CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

PYTHON.
PART 2. Activities

Computer Systems
CFGS DAW

Vicent Bosch
vicent.bosch@ceedcv.es
2020/2021
Versión:12/1/2020.11:40 AM

Licencia

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

Actividad opcional. Normalmente hace referencia a un contenido que se ha comentado en la documentación por encima o que no se ha hecho, pero es interesante que le alumno investigue y practique. Son tipos de actividades que no entran para examen

Atención. Hace referencia a un tipo de actividad donde los alumnos suelen cometer equivocaciones.

PYTHON - PART 2
Activities

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

```
import subprocess
output = subprocess.check_output('pwd', shell=True)
print(output)
```

(2) Create a program that displays the contents of the current folder.

```
import subprocess
output = subprocess.check_output('ls', shell=True)
print(output)
```

(3) Create a program that displays the contents of the current folder including hidden files.

```
import subprocess
output = subprocess.check_output("ls -a", shell=True)
print(output)
```

(4) Create a program that creates a folder named SIN-Python-Block2.

```
import subprocess
output = subprocess.check_output("mkdir SIN-Python-Block", shell=True)
```

(5) 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.

```
import subprocess
import platform
sistema = platform.system()
if (sistema == 'Linux') | (sistema == 'MacOS'):
    comando = "pwd"
else:
    comando = "dir"
output = subprocess.check_output(comando, shell=True)
print(output)
```

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

```
import subprocess
import platform
sistema = platform.system()
if (sistema == 'Linux') | (sistema == 'MacOS'):
    comando = 'mkdir'
else:
    comando = 'md'
output = subprocess.check_output(comando + " SIN-Python-Block", shell=True)
```

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

```
import subprocess
output = subprocess.run("ls -l", shell=True, universal_newlines=True,
                        stdout=subprocess.PIPE, stderr=subprocess.PIPE, )
print(chr(27) + "[0;32m" + output.stdout)
```

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

SIN Python

- Block2
  - Activity1
- Block3
  - Activity1

```
import subprocess
comando = ["mkdir SIN_Python", "mkdir SIN_Python/Block2",
           "mkdir SIN_Python/Block3", "mkdir SIN_Python/Block2/Activity1",
           "mkdir SIN_Python/Block3/Activity1"]
for i in range(5):
    output = subprocess.run(comando[i], shell=True, universal_new-
lines=True,
                            stdout=subprocess.PIPE, stderr=subprocess.PIPE,
)
```

(9) Create a program that generates the directory structure of the exercise8 that works on Linux/MacOS and Windows platforms.

```
import subprocess
import platform
sistema = platform.system()
if (sistema == 'Linux') | (sistema == 'MacOS'):
    comando = "mkdir "
else:
    comando = "md "
argumento = ["SIN_Python", "SIN_Python/Block2", "SIN_Python/Block3",
             "SIN_Python/Block2/Activity1", "SIN_Python/Block3/Activity1"]
for i in range(5):
    output = subprocess.run(comando + argumento[i], shell=True,
                            universal_newlines=True, stdout=subprocess.PIPE,
                            stderr=subprocess.PIPE, )
```