', 'Gmail']

>>> names
```

```
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon', 'Gmail']
>>> names[-]
SyntaxError: invalid syntax
>>> names[-1]
'Gmail'
>>> names.append('instagram')
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon', 'Gmail',
'instagram']
>>> name='steve'
>>> name.upper()
'STEVE'
>>> name
'steve'
>>> names.append('instagram')
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon', 'Gmail',
'instagram', 'instagram']
>>> name
'steve'
>>> type(name)
<class 'str'>
>>> name.append('jobs')
Traceback (most recent call last):
  File "<pyshell#113>", line 1, in <module>
    name.append('jobs')
AttributeError: 'str' object has no attribute 'append'
```

```
>>> names.append('Gmail')
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon', 'Gmail',
'instagram', 'instagram', 'Gmail']
>>> names.upper()
Traceback (most recent call last):
  File "<pyshell#116>", line 1, in <module>
    names.upper()
AttributeError: 'list' object has no attribute 'upper'
>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names.apppend(23)
Traceback (most recent call last):
  File "<pyshell#119>", line 1, in <module>
    names.apppend(23)
AttributeError: 'list' object has no attribute 'apppend'
>>> names.append(23)
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon', 23]
>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names.insert(0,'Gmail')
>>> names
['Gmail', 'Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names.insert(3,'facebook')
>>> names
['Gmail', 'Apple', 'Google', 'facebook', 'Yahoo', 'Microsoft', 'TestYantra',
'Amazon']
>>> names.extend(['watsup','TCS','Wipro'])
>>> names
['Gmail', 'Apple', 'Google', 'facebook', 'Yahoo', 'Microsoft', 'TestYantra',
'Amazon', 'watsup', 'TCS', 'Wipro']
>>> names.extend('TCS')
>>> names
['Gmail', 'Apple', 'Google', 'facebook', 'Yahoo', 'Microsoft', 'TestYantra',
'Amazon', 'watsup', 'TCS', 'Wipro', 'T', 'C', 'S']
>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> len(names)
6
>>> names[0]
'Apple'
>>> names[0]='Gmail'
>>> names
['Gmail', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Amazon']
>>> names[-1]='Gmail'
>>> names
['Gmail', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Gmail']
>>> names[0]
'Gmail'
>>> names[0]='Apple'
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'TestYantra', 'Gmail']
>>> names[-2]='Amazon'
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']
>>> dir(list)
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__',
'__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',
'__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert',
'pop', 'remove', 'reverse', 'sort']
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']
>>> names.pop()
'Gmail'
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon']
>>> names.pop()
'Amazon'
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft']
>>> names.remove('Apple')
>>> names
['Google', 'Yahoo', 'Microsoft']
>>> names.pop(1)
```

```
'Yahoo'
>>> names
['Google', 'Microsoft']
>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']
>>> names.pop()
'Gmail'
>>> names.pop(2)
'Yahoo'
>>> names
['Apple', 'Google', 'Microsoft', 'Amazon']
>>> names.remove('Amazon')
>>> names
['Apple', 'Google', 'Microsoft']
>>> names.remove('Amazon')
Traceback (most recent call last):
  File "<pyshell#161>", line 1, in <module>
    names.remove('Amazon')
ValueError: list.remove(x): x not in list
>>> names
['Apple', 'Google', 'Microsoft']
>>> names.clear()
>>> names
[]
>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']
```

```
>>> names.clear()

>>> names

[]

>>> del

SyntaxError: invalid syntax

>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']

>>> del names

>>> names

Traceback (most recent call last):

  File "<pyshell#172>", line 1, in <module>

    names

NameError: name 'names' is not defined

>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']

>>> names

['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']

>>> del names

>>> names

Traceback (most recent call last):

  File "<pyshell#176>", line 1, in <module>

    names

NameError: name 'names' is not defined

>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']

>>> del names[0]

>>> names

['Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']

>>> del names[-1]
>>> names
['Google', 'Yahoo', 'Microsoft', 'Amazon']
>>> names=['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']
>>> names
['Apple', 'Google', 'Yahoo', 'Microsoft', 'Amazon', 'Gmail']
>>> names.sort()
>>> names
['Amazon', 'Apple', 'Gmail', 'Google', 'Microsoft', 'Yahoo']
>>> alpha=[q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m]
Traceback (most recent call last):
  File "<pyshell#186>", line 1, in <module>
    alpha=[q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m]
NameError: name 'q' is not defined
>>> alpha=[q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m]
Traceback (most recent call last):
  File "<pyshell#187>", line 1, in <module>
    alpha=[q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m]
NameError: name 'q' is not defined
>>> alpha=[,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m]
SyntaxError: invalid syntax
>>> alpha=[w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m]
Traceback (most recent call last):
  File "<pyshell#189>", line 1, in <module>
    alpha=[w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m]
NameError: name 'w' is not defined
>>> alpha=['w','e','r','y','a','g','b']
>>> alpha
['w', 'e', 'r', 'y', 'a', 'g', 'b']
>>> alpha.sort()
>>> alpha
['a', 'b', 'e', 'g', 'r', 'w', 'y']
>>> number=[1,5,2,7,2,9]
>>> number
[1, 5, 2, 7, 2, 9]
>>> number.sort()
>>> number
[1, 2, 2, 5, 7, 9]
>>> names
['Amazon', 'Apple', 'Gmail', 'Google', 'Microsoft', 'Yahoo']
>>> names.reverse()
>>> names
['Yahoo', 'Microsoft', 'Google', 'Gmail', 'Apple', 'Amazon']
>>> number.reverse()
>>> number
[9, 7, 5, 2, 2, 1]
>>> names
['Yahoo', 'Microsoft', 'Google', 'Gmail', 'Apple', 'Amazon']
>>> names.sort()
>>> names
['Amazon', 'Apple', 'Gmail', 'Google', 'Microsoft', 'Yahoo']
>>> names.reverse()
>>> names
```

```
['Yahoo', 'Microsoft', 'Google', 'Gmail', 'Apple', 'Amazon']

>>> dir(list)

['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__',
'__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',
'__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert',
'pop', 'remove', 'reverse', 'sort']

>>> names

['Yahoo', 'Microsoft', 'Google', 'Gmail', 'Apple', 'Amazon']

>>> names.count('Yahoo')

1

>>> number

[9, 7, 5, 2, 2, 1]

>>> number.count(2)

2

>>> names.count('Google')

1

>>> names.index('Yahoo')

0

>>> names.index('Gmail')

3

>>> names.index('microsoft')

Traceback (most recent call last):

  File "<pyshell#216>", line 1, in <module>

    names.index('microsoft')
```

```
ValueError: 'microsoft' is not in list
>>> names.index('Microsoft')
1
>>> number
[9, 7, 5, 2, 2, 1]
>>> number.index(2)
3
>>> number.rindex(2)
Traceback (most recent call last):
  File "<pyshell#220>", line 1, in <module>
    number.rindex(2)
AttributeError: 'list' object has no attribute 'rindex'
>>> names
['Yahoo', 'Microsoft', 'Google', 'Gmail', 'Apple', 'Amazon']
>>> names.copy()
['Yahoo', 'Microsoft', 'Google', 'Gmail', 'Apple', 'Amazon']
>>> new_names=names.copy()
>>> new_names
['Yahoo', 'Microsoft', 'Google', 'Gmail', 'Apple', 'Amazon']
>>> x=names
>>> x
['Yahoo', 'Microsoft', 'Google', 'Gmail', 'Apple', 'Amazon']
>>>
'''
```