2. Noções básicas de programação em Python

2.5 - Regular Expressions and File Manipulation

Cristiano Bacelar de Oliveira

Julho, 2020

Universidade Federal do Ceará

Summary

- Regular Expressions
- The re Module
- File Manipulation

Regular Expressions

Regular Expressions

A Regular Expression (or regex) is a coded pattern used for matching strings to rules specified by the programmer.

Regular expressions may be applied in many tasks, such as:

- To split or to modify strings
- To find sentences
- To pre-process text
- To identify numbers or other symbols

Regular Expressions (cont.)

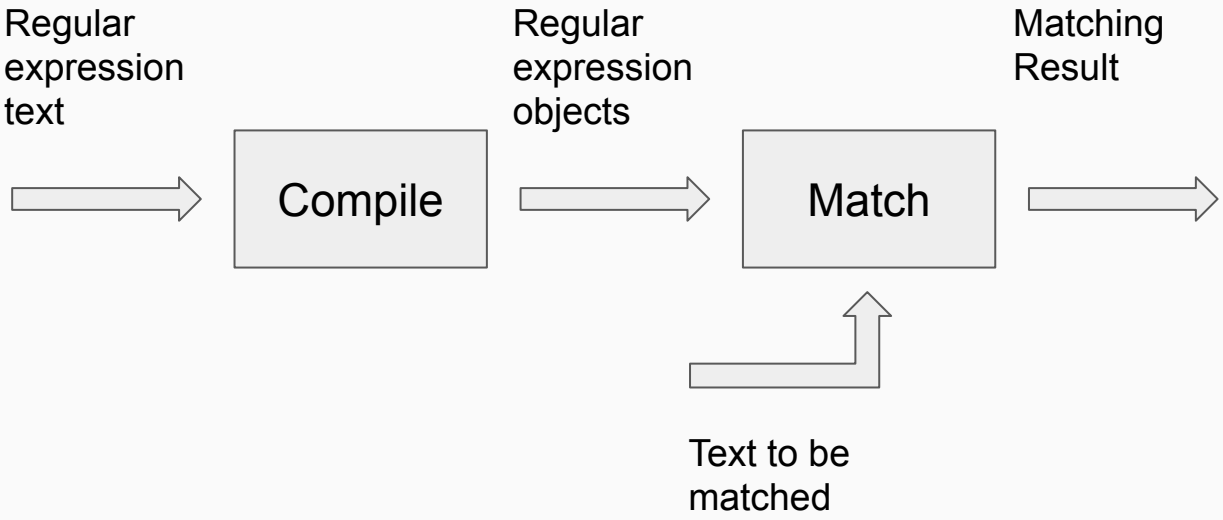
Regular expressions work like a small programming language in the main code.

- For one to use REs is necessary to apply a proper syntax.
  - Use of symbols . ^ $ * + ? { } [ ] \ | ( )
  - Ex. [0-9]
  - Ex. [a-z]
  - Ex. a{5}

Python includes the **re** module in order to make available full regular expression functionality.

The re Module

Matching Process



The re Module

```
import re

text_to_match = ‘hello world!’

pattern = re.compile(r'hello')
match = pattern.match(text_to_match)

if match:
        print(match.group())
```

re Module Functions

The re module provides the following functions[1]:

- *match* → Match a regular expression pattern to the beginning of a string.
- *fullmatch* → Match a regular expression pattern to all of a string.
- *search* → Search a string for the presence of a pattern.
- *sub* → Substitute occurrences of a pattern found in a string.
- *subn* → Same as sub, but also return the number of substitutions made.
- *split* → Split a string by the occurrences of a pattern.
- *findall* → Find all occurrences of a pattern in a string.
- *finditer* → Return an iterator yielding a Match object for each match.
- *compile* → Compile a pattern into a Pattern object.
- *purge* → Clear the regular expression cache.
- *escape* → Backslash all non-alphanumerics in a string.

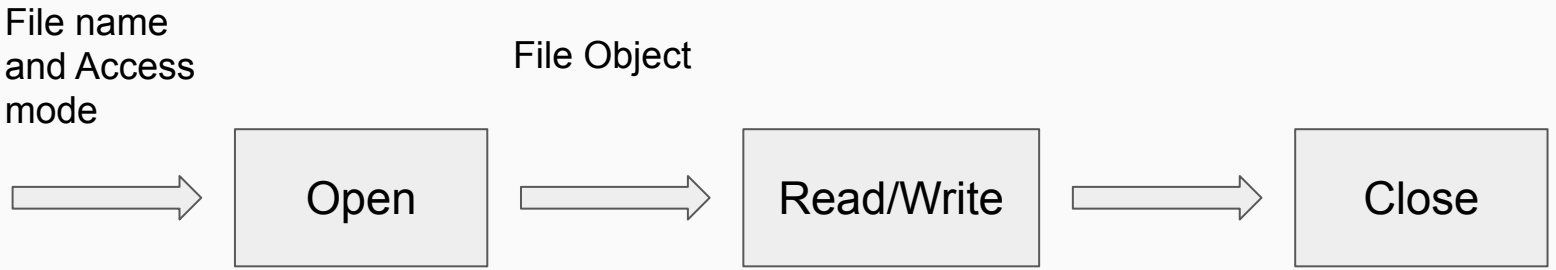
[1] For more details use help(re) in the python console

HUAWEI

DEMO

File Manipulation

File Manipulation



File Access Modes

A file can be opened by using one of the following access modes:

- *'**w**'* → Creates a new file for writing only. Overwrites the file if it already exists.
- *'**r**'* → Reading only.
- '***a***' → Opens an existing file for writing. Works as 'w' if file does not exist.
- '***w+***' → Creates a new file for reading and writing.
- '***r+***' → Opens a file for reading and writing.
- '***a+***' → Opens an existing file for reading and writing.
- '***wb***', '***rb***', '***ab***', '***wb+***', '***rb+***' and '***ab+***' → Same as '***w***', '***r***', '***a***', '***w+***', '***r+***' and '***a+***', respectively, but for binary files.

Examples

File writing

```
f  = open('filename.txt', 'w')
f.write('hello world\n!')
f.close()
```

File reading

```
f  = open('filename.txt', 'r')
lines = f.readlines()
for line in lines:
        print(line)
f.close()
```

Thank You!

Next: 3.1 - Overview on Linear Algebra