**List**

* a list is a versatile and mutable data structure that allows you to store and organize a collection of point values.
* These items can be of any data type, including numbers, strings, or even other lists.
* Lists are defined by enclosing the elements in square brackets [] and separating them with commas.

example

num = [1,2,3,4,5]

num

[1, 2, 3, 4, 5]

characteristics of list

1. **Mutable:**

can be modified , updated with built in methods in directory ,dir

example:

dir(list)

['\_\_add\_\_',

'\_\_class\_\_',

'\_\_class\_getitem\_\_',

'\_\_contains\_\_',

'\_\_delattr\_\_',

'\_\_delitem\_\_',

'\_\_dir\_\_',

'\_\_doc\_\_',

'\_\_eq\_\_',

'\_\_format\_\_',

'\_\_ge\_\_',

'\_\_getattribute\_\_',

'\_\_getitem\_\_',

'\_\_getstate\_\_',

'\_\_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']

**lets apply some methods**

* append can add single element at the last of the list

num.append(6)

num

[1, 2, 3, 4, 5, 6]

* extend can add the more than one elements in the end of the list

num.extend([7,8,9])

num

[1, 2, 3, 4, 5, 6, 7, 8, 9]

* remove works with the value of the element

num.remove(9)

num

[1, 2, 3, 4, 5, 6, 7, 8]

* del works with the index of the element

del num[0]

num

[2, 2, 3, 4, 5, 5, 6, 7, 8]

* insert use to put an element in between

num.insert([4],5)

num

[1, 2, 3, 4, 5, 5, 6, 7, 8]

1. **heterogenous**

* lists are heterogenous, may contain elements of different data types eg:list, string , integers etc

example:

st1 = ['aamir', 25 , 5.2, ]

print(type(st1[0]))

= class str

print(type(st1[1]))

=class int

print(type(st1[2]))

=classfloat

1. **Indexing**

* position of an element in the list
* we can access the individual element in the list due to its unique index (position)
* python uses zero index language (starts from 0)
* uses [] to access the single element with its position in between

|  |  |
| --- | --- |
| positive  indexing | negative  indexing |
| * left to right | * **right to left** |
| * strats from 0 , and so on (forward) | * **starts from -1 and backwards** |
| * Example   num =[1, 2, 3, 4, 5, 5, 6, 7, 8]  print(num[0])  = 1 | * **Example**   **num[1,2,3,4,5,6,7,8]**  **Print(num[-1])**  **= 8** |

**#lets suppose we have to insert 2 in thr the list**

Example:

​num.insert(1,2) #here we used indexing to romeve the referred elemnt in the list

num= [1, 2, 2, 2, 3, 4, 5, 5, 6, 7, 8]

1. **length**

* we can find the length of the list with len function .

example

len(num)

10

1. **memberships**

* we can find the presence and absence of an elemnts with this keyword
* in
* not in

example

4 in num

True

44 in num #bool data type

False

44 not in num

True

1. **slicing**

* continuous part of the list
* create a new list.
* Use the colon : to specify a range of indices.
* The syntax is sarting\_index : end\_index
* it includes elements from start\_index to end index , escluding the elements end\_index

For example:

slice\_of\_num = num[2:-1]

slice\_of\_num

[2, 3, 4, 5, 5, 6, 7]

1. **list of lists(nested list)**

* nested list
* having multiple list in single list

example

st1 =["aamir", "DS", 100 ]

st2 =["beenish", "DS", 100]

​

students= [st1 , st2, ["daanish", "DS", 100]]

students

[['aamir', 'DS', 100], ['beenish', 'DS', 100], ['daanish', 'DS', 100]]

len(students) #having 3 elemnts (3 list)

3

students = [students , ["fasih", "DS" , 100]]

students

[[['aamir', 'DS', 100], ['beenish', 'DS', 100], ['daanish', 'DS', 100]],

['fasih', 'DS', 100]]

#to extract list from a list of list

* use index of the list

example

​students[1]

['fasih', 'DS', 100]

#extract single elemnts from the list which is present in the nested list

* use index to access the list then again use index to access element

example

students[1][0]

'fasih'