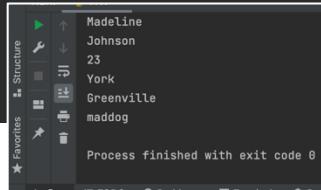


## Exercise 4

#1

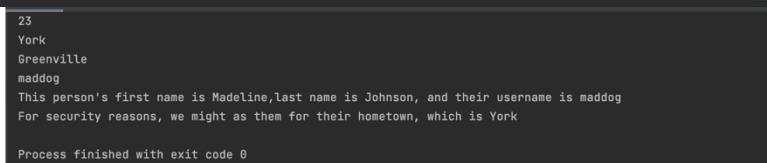
```
1 sister = {'f_name': 'Madeline', 'l_name': 'Johnson', 'age': 23, 'hometown': 'York', 'current_city': 'Greenville', 'username': 'maddog'}
2 print(sister['f_name'])
3 print(sister['l_name'])
4 print(sister['age'])
5 print(sister['hometown'])
6 print(sister['current_city'])
7 print(sister['username'])
```



#2

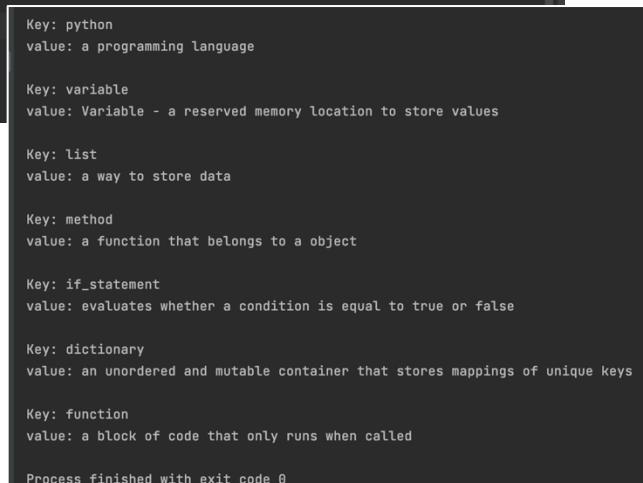
```
1 sister = {'f_name': 'Madeline', 'l_name': 'Johnson', 'age': 23, 'hometown': 'York', 'current_city': 'Greenville', 'username': 'maddog'}
2 print(sister['f_name'])
3 print(sister['l_name'])
4 print(sister['age'])
5 print(sister['hometown'])
6 print(sister['current_city'])
7 print(sister['username'])

8
9 print(f"This person's first name is {sister['f_name']}, last name is {sister['l_name']}, and their username is {sister['username']}")
10 print(f"For security reasons, we might as them for their hometown, which is {sister['hometown']}")
```



#3 & #4

```
1 definitions = {
2     'python': 'a programming language',
3     'variable': 'Variable - a reserved memory location to store values',
4     'list': 'a way to store data',
5     'method': 'a function that belongs to a object',
6     'if_statement': 'evaluates whether a condition is equal to true or false',
7     'dictionary': 'an unordered and mutable container that stores mappings of unique keys',
8     'function': 'a block of code that only runs when called',
9 }
10
11 for key, value in definitions.items():
12     print(f"\nKey: {key}")
13     print(f"Value: {value}")
```



#5

The screenshot shows a Python code editor with the following code:

```
1  South_Carolina = {  
2      'York County':'York',  
3      'Pickens County':'Pickens',  
4      'Richland County':'Columbia',  
5      'Saluda County':'Saluda',  
6      'Spartanburg County':'Spartanburg',  
7      'Sumter County':'Sumter',  
8  }  
9  for key, value in South_Carolina.items():  
10     print(f"\nKey: {key}")  
11     print(f"Value:{value}")
```

To the right of the code, the terminal output is displayed:

```
Key: York County  
Value:York  
  
Key: Pickens County  
Value:Pickens  
  
Key: Richland County  
Value:Columbia  
  
Key: Saluda County  
Value:Saluda  
  
Key: Spartanburg County  
Value:Spartanburg  
  
Key: Sumter County  
Value:Sumter  
  
Process finished with exit code 0
```

#6

The screenshot shows a Python code editor with the following code:

```
blic ~/Public  
Drop Box  
.localized  
CYB 220.py  
HW.py  
In class.py  
ternal Libraries  
ratches and Cor  
1  South_Carolina = {  
2      'York County':'York',  
3      'Pickens County':'Pickens',  
4      'Richland County':'Columbia',  
5      'Saluda County':'Saluda',  
6      'Spartanburg County':'Spartanburg',  
7      'Sumter County':'Sumter',  
8  }  
9  more = ['Oconee','Spartanburg','Richland','lee','laurens','York','Pickens','Saluda','sumter','Newb  
10  for county in South_Carolina.values():  
11      print(f"County Name is in our dictionary, and the capital is {county.title()}")  
12  
13  if county in more:  
14      print(f"County name is not in our dictionary. We will add this county shortly. Thanks!")  
for county in South_Carolina.va... > if county in more
```

Below the code editor, a terminal window titled "HW" shows the execution of the script:

```
/usr/local/bin/python3.8 /Users/audreyjohnson/Public/HW.py  
County Name is in our dictionary, and the capital is York  
County name is not in our dictionary. We will add this county shortly. Thanks!  
County Name is in our dictionary, and the capital is Pickens  
County name is not in our dictionary. We will add this county shortly. Thanks!  
County Name is in our dictionary, and the capital is Columbia  
County Name is in our dictionary, and the capital is Saluda  
County name is not in our dictionary. We will add this county shortly. Thanks!  
County Name is in our dictionary, and the capital is Spartanburg  
County name is not in our dictionary. We will add this county shortly. Thanks!  
County Name is in our dictionary, and the capital is Sumter  
  
Process finished with exit code 0
```

#7

I have no clue what is going on here. But I mainly do not understand why Sumter is the only one that is “not” a capital. Like what ???

```
blic ~/Public
Drop Box
.localized
CYB 220.py
HW.py
In class.py
ternal Libraries
atches and Cor
1   South_Carolina = {
2       'York County': 'York',
3       'Pickens County': 'Pickens',
4       'Richland County': 'Columbia',
5       'Saluda County': 'Saluda',
6       'Spartanburg County': 'Spartanburg',
7       'Sumter County': 'Sumter',
8   }
9   city = ['York', 'Rock Hill', 'Fort Mill', 'Newport', 'Lake Wylie']
10  for city in South_Carolina.values():
11      print(f"{city} is the County Name in South Carolina")
12
13  if city not in South_Carolina:
14      print(f"{city} is not a capital of any South Carolina county")

HW x
/usr/local/bin/python3.8 /Users/audreyjohnson/Public/HW.py
York is the County Name in South Carolina
Pickens is the County Name in South Carolina
Columbia is the County Name in South Carolina
Saluda is the County Name in South Carolina
Spartanburg is the County Name in South Carolina
Sumter is the County Name in South Carolina
Sumter is not a capital of any South Carolina county

Process finished with exit code 0
```

#8 in class

#9

I guess I don't know how to get it to do each individual one?

```
blic ~/Public
Drop Box
.localized
CYB 220.py
HW.py
In class.py
ternal Libraries
atches and Cor
1   sc_counties = {
2       'York': ['rock hill', 'fort mill', 'lake wylie'],
3       'Greenville': ['greenville', 'maudlin', 'taylors'],
4       'Pickens': ['Easley', 'Arial', 'Liberty'],
5       'Richland': ['Columbia', 'Gadsen', 'Hopkins'],
6       'Oconee': ['Seneca', 'Fair play', 'Walhalla']
7   }
8
9   print(f"\nIn {sc_counties.keys()}, \nthe largest cities are"
10     f"\n{sc_counties.values()}")
```

```
HW x
/usr/local/bin/python3.8 /Users/audreyjohnson/Public/HW.py

In dict_keys(['York', 'Greenville', 'Pickens', 'Richland', 'Oconee']),
the largest cities aredict_values([['rock hill', 'fort mill', 'lake wylie'], ['greenville', 'maudlin']])
```

## User Input:

#1

```
public ~/Public 1 message = input("Hi! Welcome to Anderson University! What is your name?") ▲1 ⌂ ⌂
Drop Box 2 print(message)
.localized CYB 220.py
HW.nv
HW ×
/usr/local/bin/python3.8 /Users/audreyjohnson/Public/HW.py
Hi! Welcome to Anderson University! What is your name? Audrey
Audrey

Process finished with exit code 0
```

#2

```
public ~/Public 1 number = input("how much money do you have?")
Drop Box 2 number = int(number)
.localized CYB 220.py
HW.py
In class.py
Internal Libraries
Scratches and Cor 7
if number >= 150:
    print(f"You have enough money to buy a i3 processor")
if number >= 200:
    print(f"You have enough money to buy a i5 processor")
if number >= 300:
    print(f"You have enough money to buy a i7 processor")
if number >= 300

How ×
/usr/local/bin/python3.8 /Users/audreyjohnson/Public/HW.py
how much money do you have? 310
You have enough money to buy a i3 processor
You have enough money to buy a i5 processor
You have enough money to buy a i7 processor

Process finished with exit code 0
```

#3

```
public ~/Public 1 number = input("Enter a number, and I'll tell you if it is odd or even:")
Drop Box 2 number = int(number)
.localized CYB 220.py
HW.py
In class.py
Internal Libraries
Scratches and Cor 6
if number % 2 == 0:
    print(f"\nThe number {number} is even.")
else:
    print(f"\nThe number {number} is odd")
else

How ×
/usr/local/bin/python3.8 /Users/audreyjohnson/Public/HW.py
Enter a number, and I'll tell you if it is odd or even: 3

The number 3 is odd
```

#4

I couldn't figure out how to have an active variable and control it so I commented it out.

```
blic ~/Public  
Drop Box  
.localized  
CYB 220.py  
HW.py  
In class.py  
Internal Libraries  
ratches and Cor  
1   prompt = "\nTell me something, and I will repeat it back to you:  
2   prompt += "\nEnter 'quit' to end the program"  
3  
4   message = ""  
5   a = 1  
6   #while a <= 25:  
7   while message != 'quit':  
8       message = input(prompt)  
9       if message != 'quit':  
10           print(message)  
11  
if message != 'quit'  
  
HW x  
/usr/local/bin/python3.8 /Users/audreyjohnson/Public/HW.py  
  
Tell me something, and I will repeat it back to you:  
Enter 'quit' to end the program hi  
  
Tell me something, and I will repeat it back to you:  
Enter 'quit' to end the programhi  
  
Tell me something, and I will repeat it back to you:  
Enter 'quit' to end the programquit  
  
Process finished with exit code 0
```

#5

```
1 responses = {}
2
3 polling_active = True
4 while polling_active:
5     username = input("\nWhat is your username?")
6     response = input("What is your favorite Ubuntu flavour?")
7     responses[username] = response
8     repeat = input("Another user? (yes/no)")
9     if repeat == 'no':
10         polling_active = False
11
12 print("\n ---Poll Results---")
13 for username, response in responses.items():
14     print(f"{username} 's favorite Ubuntu flavour is {response}")
15
16 print(responses)
```