

# Exercises chapter 3: Introducing Lists

Saul SL

June 2023

## 1 Exercise 3-1 Names

Store the names of a few of your friends in a list called `names`. Print each person's name by accessing each element in the list, one at a time.

```
1 names = ['tim', 'tom', 'sam']
2 for i in names:
3     print(i.title())
```

## 2 Exercise 3-2 Greetings

Start with the list you used in Exercise 3-1, but instead of just printing each person's name, print a message to them. The text of each message should be the same, but each message should be personalized with the person's name.

```
1 names = ['tim', 'tom', 'sam']
2 for name in names:
3     print(f"Hello {name.title()}, welcome back!")
```

## 3 Exercise 3-3 Your Own List

Think of your favorite mode of transportation, such as a motorcycle or a car, and make a list that stores several examples. Use your list to print a series of statements about these items, such as "I would like to own a Honda motorcycle."

```
1 transportation_modes = ['bicycle', 'train', 'bus']
2 print(f"I rarely move by {transportation_modes[0]}")
3 print(f"I used to ride the {transportation_modes[1]} a lot")
4 print(f"Nowadays I mainly use the {transportation_modes[2]}")
```

## 4 Exercise 3-4 Guest list

If you could invite anyone, living or deceased, to dinner, who would you invite? Make a list that includes at least three people you'd like to invite to dinner. Then use your list to print a message to each person, inviting them to dinner.

```
1 guests = ['linus', 'tim', 'tom']
2 for guest in guests:
3     print(f"Dear {guest.title()}, I would like to invite you to have dinner")
```

## 5 Exercise 3-5 Changing Guest List

You just heard that one of your guests can't make the dinner, so you need to send out a new set of invitations. You'll have to think of someone else to invite.

- Start with your program from Exercise 3-4. Add a `print()` call at the end of your program stating the name of the guest who can't make it.
- Modify your list, replacing the name of the guest who can't make it with the name of the new person you are inviting.
- Print a second set of invitation messages, one for each person who is still in your list.

```
1 guests = ['linus', 'tim', 'tom']
2 for guest in guests:
3     print(f"Dear {guest.title()}, I would like to invite you to have dinner")
4
5 print(f"Unfortunately, {guests[0].title()} won't come for dinner")
6 # guests.replace('linus', 'steve') # Invalid
7 guests.remove('linus')
8 guests.append('steve')
9 print(f"Dear {guests[len(guests)-1].title()},\
10 I would like to invite you to have dinner")
```

## 6 Exercise 3-6 More Guests

You just found a bigger dinner table, so now more space is available. Think of three more guests to invite to dinner.

- Start with your program from Exercise 3-4 or Exercise 3-5. Add a `print()` call to the end of your program informing people that you found a bigger dinner table.
- Use `insert()` to add one new guest to the beginning of your list.
- Use `insert()` to add one new guest to the middle of your list.
- Use `append()` to add one new guest to the end of your list.
- Print a new set of invitation messages, one for each person in your list.

```
1 guests = ['linus', 'tim', 'tom']
2 for guest in guests:
3     print(f"Dear {guest.title()}, I would like to invite you to have dinner")
4
5 print("We found a bigger place to host the dinner")
6 guests.insert(0, "tina")
7 guests.insert(2, "anna")
8 guests.insert(len(guests), "maya")
9 for guest in guests:
10     print(f"Dear {guest.title()}, I would like to invite you to have dinner")
```

## 7 Exercise 3-7 Shrinking Guest List

You just found out that your new dinner table won't arrive in time for the dinner, and you have space for only two guests.

- Start with your program from Exercise 3-6. Add a new line that prints a message saying that you can invite only two people for dinner.

- Use `pop()` to remove guests from your list one at a time until only two names remain in your list. Each time you pop a name from your list, print a message to that person letting them know you're sorry you can't invite them to dinner.
- Print a message to each of the two people still on your list, letting them know they're still invited.
- Use `del` to remove the last two names from your list, so you have an empty list. Print your list to make sure you actually have an empty list at the end of your program.

```

1  guests = ['linus', 'tim', 'tom']
2  for guest in guests:
3      print(f"Dear {guest.title()}, I would like to invite you to have dinner")
4
5  print("We found a bigger place to host the dinner")
6  guests.insert(0, "tina")
7  guests.insert(2, "anna")
8  guests.insert(len(guests), "maya")
9  for guest in guests:
10     print(f"Dear {guest.title()}, I would like to invite you to have dinner")
11
12 print("Change of plans, now only two people can come to dinner")
13 while len(guests) > 2:
14     out_guest = guests.pop()
15     print(f"Sorry we can't have you, {out_guest.title()}")
16
17 for guest in guests:
18     print(f"Dear {guest.title()}, please come to dinner")
19
20 # guest_n = len(guests)
21 while guests:
22     del guests[0]
23
24 print(guests)

```

## 8 Exercise 3-8 Seeing the World

Think of at least five places in the world you'd like to visit.

- Store the locations in a list. Make sure the list is not in alphabetical order.
- Print your list in its original order. Don't worry about printing the list neatly, just print it as a raw Python list.
- Use `sorted()` to print your list in alphabetical order without modifying the actual list.
- Show that your list is still in its original order by printing it.
- Use `sorted()` to print your list in reverse alphabetical order without changing the order of the original list.
- Show that your list is still in its original order by printing it again.
- Use `reverse()` to change the order of your list. Print the list to show that its order has changed.
- Use `reverse()` to change the order of your list again. Print the list to show it's back to its original order.
- Use `sort()` to change your list so it's stored in alphabetical order. Print the list to show that its order has been changed.
- Use `sort()` to change your list so it's stored in reverse alphabetical order. Print the list to show that its order has changed.

```

1  itravel = ['thailand', 'colombia', 'chile', 'tanzania', 'madagascar']
2  for place in itravel:
3      print(f"- {place.title()}")
4
5  # sorted without modification
6  for place in sorted(itravel):
7      print(f"- {place.title()}")
8
9  for place in itravel:
10     print(f"- {place.title()}")
11
12 # reverse sorted without modification
13 for place in sorted(itravel, reverse=True):
14     print(f"- {place.title()}")
15
16 for place in itravel:
17     print(f"- {place.title()}")
18
19 # Reverse list
20 itravel.reverse()
21 for place in itravel:
22     print(f"- {place.title()}")
23
24 # Re-reverse list
25 itravel.reverse()
26 for place in itravel:
27     print(f"- {place.title()}")
28
29 # Sort list
30 itravel.sort()
31 for place in itravel:
32     print(f"- {place.title()}")
33
34 # Sort in reverse order
35 itravel.sort(reverse=True)
36 for place in itravel:
37     print(f"- {place.title()}")

```

## 9 Exercise 3-9 Dinner Guests

Working with one of the programs from Exercises 3-4 through 3-7 (page 42), use `len()` to print a message indicating the number of people you are inviting to dinner.

```

1  guests = ['linus', 'tim', 'tom']
2  for guest in guests:
3      print(f"Dear {guest.title()}, I would like to invite you to have dinner")
4
5  print("We found a bigger place to host the dinner")
6  guests.insert(0, "tina")
7  guests.insert(2, "anna")
8  guests.insert(len(guests), "maya")
9
10 print(f"There will be {len(guests)} people at the dinner")

```

## 10 Exercise 3-10 Every Function

Think of something you could store in a list. For example, you could make a list of mountains, rivers, countries, cities, languages, or any- thing else you'd like. Write a program that creates a list containing these items and then uses each function introduced in this chapter at least once.

```

1 languages = ['spanish', 'chinese', 'english', 'korean', 'swahili']
2
3 # length of a list
4 print(f"The list has {len(languages)} elements")
5 print(f"The first item is: {languages[0].title()}")
6 languages[0] = 'quechua'
7 print(f"Now the first element is: {languages[0].title()}")
8 languages.append('aymara')
9 languages.insert(0, 'guarani')
10 print(f"Now the first element is: {languages[0].title()}")
11 print(f"And the last is: {languages[-1].title()}")
12
13 # Removing elements
14 print(languages)
15 print("I will now remove the 4th, the 1st and last elements")
16 del languages[3]
17 languages.pop()
18 languages.pop(0)
19 print(languages)
20
21 # sort
22 print(sorted(languages))
23 languages.sort(reverse=True)
24 print(languages)

```