THE UNIVERSITY OF MELBOURNE

COMP10001

Foundations of Computing

Semester 1, 2021

Tutorial 5

Andrew Naughton

andrew.naughton@unimelb.edu.au

THE UNIVERSITY OF
MELBOURNE

Outline

- Dictionaries
- Sets
- None
- sorted() and .sort
- Namespaces and Scope
- Exercises

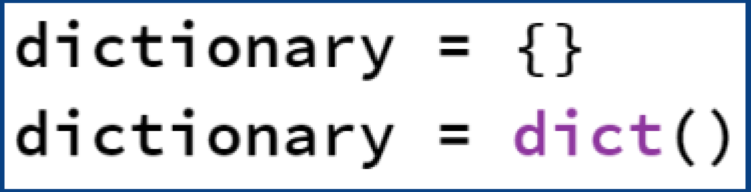
Dictionaries

- Analogous to real-world dictionaries, in which we store (unique) words and their meanings
- A dict is a data structure that stores an unordered collection of key-value pairs
  - The keys are unique and must be of an immutable type (bool, int, float, str, tuple)
  - The values can be of any type

THE UNIVERSITY OF MELBOURNE

Dictionaries

- ❖ Useful when:
  - ❖ Counting frequencies
  - ❖ Storing information related to different objects in our code
- ❖ To define a new empty dictionary:



```
dictionary = {}
dictionary = dict()
```

dict operations

❖ Accessing a value associated with a particular key:	`dict[key]`
❖ Accessing all the keys (returns a list):	`dict.keys()`
❖ Accessing all the values (returns a list):	`dict.values()`
❖ Accessing all the key-value pairs (returns a list)	`dict.items()`
❖ Make a copy	`dict.copy()`

dict operations

- Adding a new key: `dictionary[new_key] = associated_value`
- Updating a key with a new value: `dictionary[key_to_update] = new_value`
- Remove a single key-value pair `dictionary.pop(key)`
- Removing all key-value pairs `dictionary.clear()`

THE UNIVERSITY OF
MELBOURNE

Sets

- Essentially represents a mathematical set
- A data structure that stores an unordered collection of unique items
  - The items of a set must be of an immutable type (bool, int, float, str, tuple)

THE UNIVERSITY OF
MELBOURNE

Sets

- Useful when:
  - We store a mathematical set of numbers
  - We want to remove duplicates from some other sequence
  - We want to combine sets with set operations (e.g. set union, set intersection)
- To define a new empty set:

```
my_set = set()
```

set operations

- Adding a new element:
  - Since all elements must be unique, adding an element that already exists has no effect

```
set.add(new_elem)
```

- Removing an element:
  - Removing (and retrieving) a random element:

```
set.pop()
```

  - Removing a specific element:

```
set.remove(elem_to_remove)
```

set operations

- Set Intersection
  - The elements common to both sets

```
set1.intersect(set2)
or set1 & set2
```

- Set Union
  - The unique elements from both sets

```
set1.union(set2)
or set1 | set2
```

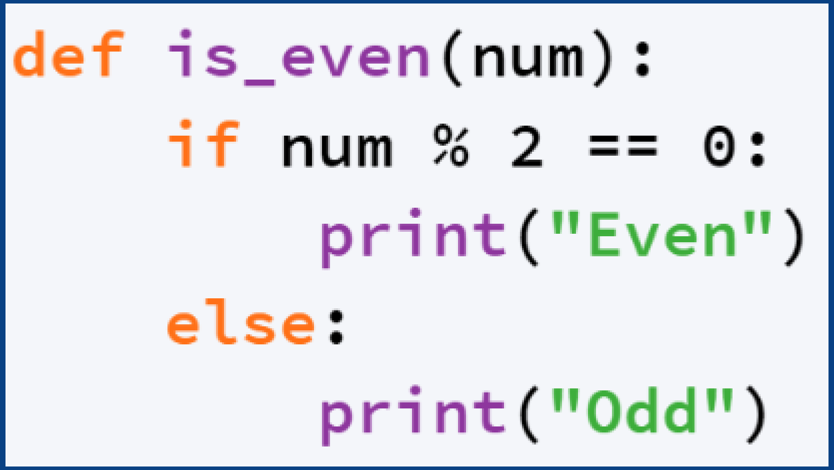
- Set Difference
  - The elements in set1 that aren't in set2

```
set1.difference(set2)
or set1 - set2
```

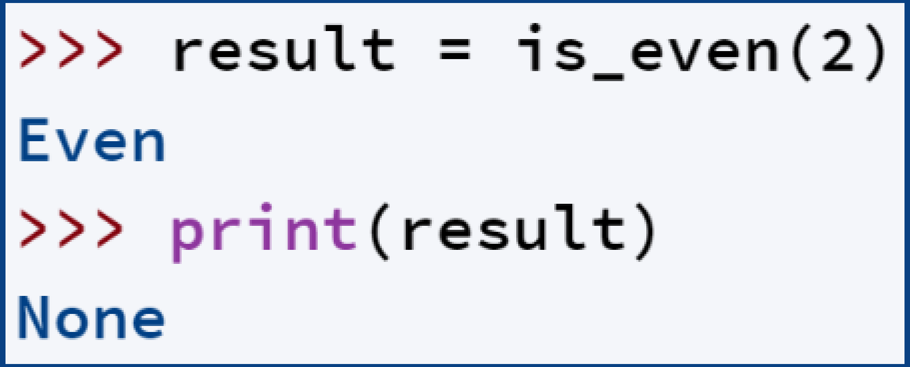
- Also: .copy(), clear(), issubset()

None

- A special value in Python
- Belongs to it own data type – NoneType
- None is the default return value of a function when we do not specify a return value ourselves:



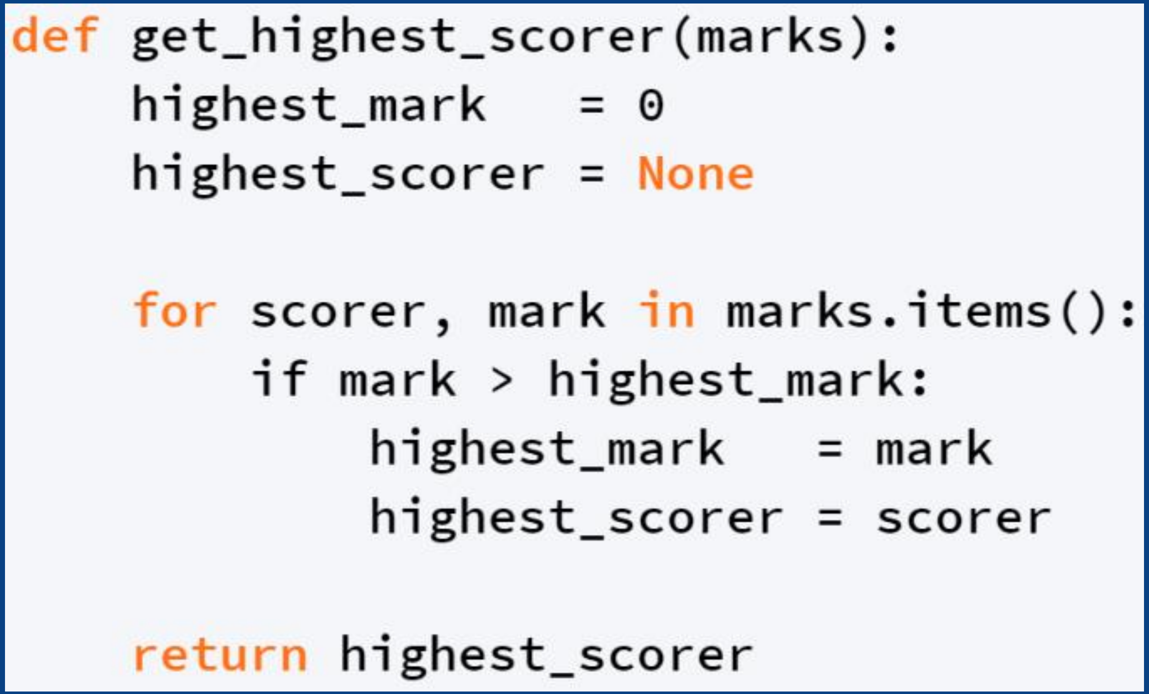
```
def is_even(num):
    if num % 2 == 0:
        print("Even")
    else:
        print("Odd")
```



```
>>> result = is_even(2)
Even
>>> print(result)
None
```

None

- Useful when:
  - Initialising a variable we have not found yet



```python
def get_highest_scorer(marks):
    highest_mark   = 0
    highest_scorer = None

    for scorer, mark in marks.items():
        if mark > highest_mark:
            highest_mark   = mark
            highest_scorer = scorer

    return highest_scorer
```

sorted() and .sort()

- sorted() is a function that takes a collection (commonly a list) as input and returns a new list of sorted elements
- .sort() is a list method that sorts a list *in-place*, i.e. it mutates the original list

THE UNIVERSITY OF
MELBOURNE

Namespaces

- A mapping from names (of variables or functions) to objects
- Defines the variables and functions that can be used in a certain part of your program
- The global namespace is the group of variables and functions available outside of any function in a program
- When a function is called, it will have a local namespace, which is unique to that function’s execution and forgotten once it terminates

THE UNIVERSITY OF
MELBOURNE

Scope

- The area of a program where a particular namespace is used:
  - Variables in a function’s local namespace are said to be in the function’s scope
  - Python looks in the most local namespace first, and if it can’t be found there, proceeds to check the global namespace

THE UNIVERSITY OF MELBOURNE

Exercises