Lists

What is List in Python?

List is one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.

Eg:

fruits = [“apple”, “orange”, “lemon”]

List Items

- 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.
- When we say that lists are ordered, it means that the items have a defined order, and that order will not change. If you add new items to a list, the new items will be placed at the end of the list.
- The list is changeable(mutable), meaning that we can change, add, and remove items in a list after it has been created.
- Since lists are indexed, lists can have items with the same value

List Items - Data Types

- List items can be of any data type : [String, int, or Boolean]

  list_1 = [“apple”, “banana”, “orange”]

  list_2 = [1, 2, 3, 4, 5]

  list_3 = [True, False, False, True]

  or

  list_4 = [“apple”, 1, 2, False, True, “orange”]

Access List Items

- Indexing

  List items are indexed and you can access them by referring to the index number. The first item has index 0.

- Negative Indexing

  Negative indexing means start from the end.

List slicing

- Same as string slicing :)
- Guess the output of the following code :

list_ = [1, 2, 3, 4, 5, “hello”, “python is easy”]

print(list_[2:])

print(list_[-5:-1])

print(list_[-1:-5])

Change List Items

- Changing item values with index

  Eg : list_1 = [5, 10, 15, 20, 25]

  list_1[3] = 90

  updated list : [5, 10, 15, 90, 15]

- With the insert() method : add item without replacing the old value with the new one.

  list_1.insert(#index, #value)

  - Try #insert() method yourself

Add List Items

- To add an item to the end of the list, use the append() method

```
my_list = [1, 3, 5, 7]
my_list.append(9)
print(my_list) #output : >> [1, 3, 5, 7, 9]
```

- You can also use insert() method to add items in the list.

Remove List Items

- Remove Specified Item with remove() method

```
my_list = [“Alice”, “Bob”, “Harry”]
my_list.remove(“Bob”)
print(my_list) #Output : >> [“Alice”, “Harry”]
```

- Remove Specified Index with pop() method

```
my_list = [“Alice”, “Bob”, “Harry”]
my_list.pop(1)
print(my_list) #Output : >> [“Alice”, “Harry”]
```

- Note: If you do not specify the index, the pop() method removes the last item.

Python - Loop Lists

- Loop Through a List using for loop:

```
my_list = [1, 2, 3, 4, 5]
for i in my_list:
    print(i)
```

or

```
for i in range(len(my_list)):
    print(my_list[i])
#Try this with while loop
```

Sorting a list

- List objects have a sort() method that will sort the list alphanumerically, ascending, by default:

```
my_list = [1, 5, 2, 9, 0, 11]

print(my_list.sort())

#Output : >> [0, 1, 2, 5, 9, 11]


#Try this with list of strings

#Try reverse() method and observe the output
```

Copying Lists

- Using copy() method

```
my_list = [1, 2, 3, 4, 5]
my_new_list = my_list.copy()
print(my_new_list) #Output : >> [1, 2, 3, 4, 5]
```

- Using list() method and Using list slicing

```
my_new_list = list(my_list) #1
my_new_list = list[:] #2
```

Both #1 and #2 will copy my_list into my_new_list

Joining Lists

- With + operator

```
list_1 = [1, 2, 3, 4, 5]
list_2 = [10, 20, 30, 40, 50]
list_3 = list_1 + list_2
```

- With extend() method

```
list_1.extend(list_2)
print(list_1) #Output : >> [1, 2, 3, 4, 5, 10,..., 50]
# Try using append() method and observe the output
```

List Comprehension

- List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list.

  li_1 = [1, 2, 3, 4, 5]

  li_2 = [i for i in li_1] #Output : >> [1, 2, 3, 4, 5]

- Understood? No? I know :)

  Let’s do some examples....

List methods

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value

List methods

Method	Description
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

References

https://www.w3schools.com/python/python_lists.asp

https://www.tutorialspoint.com/python/python_lists.htm

Thank You