Python – Lists and its Operations

- A list is a collection which is ordered and changeable. In Python lists are written with square brackets.

Example

Create a List:

```
thislist = ["apple", "banana", "cherry", “Mango”]
print(thislist)
```

```
['apple', 'banana', 'cherry', 'Mango']
```

Access Items

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

Example

```
thislist = ["apple", "banana", "cherry"]
print(thislist[0])
```

```
apple
```

Negative Indexing

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

Example

Print the last item of the list:

```
thislist = ["apple", "banana", "cherry"]
print(thislist[-1])
```

```
cherry
```

Range of Indexes

You can specify a range of indexes by specifying where to start and where to end the range.

When specifying a range, the return value will be a new list with the specified items.

Example

```
student_list = ["Devi Kalyani", "Ram", "Selin", "Karthik", "Mahesh", "Lakshmi", "Gopi"]

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

```
['Selin', 'Karthik', 'Mahesh']
```

Note: The search will start at index 2 (included) and end at index 5 (not included).

Remember that the first item has index 0.

Example

By leaving out the start value, the range will start at the first item:

```
student_list = ["Devi Kalyani", "Ram", "Selin", "Karthik", "Mahesh", "Lakshmi", "Gopi"]
print(student_list[:4])
```

```
['Devi Kalyani', 'Ram', 'Selin', 'Karthik']
```

Example

By leaving out the end value, the range will go on to the end of the list:

This example returns the items from "cherry" and to the end:

```
student_list = ["Devi Kalyani", "Ram", "Selin", "Karthik", "Mahesh", "Lakshmi", "Gopi"]

print(student_list[3:])
```

```
['Karthik', 'Mahesh', 'Lakshmi', 'Gopi']
```

Range of Negative Indexes

Specify negative indexes if you want to start the search from the end of the list:

Example

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

```
thislist =
["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[-4:-1])
```

```
['orange', 'kiwi', 'melon']
```

Change Item Value

To change the value of a specific item, refer to the index number:

Example

```
student_list = ["Devi Kalyani", "Ram", "Selin", "Karthik", "Mahesh", "Lakshmi", "Gopi"]

print(student_list)

student_list[1] = "Sachin"

print(student_list)
```

```
['Devi Kalyani', 'Ram', 'Selin', 'Karthik', 'Mahesh', 'Lakshmi', 'Gopi']
['Devi Kalyani', 'Sachin', 'Selin', 'Karthik', 'Mahesh', 'Lakshmi', 'Gopi']
```

Loop Through a List

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

Example

Print all students in the list, one by one:

```
student_list = ["Devi Kalyani", "Ram", "Selin", "Karthik", "Mahesh", "Lakshmi", "Gopi"]
for x in student_list:
  print(x)
```

```
Devi Kalyani
Ram
Selin
Karthik
Mahesh
Lakshmi
Gopi
```

Check if Item Exists

To determine if a specified item is present in a list use the in keyword:

Example

Check if "Ram" is present in the list:

```
student_list = ["Kalam", "Devi Kalyani", "Ram", "Selin", "Karthik", "Mahesh", "Lakshmi", "Gopi", "Haneef"]

if "Ram" in student_list:
  print("Yes, 'Ram' is in the students list")
else:
  print("No, 'Ram' is not in the students list")
```

```
Yes, 'Ram' is in the students list
```

To determine if a specified item is not present in a list, we can use the not in keyword:

```
student_list = ["Kalam", "Devi Kalyani", "Ram", "Selin", "Karthik", "Mahesh", "Lakshmi", "Gopi", "Haneef"]

if "Kalam" not in student_list:
  print("No, 'Kalam' is not in the students list")
else:
  print("Yes, 'Kalam' is in the students list")
```

```
Yes, 'Kalam' is in the students list
```

List Length

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

Example

Print the number of items in the list:

```
subject_list = ["Chemistry", "Mathematics", "Physics", "Biology", "Python", "Data Structures"]

print(subject_list)

print(len(subject_list))
```

```
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']
6
```

Add Items

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

Example

Using the append() method to append an item:

```
subject_list = ["Chemistry", "Mathematics", "Physics", "Biology", "Python", "Data Structures"]
print(subject_list)

subject_list.append("Software Engineering")
print(subject_list)
```

```
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures', 'Software Engineering']
```

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

Example

Insert an item as the second position:

```
subject_list = ["Chemistry", "Mathematics", "Physics", "Biology", "Python", "Data Structures"]
print(subject_list)
subject_list.insert(1, "History")
print(subject_list)
```

```
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']
['Chemistry', 'History', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']
```

Remove Item

There are several methods to remove items from a list:

Example

The remove() method removes the specified item:

```
subject_list = ["Chemistry", "Mathematics", "Physics", "Biology", "Python", "Data Structures"]
print(subject_list)
subject_list.remove("Physics")
print(subject_list)
```

```
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']
['Chemistry', 'Mathematics', 'Biology', 'Python', 'Data Structures']
```

Example

```
subject_list = ["Chemistry", "Mathematics", "Physics", "Biology", "Python", "Data Structures"]
print(subject_list)
subject_list.remove("Physics")
print(subject_list)
print("\n\n")
subject_list.remove(1) *****
print(subject_list)
```

```
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']
['Chemistry', 'Mathematics', 'Biology', 'Python', 'Data Structures']



Traceback (most recent call last):
  File "prgm16-lists.py", line 92, in <module>
    subject_list.remove(1)
ValueError: list.remove(x): x not in list
```

Example

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

```
subject_list = ["Chemistry", "Mathematics", "Physics", "Biology", "Python", "Data Structures"]
print(subject_list)
subject_list.pop()
print(subject_list)
```

```
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python']
```

Example

```
subject_list = ["Chemistry", "Mathematics", "Physics", "Biology", "Python", "Data Structures"]
print(subject_list)
print("\n")
subject_list.pop(0)
print(subject_list)
print("\n")
subject_list.pop()
print(subject_list)
print("\n")
subject_list.pop(-1)
print(subject_list)
```

```
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']

['Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']

['Mathematics', 'Physics', 'Biology', 'Python']

['Mathematics', 'Physics', 'Biology']
```

Example

The del keyword removes the specified index:

```
subject_list = ["Chemistry", "Mathematics", "Physics", "Biology", "Python", "Data Structures"]
print(subject_list)
del subject_list[2]
print(subject_list)
```

```
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']
['Chemistry', 'Mathematics', 'Biology', 'Python', 'Data Structures']
```

Example

The del keyword can also delete the list completely:

```
subject_list = ["Chemistry", "Mathematics", "Physics", "Biology", "Python", "Data Structures"]
print(subject_list)
del subject_list
print(subject_list)
```

```
['Chemistry', 'Mathematics', 'Physics', 'Biology', 'Python', 'Data Structures']



Traceback (most recent call last):
  File "prgm16-lists.py", line 125, in <module>
    print(subject_list)
NameError: name 'subject_list' is not defined
```