

Homework Problems

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

```
In [1]: names = ['Mack', 'Dug', 'Chester', 'Maeve', 'Remi']
```

```
# manual example:  
print(names[0])  
print(names[1])  
print(names[2])  
print(names[3])  
print(names[4])  
print("\n")
```

```
# for loop example  
for name in names:  
    print(name)
```

```
Mack  
Dug  
Chester  
Maeve  
Remi
```

```
Mack  
Dug  
Chester  
Maeve  
Remi
```

3-2. Greetings: Start with the list you used above, 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

```
In [2]: # manual example
print(f"Hello {names[0]}!")
print(f"Hello {names[1]}!")
print(f"Hello {names[2]}!")
print(f"Hello {names[3]}!")
print(f"Hello {names[4]}!")
print("\n")

# for loop example
for name in names:
    print(f"Hello {name}!")
```

Hello Mack!
Hello Dug!
Hello Chester!
Hello Maeve!
Hello Remi!

Hello Mack!
Hello Dug!
Hello Chester!
Hello Maeve!
Hello Remi!

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"

```
In [3]: modes_of_transportation = ['car', 'bike', 'scooter']

print(f"I would like to own a {modes_of_transportation[0]}")
print(f"I would like to ride a {modes_of_transportation[1]}")
print(f"I would like to buy a {modes_of_transportation[2]}")
```

I would like to own a car
I would like to ride a bike
I would like to buy a scooter

3-4. Guest List: If you could invite any famous Naval Officer, living or deceased, to a dinner, who would you invite? Make a list that includes at least three officers you'd like to invite to dinner. Then use your list to print a message to each person, inviting them to dinner.

```
In [4]: naval_officers = ['Wendy B. Lawrence', 'Alfred Thayer Mahan', 'John Paul Jones']
print(f"{naval_officers[0]}, you are invtd to dinner!")
print(f"{naval_officers[1]}, you are invtd to dinner!")
print(f"{naval_officers[2]}, you are invtd to dinner!")
```

Wendy B. Lawrence, you are invtd to dinner!
Alfred Thayer Mahan, you are invtd to dinner!
John Paul Jones, you are invtd to dinner!

3-5. Changing Guest List: One of your guests cannot make it, which is ok because you forgot to invite someone from the Marines.

- Start with the code from above. Add a `print()` call at the end of the program stating the name of the guest who can't make it.
- Modify the list, replacing the guest who can't make it with a famous Marine Officer.
- Print a second set of invitation messages, one for each person who is still in your list

```
In [5]: print(f"{naval_officers[-1]} can't make it\n")
        naval_officers[-1] = 'John A. Lejeune'

        #manual example
        print(f"{naval_officers[0]}, you are invited to dinner!")
        print(f"{naval_officers[1]}, you are invited to dinner!")
        print(f"{naval_officers[2]}, you are invited to dinner!")
        print("\n")

        # for example
        for officer in naval_officers:
            print(f"{officer}, you are invtd to dinner!")
```

John Paul Jones can't make it

Wendy B. Lawrence, you are invited to dinner!
 Alfred Thayer Mahan, you are invited to dinner!
 John A. Lejeune, you are invited to dinner!

Wendy B. Lawrence, you are invtd to dinner!
 Alfred Thayer Mahan, you are invtd to dinner!
 John A. Lejeune, you are invtd to dinner!

3-6. More Guests: You just found a bigger dinner table, so now more space is available. Think of three famous Army Officers to invite to dinner. A joint operation!

- Start with the code from above. 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.

```
In [6]: print("We found a bigger dinner table!\n")
        officers = naval_officers
        officers.insert(0, 'Douglas MacArthur')
        officers.insert(int(len(officers)/2), 'Dwight D. Eisenhower')
        officers.append('George S. Patton')

        print(f"{officers[0]}, you are invited to dinner!")
        print(f"{officers[1]}, you are invited to dinner!")
        print(f"{officers[2]}, you are invited to dinner!")
        print(f"{officers[3]}, you are invited to dinner!")
        print(f"{officers[4]}, you are invited to dinner!")
        print(f"{officers[5]}, you are invited to dinner!")
```

We found a bigger dinner table!

Douglas MacArthur, you are invited to dinner!
Wendy B. Lawrence, you are invited to dinner!
Dwight D. Eisenhower, you are invited to dinner!
Alfred Thayer Mahan, you are invited to dinner!
John A. Lejeune, you are invited to dinner!
George S. Patton, you are invited to dinner!

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 the program above. 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 are 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.

```
In [7]: print("We can only invite two people for dinner\n")
print(f"Sorry, {officers.pop()}, you are no longer invited.")
print(f"Sorry, {officers.pop()}, you are no longer invited.")
print(f"Sorry, {officers.pop()}, you are no longer invited.")
print(f"Sorry, {officers.pop()}, you are no longer invited.")
print(f"{officers[0]}, you are still invited!")
print(f"{officers[1]}, you are still invited!")
# have to delete at index zero because once you delete the first officers[0],
del officers[0], officers[0]
print(officers)
```

We can only invite two people for dinner

Sorry, George S. Patton, you are no longer invited.
 Sorry, John A. Lejeune, you are no longer invited.
 Sorry, Alfred Thayer Mahan, you are no longer invited.
 Sorry, Dwight D. Eisenhower, you are no longer invited.
 Douglas MacArthur, you are still invited!
 Wendy B. Lawrence, you are still invited!
 []

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 in its original order.
- Use `sort()` to change your list so it's stored in reverse alphabetical order. Print the list to show that its order has changed.

```
In [8]: locations = ['Paris', 'Tokyo', 'New York', 'Manila']
print(locations)
print(sorted(locations))
print(locations)
print(sorted(locations, reverse=True))
print(locations)
locations.reverse()
print(locations)
locations.reverse()
print(locations)
locations.sort(reverse=True)
print(locations)
```

```
['Paris', 'Tokyo', 'New York', 'Manila']
['Manila', 'New York', 'Paris', 'Tokyo']
['Paris', 'Tokyo', 'New York', 'Manila']
['Tokyo', 'Paris', 'New York', 'Manila']
['Paris', 'Tokyo', 'New York', 'Manila']
['Manila', 'New York', 'Tokyo', 'Paris']
['Paris', 'Tokyo', 'New York', 'Manila']
['Tokyo', 'Paris', 'New York', 'Manila']
```

3-9. Dinner Guests: Working with one of the programs from **3-4** through **3-7**, use `len()` to print a message indicating the number of people you are inviting to dinner.

```
In [9]: print(f"I will be inviting {len(officers)} people to dinner")
```

I will be inviting 0 people to dinner

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 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.

```
In [10]: colors = ['red', 'yellow', 'green', 'blue']
print(colors)
print(sorted(colors))
print(len(colors))
colors.append('purple')
print(colors)
colors.insert(3, 'orange')
print(colors)
print(colors.pop())
print(colors)
colors.remove('yellow')
print(colors)
colors.sort()
print(colors)
colors.reverse()
print(colors)
del colors[3]
print(colors)
```

```
['red', 'yellow', 'green', 'blue']
['blue', 'green', 'red', 'yellow']
4
['red', 'yellow', 'green', 'blue', 'purple']
['red', 'yellow', 'green', 'orange', 'blue', 'purple']
purple
['red', 'yellow', 'green', 'orange', 'blue']
['red', 'green', 'orange', 'blue']
['blue', 'green', 'orange', 'red']
['red', 'orange', 'green', 'blue']
['red', 'orange', 'green']
```

3-11. Intentional Error: Change an index in one of your programs to produce an index error. Learning how to read and understand Python error messages can significantly speed up your code debugging.

```
In [11]: trees = ['Oak', 'Maple', 'Spruce', 'Dogwood']
print(trees)
trees[10] = 'Redwood'
```

```
['Oak', 'Maple', 'Spruce', 'Dogwood']
```

```
-----
IndexError                                Traceback (most recent call last)
Cell In[11], line 3
      1 trees = ['Oak', 'Maple', 'Spruce', 'Dogwood']
      2 print(trees)
----> 3 trees[10] = 'Redwood'

IndexError: list assignment index out of range
```

