

Prueba Módulo 2

Puntos totales 16/16 ?

Empecemos por el principio...

0 de 0 puntos

Nombre y apellidos *

Jose Antonio Cerván García

Preguntas

16 de 16 puntos

Responde marcando la opción correcta, escribiendo texto o marcando más de una casilla según corresponda

✓ ¿Qué instrucción nos permite tener esta salida?

1/1

```
1  a = {1,2,3}
2  b = {4,5,6}
3  |
{1, 2, 3, 4, 5, 6}
```

- ☐ a.append(b)
- ☐ a.append(b) tres veces seguidas
- ☒ a.union(b)
- ☐ b.union(a)



- ✓ Completa el código con una instrucción más para obtener la salida que se en la imagen 1/1

```
1 class Dog:
2     def woof(self):
3         return 'woof!'
4
5 t = Dog()
```

woof!

t.woof()



- ✓ ¿Cuál es la instrucción que debe ir en el hueco en este script? 1/1

```
1 ____ open('hello.txt', 'w') as file:
2     file.write("hello!")
3
4 print(file.closed)
```

True

- ☐ import
- ☒ with
- ☐ in
- ☐ load



- ✓ ¿Qué debe haber en el hueco del código para que su ejecución nos de la 1/1 salida de la imagen?

```
1  class Dog:
2      def __init__(self):
3          pass
4
5      def bark(self):
6          return "bark bark bark bark bark bark..."
7
8  d = Dog()
9  d._____()
```

bark bark bark bark bark bark...

- ☐ self
- ☐ Dog.bark
- ☒ bark
- ☐ Ninguna es correcta



- ✓ Dadas las dos funciones decoradoras del script, marca las opciones que sean correctas 1/1

```
1  @square_it
2  @document_it
3  def add_ints(a, b):
4      return a + b
5
6  add_ints(3,5)
```

- ☒ Se aplicará primero la funcionalidad de document_it sobre add_ints y después, sobre el resultado, se aplicará square_it ✓
- ☐ Se aplicará primero sobre add_ints la funcionalidad de square_it, que ha sido llamada primero, y luego se aplicará document_it sobre el resultado
- ☐ A las decoradoras les faltan los paréntesis. Deberían ser @square_it() y @document_it()
- ☒ Las decoradoras tienen, como argumento, la función add_ints ✓



✓ ¿Qué es necesario poner en el hueco del código para obtener esa salida? 1/1

```
1 packages = ["numpy","pandas","scipy"]
2
3 for i in packages:
4     print(__)
```

```
numpy
pandas
scipy
```

- ☒ i ✓
- ☐ packages
- ☐ "i"
- ☐ "numpy","pandas","scipy"

✓ Completa el código utilizando la instrucción correcta en el hueco

1/1

```
1 w = 'python'
2
3 w_iterator = iter(w)
4
5 _____(w_iterator)
```

p

next



✓ ¿Qué parámetro es necesario poner en la definición de la función?

1/1

```
1 def make_dict(____):  
2     return kwargs  
3  
4 make_dict(a = 1, b = 2)  
  
{'a': 1, 'b': 2}
```

- ☐ **args
- ☐ a = x, b = y
- ☐ *kwargs
- ☒ **kwargs



✓ Escribe la instrucción que permite eliminar el item correspondiente a 'format' en el diccionario book

1/1

```
1 book = {  
2     'title': 'The Giver',  
3     'author': 'Lois Lowry',  
4     'rating': 4.13,  
5     'format': 'paperback'  
6 }  
7  
8 _____  
9  
10  
11 print(book)  
  
{'title': 'The Giver', 'author': 'Lois Lowry', 'rating': 4.13}
```

book.pop('format')



✓ Completa el código para obtener la salida mostrada en la imagen

1/1

```
1  def add_many(*args):  
2      s = 0  
3      for _____:  
4          s += n  
5      print(s)  
6  
7  add_many(100, 50, 3)
```

153

n in args



✓ Dado el script de la imagen, marca todas las afirmaciones correctas

1/1

```
1  import sqlite3
2  from sqlite3 import Error
3
4  def create_connection(path):
5      connection = None
6      try:
7          connection = sqlite3.connect(path)
8          print(f"Conexión a {path} realizada")
9      except Error as e:
10         print(f"Error '{e}'")
11     return connection
12
13 connection = create_connection("curso.db")
```

- ☐ El script establece la conexión a una base de datos curso.db que debe haber sido creada previamente. Si la base de datos no existe, el script dará error
- ☒ El bloque except se ejecuta solo si el código del bloque try no se puede ejecutar por algún error ✓
- ☐ Tanto al crear una base de datos con SQLite como con MySQL es necesario conectar primero con un servidor
- ☒ Para poder trabajar con la base de datos, será necesario crear un objeto cursor con connection.cursor() ✓



✓ Escribe las palabras que faltan, separadas por una coma y sin espacios 1/1

```
1 l = [i * 2 __ i __ range(5)]
2 print(l)
```



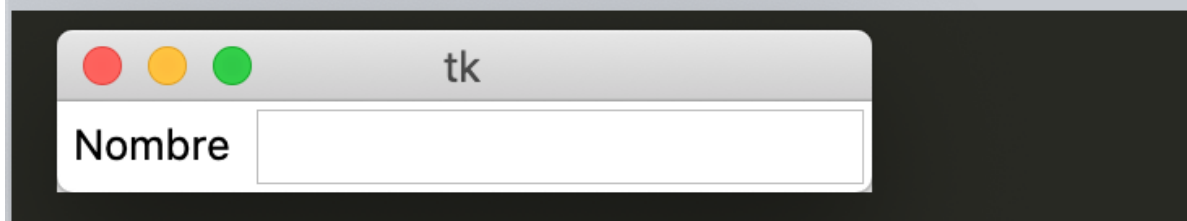
```
[0, 2, 4, 6, 8]
```

for,in



✓ Escribe la palabra que debe ir en los dos huecos de la imagen para que se genere la interfaz mostrada 1/1

```
1 import tkinter as tk
2 root = tk.Tk()
3 label = tk.Label(root, text="Nombre")
4 label.___(side="left")
5 entry = tk.Entry(root)
6 entry.___(side="right")
7 root.mainloop()
```



pack



✓ ¿Qué debería ir en el hueco de esta expresión lambda?

1/1

```
1 suma = lambda _____: x + n
2
3 print(suma(2))
```

5

- ☐ x, n
- ☐ x=3, n
- ☒ x, n=3
- ☐ n=3



✓ ¿Escribe lo que debería ir en el hueco para que el código se ejecute correctamente

1/1

```
1 class Planet:
2     def __init__(self, name):
3         self.name = name
4
5 m = Planet(_____)
6
7 print(m.name)
```

mercury

'mercury'



✓ ¿Que ocurre cuando utilizas la función integrada any() sobre una lista? 1/1

- ☐ La función any() devuelve un valor Booleano que responde a la pregunta "¿Hay algún elemento en esta lista?"
- ☒ La función any() devuelve True si cualquier elemento de la lista es evaluado como True. Si no, devuelve False ✓
- ☐ La función any() devuelve un item al azar de la lista
- ☐ La función any() toma como argumentos una lista y un item. Si alguno de los items de la lista coincide con el item pasado como argumento, la función devuelve True

Pruebas técnicas

0 de 0 puntos

Ahora te toca programar y darlo todo. La puntuación de las pruebas técnicas se publicará tras su revisión



Escribe una función que admita dos cadenas y devuelva True si las dos cadenas son anagramas

Copia aquí el código además de enviarlo por email a carmenbvg@gmail.com

```
>>> is_anagram("tea", "eat")
True
>>> is_anagram("tea", "treat")
False
>>> is_anagram("sinks", "skin")
False
>>> is_anagram("Listen", "silent")
True
```

```
def is_anagram(s1, s2):
    """
    :param s1: string 1
    :param s2: string 2
    :return: boolean value (are they anagrams?)
    """
    return ".join(sorted(s1.lower())) == ".join(sorted(s2.lower()))
```

```
print(is_anagram('tea', 'eat'))
print(is_anagram('tea', 'treat'))
print(is_anagram('sinks', 'skin'))
print(is_anagram('Listen', 'silent'))
```



Crea una función que determine el día que se debe celebrar el evento mensual de la asociación "Programadores Compulsivos". Esta asociación ha decidido reunirse siempre el segundo jueves de cada mes. La función debe admitir como argumentos el año y el mes, y debe devolver un objeto `datetime.date` con la fecha propuesta

Copia aquí el código además de enviarlo por email a carmenbyg@gmail.com

```
>>> meetup_date(2012, 3)
datetime.date(2012, 3, 22)
>>> meetup_date(2015, 2)
datetime.date(2015, 2, 26)
>>> meetup_date(2018, 6)
datetime.date(2018, 6, 28)
>>> meetup_date(2020, 1)
datetime.date(2020, 1, 23)
```

```
import datetime
```

```
def meetup_date(year, month):
    assert year > 0
    assert 1 <= month <= 12

    new_date = datetime.date(year, month, 1)
    while new_date.strftime('%A') != 'Thursday':
        new_date += datetime.timedelta(days=1)

    new_date += datetime.timedelta(days=7)
    return new_date
```

```
print(meetup_date(2012, 3))
print(meetup_date(2015, 2))
print(meetup_date(2018, 6))
print(meetup_date(2020, 1))
```

Este contenido no ha sido creado ni aprobado por Google. - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios

