DATA 602 Assignment 2

Author: Kevin Havis

Q1

What will the following code display?

```
numbers = [1, 2, 3, 4, 5]
print(numbers[1:-5])
```

Can you debug and fix the output? The code should return the entire list

The slice is starting at the 2nd element through the 1st element, which is invalid.

In [1]:
```
numbers = [1, 2, 3, 4, 5]
print(numbers[1:-5])
```

```
[]
```

In [2]:
```
# The fix
print(numbers[:])
```

```
[1, 2, 3, 4, 5]
```

Q2

Design a program that asks the user to enter a store's sales for each day of the week. The amounts should be stored in a list. Use a loop to calculate the total sales for the week and display the result.

In [3]:
```
def store_sales():

    # initiate the list
    sales = []
    total_sales = 0

    # define the days we expect sales for
    week_days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']

    # get sales from user input per day
    for day in week_days:
        sales.append(int(input(f'Enter the sales for {day}: ')))

    # total the sales
    for sale in sales:
        total_sales += sale
```

```
    print(f'Total sales for the week: {total_sales}')

    return None

store_sales()
```

```
Total sales for the week: 500
```

Q3

Create a list with at least 5 places you'd like to travel to. Make sure the list isn't in alphabetical order

- Print your list in its original order.
- Use the sort() function to arrange your list in order and reprint your list.
- Use the sort(reverse=True) and reprint your list.

In [14]:
```
def places():

    # Create list of places
    places = [
        'Italy',
        'Finland',
        'Iceland',
        'Carribbean',
        'Alaska'
    ]

    print(places)

    # Sort in alphabetical order
    places.sort() # Sorts in place
    print(places)

    # Sort in reverse alphabetical order
    places.sort(reverse=True)
    print(places)

    return None

places()
```

```
['Italy', 'Finland', 'Iceland', 'Carribbean', 'Alaska']
['Alaska', 'Carribbean', 'Finland', 'Iceland', 'Italy']
['Italy', 'Iceland', 'Finland', 'Carribbean', 'Alaska']
```

Q4

Write a program that creates a dictionary containing course numbers and the room numbers of the rooms where the courses meet. The program should also create a dictionary containing course numbers and the names of the instructors that teach each

course. After that, the program should let the user enter a course number, then it should display the course's room number, instructor, and meeting time.

In [ ]:
```
def course_dict():

    # create lists of each data attribute
    courses = ['DATA101', 'DATA102', 'DATA103']
    rooms = ['A', 'B', 'C']
    profs = ['John Doe', 'Jane Doe', 'O Captain my captain']

    # map course numbers to room numbers
    course_room = dict(zip(courses, rooms))

    # map course numbers to instructors
    course_prof = dict(zip(courses, profs))

    # get user input
    req_course = (input("What course are you going to? (DATA101, DATA102, DA

    # print the course room and professor
    print(f'{req_course} is in Room {course_room[req_course]} with Professor

    return None

course_dict()
```

```
DATA101 is in A with Professor John Doe today. Enjoy!
```

Q5

Write a program that keeps names and email addresses in a dictionary as key-value pairs. The program should then demonstrate the four options:

- look up a person's email address,
- add a new name and email address,
- change an existing email address, and
- delete an existing name and email address.

In [ ]:
```
import time

def main():

    entries = {'Owen Wilson':'waow@gmail.com', 'Darth Vader':'hoooperrr@empi

    def print_entries()->None:
        '''Print all current entries'''
        print("Current entries:")
        for k, v in entries.items():
            print(f"Name: {k} Email: {v}\n")
        return None

    def lookup(name:str)->None:
```

```
    '''Prints email address for a given name'''
    print(f'Name: {name}\n Email: {entries[name]}')
    return None

def update_entry(name:str, email:str)->None:
    '''Adds or updates an existing entry'''
    entries[name] = email
    print(f"Updated email for {name} to {email}")
    return None

def delete_entry(name:str)->None:
    '''Deletes a name from the address book'''
    entries.pop(name)
    print(f'Removed {name} from the address book')
    return None

# start of main program loop

# Show user current entries for reference
print_entries()

# Turn loop on
run = True

while run:

    # Get user request
    request = int(input("""
                        What would you like to do?\n
                        \tEnter 1 to look up an email address\n
                        \tEnter 2 to update an entry\n
                        \tEnter 3 to delete an existing entry\n
                        \tEnter 0 to quit\n
                        """))
    # Look up name
    if request == 1:
        lookup_name = input("Please enter the name of the email you'd li
        lookup(lookup_name)

    # Edit email
    elif request == 2:
        update_name = input("Please enter the name of the email you'd li
        lookup(update_name)

        update_email = input(f"Please enter the new email address for {u
        update_entry(update_name, update_email)

    # Delete name
    elif request == 3:
        delete_name = input("Please enter the name of the entry you woul
        delete_entry(delete_name)

    # Exit program loop
    else:
        print("Exiting . . .")
        time.sleep(2) # pause before exiting for UX
```

```
            run = False

    return None

main()
```

```
Current entries:
Name: Owen Wilson Email: waow@gmail.com

Name: Darth Vader Email: hoooperrr@empire.gov

Name: Brittany Spears Email: whoops@yahoo.com

Name: Darth Vader
 Email: hoooperrr@empire.gov
Updated email for Darth Vader to tatooine@hotmail.com
Name: Darth Vader
 Email: tatooine@hotmail.com
Exiting . . .
```