

Taller de Evaluación de Funciones en Python

Fundamentos de programación

NOMBRE: DANIEL FELIPE LIZARAZO GUTIÉRREZ

Problema 1: Función Básica

Objetivo: Crear una función simple sin parámetros ni retorno.

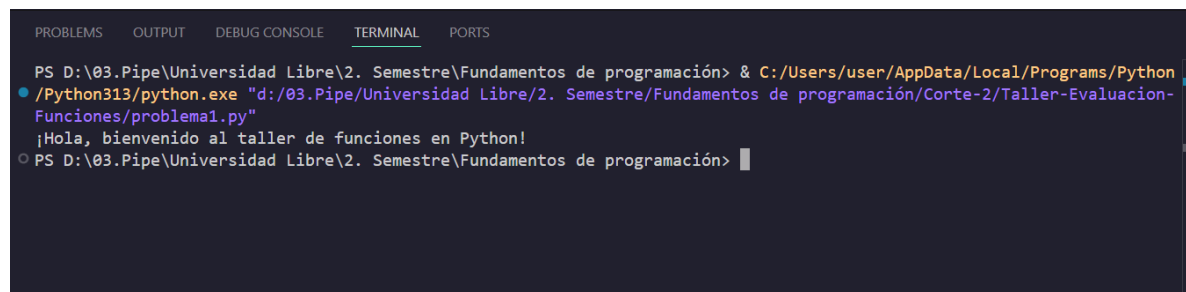
Instrucciones:

Define una función `saludar()` que imprima "¡Hola, bienvenido al taller de funciones en Python!".

CODIGO:

```
# Definimos la función
def saludar():
    print("¡Hola, bienvenido al taller de funciones en Python!")

# Llamamos a la función
saludar()
```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación & C:/Users/user/AppData/Local/Programs/Python/Python313/python.exe "d:/03.Pipe/Universidad Libre/2. Semestre/Fundamentos de programación/Corte-2/Taller-Evaluacion-Funciones/problema1.py"
¡Hola, bienvenido al taller de funciones en Python!
PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación >
```

Problema 2: Función con Parámetros

Objetivo: Crear una función que reciba parámetros y los utilice.

Instrucciones:

Define una función `calcular_area_rectangulo(base, altura)` que calcule y devuelva el área de un rectángulo.

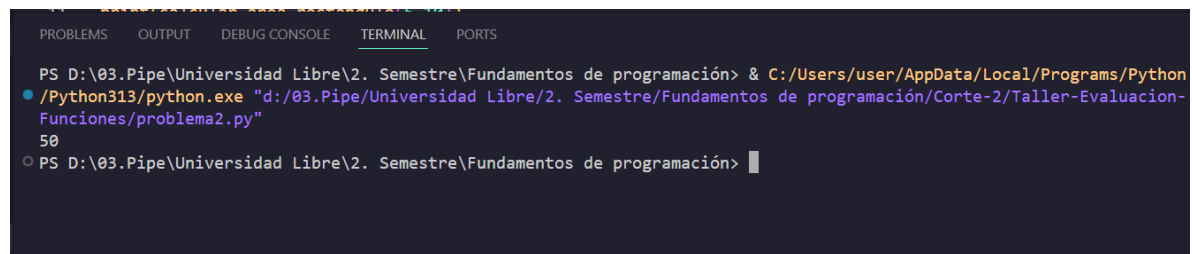
CODIGO:

```
# Definimos la función con sus parametros
def calcular_area_rectangulo(base, altura):

    # Aplicamos la formula del área
    area = base*altura

    # Devolvemos el resultado
    return area

# Imprimimos el resultado de la función con sus respectivos parametros
print(calcular_area_rectangulo(5,10))
```



```
PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación> & C:/Users/user/AppData/Local/Programs/Python/Python313/python.exe "d:/03.Pipe/Universidad Libre/2. Semestre/Fundamentos de programación/Corte-2/Taller-Evaluacion-Funciones/problema2.py"
50
PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación>
```

Problema 3: Función con Retorno de Valor

Objetivo: Definir una función que retorne un valor en lugar de imprimirlo directamente.

Instrucciones:

Crea una función `convertir_celsius_a_fahrenheit(celsius)` que reciba una temperatura en grados Celsius y retorne su equivalente en Fahrenheit.

◆ Fórmula:

$$F = (C \times 9/5) + 32$$

CODIGO:

```
# Definimos la función con el parametro de los celsius
def convertir_celsius_a_fahrenheit(celsius):

    # Aplicamos la formular de fahrenheit
    F = ((celsius*9)/5)+32

    # Devolvemos el resultado
    return F
```

```
# Imprimimos el resultado de convertir_celsius_a_fahrenheit en función de los grados
print(convertir_celsius_a_fahrenheit(10))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación & C:/Users/user/AppData/Local/Programs/Python/Python313/python.exe "d:/03.Pipe/Universidad Libre/2. Semestre/Fundamentos de programación/Corte-2/Taller-Evaluacion-Funciones/problema3.py"
50.0
PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación> █
```

Problema 4: Función Recursiva - Torres de Hanoi (Problema Clásico de Recursividad)

El problema de las **Torres de Hanoi** consiste en mover una torre de n discos desde una torre inicial a una torre destino, usando una torre auxiliar y respetando estas reglas:

1. Solo se puede mover un disco a la vez.
2. Un disco más grande no puede estar sobre uno más pequeño.
3. Se deben utilizar las tres torres.

CODIGO:

```
# Definimos la función con sus parametros
def torres_de_hanoi(n, origen, destino, auxiliar):

    # Si solo hay un disco directamente pasamos de A a C
    if n == 1:
        print(f"Mover disco 1 de {origen} a {destino}")
        return

    # Si hay más de un disco debemos empezar separando el más pequeño y luego el siguiente
    torres_de_hanoi(n - 1, origen, auxiliar, destino)
    print(f"Mover disco {n} de {origen} a {destino}")
    torres_de_hanoi(n - 1, auxiliar, destino, origen)

# Imprimimos el resultado de torres_de_hanoi cuando tenemos 2 discos
torres_de_hanoi(2, 'A', 'C', 'B')
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación> & C:/Users/user/AppData/Local/Programs/Python
/Python313/python.exe "d:/03.Pipe/Universidad Libre/2. Semestre/Fundamentos de programación/Corte-2/Taller-Evaluacion-
Funciones/problema4.py"
Mover disco 1 de A a B
Mover disco 2 de A a C
Mover disco 1 de B a C
PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación> █
```

Problema 5: Función con Parámetros Tipo Lista

Objetivo: Crear una función que trabaje con listas como parámetros.

Instrucciones:

Crear una función `calcular_promedio(lista_numeros)` que reciba una lista de números y retorne el promedio de los valores.

CODIGO:

```
# Definimos la función
def calcular_promedio(lista_numeros):
    # Inicializamos suma en 0
    suma = 0

    # Creamos un for para recorrer la lista y sumar cada uno de los
    elementos
    for i in lista_numeros:
        suma+=i

    # Calculamos el promedio
    promedio = suma/len(lista_numeros)

    # Devolvemos el promedio
    return promedio

# Definimos una lista cualquiera
lista = [1,2,3,4,5,6,7,8,9,10]

# Imprimos el valor de calcular_promedio en función de lista
print(calcular_promedio(lista))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación> & C:/Users/user/AppData/Local/Programs/Python
• /Python313/python.exe "d:/03.Pipe/Universidad Libre/2. Semestre/Fundamentos de programación/Corte-2/Taller-Evaluacion-
Funciones/problema5.py"
5.5
○ PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación> █
```

Problema 6: Función con Parámetros por Defecto y Argumentos Nombrados

Objetivo: Definir una función que tenga valores por defecto y acepte argumentos nombrados.

Instrucciones:

Crea una función `crear_mensaje(nombre, mensaje="Bienvenido al curso")` que reciba un nombre y un mensaje opcional. Si no se proporciona un mensaje, debe usar "Bienvenido al curso" por defecto.

CODIGO:

```
# Definimos la función con sus argumentos
def crear_mensaje(nombre, mensaje = "Bienvenido al curso"):

    #Devolvemos el valor (mensaje es un parametro que cambia o no
    dependiendo de si el usuario lo decide)
    return f"{nombre}, {mensaje}"

# Imprimimos la función de crear_mensaje pasandole solo el nombre
print(crear_mensaje("Daniel"))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación> & C:/Users/user/AppData/Local/Programs/Python
• /Python313/python.exe "d:/03.Pipe/Universidad Libre/2. Semestre/Fundamentos de programación/Corte-2/Taller-Evaluacion-
Funciones/problema6.py"
Daniel, Bienvenido al curso
○ PS D:\03.Pipe\Universidad Libre\2. Semestre\Fundamentos de programación> █
```

Reglas del Taller

- Se recomienda probar cada problema en un entorno como **python online**, o **VS Code**.
- Comenta tu código para explicar el las lineas
- Subir el taller todos los ejercicios en un archivo pdf, junto con las capturas de resultado del programa a la plataforma, revise los tiempos de entrega.
- Para la clase presencial se debe entregar el taller en una hoja examen. Abril 10.