

Lists in Python

By
Dr. Selva Rani B, SCORE

List

- a single variable, stores more than one value of different data types, separated by comma
 - a sequence of values known as elements/items
 - class "List"
1. Creation (i) Creating an empty List a) Using constructor b) Using [] (ii) Creating a non-empty List a) Using constructor b) Using []
 2. Accessing Elements (i) Indexing (Positive, Negative) (ii) Slicing
 3. Builtin Functions (i) len() (ii) max() (iii) min() (iv) sum() (v) random.shuffle()
 4. Operators (i) + (ii) * (iii) in (iv) is (v) del
 5. List Comprehensions
 6. List Methods (i) append() (ii) extend() (iii) count() (iv) copy() (v) clear() (vi) index() (vii) insert() (viii) pop() (ix) remove() (x) reverse() (xi) sort()
 7. Nested Lists

```
In [ ]: ▶ # 1.(i) Creating an empty List
        # (a) Using constructor
        L1 = list()
        print(L1)
        print(type(L1))
```

```
In [ ]: ▶ # 1.(i) Creating an empty List
        # (b) Using []
        L2 = []
        print(L2)
        print(type(L2))
```

```
In [ ]: ▶ # 1.(ii) Creating a non-empty List
        # (a) Using constructor
        L3 = list("VIT")
        L3
```

```
In [ ]: ▶ L4 = list(range(0, 11, 2))
        L4
```

```
In [ ]: ▶ # 1.(ii) Creating a non-empty List
# (b) Using []
Series = ["Money Heist", "Stranger Things", "13 Reasons Why", "911", "D
"Citadel", "The Night Agent", "Silo", "From", "Manifest", "Th
Novels = ["You have reached Sam", "Ugly Love", "It starts with us", "It
"Twisted Games", "Twisted Hate", "Twisted Love", "Verity", "No
Movies = ["Kill", "Kishkindha Kaandam", "Aavesham", "Amaran", "Laapata
"Maharaja", "Mr and Mrs Mahi"]
print(Series)
print(Novels)
print(Movies)
```

```
In [ ]: ▶ # 2.Accessing Elements
# (a) Indexing (Positive)
Series[8]
```

```
In [ ]: ▶ L3[0]
```

```
In [ ]: ▶ Novels[2]
```

```
In [ ]: ▶ # (a) Indexing (Negative)
Series[-6]
```

```
In [ ]: ▶ L3[-2]
```

```
In [ ]: ▶ # 2.Accessing Elements
# (b) Slicing
L3[:]
```

```
In [ ]: ▶ Series[0:]
```

```
In [ ]: ▶ Series[:5]
```

```
In [ ]: ▶ Novels[2:6]
```

```
In [ ]: ▶ Movies[::2]
```

```
In [ ]: ▶ L = [10, 20, 30, 40, 50]
L[::-1]
```

```
In [ ]: ▶ L[-1:0:-1]
```

```
In [ ]: ▶ # 3.Builtin Functions
# (i) Len()
len(Movies)
```

```
In [ ]: ▶ # (ii) max()
max(L)
```

```
In [ ]: ▶ # (iii) min()
min(L)
```

```
In [ ]: ▶ # (iv) sum()
sum(L)
```

```
In [ ]: ▶ # (v) random.shuffle()
import random
random.shuffle(L)
print(L)
```

```
In [ ]: ▶ # 4.Operators
# (i) '+' Concatenation
Movies + Novels
```

```
In [ ]: ▶ # (ii) '*' Repetition
L * 3
```

```
In [ ]: ▶ # (iii) 'in' / 'not in'
print("Money Heist" in Series)
print("911" not in Novels)
```

```
In [ ]: ▶ # (iv) 'is'
x = "Python"
y = "Python"
print(x is y)
```

```
In [ ]: ▶ '''Note:
    If two objects are identical then they are equivalent,
    If two objects are equivalent then it is not necessary
    that they will also be identical'''
A = [1, 2, 3]
B = [1, 2, 3]
print(A is B)
```

```
In [ ]: ▶ # (v) del
print(L)
del L[0]
print(L)
del L[-1]
print(L)
del L[:]
print(L)
```

```
In [ ]: ▶ # 5.List Comprehension
Num = [1, 2, 3, 4, 5]
L = [i * i for i in Num]
print(L)
```

```
In [ ]: ▶ L = [9, 1, 0, 4, 7, 12, 5, 13, 6]
Even_L = [i for i in L if i % 2 == 0]
print(Even_L)
```

```
In [ ]: ▶ L = [9, 1, 0, 4, 7, 12, 5, 13, 6]
Odd_L = [i for i in L if i % 2 != 0]
print(Odd_L)
```

```
In [ ]: ▶ # 6.List Methods
# (i) append() - append an element at the end of the list
print(Novels)
Novels.append("Life is what you make it")
print(Novels)
```

```
In [ ]: ▶ # 6.List Methods
# (ii) extend()
Holly = ["Mission Impossible", "Taken", "John Wick"]
print(Holly)
Movies.extend(Holly)
print(Movies)
```

```
In [ ]: ▶ # 6.List Methods
# (iii) count()
Movies.count('Taken')
```

```
In [ ]: ▶ # 6.List Methods
# (iv) copy()
FilmSeries = []
print(FilmSeries)
FilmSeries=Holly.copy()
print(FilmSeries)
```

```
In [ ]: ▶ # 6.List Methods
# (v) clear()
print(L)
L.clear()
print(L)
```

```
In [ ]: ▶ # 6.List Methods
# (vi) index()
print(Holly.index("Taken"))
```

```
In [ ]: ▶ # 6.List Methods
# (vii) insert()
print(Holly)
Holly.insert(1, "Twilight")
print(Holly)
```

```
In [ ]: ▶ # 6.List Methods
# (viii) pop()
print(Movies)
Movies.pop()
print(Movies)
```

```
In [ ]: ▶ # 6.List Methods
# (ix) remove()
print(Movies)
Movies.remove('Mr and Mrs Mahi')
print(Movies)
```

```
In [ ]: ▶ # 6.List Methods
# (x) reverse()
print(Series)
Series.reverse()
print(Series)
```

```
In [ ]: ▶ # 6.List Methods
# (xi) sort()
print(Novels)
Novels.sort()
print(Novels)
```

```
In [ ]: ▶ # 7.Nested Lists
Nest = [1, [2,3,4], 5, [6,7], 8, [9,10]]
```

```
In [ ]: ► Color = ['Red', 'Blue']
        Color.append('Green')
        print(Color)
        Color.extend('White')
        print(Color)
```

```
In [ ]: ► A = [111, 222, 333, 444, 555]
        B = ['X', 'Y', 'Z']
        Ans = list(zip(A, B))
        print(Ans)
```

```
In [ ]: ► Words = ['Cat', 'Note', 'Bat', 'Pen', 'Book', 'Apple']
        L = [w for w in Words if len(w) == 3]
        print(L)
```

```
In [ ]: ► Courses = ["python", "c", "cpp", "pascal", "basic", "fortran"]
        Courses_U = [c.upper() for c in Courses]
        print(Courses_U)
        Courses_C = [c.capitalize() for c in Courses]
        print(Courses_C)
```

```
In [ ]: ► A = [111, 222, 333]
        B = ['x', 'y', 'z']
        AnsT = tuple(zip(A, B))
        print(AnsT)
```

```
In [ ]: ► Colours = ['RED', 'GREEN', 'BLUE']
        Val = [128, 150, 200]
        LCol = list(zip(Colours, Val))
        for x, y in LCol:
            print(x, y)
```

```
In [ ]: ► Ques = ['Place', 'Food', 'Color', 'Language']
        Ans = ['Pondy', 'Fish', 'Green', 'Python']
        QandA = tuple(zip(Ques, Ans))
        for Q, A in QandA:
            print("What is your favourite", Q, "?")
            print("My Favourite", Q, "is", A)
```

```
In [ ]: ▶ N = int(input("Enter the size of the list: "))
L = []
Temp = []
for i in range(0, N):
    t = int(input("Enter the element:"))
    L.append(t)
print(L)

for ele in L:
    if ele not in Temp:
        Temp.append(ele)
print(Temp)
for e in Temp:
    print(e, "occurs", L.count(e), "times")
```

```
In [ ]: ▶ Matrix_A = []
Matrix_B = []
Matrix_C = []
M1 = int(input("Enter the number of rows of Matrix_A"))
N1 = int(input("Enter the number of columns of Matrix_A"))
M2 = int(input("Enter the number of rows of Matrix_B"))
N2 = int(input("Enter the number of columns of Matrix_B"))
if (M1==M2) and (N1==N2):
    print("Enter the elements of Matrix_A")
    for i in range(0, M1):
        Matrix_A.append([])
        for j in range(0, N1):
            t = int(input("Enter element:"))
            Matrix_A[i].append(t)

    print("Enter the elements of Matrix_B")
    for i in range(0, M2):
        Matrix_B.append([])
        for j in range(0, N2):
            t = int(input("Enter element:"))
            Matrix_B[i].append(t)

    for i in range(0, M1):
        Matrix_C.append([])
        for j in range(0, N1):
            t = Matrix_A[i][j] + Matrix_B[i][j]
            Matrix_C[i].append(t)

    print("Matrix_A:")
    for i in range(0, M1):
        for j in range(0, N1):
            print(Matrix_A[i][j], end=' ')
        print('\n')

    print("Matrix_B:")
    for i in range(0, M2):
        for j in range(0, N2):
            print(Matrix_B[i][j], end=' ')
        print('\n')

    print("Matrix_C:")
    for i in range(0, M1):
        for j in range(0, N1):
            print(Matrix_C[i][j], end=' ')
        print('\n')
else:
    print("Matrix Addition is not possible")
```

```
In [ ]: ▶
```