

Exercises

Python Assessment



Nijmegen, The Netherlands
www.ATComputing.net

Introduction

This assessment serves as a test for the entry level qualification for the Advanced Python course by AT Computing.

The chapters on Object Orientation and on Regular Expressions are not prerequisite knowledge for the Advanced Python course. They have been included here to make this assessment useable to test the knowledge as presented in the “regular” Python course.

The answers to the questions are available as a separate document.

All questions apply to Python 3, unless indicated otherwise.

Exercise 1: Elementary data types

1.a

Explain the difference between a string and a bytes object.

1.b

Name three differences between Python 2 and Python 3

1.c

How to suppress a newline when printing a string?

1.d

What is the maximum size of an integer?

1.e

What is the difference between a string made with double quotes (") versus a string with single quotes (')?

1.f

How is an empty list constructed?

1.g

What is the "to the power of"-operator?

1.h

Is there a difference between `and` and `&`?

1.i

What is a simple way to determine whether a certain string occurs within a larger string?

1.j

Name two mutable basic types.

Exercise 2: Basic statements

2.a

What is the maximum length of a Python statement?

2.b

What is the maximum length of a line of Python code?

2.c

How does the Python `case`-statement look like.

2.d

Which loop constructs are available in Python?

2.e

what is the meaning of the `else`-part in a Python loop?

2.f

Is it possible to generate an exception from your own code? If so, which statement does the job?

Exercise 3: Strings and lists

3.a

Is it possible to add two strings?

3.b

Can a string be added to a number? And the other way around?

3.c

How do you indicate a sub-string?

3.d

What is the result of the `sort()` string method? What happens if the string is sorted already?

3.e

How does one obtain a list of all words from a string?

3.f

What does the `strip()`-method?

3.g

How does classic formatting mark the position for substitution?

3.h

Which method is used in Python's modern formatting?

3.i

How is an element deleted from a list?

3.j

What does the `enumerate()`-function do?

3.k

Give an example of list-comprehension. Using list-comprehension, how would you generate a list with element values [0, 1, 2, 3, 4, 5, 6, 7]?

Exercise 4: Using files

4.a

Is opening a binary file a different operation in Linux vs. Windows?

4.b

How does one read 4 bytes from a binary file?

4.c

How does one read 4 bytes from a text file?

4.d

Python 3 question: does a read operation on a text file return a string or a bytes object?

4.e

Python 3 question: how does one write 3 bytes to stderr?

4.f

Which method changes the file offset?

Exercise 5: Unicode, ASCII, UTF-8 and other encodings

5.a

What are the elements of a string in Python2 and in Python 3?

5.b

How does one open a file with a non-standard encoding?

5.c

What happens if a string containing a Euro-symbol is written to a file which has been opened with ASCII encoding?

5.d

Is a bytes object mutable?

Exercise 6: Dictionaries

6.a

How is an empty dictionary created?

6.b

How is the value that belongs to a certain key obtained?

6.c

What happens if the value of a non-existing key is asked for?

6.d

Is every object suitable as a key in a dictionary?

6.e

Is it possible for a dictionary to return a default value for non-existing keys?

Exercise 7: Tuples

7.a

Is there any difference between tuples and lists?

7.b

How is an empty tuple created?

7.c

How is a tuple containing just one element created?

7.d

How is an extra element added to a tuple?

7.e

How is a tuple converted into a list?

Exercise 8: Sets

8.a

How is an empty set created?

8.b

How can the last element of a set be deleted?

8.c

What is the difference between a set and a frozenset?

8.d

How is the intersection of two sets obtained?

Exercise 9: Functions

9.a

Is a `return`-statement in a function required?

9.b

How does a function definition show that this function expects two integer arguments?

9.c

What is the purpose of a docstring?

9.d

How is the default value for an argument specified.

9.e

How is a variable number of arguments constructed?

9.f

What is the meaning of single `*` and double `**` asterisks in a function definition?

9.g

What do these arguments mean in a function call?

9.h

How is a function passed as an argument to another function?

9.i

What is the purpose of the `map()`-function?

9.j

What does a generator function do?

Exercise 10: Modules

10.a

Will immediate code in a module be executed by an `import` statement?

10.b

Which directories are searched when importing?

Exercise 11: Object orientation

11.a

Which method is called when an object is created?

11.b

What is the difference between a class variable and an instance variable?

11.c

How is it specified that a variable or method is 'private'. Private means: only visible from inside the class.

11.d

Which function is used to find out what the super-class of an object is?

11.e

Can operators like + and - and * specifically be defined for your own classes?

Exercise 12: Exceptions

12.a

Is it possible to catch multiple exceptions with one single `except`-clause?

12.b

Is it possible to pass an argument that is not a string with the `raise`-statement of an exception?

12.c

How can, in a `try`-statement, the exception-object be collected in a variable?

12.d

Is it possible to make an exception class of your own?

Exercise 13: Standard library

13.a

How does one find out which arguments were passed from the command line to the current script?

13.b

What is a clean way to stop a Python script prematurely?

13.c

Which module can be used to handle file name wildcard patterns? (e.g. file names like: *.png)?

Exercise 14: Regular expressions

14.a

Which module handles regular expressions?

14.b

What does `[^abc]` mean?

14.c

What does the `compile()` function do?

14.d

Construct a regular expression that matches (Dutch) postal codes (four digits of which the first one cannot be a 0, optionally followed by a single space, followed by two capital letters).

