Python Lists

June 10, 2024

Define List and Print it

```
thislist = ["Python", "Pascal",
"Java"]

print(thislist)
```

What are lists?

- List items are ordered, changeable, and allow duplicate values.

What are lists?

- List items are ordered, changeable, and allow duplicate values.
- List items are indexed, the first item has index [0], the second item has index [1] etc.

Ordered?

- When we say that lists are ordered, it means that the items have a defined order, and that order will not change.

Ordered?

- When we say that lists are ordered, it means that the items have a defined order, and that order will not change.

- ADD NEW ITEM to a list $\longrightarrow$ the new item will be added at the end of the list.

Changeable?

- We can change, add, and remove items in a list after it has been created.

Allow Duplicates

- lists can have items with the same value:

Allow Duplicates

- lists can have items with the same value:
- For Example, this is completely fine as a code snippet.
  myList = ["ABC", "XYZ", "ABC"]

Length of a List

To determine how many items a list has, use the len() function:

```
thislist = ["apple", "banana", "cherry"]
print(len(thislist))
```

Data Type

```
list1 = ["apple", "banana", "cherry"]
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]
```

Mixed Data Type - ALLOWED

list1 = ["apple", "banana", 3]

Mixed Data Type - ALLOWED

list1 = ["apple", "banana", 3]
list2 = ["Yellow", 5, 7, 9, 3]

Mixed Data Type - ALLOWED

list1 = ["apple", "banana", 3]
list2 = ["Yellow", 5, 7, 9, 3]
list3 = [True, False, False, "banana", 'p']

Accessing Individual Element of a list using the Index

```
Bird = ["Peacock", "Pigeon", "Parrot"]

print(Bird[2])
```

Accessing Individual Element of a list using the Index

```
Bird = ["Peacock", "Pigeon", "Parrot"]
```

```
print(Bird[2])
```

What happens when we try to extract an element at index 3?

Negative Indexing

- Negative indexing means start from the end

Negative Indexing

- Negative indexing means start from the end
- -1 refers to the last item, -2 refers to the second last item etc.

Negative Indexing

- Negative indexing means start from the end
- -1 refers to the last item, -2 refers to the second last item etc.
- Mylist=["a", "b", "c"]. What is Mylist[-2]?

Negative Indexing

- Negative indexing means start from the end
- -1 refers to the last item, -2 refers to the second last item etc.
- Mylist=["a", "b", "c"]. What is Mylist[-2]?

Range of Indices

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

print(thislist[2:5])
```

Range of Indices

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

print(thislist[2:5])
```

Prints the elements from index 2 (including 2) to index 5 (excluding 5)

Range of Indices

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

print(thislist[:4])
```

Range of Indices

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

print(thislist[:4])
```

What's the output?

Range of Indices

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

print(thislist[3:])
```

Range of Indices

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

print(thislist[3:])
```

What's the output?

Range of Indices

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

print(thislist[3:])
```

Range of Indices

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

print(thislist[3:])
```

What's the output?

Element in List?

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

if "apple" in thistuple: print("Yes, 'apple' is in the fruits tuple")

Modifying a list

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

thislist[1] = "blackcurrant"

print(thislist)
```

Modifying a list

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

thislist[1] = "blackcurrant"

print(thislist)
```

Output: ['apple', 'blackcurrant', 'cherry']

Modifying a list

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

thislist[1:2] = ["blackcurrant", "watermelon"]

print(thislist)
```

Modifying a list

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

thislist[1:2] = ["blackcurrant", "watermelon"]

print(thislist)
```

Modifying a list

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

thislist[1:2] = ["blackcurrant", "watermelon"]

print(thislist)
```

Output:['apple', 'blackcurrant', 'watermelon', 'cherry']

Modifying a list

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

thislist.insert(2, "watermelon")

print(thislist)
```

Modifying a list

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

thislist.insert(2, "watermelon")

print(thislist)
```

Output:['apple', 'banana', 'watermelon', 'cherry']

Remove List Items

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

thislist.append("orange")

print(thislist)
```

Remove List Items

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

thislist.append("orange")

print(thislist)
```

Output: ['apple','banana','cherry','orange']

Remove List Items

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

thislist.insert(1, "orange")

print(thislist)
```

Remove List Items

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

thislist.insert(1, "orange")

print(thislist)
```

Output:['apple','orange','banana','cherry']

Remove List Items

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

tropical = ["mango", "pineapple", "papaya"]

thislist.extend(tropical)

print(thislist)
```

Remove List Items

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

tropical = ["mango", "pineapple", "papaya"]

thislist.extend(tropical)

print(thislist)
```

Output: ['apple', 'banana', 'cherry', 'mango', 'pineapple', 'papaya']

Remove List Items

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

thislist.remove("banana")

print(thislist)
```

Remove List Items

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

thislist.remove("banana")

print(thislist)
```

Output: ['apple', 'cherry']

Remove List Items

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

thislist.remove("banana")

print(thislist)
```

Remove List Items

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

thislist.remove("banana")

print(thislist)
```

Output: ['apple','cherry','banana','kiwi']

Remove List Items

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

thislist.pop(1)

print(thislist)
```

Remove List Items

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

thislist.pop(1)

print(thislist)
```

Output: ['apple','cherry']

Remove List Items

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

thislist.pop()

print(thislist)
```

Remove List Items

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

thislist.pop()

print(thislist)
```

Output: ['apple', 'banana']

Remove List Items

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

del thislist[0]

print(thislist)
```

Remove List Items

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

del thislist[0]

print(thislist)
```

Output: ['banana', 'cherry']

Remove List Items

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

del thislist
```

Remove List Items

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

thislist.clear()

print(thislist)
```

Remove List Items

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

thislist.clear()

print(thislist)
```

Output: [ ]

Looping through a list

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

for x in thislist:

    print(x)
```

Looping through a list

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

for x in thislist:

    print(x)
```

Output:
apple
banana
cherry

Looping through a list

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

for i in range(len(thislist)):

        print(thislist[i])
```

Looping through a list

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

for i in range(len(thislist)):

        print(thislist[i])
```

Output:
apple
banana
cherry

Looping through a list

```
thislist = ["apple", "banana", "cherry"]
i = 0

while i < len(thislist):
    print(thislist[i])
    i = i + 1
```

Looping through a list

```
thislist = ["apple", "banana", "cherry"]
i = 0

while i < len(thislist):
    print(thislist[i])
    i = i + 1
```

Output:
apple
banana
cherry

A short hand for loop that will print all items in a list

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

[print(x) for x in thislist]
```

A short hand for loop that will print all items in a list

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

[print(x) for x in thislist]
```

Output:
apple
banana
cherry

List Comprehension

Shorter syntax when you want to create a new list based on the values of an existing list.

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [ ]

for x in fruits:

   if "a" in x:

       newlist.append(x)

print(newlist)
```

List Comprehension

Shorter syntax when you want to create a new list based on the values of an existing list.

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [ ]

for x in fruits:

   if "a" in x:

       newlist.append(x)

print(newlist)
```

Output: ['apple', 'banana', 'mango']

List Comprehension

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits if "a" in x]

print(newlist)
```

List Comprehension

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits if "a" in x]

print(newlist)
```

Output: ['apple', 'banana', 'mango']

List Comprehension Syntax

```
newlist = [expression for item in iterable if condition == True]
```

List Comprehension

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits if x != "apple"]

print(newlist)
```

List Comprehension

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits if x != "apple"]

print(newlist)
```

Output: ['banana', 'cherry', 'kiwi', 'mango']

List Comprehension

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits]

print(newlist)
```

List Comprehension

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits]

print(newlist)
```

Output: ['apple', 'banana', 'cherry', 'kiwi', 'mango']

List Comprehension

```
newlist = [x for x in range(10)]

print(newlist)
```

List Comprehension

```
newlist = [x for x in range(10)]

print(newlist)
```

Output: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

List Comprehension

```
newlist = [x for x in range(10) if x < 5]

print(newlist)
```

List Comprehension

```
newlist = [x for x in range(10) if x < 5]

print(newlist)
```

Output: [0, 1, 2, 3, 4]

List Comprehension

```
fruits = ["apple", "kiwi"]

newlist = [x.upper() for x in fruits]

print(newlist)
```

List Comprehension

```
fruits = ["apple", "kiwi"]

newlist = [x.upper() for x in fruits]

print(newlist)
```

Output: [APPLE, KIWI]

List Comprehension

```
fruits = ["apple", "kiwi"]

newlist = ["hello" for x in fruits]

print(newlist)
```

List Comprehension

```
fruits = ["apple", "kiwi"]

newlist = ["hello" for x in fruits]

print(newlist)
```

Output: ['hello', 'hello']

List Comprehension

```
fruits=["banana", "orange", "pineapple", "orange"]

newlist = [x if x != "banana" else "orange" for x in fruits]

print(newlist)

#Return the item if not equal to banana else return orange
```

List Comprehension

```
fruits=["banana", "orange", "pineapple", "orange"]

newlist = [x if x != "banana" else "orange" for x in fruits]

print(newlist)

#Return the item if not equal to banana else return orange
```

Output: ['orange', 'orange', 'pineapple', 'orange']

copying a list

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

mylist = thislist.copy()

print(mylist)
```

copying a list

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

mylist = thislist.copy()

print(mylist)
```

Output: ['apple', 'banana', 'cherry']

copying a list

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

mylist = list(thislist)

print(mylist)
```

copying a list

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

mylist = list(thislist)

print(mylist)
```

Output: ['apple', 'banana', 'cherry']

Joining two lists

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

list2 = [1, 2, 3]

list3 = list1 + list2

print(list3)
```

Joining two lists

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

list2 = [1, 2, 3]

list3 = list1 + list2

print(list3)
```

Output: ['a', 'b', 'c', 1, 2, 3]

Joining two lists

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

list2 = [1, 2, 3]

for x in list2:

        list1.append(x)

print(list1)
```

Joining two lists

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

list2 = [1, 2, 3]

for x in list2:

        list1.append(x)

print(list1)
```

Output:['a', 'b', 'c', 1, 2, 3]

Joining two lists

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

list2 = [1, 2, 3]

list1.extend(list2)

print(list1)
```

Joining two lists

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

list2 = [1, 2, 3]

list1.extend(list2)

print(list1)
```

Output: ['a', 'b', 'c', 1, 2, 3]

Sort Method

```
thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]

thislist.sort()

print(thislist)
```

Sort Method

thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]

thislist.sort()

print(thislist)

Output: ['banana', 'kiwi', 'mango', 'orange', 'pineapple']

Sort Method

```
thislist = [100, 50, 65, 82, 23]

thislist.sort()

print(thislist)
```

Sort Method

thislist = [100, 50, 65, 82, 23]

thislist.sort()

print(thislist)

Output: [23, 50, 65, 82, 100]

Sort Method

```
thislist = [100, 50, 65, 82, 23]

thislist.sort(reverse = True)

print(thislist)
```

Sort Method

thislist = [100, 50, 65, 82, 23]

thislist.sort(reverse = True)

print(thislist)

Output:[100, 82, 65, 50, 23]

Sort Method

Sort the list based on how close the number is to 50:

Sort Method

Sort the list based on how close the number is to 50:

```
def myfunc(n):

    return abs(n - 50)

thislist = [100, 50, 65, 82, 23]

thislist.sort(key = myfunc)

print(thislist)
```

Sort Method

Sort the list based on how close the number is to 50:

```
def myfunc(n):

    return abs(n - 50)

thislist = [100, 50, 65, 82, 23]

thislist.sort(key = myfunc)

print(thislist)
```

Sorting

```
thislist = ["banana", "Orange", "Kiwi"]

thislist.sort()

print(thislist)
```

Sorting

```
thislist = ["banana", "Orange", "Kiwi"]

thislist.sort()

print(thislist)
```

Guess the output?

Sorting

```
thislist = ["banana", "Orange", "Kiwi"]

thislist.sort()

print(thislist)
```

Guess the output?
Expected Output: [banana, Kiwi, Orange]

Sorting

```
thislist = ["banana", "Orange", "Kiwi"]

thislist.sort()

print(thislist)
```

Guess the output?
Expected Output: [banana, Kiwi, Orange]

Output: ['Kiwi', 'Orange', 'banana']

Case-sensitive Sorting

```
thislist = ["banana", "Orange", "Kiwi"]

thislist.sort(key = str.lower)

print(thislist)
```

Case-sensitive Sorting

```
thislist = ["banana", "Orange", "Kiwi"]

thislist.sort(key = str.lower)

print(thislist)
```

Output: ['banana', 'Kiwi', 'Orange']

Reverse Sorting

```
thislist = ["banana", "Orange", "cherry"]

thislist.reverse()

print(thislist)
```

Reverse Sorting

```
thislist = ["banana", "Orange", "cherry"]

thislist.reverse()

print(thislist)
```

Output: ['cherry', 'Orange', 'banana']

Python program to interchange first and last elements in a list

Python program to interchange first and last elements in a list

```
def swapList(newList):
    size = len(newList)


    temp = newList[0]
    newList[0] =     newList[size - 1]
    newList[size - 1] = temp
    return newList


#Main Code Begins Here
newList = [12, 35, 9, 56, 24]
print(swapList(newList))
```

Python program to print odd numbers in a List

Python program to print odd numbers in a List

```
list1 = [10, 21, 4, 45, 66, 93]

for num in list1:

    if num % 2 != 0:
       print(num, end=" ")
```

Python Program to find out the second largest element in the list

Python Program to find out the second largest element in the list

lst = []

Python Program to find out the second largest element in the list

```
lst = []
for i in range(int(input("Enter the number of elements to enter in the list: "))):
```

Python Program to find out the second largest element in the list

```
lst = []
for i in range(int(input("Enter the number of elements to enter in the list: "))):
    x = int(input("Enter the element: "))
    lst.append(x)
```

Python Program to find out the second largest element in the list

```
lst = []
for i in range(int(input("Enter the number of elements to enter in the list: "))):
    x = int(input("Enter the element: "))
    lst.append(x)

def method1(lst):

    lst.sort()
    print("The second largest element of the list is: ", lst[-2])
```

Python Program to find out the second largest element in the list

```
lst = []
for i in range(int(input("Enter the number of elements to enter in the list: "))):
  x = int(input("Enter the element: "))
  lst.append(x)

def method1(lst):

    lst.sort()
    print("The second largest element of the list is: ", lst[-2])

method1(lst)
```

Python program to find the length of the list without using the len function.

Python program to find the length of the list without using the len function.

```
my_list = [32, 4445, 67, "demo", 78]

count = 0

while(my_list):

    my_list.pop()

    count += 1

print("The length of the list is : ", count)
```

Python program to replace a given fixed character with another fixed character in a list (containing english language alphabets)

Python program to replace a given fixed character with another fixed character in a list (containing english language alphabets)

```
test_list = ["G", "F", "G", "I", "S", "B", "E", "S",
"T"]

print(test_list)

repl_chr = "$"

ret_chr = "G"

res = [ele if ele != ret_chr else repl_chr for ele in
test_list]

print(res)
```