Sistemas Informáticos (Computer Systems)

Scripting in Python 03. Activities 01

Authors: Sergi García, Alfredo Oltra

Updated October 2023

Scripting in Python - Part 03

1. Previous information

The objective of this unit is to make calls to the OS, so it is necessary to know the possible actions that the OS can perform. In the next units, we will know many of the commands that the terminal (both Linux/MacOs and Windows) has, but there are some basics that we can already know.

Keep in mind that the same functionality does not have to be done in the same way in Windows and Linux/MacOs so you will indicate the necessary commands to do the activities in both OS. Unless otherwise indicated, you must perform the activity for one or another system.

2. Exercise 01

Create a single program that shows what is the current directory.

In Linux/MacOS you have to use the command "*pwd*".

In Windows you have to use the command cd. This command actually exists on both (Linux/MacOS and Windows) and is used (along with a parameter) to change directories, but on Windows, if used without parameters returns the current directory.

3. Exercise 02

Create a program that displays the contents of the current folder. In Linux/MacOS you have to use the command "*ls*". On Windows you have to use the command "*dir*".

4. Exercise 03

Create a program that displays the contents of the current folder, including hidden files.

The vast majority of OS commands need, or at least support, parameters. These parameters are information that is added to the command to particularize its action. In our case, both ("ls" and "dir") have many of them. You can see all available executing in Linux/MacOS "*man ls*" and on Windows with "*help dir*".

In Linux/MacOS the hidden files start with . , for example "*.file*".

5. Exercise 04

Create a program that creates a folder named "*SIN-Python-Block2*".

In Linux/MacOS you have to use the command "*mkdir*".

On Windows you have to use the command "*md*". ("*mkdir*" is also valid).

6. Exercise 05

The solution to problem 1 has the disadvantage that you have to create different versions depending on the operating system. But Python provides other functions to be able to perform that operation in any OS. Create a new program that works on both platforms.

SISTEMAS INFORMÁTICOS (COMPUTERS SYSTEMS)

7. EXERCISE 06

Repeat the exercise number 4 but creating a program that works on Linux/MacOS and Windows (without using "*mkdir*" command).

8. EXERCISE 07

Creates a program that displays on screen a list of all files in the current directory in green.

In Linux/MacOS you have to use the command "*ls*" with the parameter "*-l*".

On Windows, you have to use the command "*dir*".

The run function returns a special variable called object. Although the explanation of what an object is not trivial, in a simplified way it could be said that it is a variable in which its information is distributed in sections.

In our case, among others, there are 3 sections that interest us:
- **"stdout"**: the information generated by the execution of the command.
- "**stderr**": information on the error, if the execution was not correct.
- "**returncode**": zero in case everything is right, or a number referencing an error.

To access each section you must use a dot. For example, if "*myObject*" is the name of the variable, "*myBobjet.sdterr*" allows access to the sdterr value of the "*myObject*" object.

In order to get run to return the data rightly, it is necessary to indicate as parameters "*universal_newlines=True,stdout=subprocess.PIPE and stderr=subprocess.PIPE*".

9. EXERCISE 08

Create a program that generates the following directory structure using subprocess module:

```
SIN Python
└ Block2
└ Activity1
└ Block3
└ Activity1
```

In Linux/MacOS and Windows, the command "*cd*" allows changing directory.

10. EXERCISE 09

Create a program that generates the directory structure of the exercise 8 that works on Linux/MacOS and Windows platforms.

11. EXERCISE 10

Write a program that requests the user's name, age and number of years of membership in the company. With this information, create a folder with the user's name and, if the sum of the age and years in the company is greater than 35, a sub-folder called "private".