# Sets and Dictionaries

## Exercises

### Week 7

Prior to attempting these exercises ensure you have read the lecture notes and/or viewed the video, and followed the practical. You may wish to use the Python interpreter in interactive mode to help work out the solutions to some of the questions.

Download and store this document within your own filespace, so the contents can be edited. You will be able to refer to it during the test in Week 6.

Enter your answers directly into the highlighted boxes.

For more information about the module delivery, assessment and feedback please refer to the module within the MyBeckett portal.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## ©2020 Mark Dixon / Tony Jenkins

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Specify two ways in which a Set varies from a List.

*Answer:*

A set does not contain any duplicate values.

The items are stored in an unordered manner.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a Python statement that uses the set() *constructor* to produce the same Set as the following -

languages = { "C++", "Java", "C#", "PHP", "JavaScript" }

*Answer:*

languages = set( "C++", "Java", "C#", "PHP", "JavaScript")

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Is a Set **mutable** or **immutable**?

*Answer:*

A set would be mutable, as it can be modified by adding or removing elements.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Why does a Set not support *indexing* and *slicing* type operations?

*Answer:*

They are unordered and can appear randomly each time they are used by the program, therefore it would be impossible to index and slice.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Why is a frozenset() different from a regular set?

*Answer:*

A frozenset() cannot be modified, whereas a regular set can be modified. Items cannot be added or removed from a frozenset.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

How many elements would exist in the following set?

names = set("John", "Eric", "Terry", "Michael", "Graham", "Terry")

*Answer:*

5 – There would only be 5 elements within this set as there cannot be any duplicates, and ‘Terry’ is repeated.

And how many elements would exist in this set?

vowels = set("aeiou")

*Answer:*

5 – Each letter would represent its own element, and as there are no duplicates, none would be removed.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What is the name given to the following type of expression which can be used to programmatically populate a set?

chars = {chr(n) for n in range(32, 128)}

*Answer:*

The name for an expression to programatically populate a set would be Set Comprehension.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What **operator** can be used to calculate the intersection (common elements) between two sets?

*Answer:*

The & operator could be used to find the common elements between 2 sets.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What **operator** can be used to calculate the difference between two sets?

*Answer:*

The – operator could be used to calculate the difference between 2 sets.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would be the result of each of the following expressions?

{ "x", "y", "z" } < { "z" , "u", "t", "y", "w", "x" }

*Answer:*

True

{ "x", "y", "z" } < { "z", "y", "x" }

*Answer:*

False

{ "x", "y", "z" } <= { "y", "z", "x" }

*Answer:*

True

{ "x" } > { "x" }

*Answer:*

False

{ "x", "y" } > { "x" }

*Answer:*

True

{ "x", "y" } == { "y", "x" }

*Answer:*

True

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a Python statement that uses a **method** to perform the equivalent of the following operation -

languages = languages | { "Python" }

*Answer:*

languages.union({"Python"})

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Do the elements which are placed into a set always remain in the same position?

*Answer:*

No, the elements are unordered and therefore are randomly positioned within the set. They do not have a fixed position or order, hence why we cannot use indexing or slicing.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Is the following operation a **mutator** or an **accessor**?

languages &= oo\_languages

*Answer:*

It would be a mutator operation as it modifies the languages list by removing elements that are not present in both sets and leaving only the common elements

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What term is often used to refer to each *pair* of elements stored within a **dictionary**?

*Answer:*

A pair of elements within a dictionary would be a key:value pair.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Is it possible for a dictionary to have more than one **key** with the same value?

*Answer:*

A dictionary cannot have more than 1 key with the same value, they must all be unique.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Is it possible for a dictionary to have the same **value** appear more than once?

*Answer:*

A dictionary can have values appearing more than once.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Is a Dictionary **mutable** or **immutable**?

*Answer:*

A dictionary would be mutable as items can be added and removed by the users wish.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Are the **key** values within a dictionary **mutable** or **immutable**?

*Answer:*

The keys are immutable as they cannot be changed within the program due to the need for it to remain the same as it is ran.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

How many *elements* exist in the following dictionary?

stock = {"apple":10, "banana":15, "orange":11}

*Answer:*

3 elements exist within the above dictionary as there are 3 different pairs.

And, what is the data-type of the **keys**?

*Answer:*

Integers

And, what output would be displayed by executing the following statement -

print(stock["banana"])

*Answer:*

‘15’ would be printed as it is checking the value which is matched with ‘banana’.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a Python statement that uses the dictionary() *constructor* to produce the same dictionary as the following -

lang\_gen = { "Java":3, "Assembly":2, "Machine Code":1 }

*Answer:*

lang\_gen = dict({"Java": 3, "Assembly": 2, "Machine Code": 1})

Now write a simple expression that tests whether the word "Assembly" is a member of the dictionary.

*Answer:*

if 'Assembly' in lang\_gen:  
 print('True')

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write some Python code that uses a for statement to iterate over a dictionary called module\_stats and print only its **values** (i.e. do not output any keys) -

*Answer:*

for value in module\_stats.values():  
 print(value)

Now write another loop which prints the only the **keys** -

*Answer:*

for keys in module\_stats.keys():  
 print(keys)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Is it possible to construct a dictionary using a **comprehension** style expression, as supported by lists and sets?

*Answer:*

Yes it is possible to create a comprehension style expression, such as in lists and sets, and it is done in quite a similar way.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

When a Dictionary type value is being passed as an argument to a function, what characters can be used as a prefix to force the dictionary to be **unpacked** prior to the call being made?

*Answer:*

The \*\* prefix should be added when passing a dictionary as an argument to a function as they force it to be unpacked beforehand, meaning that the function can fully understand and work with it.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Exercises are complete**

Save this logbook with your answers. Then ask your tutor to check your responses to each question.