**Section 3: Python Object and Data Structure Basics**

**01.02.**

**11. Introduction to Python Data Types**

Lists and Tuples are ordered because:

* we can access the elements by using an index
* the insertion order is maintained

**13. Numbers – FAQ**

0.1 + 0.2 – 0.3 != 0.0 because in memory each number is represented as base 2 fractions

**14. Variable Assignments**

Python uses Dynamic Typing, so we can reassign variables to different data types

Use type() for type checking

**15. Introduction to Strings**

Because strings are ordered sequences, we can use indexing and slicing

There a reverse indexing

Slicing has the following syntax: [start:stop:step]

**16. Indexing and Slicing with Strings**

Reversing a string: my\_string[::-1]

**02.02.**

**17. String Properties and Methods**

Strings are immutable

Methods: .upper(), .lower(), .capitalize(), .split()

**19. Print Formatting with Strings**

String interpolation is the process of substituting values of variables into placeholders in a string

There are 2 widely used methods:

* .format()
* f-strings

Examples for .format():

print(‘This {1} {0} {2}’.format(‘a’, ‘is’, ‘text’)) – access the strings by index

print(‘This {i} {a} {t}’.format(a=‘a’, i=‘is’, t=‘text’)) – access the strings as variables

print(‘Price is {value:width.precision f}”.format(value=results)) – floating point formatting

Examples for f-strings:

print(f‘This {var\_one} {var\_two} {var\_three}’)

**20. Print Formatting FAQs**

Resource for formatting: <https://pyformat.info/>

**21. Lists in Python**

A list has the following methods:

* Clear
* Copy
* Count – counts the number of apparitions of the passed argument
* Extend – adds the elements of an iterable (list, tuple, string) at the end of the list
* Index – get the first index of the passed element
* Insert – insert an element at a position
* Pop – removes an element placed at a given index and returns it, by default the last one
* Remove – removes the passed element
* Reverse – doesn’t return anything (is an in-place method)
* Sort – doesn’t return something and can be applied on lists with numbers / strings
* Append

To sort a list in place and to return the sorted version we can use sorted(list\_name)

**03.02.**

**24. Dictionaries – FAQ**

Keys can also be of different types, but need to be immutable (strings, numbers or tuples)

<https://softwaremaniacs.org/blog/2020/02/05/dicts-ordered/>

Dictionaries are:

* bigger than list by an order of magnitude
* ordered starting with Python 3.7, for older versions we can use orderedict
* mutable

Have started to can be sorted by Python 3.7.

The main methods are:

* .keys()
* .values()
* .items()

**25. Tuples with Python**

Are very similar to lists, but are immutable

They ensure that the objects inside won’t be replaces, thus it ensures data integrity

To define a tuple, we use parenthesis: (1, 2, 3)

They have just 2 methods, that work the same as the lists’ ones:

* .count()
* .index()

You can’t add or remove elements in a tuple

Index and slicing can be used on tuples

**26. Sets in Python**

Sets are unordered collections of unique elements

A set can be defined as:

* set\_name = set()
* set\_name = {‘a’, 1, “st”}

Defining as {}, will be recognized as a dictionary

[**https://www.programiz.com/python-programming/set**](https://www.programiz.com/python-programming/set)

Sets are mutables, but we can’t access elements via indices

We can add items in a set with:

* .add() – we add an element
* .update() – we add an iterable (list, tuple, etc)

We can remove items from a set with:

* .remove() – gives an error if we don’t have the element that we want to remove
* .discard() – does the same thing, but without throwing an error

There also are a lot of operations with sets

**07.02.**

**27. Booleans in Python**

The type(Boolean\_variable) is bool

**28. I/O with Basic Files in Python**

Files can be written directly from notebook with special commands:

%%writefile file\_name.txt

text\_for\_file

We can open files in 2 ways:

* f = open(file\_name)
* …
* f.close()
* with open(file\_name) as f:
* …

Functions used:

* f.read()
* f.readlines()
* f.seek(position)
* f.write()

The open function has a mode that can has a value from the following ones:

* r – read only
* w – write only (overwrite / create new)
* a - append
* r+ - reads and writes
* w+ - writes (overwrite / create new) and read

With shift + tab -> check the documentation for a function in a notebook

**29. Resources for More Basic Practice**

[**http://www.pythonchallenge.com/**](http://www.pythonchallenge.com/)

[**https://www.reddit.com/r/dailyprogrammer/new/**](https://www.reddit.com/r/dailyprogrammer/new/)

[**http://www.codeabbey.com/index/task\_list**](http://www.codeabbey.com/index/task_list)

[**https://projecteuler.net/archives**](https://projecteuler.net/archives)

[**https://codingbat.com/python**](https://codingbat.com/python)

**08.02.**

**30. Python Objects and Data Structures Assessment Test Overview**

3.0 == 3 is True

4\*\*0.5 != 2 is False