

Ejercicios – Clase 1

Ejercicio 1

Escribí un programa que solicite al usuario ingresar su nombre y su edad. El programa debe realizar las siguientes validaciones:

- El nombre no debe estar vacío.
- La edad debe ser un número entero positivo.
- Si alguna de las validaciones falla, se debe mostrar un mensaje de error específico y pedir al usuario que ingrese los datos nuevamente.
- Si ambas validaciones son exitosas, muestra un mensaje de confirmación con el nombre y la edad ingresados: "¡Hola [nombre]! Tu edad es [edad]."

Ejercicio 2

Escribí una función llamada `dividir_seguro` que tome dos números como argumentos y realice la división. Sin embargo, la función debe manejar las siguientes situaciones:

- Si alguno de los números no es un valor numérico, debe generar una excepción `ValueError` con el mensaje "Ambos argumentos deben ser números".
- Si el segundo número es cero, debe generar una excepción `ZeroDivisionError` con el mensaje "No se puede dividir por cero".

Ejercicio 3

Escribí un programa que lea el contenido de un archivo de texto y realice las siguientes tareas:

- Si el archivo no existe, genera una excepción `FileNotFoundError` con el mensaje "El archivo no existe".
- Si se produce un error al leer el archivo, genera una excepción `IOError` con el mensaje "Error al leer el archivo".

EVOLUCIÓN CONTINUA

- Si el archivo existe y se puede leer correctamente, muestra su contenido por pantalla.

Ejercicio 4

- Crea un archivo llamado `mathlib.py` en el mismo directorio que tu programa principal.
- En el archivo `mathlib.py`, define al menos tres funciones que realicen cálculos matemáticos diferentes. Puedes considerar funciones como cálculo de área de figuras geométricas, operaciones con números complejos, generación de secuencias matemáticas, etc. Asegúrate de agregar comentarios descriptivos en cada función para explicar qué hace y qué parámetros espera.
- En el archivo `main.py`, importa el módulo `mathlib` utilizando la declaración `import`.
- Utiliza al menos dos funciones del módulo `mathlib` en el programa principal para realizar operaciones matemáticas específicas. Puedes solicitar datos al usuario o utilizar valores predefinidos.
- Muestra los resultados de las operaciones por pantalla.
- Ejecuta el archivo `main.py` y verifica que las funciones importadas desde `mathlib.py` se utilicen correctamente y produzcan los resultados esperados.
- Por ejemplo, supongamos que en el archivo `mathlib.py` has definido las siguientes funciones: `calcular_area_rectangulo(base, altura)`, `calcular_area_circulo(radio)`, y `calcular_factorial(numero)`. En el archivo `main.py`, podrías importar el módulo `mathlib` y utilizar las funciones de la siguiente manera:

```
import mathlib
```

```
base = 5
```

```
altura = 10
```

```
area_rectangulo = mathlib.calcular_area_rectangulo(base, altura)
```

```
print("El área del rectángulo es:", area_rectangulo)
```

```
radio = 3
```

```
area_circulo = mathlib.calcular_area_circulo(radio)
```

```
print("El área del círculo es:", area_circulo)
```

```
numero = 5
```

```
factorial = mathlib.calcular_factorial(numero)
```

```
print("El factorial de", numero, "es:", factorial)
```

Ejercicio 5

- Crea una carpeta llamada stats en el mismo directorio que tu programa principal.
- Dentro de la carpeta stats, crea los archivos descriptives.py y visualizations.py.
- En el archivo descriptives.py, define al menos dos funciones o clases que realicen cálculos descriptivos sobre conjuntos de datos, como calcular la media, la mediana, la desviación estándar, etc.
- En el archivo visualizations.py, define al menos dos funciones o clases que generen visualizaciones de datos, como gráficos de dispersión, histogramas, etc.
- En el archivo main.py, importa el paquete stats utilizando la declaración import.
- Utiliza las funciones y clases del paquete stats en el programa principal para realizar análisis y visualizaciones de datos específicos. Puedes utilizar conjuntos de datos predefinidos o importar datos desde archivos.
- Muestra los resultados de los análisis y las visualizaciones por pantalla o guarda las visualizaciones en archivos.
- Ejecuta el archivo main.py y verifica que las funciones y clases importadas desde el paquete stats se utilicen correctamente y produzcan los resultados y visualizaciones esperados.
- Por ejemplo, supongamos que en el archivo descriptives.py has definido las siguientes funciones: calcular_media(datos), calcular_mediana(datos), y en el archivo visualizations.py has definido las funciones: grafico_dispersion(x, y), histograma(datos). En el archivo main.py, podrías importar el paquete stats y utilizar las funciones del paquete de la siguiente manera:

```
import stats.descriptives as desc
```

```
import stats.visualizations as vis
```

```
datos = [2, 4, 6, 8, 10, 12]
```

```
media = desc.calcular_media(datos)
```

```
print("La media de los datos es:", media)
```

```
mediana = desc.calcular_mediana(datos)
```



Escuela de
INNOVACIÓN

EVOLUCIÓN CONTINUA

```
print("La mediana de los datos es:", mediana)
```

```
vis.histograma(datos)
```

```
x = [1, 2, 3, 4, 5]
```

```
y = [10, 8, 6, 4, 2]
```

```
vis.grafico_dispersion(x, y)
```