

Python Módulo I – Examen final Parte práctica

Se tomará en cuenta las **buenas prácticas**, las **tabulaciones o sangrados respectivos** y así como el uso de **flake8** para el adecuado diseño de código de nuestros problemas

PREGUNTA 1

```
class persona:
    def __init__(self, nombre, edad, nacionalidad = "peruana", saldo=0.0):
        self.nombre = nombre
        self.edad = edad
        self.nacionalidad = nacionalidad
        self.saldo = saldo

    def actualizar_nombre(self):
        self.nombre = self.nombre
        print("El nombre es: {}".format(self.nombre))

    def actualizar_edad(self):
        if edad > 0:
            self.edad = self.edad
        else:
            print("La edad tiene que ser mayor que 0")

    def cumplir_anios(self):
        self.edad += 1
        print ("La edad cumplida es: {}".format (self.edad))

    def mostrar_saldo(self):
```

```
class empleado(persona):
    def __init__(self, nombre, edad, sueldo, nacionalidad = "peruana", saldo=0.0):
        super().__init__(nombre, edad, nacionalidad, saldo)
        self.sueldo = sueldo
        print (sueldo)

    def aumento_sueldo (self, porcentaje=0.30):
        incremento = self.sueldo * porcentaje
        self.sueldo = self.sueldo + incremento
        print ("Para {}, el sueldo aumentó en {:.0f}%, siendo el nuevo sueldo {:.2f}".format( "args: {}".format(self.nombre, porcentaje*100, self.sueldo))

    def transferir(self, destino, monto):
        if self.saldo < monto:
            print ("Saldo insuficiente")
        else:
            self.saldo = self.saldo - monto
            destino.saldo = destino.saldo + monto
            print ("La transferencia se ha realizado de {} hacia {}".format (self.nombre, destino.nombre))

    def mostrar_saldo(self):
        self.saldo = self.saldo
        print (" El saldo actual es: {}".format(self.saldo))
```

```
class Empleado(persona):
    def aumentar_sueldo(self, porcentaje=0.30):
        incremento = self.sueldo * porcentaje
        self.sueldo = self.sueldo + incremento
        print(f"Para {self.nombre}, el sueldo aumento en {porcentaje*100}% a {self.sueldo}")

    def prediccion(self, anio_objetivo, edad_parametro=None):
        anio_actual = 2025
        anio_faltantes = anio_objetivo - anio_actual
        if edad_parametro is not None:
            if edad_parametro < self.edad:
                print(f"La persona {self.nombre}, en el año {self.edad} tendrá {anio_faltantes} años.")
            else:
                edad_futura = self.edad + anio_faltantes
                print(f"La persona {self.nombre}, en el año {self.edad} tendrá {anio_faltantes} años.")

emp1 = Empleado(nombre="Lucas", edad=32, sueldo=1000, nacionalidad="Peruana")
emp2 = Empleado(nombre="Juan", edad=20, sueldo=2000, nacionalidad="Peruana")
```

Run console output:

```
C:\Users\Sarai\AppData\Local\Programs\Python\Python313\python.exe "D:\CURSOS\PYTHON\BASICO 1\Exámenes\Examen Final\examen_final_Sarai_Cisneros\EF_01.py"
1000
2000
Process finished with exit code 0
```

PREGUNTA 2

```
import random

def gen_lista(num):
    lista = []
    for n in range(num):
        lista.append(random.randint(1, 50))
    print(lista)
    return lista

def no_repet(lista):
    no_repet = list(set(lista))
    print("Sin números repetidos:", no_repet)
    return no_repet

def ordenado(lista):
    ascend = sorted(lista)
    descend = sorted(lista, reverse=True)
    print("Descendente:", descend)
    print("Ascendente:", ascend)
    return descend, ascend
```

Run console output:

```
C:\Users\Sarai\AppData\Local\Programs\Python\Python313\python.exe "D:\CURSOS\PYTHON\BASICO 1\Exámenes\Examen Final\examen_final_Sarai_Cisneros\EF_002\main.py"
Ingrese Cantidad: 3
[48, 21, 15]
Sin números repetidos: [48, 21, 15]
Descendente: [48, 21, 15]
Ascendente: [15, 21, 48]
Process finished with exit code 0
```

Project

BASICO 1

D:\CURSOS\PYTHON\BASICO 1

Clases

Clase 1 (24.08.25)

Clase 2 (31.08.25)

Clase 3 (14.09.25)

Clase 4 (21.09.25)

Clase 4 - II

Clase 5

ejercicio_01.py

ejercicio_02.py

ejercicio_03.py

ejercicio_04.py

ejercicio_05.py

ejercicio_06.py

Clase 5 - II

Exámenes

Examen Final

examen_final_Sara1_Cisneros

EF_002

funciones.py

main.py

19

-

20

Crear una función para indicar cuál es el mayor número par de

21

la lista (lista de la regla 2), retornar este valor e imprimirlo por consola.

22

-

23

Crear el archivo main.py,

24

25

26

from funciones import gen_lista, no_repet, ordenado

27

28

try:

29

total = int(input("Ingrese Cantidad: "))

30

except ValueError:

31

print("Ingresar número entero.")

32

33

lista = gen_lista(total)

34

35

lista_sin_repetidos = no_repet(lista)

36

37

ordenado(lista_sin_repetidos)

38

39

-

Run

main

C:\Users\Sara1\AppData\Local\Programs\Python\Python313\python.exe "D:\CURSOS\PYTHON\BASICO 1\Exámenes\Examen Final\examen_final_Sara1_Cisneros\EF_002\main.py"

Ingrese Cantidad: 3

[48, 21, 15]

Sin números repetidos: [48, 21, 15]

Descendente: [48, 21, 15]

Ascendente: [15, 21, 48]

Process finished with exit code 0

BASICO 1

>

Exámenes

>

Examen Final

>

examen_final_Sara1_Cisneros

>

EF_002

>

main.py

28.5

CRLF

UTF-8

4 spaces

Python 3.13

PREGUNTA 3

PREGUNTA 4