**Simple Exercises on Lists**

* [1-1: Names](https://ehmatthes.github.io/pcc/solutions/chapter_3.html#3-1-names)
* [1-2: Greetings](https://ehmatthes.github.io/pcc/solutions/chapter_3.html#3-2-greetings)
* [1-4: Guest List](https://ehmatthes.github.io/pcc/solutions/chapter_3.html#3-4-guest-list)
* [1-5: Changing Guest List](https://ehmatthes.github.io/pcc/solutions/chapter_3.html#3-5-changing-guest-list)
* [1-6: More Guests](https://ehmatthes.github.io/pcc/solutions/chapter_3.html#3-6-more-guests)
* [1-7: Shrinking Guest List](https://ehmatthes.github.io/pcc/solutions/chapter_3.html#3-7-shrinking-guest-list)
* [1-8: Seeing the World](https://ehmatthes.github.io/pcc/solutions/chapter_3.html#3-8-seeing-the-world)

1-1: 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.

Output:

ron

tyler

dani

1-2: Greetings

Start with the list you used in Exercise 1-1, 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.

Output:

Hello, Ron!

Hello, Tyler!

Hello, Dani!

1-4: 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.

Output:

Guido Van Rossum, please come to dinner.

Jack Turner, please come to dinner.

Lynn Hill, please come to dinner.

1-5: 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 1-4. Add a print statement 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.

Output:

Guido Van Rossum, please come to dinner.

Jack Turner, please come to dinner.

Lynn Hill, please come to dinner.

Sorry, Jack Turner can't make it to dinner.

Guido Van Rossum, please come to dinner.

Gary Snyder, please come to dinner.

Lynn Hill, please come to dinner.

1-6: 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 statement 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.

Output:

Guido Van Rossum, please come to dinner.

Jack Turner, please come to dinner.

Lynn Hill, please come to dinner.

Sorry, Jack Turner can't make it to dinner.

Guido Van Rossum, please come to dinner.

Gary Snyder, please come to dinner.

Lynn Hill, please come to dinner.

We got a bigger table!

Frida Kahlo, please come to dinner.

Guido Van Rossum, please come to dinner.

Reinhold Messner, please come to dinner.

Gary Snyder, please come to dinner.

Lynn Hill, please come to dinner.

Elizabeth Peratrovich, please come to dinner.

1-7: 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 1-6. 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.

Output:

Guido Van Rossum, please come to dinner.

Jack Turner, please come to dinner.

Lynn Hill, please come to dinner.

Sorry, Jack Turner can't make it to dinner.

Guido Van Rossum, please come to dinner.

Gary Snyder, please come to dinner.

Lynn Hill, please come to dinner.

We got a bigger table!

Frida Kahlo, please come to dinner.

Guido Van Rossum, please come to dinner.

Reinhold Messner, please come to dinner.

Gary Snyder, please come to dinner.

Lynn Hill, please come to dinner.

Elizabeth Peratrovich, please come to dinner.

Sorry, we can only invite two people to dinner.

Sorry, Elizabeth Peratrovich there's no room at the table.

Sorry, Lynn Hill there's no room at the table.

Sorry, Gary Snyder there's no room at the table.

Sorry, Reinhold Messner there's no room at the table.

Frida Kahlo, please come to dinner.

Guido Van Rossum, please come to dinner.

1-8: 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.

Output:

Original order:

['himalaya', 'andes', 'tierra del fuego', 'labrador', 'guam']

Alphabetical:

['andes', 'guam', 'himalaya', 'labrador', 'tierra del fuego']

Original order:

['himalaya', 'andes', 'tierra del fuego', 'labrador', 'guam']

Reverse alphabetical:

['tierra del fuego', 'labrador', 'himalaya', 'guam', 'andes']

Original order:

['himalaya', 'andes', 'tierra del fuego', 'labrador', 'guam']

Reversed:

['guam', 'labrador', 'tierra del fuego', 'andes', 'himalaya']

Original order:

['himalaya', 'andes', 'tierra del fuego', 'labrador', 'guam']

Alphabetical

['andes', 'guam', 'himalaya', 'labrador', 'tierra del fuego']

Reverse alphabetical

['tierra del fuego', 'labrador', 'himalaya', 'guam', 'andes']