COLLECTION

SEQUENCE

LIST

* **List** is a collection which is ordered and changeable. Allows duplicate members.

CREATE A LIST [In Python lists are written with square brackets.]

thislist = ["apple", "banana", "cherry"]

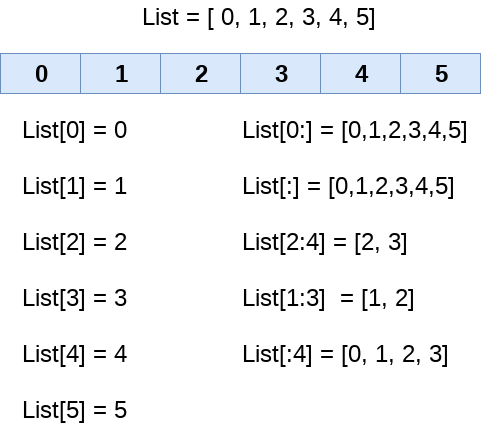
print(thislist)

## **Access Items**

You access the list items by referring to the index number:

thislist = ["apple", "banana", "cherry"]

print(thislist[1])



Python Lists

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[-4:-1])

#Negative indexing means starting from the end of the list.

#This example returns the items from index -4 (included) to index -1 (excluded)

## **Updating List values (LIST IS MUTABLE)**

List = [1, 2, 3, 4, 5, 6]

**print**(List)

List[2] = 10;

**print**(List)

List[1:3] = [89, 78]

**print**(List)

**Output:**

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

[1, 2, 10, 4, 5, 6]

[1, 89, 78, 4, 5, 6]

## **Loop Through a List**

You can loop through the list items by using a for loop:

thislist = ["apple", "banana", "cherry"]

for x in thislist:

print(x)

## **Check if Item Exists**

thislist = ["apple", "banana", "cherry"]

if "apple" in thislist:

print("Yes, 'apple' is in the fruits list")

## **List Length**

thislist = ["apple", "banana", "cherry"]

print(len(thislist))

## **Add Items**

Using the append() method to append an item:

thislist = ["apple", "banana", "cherry"]

thislist.append("orange")

print(thislist)

To add an item at the specified index, use the insert() method:

thislist = ["apple", "banana", "cherry"]

thislist.insert(1, "orange")

print(thislist)

## **Remove Item**

thislist = ["apple", "banana", "cherry"]

thislist.remove("banana")

print(thislist)

The pop() method removes the specified index, (or the last item if index is not specified):

thislist = ["apple", "banana", "cherry"]

thislist.pop()

print(thislist)

The del keyword removes the specified index:

thislist = ["apple", "banana", "cherry"]

del thislist[0]

print(thislist)

The del keyword can also delete the list completely:

thislist = ["apple", "banana", "cherry"]

del thislist

print(thislist) #this will cause an error because you have succsesfully deleted "thislist".

The clear() method empties the list:

thislist = ["apple", "banana", "cherry"]

thislist.clear()

print(thislist)

## **Copy a List**

thislist = ["apple", "banana", "cherry"]

mylist = thislist.copy()

print(mylist)

Another way to make a copy is to use the built-in method list().

thislist = ["apple", "banana", "cherry"]

mylist = list(thislist)

print(mylist)

## **Join Two Lists**

list1 = ["a", "b" , "c"]

list2 = [1, 2, 3]

list3 = list1 + list2

print(list3)

Another way to join two lists are by appending all the items from list2 into list1, one by one:

list1 = ["a", "b" , "c"]

list2 = [1, 2, 3]

for x in list2:

list1.append(x)

print(list1)

Or you can use the extend() method, which purpose is to add elements from one list to another list:

list1 = ["a", "b" , "c"]

list2 = [1, 2, 3]

list1.extend(list2)

print(list1)

# **Python List index() Method**

fruits = ['apple', 'banana', 'cherry']

x = fruits.index("cherry")

print(x)

…………..

fruits = [4, 55, 64, 32, 16, 32]

x = fruits.index(32)

print(x)

# **Python List count() Method**

fruits = ["apple", "banana", "cherry"]

x = fruits.count("cherry")

print(x)

…………….

fruits = [1, 4, 2, 9, 7, 8, 9, 3, 1]

x = fruits.count(9)

print(x)

# **Python List reverse() Method**

fruits = ['apple', 'banana', 'cherry']

fruits.reverse()

print(fruits)

# **Python List sort() Method**

cars = ['Ford', 'BMW', 'Volvo']

cars.sort()

print(cars)

…………..

cars = ['Ford', 'BMW', 'Volvo']

cars.sort(reverse=True)

print(cars)

………………..

1. Which of the following commands will create a list?  
   a) list1 = list()  
   b) list1 = []   
   c) list1 = list([1, 2, 3])  
   d) all of the mentioned
2. 2. What is the output when we execute list(“hello”)?  
   a) [‘h’, ‘e’, ‘l’, ‘l’, ‘o’]   
   b) [‘hello’]   
   c) [‘llo’]   
   d) [‘olleh’]
3. 3. Suppose listExample is [‘h’,’e’,’l’,’l’,’o’], what is len(listExample)?  
   a) 5  
   b) 4  
   c) None  
   d) Error
4. 5. Suppose list1 is [3, 5, 25, 1, 3], what is min(list1)?  
   a) 3  
   b) 5  
   c) 25  
   d) 1
5. 8. Suppose list1 is [4, 2, 2, 4, 5, 2, 1, 0], Which of the following is correct syntax for slicing operation?  
   a) print(list1[0])  
   b) print(list1[:2])  
   c) print(list1[:-2])  
   d) all of the mentioned
6. Suppose list1 is [2, 33, 222, 14, 25], What is list1[:-1]?  
   a) [2, 33, 222, 14]   
   b) Error  
   c) 25  
   d) [25, 14, 222, 33, 2]
7. What will be the output of the following Python code?

names = ['Amir', 'Bear', 'Charlton', 'Daman']

print(names[-1][-1])

1. Suppose list1 is [1, 3, 2], What is list1 \* 2?  
   a) [2, 6, 4]   
   b) [1, 3, 2, 1, 3]   
   c) [1, 3, 2, 1, 3, 2]   
   d) [1, 3, 2, 3, 2, 1]
2. What will be the output of the following Python code?

list1 = [11, 2, 23]

list2 = [11, 2, 2]

list1 < list2 is

1. To add a new element to a list we use which command?  
   a) list1.add(5)  
   b) list1.append(5)  
   c) list1.addLast(5)  
   d) list1.addEnd(5)
2. To insert 5 to the third position in list1, we use which command?  
   a) list1.insert(3, 5)  
   b) list1.insert(2, 5)  
   c) list1.add(3, 5)  
   d) list1.append(3, 5)
3. To remove string “hello” from list1, we use which command?  
   a) list1.remove(“hello”)  
   b) list1.remove(hello)  
   c) list1.removeAll(“hello”)  
   d) list1.removeOne(“hello”)
4. Suppose list1 is [3, 4, 5, 20, 5], what is list1.index(5)?  
   a) 0  
   b) 1  
   c) 4  
   d) 2
5. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1.count(5)?  
   a) 0  
   b) 4  
   c) 1  
   d) 2
6. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after list1.reverse()?  
   a) [3, 4, 5, 20, 5, 25, 1, 3]   
   b) [1, 3, 3, 4, 5, 5, 20, 25]   
   c) [25, 20, 5, 5, 4, 3, 3, 1]   
   d) [3, 1, 25, 5, 20, 5, 4, 3]
7. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.pop(1)?  
   a) [3, 4, 5, 20, 5, 25, 1, 3]   
   b) [1, 3, 3, 4, 5, 5, 20, 25]   
   c) [3, 5, 20, 5, 25, 1, 3]   
   d) [1, 3, 4, 5, 20, 5, 25]
8. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.pop()?  
   a) [3, 4, 5, 20, 5, 25, 1]   
   b) [1, 3, 3, 4, 5, 5, 20, 25]   
   c) [3, 5, 20, 5, 25, 1, 3]   
   d) [1, 3, 4, 5, 20, 5, 25]
9. What will be the output of the following Python code?

a=**"welcome to python"**print(a.split())

1. [“Welcome”, “to”, “Python”]   
   b) (“Welcome”, “to”, “Python”)  
   c) {“Welcome”, “to”, “Python”}  
   d) “Welcome”, “to”, “Python”
2. What will be the output of the following Python code?

myList = [1, 5, 5, 5, 5, 1]  
max = myList[0]  
indexOfMax = 0  
**for** i **in** range(1, len(myList)):  
 **if** myList[i] > max:  
 max = myList[i]  
 indexOfMax = i  
   
print(indexOfMax)

1. 1  
   b) 2  
   c) 3  
   d) 4
2. What will be the output of the following Python code?

**def** f(values):  
 values[0] = 44  
   
v = [1, 2, 3]  
f(v)  
print(v)

1. [1, 44]   
   b) [1, 2, 3, 44]   
   c) [44, 2, 3]   
   d) [1, 2, 3]
2. What will be the output of the following Python code?

names1 = [**'Amir'**, **'Bala'**, **'Chales'**]  
   
**if 'amir' in** names1:  
 print(1)  
**else**:  
 print(2)

1. None  
   b) 1  
   c) 2  
   d) Error
2. What is the output when following code is executed ?

list1 = [1, 3]  
list2 = list1  
list1[0] = 4  
print(list2)

|  |  |
| --- | --- |
| **a.** | [1, 3]. |
| **b.** | [4, 3]. |
| **c.** | [1, 4]. |
| **d.** | [1, 3, 4]. |

## **Python List Operations**

## **Python List Operations**

The concatenation (+) and repetition (\*) operator work in the same way as they were working with the strings.

Lets see how the list responds to various operators.

1. Consider a List l1 = [1, 2, 3, 4], **and** l2 = [5, 6, 7, 8]

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| Repetition | The repetition operator enables the list elements to be repeated multiple times. | L1\*2 = [1, 2, 3, 4, 1, 2, 3, 4] |
| Concatenation | It concatenates the list mentioned on either side of the operator. | l1+l2 = [1, 2, 3, 4, 5, 6, 7, 8] |
| Membership | It returns true if a particular item exists in a particular list otherwise false. | print(2 in l1) prints True. |
| Iteration | The for loop is used to iterate over the list elements. | for i in l1:  print(i)  **Output**  1  2  3  4 |
| Length | It is used to get the length of the list | len(l1) = 4 |

## **Removing elements from the list**

1. List = [0,1,2,3,4]
2. **print**("printing original list: ");
3. **for** i **in** List:
4. **print**(i,end=" ")
5. List.remove(0)
6. **print**("\nprinting the list after the removal of first element...")
7. **for** i **in** List:
8. **print**(i,end=" ")

**Output:**

printing original list:

0 1 2 3 4

printing the list after the removal of first element...

1 2 3 4

## **Python List Built-in functions**

Python provides the following built-in functions which can be used with the lists.

|  |  |  |
| --- | --- | --- |
| **SN** | **Function** | **Description** |
| 1 | cmp(list1, list2) | It compares the elements of both the lists. |
| 2 | len(list) | It is used to calculate the length of the list. |
| 3 | max(list) | It returns the maximum element of the list. |
| 4 | min(list) | It returns the minimum element of the list. |
| 5 | list(seq) | It converts any sequence to the list. |