

Actividad II Programación Orientada a Objetos con Python

Responde a los todos los Quizes hasta obtener el 100% de las respuestas correctas, no se te olvide tomar como evidencia imagen de perfil de usuario

- PE2 Módulo 1 Quiz

Python INSTITUTE
Examén Módulo 1

Progress (100%)

Your score: 18/18

100%

Congratulations, you've passed the test!

SECTION ANALYSIS

| Section | Score |
|----------------------|-------|
| PE2 -- Module 1 Test | 100% |

[Retake Test](#) [Review Test](#)

Cristopher

- My Profile
- Certificates/Letters
- Discounts
- Badges
- Log Out

| Criteria | Status |
|---------------------|--------|
| Completion status | No |
| Yet started | No |
| Below are required: | 0 of 2 |

- PE2 Módulo 2 Quiz

Python INSTITUTE
Examén Módulo 2

Progress (100%)

Your score: 15/15

100%

Congratulations, you've passed the test!

SECTION ANALYSIS

| Section | Score |
|----------------------|-------|
| PE2 -- Module 2 Test | 100% |

[Retake Test](#) [Review Test](#)

Cristopher

- My Profile
- Certificates/Letters
- Discounts
- Badges
- Log Out

| Criteria | Status |
|---------------------|--------|
| Completion status | No |
| Yet started | No |
| Below are required: | 0 of 2 |

- PE2 Módulo 3 Quiz

Python INSTITUTE
Examén Módulo 3

Progress (100%)

Your score: 16/16

100%

Congratulations, you've passed the test!

SECTION ANALYSIS

| Section | Score |
|----------------------|-------|
| PE2 -- Module 3 Test | 100% |

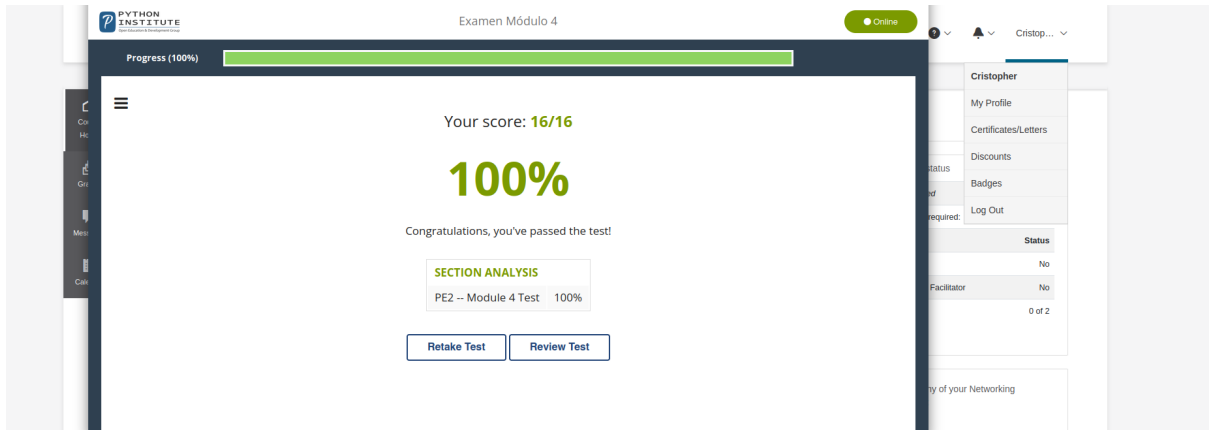
[Retake Test](#) [Review Test](#)

Cristopher

- My Profile
- Certificates/Letters
- Discounts
- Badges
- Log Out


| Criteria | Status |
|---------------------|--------|
| Completion status | No |
| Yet started | No |
| Below are required: | 0 of 2 |

- PE2 Módulo 4 Quiz



Realiza los laboratorios del módulo y documentalos

2.3.1.18


PYTHON
INSTITUTE
AN ASSOCIATION OF UNIVERSITIES

MODULE (50%)

SECTION (100%)

2.3.1.18 Tu propio split »

20-25 minutos

Nivel de dificultad

Medio

Objetivos

- Mejorar las habilidades de programación.
- Utilizar los métodos de Python.

Escenario

Ya sabes como funciona el método `split()`.

Tu tarea es **escribir tu propia función** que devuelva una lista de palabras creadas a partir de la cadena, dividida en los lugares donde la cadena contiene espacios en blanco.

- Debe aceptar únicamente un argumento: una cadena.
- Debe devolver una lista de palabras creadas a partir de la cadena, dividida en los lugares donde la cadena contiene espacios en blanco.
- Si la cadena está vacía, la función debería devolver una lista vacía.
- Su nombre debe ser `mysplit()`.

Utiliza la plantilla en el editor. Prueba tu código con cuidado.

Salida esperada


```
[
  'Ser', 'o', 'no', 'ser', 'esa', 'es', 'la', 'pregunta']
[
  'Ser', 'o', 'no', 'ser,esa', 'es', 'la', 'pregunta']
[
  ]
[
  'abc' ]
[
  ]
```

1


```
"""
2      Elaborado por: Christopher López Jiménez
3      Descripción: Laboratorios R1
4      """
5      def mysplit(strng):
6          return strng.strip().split()
7
8      print(mysplit("Ser o no ser, esa es la pregunta"))
9      print(mysplit("Ser o no ser,esa es la pregunta"))
10     print(mysplit(" "))
11     print(mysplit(" abc "))
12     print(mysplit(""))
13
```

Console >_


```
[
  'Ser', 'o', 'no', 'ser', 'esa', 'es', 'la', 'pregunta']
[
  'Ser', 'o', 'no', 'ser,esa', 'es', 'la', 'pregunta']
[
  ]
[
  'abc' ]
[
  ]
```

2.4.1.6

« 2.4.1.6 LABORATORIO: Un Display LED »

SECTION (100%)

My Profile | Networking Academy — Mozilla Firefox

My Profile | Networking Academy

Cristopher López Jiménez

Networking Academy ID: 1041497772

alumno al definir funciones.
proporcionar un entorno de entrada más seguro.

```
1 """
2     Elaborado por:  Cristopher López Jiménez
3     Descripción:    Laboratorios R1
4 """
5 dict1 = {
6     '0': ('###', '# #', '# #', '# #', '###'),
7     '1': ('# #', '# #', '# #', '# #', '# #'),
8     '2': ('###', '# #', '###', '# #', '###'),
9     '3': ('###', '# #', '###', '# #', '###'),
10    '4': ('# #', '# #', '###', '# #', '# #'),
11    '5': ('###', '# #', '###', '# #', '###'),
12    '6': ('###', '# #', '###', '# #', '###'),
13    '7': ('###', '# #', '# #', '# #', '# #'),
14    '8': ('###', '# #', '###', '# #', '###'),
15    '9': ('###', '# #', '###', '# #', '###'),
16    '.': ('.', '# #', '# #', '# #', '# #'),
17 }
18 number = input("Ingresa un numero: ")
19 def seven_segment(number):
20     for i in range(5):
21         digits = [dict1[digit] for digit in str(number)]
22         print(" ".join(segment[i] for segment in digits))
23 seven_segment(number)
```

Console >_

Ingresa un numero: 123

###

#

###

#

###

2.8.1.4

« 2.8.1.4 Leer enteros de forma segura »

SECTION (67%)

My Profile | Networking Academy — Mozilla Firefox

My Profile | Networking Academy

Cristopher López Jiménez

Networking Academy ID: 1041497772

alumno al definir funciones.
proporcionar un entorno de entrada más seguro.

```
1 """
2     Elaborado por:  Cristopher López Jiménez
3     Descripción:    Laboratorios R1
4 """
5 def read_int(prompt, min, max):
6     while True:
7         try:
8             response = int(input(prompt))
9             if min <= response <= max:
10                return response
11         except:
12             print("Error: el valor no esta dentro del rango " + str(min) + "..." + str(max))
13             print("Error, entrada equivocada")
14
15 v = read_int("Ingresa un numero de -10 al 10: ", -10, 10)
16
17 print("El numero es:", v)
```

3.2.1.14

« 3.2.1.14 Pila Contadora »

SECTION (88%)

Elaborado por: Cristopher López Jiménez

Descripción: Laboratorios R1

```
1 """
2
3
4 """
5 class Stack:
6     def __init__(self):
7         self.__stk = []
8
9
10
11
12     def push(self, val):
13         self.__stk.append(val)
14
15
16
17     def pop(self):
18         val = self.__stk[-1]
19         del self.__stk[-1]
20         return val
21
22
23
24
25 class CountingStack(Stack):
26     def __init__(self):
27         # Llena el constructor con acciones apropiadas.
28         Stack.__init__(self)
29         self.__counter = 0
30
31
32     def get_counter(self):
33         # Presenta el valor actual del contador al mundo.
34         return self.__counter
35
36
37
38
39     def pop(self):
40         # Haz un pop y actualiza el contador.
```

Console >_

100

My Profile | Networking Academy — Mozilla Firefox

My Profile | Networking Academy

Cristopher López Jiménez

Networking Academy ID: 1041497772

Elaborado por: Cristopher López Jiménez

Descripción: Laboratorios R1

class Stack:

def __init__(self):

self.__stk = []

def push(self, val):

self.__stk.append(val)

def pop(self):

val = self.__stk[-1]

del self.__stk[-1]

return val

class CountingStack(Stack):

def __init__(self):

Llena el constructor con acciones apropiadas.

Stack.__init__(self)

self.__counter = 0

def get_counter(self):

Presenta el valor actual del contador al mundo.

return self.__counter

def pop(self):

Haz un pop y actualiza el contador.

Console >_

100

3.2.1.15

« 3.2.1.15 Colas alias FIFO »

SECTION (94%)

Elaborado por: Cristopher López Jiménez

Descripción: Laboratorios R1

```
1 """
2
3
4 """
5
6 class QueueError(IndexError):
7     pass
8
9 class Queue:
10     def __init__(self):
11         self.queue = []
12
13     def put(self, elem):
14         self.queue.insert(0, elem)
15
16     def get(self):
17         if len(self.queue) > 0:
18             elem = self.queue[-1]
19             del self.queue[-1]
20             return elem
21         else:
22             raise QueueError
23
24 que = Queue()
25 que.put(1)
26 que.put("perro")
27 que.put(False)
28 try:
29     for i in range(4):
30         print(que.get())
31 except:
32     print("Error de cola")
```

Console >_

1
perro
False
Error de cola

My Profile | Networking Academy — Mozilla Firefox

My Profile | Networking Academy

Cristopher López Jiménez

Networking Academy ID: 1041497772

Elaborado por: Cristopher López Jiménez

Descripción: Laboratorios R1

class QueueError(IndexError):

pass

class Queue:

def __init__(self):

self.queue = []

def put(self, elem):

self.queue.insert(0, elem)

def get(self):

if len(self.queue) > 0:

elem = self.queue[-1]

del self.queue[-1]

return elem

else:

raise QueueError

que = Queue()

que.put(1)

que.put("perro")

que.put(False)

try:

for i in range(4):

print(que.get())

except:

print("Error de cola")

Console >_

1

perro

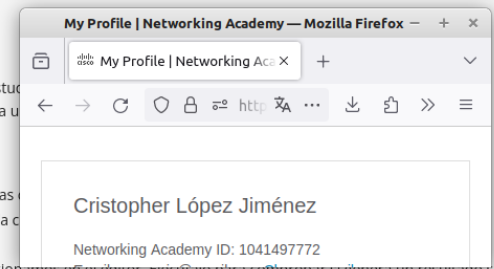
False

Error de cola

3.2.1.16

« 3.2.1.16 Colas alias FIFO: parte 2 »

SECTION (100%)



```
1 """
2     Elaborado por:  Christopher López Jiménez
3     Descripción:    Laboratorios R1
4 """
5 class QueueError(IndexError):
6     pass
7
8
9 class Queue:
10     def __init__(self):
11         self.queue = []
12     def put(self, elem):
13         self.queue.insert(0, elem)
14     def get(self):
15         if len(self.queue) > 0:
16             elem = self.queue[-1]
17             del self.queue[-1]
18             return elem
19         else:
20             raise QueueError
21
22 class SuperQueue(Queue):
23     def isempty(self):
24         return len(self.queue) == 0
25
26 que = SuperQueue()
27 que.put(1)
28 que.put("perro")
29 que.put(False)
30 for i in range(4):
31     pass
```

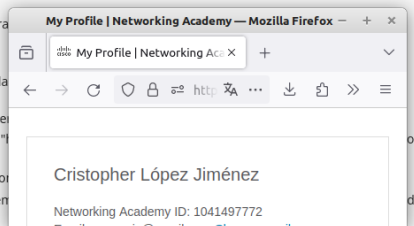
Console >_

Warning: The pty module has been disabled for security reasons.

3.4.1.12

« 3.4.1.12 La clase Timer »

SECTION (80%)



```
1 """
2     Elaborado por:  Christopher López Jiménez
3     Descripción:    Laboratorios R1
4 """
5
6 class Timer:
7     def __init__( self, a,b,c ):
8         self.num = a*60*60 + b*60 + c
9
10     def __str__(self):
11         return str(int(self.num/(60*60)))+":"+str(int(self.num/60%60))+":"+str(self.num%60)
12
13     def next_second(self):
14         self.num = self.num + 1
15         if(self.num >= 60*60*24):
16             self.num = 0
17
18     def prev_second(self):
19         self.num = self.num - 1
20         if(self.num < 0):
21             self.num = 60*60*24 - 1
22
23 timer = Timer(23, 59, 59)
24 print(timer)
25 timer.next_second()
26 print(timer)
27 timer.prev_second()
28 print(timer)
29
```

Console >_

23:59:59
0:0:0
23:59:59

3.1.4.13

« 3.4.1.13 Días de la semana »

SECTION (87%)

ase llamada `Weeker`. Si, tus ojos no te engañan, este nombre proviene del hecho
: podrán almacenar y manipular los días de la semana.

a un a
:ptabl

Dom

rgume
mo; n
porci

en se
a que

da con métodos de un parámetro llamados `add_days(n)` y
do `n` un número entero que actualiza el día de la semana almacenado dentro
nero de días indicado, hacia adelante o hacia atrás.
objeto deben ser privadas.

porcionamos en el editor, ejecuta su código y verifica si tu salida se ve igual que la

der tu solicitud.

```
7
8
9- class Weeker:
10     __names = ['Lun', 'Mar', 'Mie', 'Jue', 'Vie', 'Sab', 'Dom']
11
12     def __init__(self, day):
13         try:
14             self.__current = Weeker.__names.index(day)
15         except ValueError:
16             raise WeekDayError
17
18     def __str__(self):
19         return Weeker.__names[self.__current]
20
21     def add_days(self, n):
22         self.__current = (self.__current + n) % 7
23
24     def subtract_days(self, n):
25         self.__current = (self.__current - n) % 7
26
27
28- try:
29     weekday = Weeker('Lun')
30     print(weekday)
31     weekday.add_days(15)
32     print(weekday)
33     weekday.subtract_days(23)
34     print(weekday)
35     weekday = Weeker('Lun')
36- except WeekDayError:
37     print('Lo siento, no puedo atender tu solicitud.')
38
```

```
Console >_
Lun
Mar
Dom
```

3.4.1.14

« 3.4.1.14 Puntos en un plano »

SECTION (93%)

que `__repr__` y `__str__` se llaman `__repr__` y `__str__`.

puede describirse como un par de coordenadas habitualmente llamadas `x` y `y`.
e en Python que almacene ambas coordenadas como números flotantes.
os de esta clase evalúen las distancias entre cualquiera de los dos puntos

bas l
sa de
'docs

rgun
n se
los s

ordenadas se almacenan de forma privada, por lo que no se puede acceder a
objeto).

todo llamado `distance_from_xy(x, y)`, que calcula y devuelve la distancia
dentro del objeto y el otro punto dado en un par de números flotantes.
todo llamado `distance_from_point(point)`, que calcula la distancia (como
ubicación del otro punto se da como otro objeto de clase `Point`.

orcionamos en el editor, ejecuta tu código y verifica si tu salida se ve igual que la

```
1
2
3     Elaborado por: Christopher López Jiménez
4     Descripción: Laboratorios R1
5
6
7
8- class Point:
9     def __init__(self, x=0.0, y=0.0):
10         self.x = x
11         self.y = y
12
13     def getx(self):
14         return self.x
15
16     def gety(self):
17         return self.y
18
19     def distance_from_xy(self, x, y):
20         return math.sqrt((self.y-y)**2+(self.x-x)**2)
21
22     def distance_from_point(self, point):
23         return math.sqrt((self.y-point.y)**2+(self.x-point.x)**2)
24
25
26 point1 = Point(0, 0)
27 point2 = Point(1, 1)
28 print(point1.distance_from_point(point2))
29 print(point2.distance_from_xy(2, 0))
```

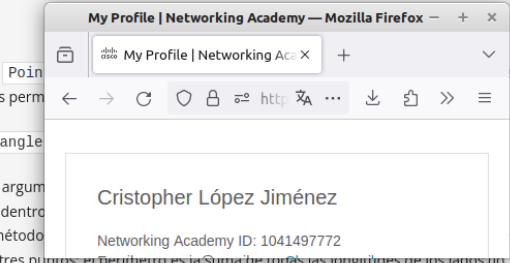
```
Console >_
1.4142135623730951
1.4142135623730951
```

3.4.1.15

« 3.4.1.15 Triángulo »

SECTION (100%)

el estudiante para definir clases desde cero.



el código de la clase `Point`, el cual se utilizó en el laboratorio anterior:

```
1 """
2     Elaborado por:  Cristopher López Jiménez
3     Descripción:   Laboratorios R1
4 """
5 import math
6
7 class Point:
8     def __init__(self, x=0.0, y=0.0):
9         self.x = x
10        self.y = y
11
12    def getx(self):
13        return self.x
14
15    def gety(self):
16        return self.y
17
18    def distance_from_xy(self, x, y):
19        return math.sqrt((self.y-y)**2+(self.x-x)**2)
20
21    def distance_from_point(self, point):
22        return math.sqrt((self.y-point.y)**2+(self.x-point.x)**2)
23
24 class Triangle:
25     def __init__(self, vertice1, vertice2, vertice3):
26         self.a = vertice1
27         self.b = vertice2
28         self.c = vertice3
29     def perimeter(self):
30         return (
31             self.a.distance_from_point(self.b) +
32             self.b.distance_from_point(self.c) +
```

Console >_

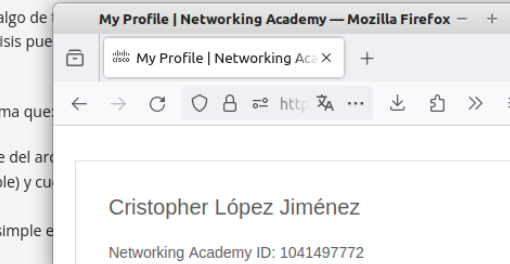
3.414213562373095

4.3.1.15

« 4.3.1.15 LABORATORIO: Histograma de frecuencia de caracteres »

SECTION (83%)

el estudiante al operar con la lectura archivos.
atos para contar datos numerosos.



ra tu código y verifica si tu histograma contiene resultados validos.

prueba contiene solo una línea con:

samplefile.txt

verse de la siguiente manera:

salida

```
1 """
2     Elaborado por:  Cristopher López Jiménez
3     Descripción:   Laboratorios R1
4 """
5
6 from os import strerror
7
8 filename = input("¿Cual es el nombre del archivo? ")
9
10 try:
11     file = open(filename, 'r')
12 except IOError as e:
13     print("No se puede abrir.", strerror(e.errno))
14     exit()
15
16 store = {}
17 count = 0
18
19 read = file.read()
20 for char in read:
21     if char.isalpha() == True:
22         if char not in store.keys():
23             count = 1
24             store.update({char.lower(): count})
25         else:
26             count += 1
27             store.update({char.lower(): count})
28
29 clean_name = filename.split('.')[0]
30 output = open(clean_name + '.hist', 'w+t')
31
32
```

Console >_

¿Cual es el nombre del archivo? aBc
No se puede abrir. No such file or directory

4.3.1.16

« 4.3.1.16 LABORATORIO: Histograma de frecuencia de caracteres ordenado »

SECTION (89%)

Elaborado por: Cristopher López Jiménez
Descripción: Laboratorios R1

```
from collections import defaultdict

def generar_histograma(archivo):
    recuento_letras = defaultdict(int)

    try:
        with open(archivo, 'r') as file:
            contenido = file.read()

        for caracter in contenido:
            if caracter.isalpha():
                letra = caracter.lower()
                recuento_letras[letra] += 1

        ordenado = sorted(recuento_letras.items(), key=lambda x: x[1], reverse=True)

        for letra, recuento in ordenado:
            print(f"{letra} -> {recuento}")

        nombre_archivo_salida = archivo.split('.')[0] + '.hist'
        with open(nombre_archivo_salida, 'w') as output_file:
            for letra, recuento in ordenado:
                output_file.write(f"{letra} -> {recuento}\n")

        print(f"Histograma guardado en {nombre_archivo_salida}")
```

Console >_

```
a -> 3
b -> 2
c -> 1
Histograma guardado en samplefile.hist
```

My Profile | Networking Academy — Mozilla Firefox

My Profile | Networking Academy

Cristopher López Jiménez

Networking Academy ID: 1041497772
Email: azzerman@gmail.com [Change email](#)

samplefile.txt

salida

4.3.1.17

« 4.3.1.17 LABORATORIO: Evaluando los resultados de los estudiantes »

SECTION (94%)

Elaborado por: Cristopher López Jiménez
Descripción: Laboratorios R1

```
class StudentsDataException(Exception):
    pass

class WrongLine(StudentsDataException):
    def __init__(self, line):
        super().__init__()
        self.line = line

class FileEmpty(StudentsDataException):
    pass

class WrongLine(Exception):
    def __init__(self, line):
        self.line = line

class FileEmpty(Exception):
    pass

def main():
    try:
        filename = input("Introduce el nombre del archivo: ")
        students = {}
        with open(filename, "r") as f:
            for line in f:
                line = line.strip()
                if len(line.split()) != 3:
```

Console >_

```
Warning: The pty module has been disabled for security reasons.
```

My Profile | Networking Academy — Mozilla Firefox

My Profile | Networking Academy

Cristopher López Jiménez

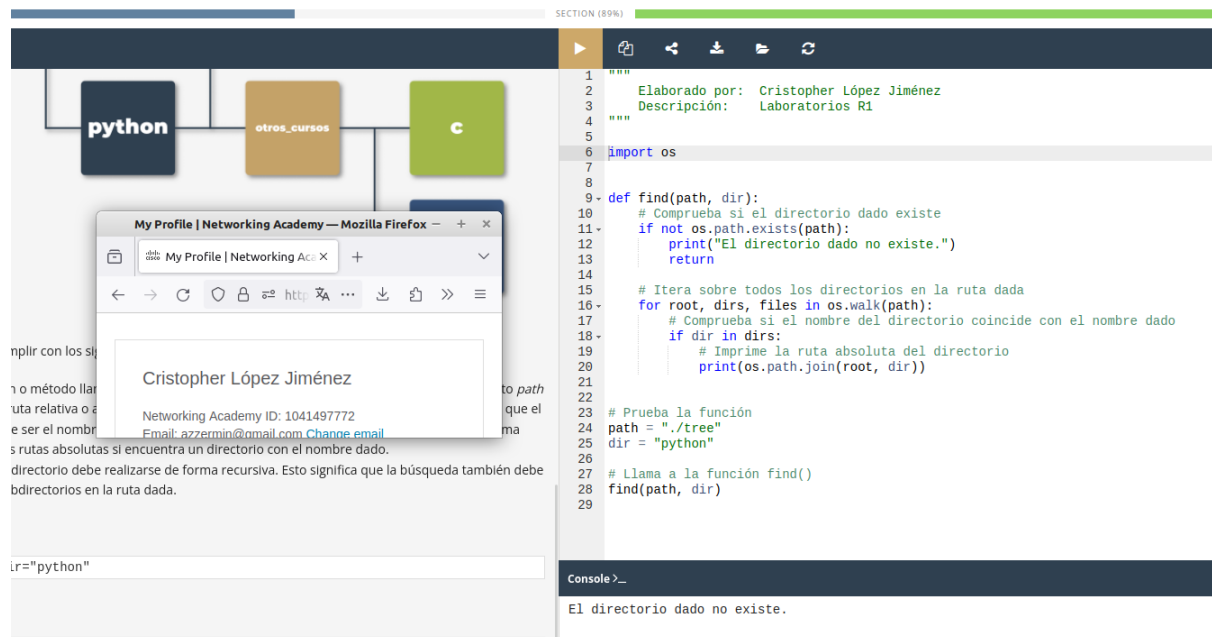
Networking Academy ID: 1041497772
Email: azzerman@gmail.com [Change email](#)

samplefile.txt

salida

4.4.1.8

« 4.4.1.8 LABORATORIO: El módulo os »



The screenshot shows a Jupyter Notebook interface. On the left, a file explorer displays a directory structure with folders named 'python', 'otros_cursos', and 'C'. Below it, a browser window shows a profile page for 'Cristopher López Jiménez' with details like 'Networking Academy ID: 1041497772' and 'Email: azzermi@gmail.com'. The main area contains a code editor with Python code using the 'os' module to check if a directory exists. The console output shows the message 'El directorio dado no existe.'.

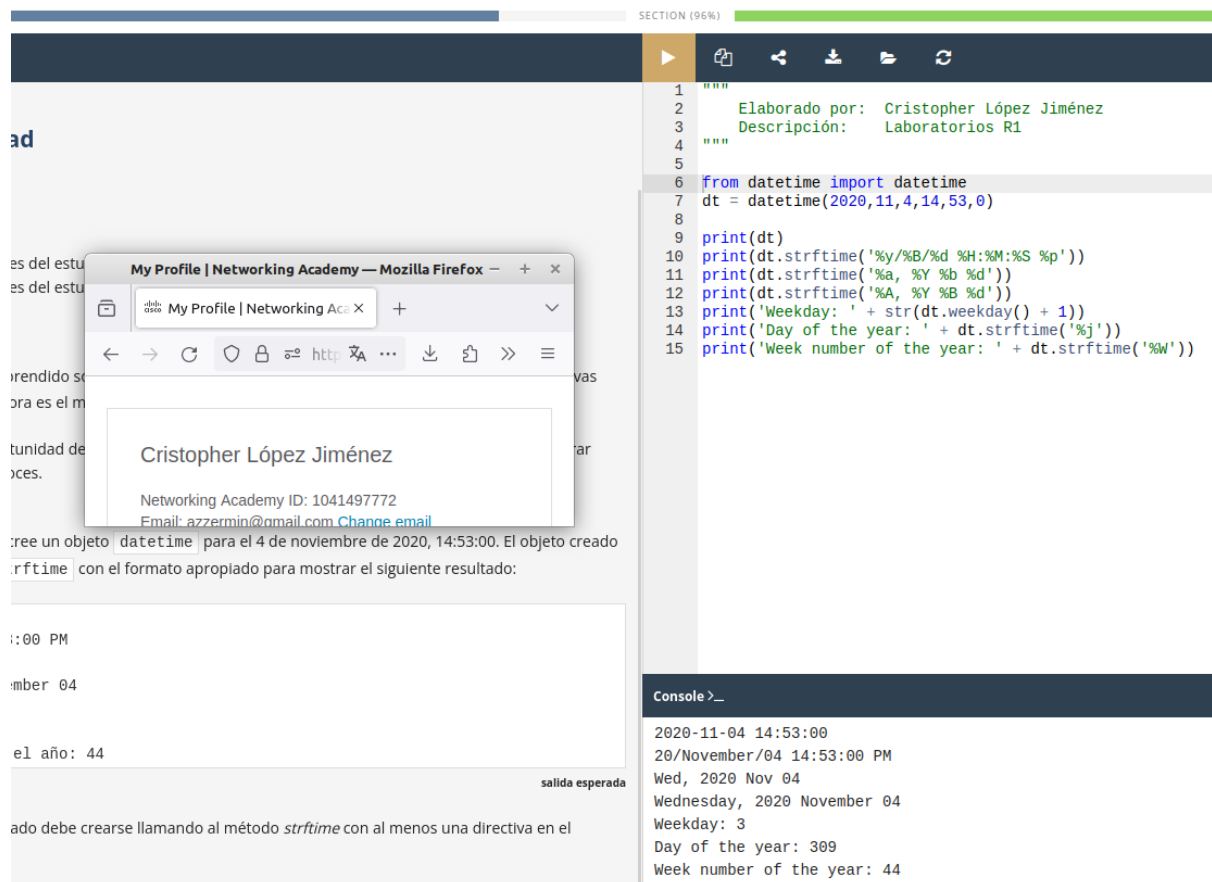
```
1 """
2     Elaborado por: Cristopher López Jiménez
3     Descripción: Laboratorios R1
4 """
5
6 import os
7
8
9 def find(path, dir):
10     # Comprueba si el directorio dado existe
11     if not os.path.exists(path):
12         print("El directorio dado no existe.")
13     return
14
15     # Itera sobre todos los directorios en la ruta dada
16     for root, dirs, files in os.walk(path):
17         # Comprueba si el nombre del directorio coincide con el nombre dado
18         if dir in dirs:
19             # Imprime la ruta absoluta del directorio
20             print(os.path.join(root, dir))
21
22
23 # Prueba la función
24 path = "./tree"
25 dir = "python"
26
27 # Llama a la función find()
28 find(path, dir)
```

Console >_

El directorio dado no existe.

4.5.1.22

« 4.5.1.22 LABORATORIO: Los módulos datetime y time »



The screenshot shows a Jupyter Notebook interface. On the left, a file explorer displays a directory structure. Below it, a browser window shows a profile page for 'Cristopher López Jiménez' with details like 'Networking Academy ID: 1041497772' and 'Email: azzermi@gmail.com'. The main area contains a code editor with Python code using the 'datetime' module to create a datetime object and print its details. The console output shows the date and time '2020-11-04 14:53:00' and other details like 'Wednesday, 2020 November 04'.

```
1 """
2     Elaborado por: Cristopher López Jiménez
3     Descripción: Laboratorios R1
4 """
5
6 from datetime import datetime
7 dt = datetime(2020,11,4,14,53,0)
8
9
10 print(dt)
11 print(dt.strftime('%y/%B/%d %H:%M:%S %p'))
12 print(dt.strftime('%a, %Y %b %d'))
13 print(dt.strftime('%A, %Y %B %d'))
14 print('Weekday: ' + str(dt.weekday() + 1))
15 print('Day of the year: ' + dt.strftime('%j'))
16 print('Week number of the year: ' + dt.strftime('%W'))
```

Console >_

2020-11-04 14:53:00
20/November/04 14:53:00 PM
Wed, 2020 Nov 04
Wednesday, 2020 November 04
Weekday: 3
Day of the year: 309
Week number of the year: 44

4.6.1.13

« 4.6.1.13 LABORATORIO: El módulo calendar »

Un breve vistazo a la clase `Calendar`. Tu tarea ahora es ampliar su funcionalidad do `count_weekday_in_year`, que toma un año y un día de la semana como e el número de ocurrencias de un día de la semana específico en el año.

is:

MyCalendar

count_weekday_in_year

entre 0 y 6,

ro enteros

usa el m

is espera

```
1 """
2     Elaborado por:  Christopher López Jiménez
3     Descripción:   Laboratorios R1
4 """
5
6 import calendar
7
8 class Calendar:
9     pass
10
11 class MyCalendar(Calendar):
12     def __init__(self):
13         Calendar.__init__(self)
14
15     def __str__(self, result):
16         print(str(result))
17
18     def count_weekday_in_year(self, year, weekday):
19         y = year
20         wd = weekday
21         cal = calendar.Calendar()
22         weekday_counter = 0
23
24         for month in range(12):
25             month += 1
26             for week in cal.monthdays2calendar(y, month):
27                 for day in week:
28                     if day[1] == wd and day[0] != 0:
29                         weekday_counter += 1
30
31         self.__str__(weekday_counter)
32
```

Console >_

53