

Practical

Lists

1. Create the list **list1**, which contains the values [**“hello”**, **1**, **True**]. Get some values from the user, using the `sys` module, and add those values to **list1**: Print **list1** before and after adding the values.
2. Do the previous exercise creating a new list which contains both the values from **list1** and the new values given by the user, don't make changes to the list **list1**.
3. Create the list **list2**, which contains some values of your choice. Get one value as a positional command line argument from the user (using the module `argparse`) and check how many times the given value appears in the list **list2**. Print the result in the following format:

(in this example user input is 2, the list2 is [0, 'hi', 2, 100, 300, 2])

```
list2 = [0, 'hi', 2, 100, 300, 2]
```

Number of 2s = 2

4. Create the list **list4**, which contains some values of your choice. Get one value as a positional command line argument from the user (using the module `argparse`) and delete the first occurrence of the value from the list **list4**. Print the list **list4** before and after the change.
5. Create the list **list5**, which contains some values of your choice. Delete the values at indexes **0**, **4** and **5** from the list. Print the list **list5** before and after the change.
6. Do the previous exercise creating a new list and making the required changes on it, without changing the list **list5**.
7. Create the lists **l1** and **l2**, which contain some values of your choice. Replace the last value of list **l1** by the list **l2**. Print the list **l1** before and after the change.

Sets

8. Create the set **set1**, which contains some values of your choice: Get one value as a positional command line argument from the user (using the module `argparse`) and add the value to the set **set1**. Print the set **set1** before and after the change.

9. Create the set **set2**, which contains some values of your choice: Get one value as a positional command line argument from the user (using the module `argparse`) and remove the value from the set **set2**. Print the set **set2** before and after the change.
10. Do the previous exercise taking into consideration that the value given by the user may not be in the set **set2**, and you shouldn't get an error while removing the value anyways.
11. Create the sets **set1** and **set2** containing some values of your choice. Print the union and the intersection of the 2 sets.
12. Create the set **set3** containing the values of type `int` of your choice. Get one value as a positional command line argument from the user (using the module `argparse`) and check if the value is between the minimum and maximum values of the set **set3** (`min<value<max`). Print `True` (it is) or `False` (it is not) accordingly.

Tuples

13. Create the tuple **t1** containing some values of your choice. Print the values at indices **4** and **4** from the end.
14. Create the tuple **t1** containing some values of your choice. Replace the value at index **4** by the value `"hello"`. Print the tuple **t1** before and after the change. Note that the tuples are immutable i.e. cannot be modified.

Dictionary

15. Create the dictionary **dict1** containing some values of your choice. Get a value **key** of type `String` and a value **value** of type `String` as positional command line arguments from the user (using the module `argparse`). Add the values as a **key:value** pair to the dictionary **dict1**. Print the dictionary **dict1** before and after adding the value.
16. Create a list of tuples **l1** with the following values: `[(1, "a"), (2, "b"), (3, "c")]`. Create the dictionary **d1**, the keys of the dictionary should be the first values of all the tuples in **l1** and the values should be the second values of all the tuples in **l1**. You should get the following dictionary: `{1: "a", 2: "b", 3: "c"}`:

Homework

Lists:

- 1) Create the list **a** with the following values: **1, 4, 5, 7, 8, -2, 0, -1**
- 2) Print the values of list **a** at indices **3** and **5**
- 3) Sort the list **a** in a decreasing order and assign the newly obtained list to the variable **a_sorted**, the list **a** should not be changed
- 4) Print the 2 sublists of the list **a_sorted** containing the indices 1...3 and 2...6
- 5) Delete the values at indices **2** and **3** from **a_sorted**
- 6) Print the list **a_sorted**
- 7) Create the list **b** with the following values: **"grapes", "Potatoes", "tomatoes", "Orange", "Lemon", "Broccoli", "Carrot", "Sausages"**
- 8) Sort the list **b** in an increasing order and assign the newly obtained list to the variable **b_sorted**, the list **b** should not be changed
- 9) Create a new list **c**. The first 3 values of the list **c** should be the elements from the list **a** at indices 1...3 and the last values of the list **c** should be the elements from the list **b** at indices 4...6.
- 10) Print the list **c**

Sets:

- 1) Create the list **a1** with the following values: **"Cookies", "Chocolate", 8, True, -3, -5, "Chocolate", 8, False, 8**
- 2) Create the list **b1** with the following values: **8, True, 10, 14, "Chocolate", "Milk", "Jelly", True, False, True**
- 3) Convert the list **a1** into a set and assign the result to the variable **set_a**
- 4) Convert the list **b1** into a set and assign the result to the variable **set_b**
- 5) Find the union of the sets **set_a** and **set_b** and assign the result to the variable **union_ab**
- 6) Find the intersection of the sets **set_a** and **set_b** and assign the result to the variable **intersection_ab**
- 7) Add the values "Kit-Kat" and "Oreo" to **union_ab** and **intersection_ab** after the change
- 8) Use the operation **or** between the sets **union_ab** and **intersection_ab** and assign the result to the variable **new_set**. Then, print the variable **new_set**.
- 9) Check if the set **new_set** contains the value **"Chocolate"**
- 10) Delete the value **"Oreo"** from the set **new_set**, then print **new_set**

Tuples:

- 1) Create the tuple **t1** with the following values: **1, True, "a", -2, "Anna"**
- 2) Delete the value **"True"** from **t1** and print **t1** afterwards
- 3) Create the tuple **t2** with the following values: **1, 2, 3, 4, 5**
- 4) Create the tuple **t3**. The first 2 values of **t3** should be the first 2 values of the tuple **t1**, and the last 3 values of **t3** should be the first 3 values of the tuple **t2**
- 5) Print the value at index **2** from the tuple **t3**
- 6) Create a list of tuples **t4** with the following value: **[(1,3,5), (8,9), ("Anna", "Bob", "Alice")]**: Print the second value of the first tuple in the list **t4**.

Dictionaries:

Create the dictionary **market** with the following values: **{"dairy": ["yogurt", "cheese"], "fruits": ["banana", "apple", "orange", "lemon", "apple", "banana", "banana"]}**. Add the key **"candies"** with a value **["mars", "kinder", "twix"]** to the dictionary **Market**: Sort the values at the key **"fruits"** in an increasing order and get rid of the duplicate values: Print the dictionary **Market** before and after the changes.