Python Basic Camp

Exercises Day 3

Python Dictionary methods: https://www.w3schools.com/python/python_ref_dictionary.asp
Python Set methods: https://www.w3schools.com/python/python_ref_set.asp

1.1 - Dictionary	Creating a dictionary
Instructions	Create a dictionary representing a person with keys 'name', 'age', and 'city'. Request the data to the user. Then print the dictionary.
Expected output	Name: >>>John Age: >>>32 City: >>>Luxembourg The content of the dictionary is: {'name': 'John', 'age': 32, 'city': 'Luxembourg'}
Solution	`name = input("Name: ")` `age = int(input("Age: "))` `city = input("City: ")` `my_dictionary = {"name":name, "age":age, "city":city}` `print("The content of the dictionary is: ", my_dictionary)`

1.2 - Dictionary	Updating a dictionary
Instructions	Update the person's age in the dictionary from the previous exercise. Then print the dictionary.
Expected output	The content of the dictionary is: {'name': 'John', 'age': 40, 'city': 'Luxembourg'}
Solution	`my_dictionary["age"] = 40` `print("The content of the dictionary is: ", my_dictionary)`

1.3 - Dictionary	Printing the content of a dictionary
Instructions	Print the contents of the dictionary from the previous exercise using these three methods and check the different outputs: ● dictionary.keys() ● dictionary.values() ● dictionary.items()
Expected output	Dictionary keys: dict_keys(['name', 'age', 'city'])

	Dictionary values: dict_values(['John', 40, 'Luxembourg']) Dictionary items: dict_items([('name', 'John'), ('age', 40), ('city', 'Luxembourg')])
Solution	`print("Dictionary keys:", my_dictionary.keys())` `print("Dictionary values:", my_dictionary.values())` `print("Dictionary items:", my_dictionary.items())`

1.4 - Dictionary	The supermarket dictionary
Instructions	Create a dictionary to store the code and name of the products of a supermarket. Start with an empty dictionary. Then ask the user if he would like to input a new product, if he would like to see the full list of products or if he would like to exit. ● If the user would like to input a new product. Ask the user for the product's code and name. Store the new product in the dictionary. ● If the user would like to see the full list of products, print the contents of the dictionary. ● If the user would like to exit, then end the program.
Expected output	Would you like to (1) input a new product, (2) see the list of products or (3) exit? >>>1 Code of the product: >>>29315 Name of the product: >>>Chocolate bar Would you like to (1) input a new product, (2) see the list of products or (3) exit? >>>1 Code of the product: >>>13215 Name of the product: >>>Vanilla Ice Cream Would you like to (1) input a new product, (2) see the list of products or (3) exit? >>>2 List of products: 29315 - Chocolate bar 13215 - Vanilla Ice Cream Would you like to (1) input a new product, (2) see the list of products or (3) exit? >>>3 Good bye!
Solution	`products_dictionary = {}` `choice = input("Would you like to (1) input a new product, (2) see the list of products or (3) exit? ")` `while choice != "3":`

	`` if choice == "1": product_code = input("Code of the product: ") product_name = input("Name of the product: ") products_dictionary[product_code] = product_name elif choice == "2": for code, name in products_dictionary.items(): print(code + " - " + name) else: print("Invalid option. Please select 1, 2 or 3") choice = input("Would you like to (1) input a new product, (2) see the list of products or (3) exit? ") print("Good bye!") ```

2.1 - Sets	Union of sets
Instructions	You have two sets of student IDs representing students who attended Math and Science classes. Use sets to find the IDs of students who attended either Math or Science class. math_students = {101, 102, 103, 104, 105} science_students = {103, 104, 106, 107, 108}
Expected output	Students that attended Math OR Science classes: {101, 102, 103, 104, 105, 106, 107, 108}
Solution	``` math_students = {101, 102, 103, 104, 105} science_students = {103, 104, 106, 107, 108} print("Students that attended Math OR Science classes:", math_students.union(science_students)) ```

2.2 - Sets	Intersection of sets
Instructions	Using the same sets from the previous exercise, find the IDs of the students who attended both, Math AND Science classes.
Expected output	Students that attended Math AND Science classes: {104, 103}
Solution	``` print("Students that attended Math AND Science classes:", math_students.intersection(science_students)) ```

2.3 - Sets	Updating sets
Instructions	You have a set of new student IDs representing new students that will attend Math class. Update the set of student IDs of the Math class from the previous exercise. Then print the updated set of student IDs of the Math class. new_students = {110,111}
Expected output	Students that attend Math class: {101, 102, 103, 104, 105, 110, 111}
Solution	`new_students = {110, 111}` `math_students.update(new_students)` `print("Students that attend Math class:", math_students)`