ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Python in practice

Lesson 7: Files

Semester 2021/22/1

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Subject matter

1. Introduction
2. Variables, operators
3. Sequence, selection, iteration
4. Programming theses
5. Strings
6. Regular expressions
7. Files
8. Object-oriented programming
9. Multithread applications
10. GUI with tkinter
11. Communications
12. SQL basics
13. SQL queries
14. Example application

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Python in practice

Working With Modules

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Defining Modules

- When your source code becomes too big and complex, you can split it into different parts, aka modules
- Makes the code more maintainable, reuseable, debuggable
- A module is a file, consisting of Python code, that can define functions, classes, and can also contain runnable code
- Example:
  - File name: test.py
  - Module name: test

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Importing Modules

- If you have a module, you have to import it in order to be able to use it. To do this, we use the ‘import’ keyword

- For example, if I have a module named ‘test’, I can import it by referencing it at the beginning of the file:
    - import test
- Now you can use any method or class that is in test.py
    - test.test_function()

- If you don’t want to import the whole file, only some methods or classes, you can use the from … import statement
    - from test import test_function, test_function2
- test_function()

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Importing Modules

- If you want to import the whole module, but still don't want to reference every single part of it by the name of the module, you can use the import * statement
    - from test import *

- You can also rename a module on import, to be more convenient to use
    - import test as t

Writing Modules

- To write your own module, you have to create a new Python file
- If you create functions and classes inside a module in the same way you did in the main file, you can reference them after importing the module

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Python in practice

Working With Files

Handling Files

- To read a text file, first you have to open it
    - f=open(“demo.txt”, “r”)
- “demo.txt” is the name of the file we want to open
- “r” is the mode that we want to open the file in
    - r – reading. If the file doesn’t exist, returns an error
    - a – append. If the file doesn’t exist, creates it
    - w – write. If the file doesn’t exist, creates it
    - x – create. If the file exists, returns an error
- In addition, you can define if the file is text or in binary
    - t – text
    - b – binary

- The default value is read text, so you don’t have to specify them
    - open(“demo.txt”, “r”) is the same as open(“demo.txt”) and as open(“demo.txt”, “rt”)

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Read Files

- To read a whole file, we can use the “read” function
  - f.read()
- Without any parameters, the function will return the whole file’s content
- If we input an integer n, it will return n characters from the file
  - f.read(5)

- To read lines from the file, we can use the “readline” function. Calling it multiple times, we can get the first multiple lines.
  - f.readline()

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Read Files

- We can get every line in the file one-by-one, by looping over the file
  - f = open(“demo.txt”)
  - for x in f:
    - print(x)

- To close a file, we will call “close” on the file
  - f.close()