**Exercise 2: Introducing Lists**

1. Store the names of a few of your favorite music artists in a list called names. Print each name by accessing each element in the list, one at a time.

names = ["George Michael", "Fleetwood Mac", "Billy Joel", "David Bowie"]

print(names[0])

print(names[1])

print(names[2])

print(names[3])

1. Start with the “names” list you used in the first exercise, but instead of just printing each name, print a message to them. The text of each message should be the same, but each message should be personalized with the music artist’s name.

names = ["George Michael", "Fleetwood Mac", "Billy Joel", "David Bowie"]

print(f"I really like {names[0]}!")

print(f"I really like {names[1]}!")

print(f"I really like {names[2]}!")

print(f"I really like {names[3]}!")

1. Think of your favorite book or movie. Make a list that stores several examples. Use your list to print a statement about your favorite book or movie. For example, “My favorite book is Pride and Prejudice”.

books = ["Pride and Prejudice", " The Secret Life of Bees", "Circe", "The Chronicle of Narnia"]

print(f"One of my favorite books is {books[0]}.")

print(f"One of my favorite books is {books[1]}.")

print(f"One of my favorite books is {books[2]}.")

print(f"One of my favorite books is {books[3]}.")

* **Stretch and Challenge:** You can go one step further, and add an author or director name! Try printing a message to the author or director on what you liked about their book or movie!

books = ["Pride and Prejudice", " The Secret Life of Bees", "Circe", "The Chronicle of Narnia"]

authors = ["Jane Austen", "Sue Monk Kidd", "Madeline Miller", "C.S. Lewis"]

print(f"One of my favorite books is {books[0]}. It was written by {authors[0]}, I wish I could thank her for writing this!")

print(f"One of my favorite books is {books[1]}. It was written by {authors[1]}, who has written a lot of other good books!")

print(f"One of my favorite books is {books[2]}. It is written by {authors[2]}, I have never read any other books by them though.")

print(f"One of my favorite books is {books[3]}. It is written by {authors[3]}, he got his inspiration from a walk in Oxford!")

The Dinner Problem -

1. 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.

invites = ["Taylor Swift", "Stevie Nicks", "Dolly Parton"]

print(f"{invites[0]}, would you like to come to my dinner party?")

print(f"{invites[1]}, would you like to come to my dinner party?")

print(f"{invites[2]}, would you like to come to my dinner party?")

1. 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 question 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.

invites = ["Taylor Swift", "Stevie Nicks", "Dolly Parton"]

print(f"{invites[0]} can't come to dinner!")

invites[0] = "David Bowie"

print(f"{invites[0]}, would you like to come to my dinner party?")

print(f"{invites[1]}, would you like to come to my dinner party?")

print(f"{invites[2]}, would you like to come to my dinner party?")

1. 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 question 4 or question 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.

invites = ["Taylor Swift", "Stevie Nicks", "Dolly Parton"]

print(f"{invites[0]} can't come to dinner!")

invites[0] = "David Bowie"

print(f"{invites[0]}, would you like to come to my dinner party?")

print(f"{invites[1]}, would you like to come to my dinner party?")

print(f"{invites[2]}, would you like to come to my dinner party?")

print("I found a bigger table! I will be inviting more people")

invites.insert(0, "Mozart")

invites.insert(2, "Billy Joel")

invites.append("Harry Styles")

print(f"{invites[0]}, could you come to the dinner as well?")

print(f"{invites[2]}, could you come to the dinner as well?")

print(f"{invites[5]}, could you come to the dinner as well?")

1. 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 question 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. Working with one of the programs from Exercises 3-4 through 3-7, use len() to print a message indicating the number of people you are inviting to dinner
2. 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.

**Stretch and Challenge**

* Think of something you could store in a list. For example, you could make a list of mountains, rivers, countries, cities, languages, or anything 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.