List:

A white background with red text

AI-generated content may be incorrect.

What is the List:

Creation:

L1=[1,2,3,4,5,6]

print(L1)

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

L2=[1.8,1.8,3.5,4.6,5.7]

print(L2)

#[1.8, 1.8, 3.5, 4.6, 5.7]

L3=["John","Angle"]

print(L3)

#['John', 'Angle']

#List(iterable)

L4=list((3,5,7,9))

print(L4)

#[3, 5, 7, 9]

L5=list("abcde")

print(L5)

#['a', 'b', 'c', 'd', 'e']

L6=[]

print(L6)

#[]

Representation

L1=[6,5,4,2,3,2]

for i in L1:

    print(i,end=' ')

#   6 5 4 2 3 2

Heterogeneous

L1=[7,3.2,"john",True,5+6j]

for i in L1:

    print(i,end=' ')

# 7 3.2 john True (5+6j)

A screenshot of a computer

AI-generated content may be incorrect.

L1=[7,3.2,"john",True,5+6j]

for i in L1:

    print(i,end=' ')

# 7 3.2 john True (5+6j)

L1[3]=5

print(L1)

# [7, 3.2, 'john', 5, (5+6j)]

L1[2]="Raghu"

print(L1)

# [7, 3.2, 'Raghu', 5, (5+6j)]

L1[2][3]='a'

print(L1)

#  TypeError: 'str' object does not support item assignment

Mutable

L1=[2,4,6,8,10,12]

L1[2]=16 #Modify

L1.append(25) #Append

print(L1)

# [2, 4, 16, 8, 10, 12, 25]

L1=[2,4,6,8,10,12]

L1[2]=16 #Modify

L1.append(25) #Append

print(L1)

# [2, 4, 16, 8, 10, 12, 25]

* Ordered collection of heterogenous elements
* It’s a mutable
* Can have duplicates

INDEXING AND SLICING

Read:

# list is mutable

#Read/Write

#Read

 # Indexing

 # slicing

#Read Indexing

L1=[3,6,9,12,15,18,21]

print(L1[4])

# 15

#Slicing

print(L1[::]) #Start,Stop,Step

# [3, 6, 9, 12, 15, 18, 21]

print(L1[:]) #Begin,#End

# [3, 6, 9, 12, 15, 18, 21]

print(L1[2:]) #2 to End

#[9, 12, 15, 18, 21]

print(L1[:6]) #0 to 5

[3, 6, 9, 12, 15, 18]

print(L1[2:6]) #2 to 5

# [9, 12, 15]

print(L1[-5:-2]) #-ve indexing

# [9, 12, 15]

#forward print

#9,12,15

print(L1[::])

#[3, 6, 9, 12, 15, 18, 21]

print(L1[::1])

# [3, 6, 9, 12, 15, 18, 21]

print(L1[::2])

#[3, 9, 15, 21]

print(L1[::3])

# [3, 12, 21]

print(L1[::-1])

# [21, 18, 15, 12, 9, 6, 3]

#Reverse,backward,indices also backwards

print(L1[4::-1])

# [15, 12, 9, 6, 3]

print(L1[4:0:-1])

# [15, 12, 9, 6]

print(L1[-3:-7:-1])

# [15, 12, 9, 6]

Indexing and Slicing (Write)

L1=[1,2,3,4,5]

L1[1]=10

print(L1)

# [1, 10, 3, 4, 5]

L1[3]=[10,11]

print(L1)

#Nested List

# [1, 10, 3, [10, 11], 5]

# L1=[1,2,3,4,5]

#L1[start:stop] #  any no of elements can insert

# L1[start:stop:step]

#step no of elements

L1=[1,2,3,4,5]

L1[0:0]=[10]

print(L1)

# [10, 1, 2, 3, 4, 5]

L1[3:3]=[10]

print(L1)

# [10, 1, 2, 10, 3, 4, 5]

L1=[1,2,3,4,5]

L1[5:5]=[10]

print(L1)

#[1, 2, 3, 4, 5, 10]

L1=[1,2,3,4,5]

L1[9:9]=[10]

print(L1)

#[1, 2, 3, 4, 5, 10]

L1=[1,2,3,4,5]

L1[3:3]=[10,11,12]

print(L1)

# [1, 2, 3, 10, 11, 12, 4, 5]

L1=[1,2,3,4,5]

L1[1:4]=[10]

#as many will values we can ive

#1 10 15 (2,3,4)--> Removed

print(L1)

# [1, 10, 5]

L1=[1,2,3,4,5]

L1[1:4]=[10,11]

print(L1)

#[1, 10, 11, 5]

L1=[1,2,3,4,5]

L1[1:4]=[10,11,12,13]

print(L1)

#[1, 10, 11, 12, 13, 5]

### step

# need to give exact number

L1=[1,2,3,4,5]

L1[::2]=[10,11,12]

#1 3 5 replace

print(L1)

# [10, 2, 11, 4, 12]

L1=[1,2,3,4,5]

L1[::-1]=[10,11,12,13,14]

print(L1)

# [14, 13, 12, 11, 10]

L1=[1,2,3,4,5]

L1[3:0:-1]=[12,13,14]

#

print(L1)

#[1, 14, 13, 12, 5]

List Concatination and repetation:

Operations on list

A screenshot of a computer screen

AI-generated content may be incorrect.

Concatination:

# #concatation

L1=[1,2,3]

L2=[8,9,10]

print(L1+L2)

#[1, 2, 3, 8, 9, 10]

# print(L1+4)

# # TypeError: can only concatenate list (not "int") to list

print(L1+[4])

# [1, 2, 3, 4]

L1=[1,2,3]

L1.extend([4,5,6])

print(L1)

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

Repetation:

#### Repetation

L1=[1,2,3]

L2=L1\*3

print(L2)

# [1, 2, 3, 1, 2, 3, 1, 2, 3]

L2=L1\*2.5

#    L2=L1\*2.5

# TypeError: can't multiply sequence by non-int of type 'float'

Membership:

# in , not in

L1=[1,2,3,4,5]

print(3 in L1)

#True

L1=[[1,2],[3,4],5]

print(3 in L1)

#False

print([3,4] in L1)

#True

L1=["Red","Green","Blue"]

for x in L1:

    print(x,end=" ")

# Red,Green,Blue

Comparision:

List comparision

L1=[1,2,3]

L2=[1,2,3]

L3=[3,2,1]

print(L1==L2)

#True

print(L1<L3)

#True

L1=[1,2,3]

L2=[1,2,3,4]

L3=[1,2,1]

print(L1<L2)

#True

print(L1<L3)

#False

L1=[2]

L3=[1,2,1]

print(L1>L3)

#True

Traverse:

L1=[1,2,3,4,5]

L1.append(6)

print(L1)

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

# L1.append(7,8)

# # #TypeError: append() takes exactly one argument (2 given)

L1=[]

L1.append(10)

print(L1)

#10

L1.append("python")

print(L1)

#[10, 'python']

L1.append([1,2])

print(L1)

#[10, 'python', [1, 2]]

L1=[1,2,3,4,5]

L1[5:5]=[10]

print(L1)

#[1, 2, 3, 4, 5, 10]

L1[len(L1):]=[12]

print(L1)

#[1, 2, 3, 4, 5, 10, 12]

Extend:

extend(iterable)

L1=[1,2,3,4]

L1.extend([5,6,7])

print(L1)

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

L1=[1,2]

L1.extend("python")

print(L1)

# only iterable

# [1, 2, 'p', 'y', 't', 'h', 'o', 'n']

L1=[1,2,3,4]

L1.extend(range(5,8))

print(L1)

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

INSERT:

##### insert

L1=[1,2,3,4]

L1.insert(0,50)

print(L1)

#[50, 1, 2, 3, 4]

L1.insert(70,"Python")

#Last index

print(L1)

# [50, 1, 2, 3, 4, 'Python']

L1=[1,2,3,4]

L1.insert(2,"python")

print(L1)

#[1, 2, 'python', 3, 4]

L1=[1,2,3,4]

L1[2:2]=[55]

print(L1)

# [1, 2, 55, 3, 4]

REMOVING ELEMENT in LIST:

Pop:

L1=[1,2,3,4,5,6]

print(L1.pop())

#6

print(L1)

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

L1=[1,2,3,4,5,6]

print(L1.pop(2))

#3

print(L1)

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

L1=[]

L1.pop()

# IndexError: pop from empty list

Remove:

L1=[1,2,3,4,5,3]

L1.remove(3)

print(L1)

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

L1.remove(3)

print(L1)

#[1, 2, 4, 5]

Clear:

L1=[1,2,3,4,5]

L1.clear()

print(L1)

#[]

Del:

L1=[1,2,3,4,5,6]

del L1[3]

print(L1)

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

L1=[1,2,3,4,5,6]

del L1[1:4]

print(L1)

# [1,5,6]

L1=[1,2,3,4,5,6]

del L1

print(L1)

# NameError: name 'L1' is not defined

INDEX and SORT/REVERSE:

Index

L1=[10,20,30,40,10,20,30,20]

print(L1.index(20))

# 1

print(L1.index(20,2))

# 5

print(L1.index(20,2,6))

# 5

Count:

L1=[10,20,30,40,10,30,20,40]

print(L1.count(20))

# #2

Reverse

L1=[10,20,30,40,50,60,70]

L1.reverse()

print(L1)

#[70,60,50,40,30,20,10]

Sort:

# # # #sort(\*,key=None,reverse=False)

# # # # key valued values

L1=[70,10,60,20,50,30,40]

L1.sort()

# increasing order

print(L1)

#[10, 20, 30, 40, 50, 60, 70]

# by default increasing order

L1=[70,10,60,20,50,30,40]

L1.sort(reverse=True)

#decreasin order

[10, 20, 30, 40, 50, 60, 70]

L1=[“coat”,”python”,”block”,”cat”]

L1.sort()

print(L1)

#[‘block’, ‘cat’, ‘coat’, ‘python’]

List Comprehensions:

L1=[x for x in range(1,5)]

print(L1)

#[1, 2, 3, 4]

L2=[x\*\*2 for x in range(1,5)]

print(L2)

#[1, 4, 9, 16]

L3=[ x.casefold() for x in "PyThoN"]

print(L3)

#['p', 'y', 't', 'h', 'o', 'n']

L4=[int(x) for x in "12345"]

print(L4)

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

L5=[x for x in "ab\*cd7e" if x.isalpha()]

print(L5)

#['a', 'b', 'c', 'd', 'e']

Nested List:

L1=[1,2,3,4]

L2=[]

L3=[[1,2],3,4,5]

L4=[[1,2],[3,4],[5,6]]

L5=[[1,2,[3,4]]]

L6=[[1,2],[3,4],[5,6]]

print(L6[0])

#[1,2]

print(L6[2][0])

#5

L1=[[1,2,3,4],[5,6,7,8],[9,10,11,12]]

print(L1)

# [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]

print(L1[0][2])

#3

print(L1[1][2])

#7

print(L1[2][3])

#12