
Practico N° 3 - Python

➤ Modularizacion

1. Escriba una función en Python que indique si un número está en un determinado rango de numeros.

Ej.:

Entrada: valor=3; lim_inferior=2; lim_superior=5

Salida: True

2. Escriba una función en Python que indique si un número es perfecto. Utilice una función auxiliar que calcule los divisores propios.

Nota: *Un número perfecto es un número entero positivo que es igual a la suma de sus divisores positivos.*

Ej.:

Entrada: 6 (1+2+3)

Salida: True

3. Escriba una función en Python que reciba como parámetro una frase y 1 carácter, y devuelva si ese carácter se encuentra dentro de la frase. Además de ello, la función debe poder indicar la cantidad de palabras que hay en la frase.

4. Escriba una función en Python que reciba una lista de valores enteros y devuelva otra lista sólo con aquellos valores pares.

Ej.:

Entrada: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Salida: [2, 4, 6, 8]

5. Escribir una función que reciba una frase y devuelva un diccionario con las palabras que contiene y su longitud.

6. Los empleados de una fábrica trabajan en dos turnos, Diurno y Nocturno. Se desea calcular el jornal diario de acuerdo a las siguientes reglas:

- La tarifa de las horas diurnas es de \$350
- La tarifa de las horas nocturnas es de \$400
- En caso de ser festivo, la tarifa se incrementa en un 10% en caso de turno diurno y en un 15% para el nocturno.

Desarrolle una función que permita ingresar por teclado la siguiente información para, al menos, 2 empleados, nombre del trabajador, el número de horas trabajadas, el turno y el tipo de día ("Festivo", "Laborable"), para ello se podría utilizar 1 "diccionario" para registrar la información y si los datos ingresados son correctos llamar a otra función que realice el cálculo del sueldo a cobrar en ese día. Mostrar por pantalla los datos ingresados y el sueldo calculado para cada empleado.

7. Realice el ejercicio 5 del practico número 2 (**listas de tuplas**), pero implementando la/s función/es necesaria/s.

8. Crea el siguiente módulo:

- El módulo se denominará **operaciones.py** y contendrá 4 funciones para realizar una **suma**, una **resta**, un **producto** y una **division** entre dos números. Todas ellas devolverán el resultado.
- En las funciones del módulo deberá de haber tratamiento e invocación manual de errores para evitar que se quede bloqueada una funcionalidad, eso incluye:
 - *ValueError*: En caso de que se envíen valores a las funciones que no sean números. Además deberá aparecer un mensaje que informe **Error: Tipo de dato no válido**.
 - *ZeroDivisionError*: En caso de realizar una división por cero. Además deberá aparecer un mensaje que informe **Error: No es posible dividir entre cero**.

Una vez diseñado el modulo, desarrolle un programa que, utilizando el modulo anterior, haga uso de todas la funciones con los parámetros ingresados por teclado

9. Cree un programa que pida por teclado el ingreso de un usuario y contraseña, luego de ello utilice las funciones *validaUsuario* y *validaClave*, ambas en un módulo llamado *seguridad*. Dichas funciones deben realizar lo siguiente:

validaUsuario:

- El nombre de usuario debe contener un mínimo de 6 caracteres y un máximo de 12.
- El nombre de usuario debe ser alfanumérico. Si no lo cumple indicar el mensaje: *“El nombre de usuario puede contener solo letras y números”*
- Nombre de usuario válido, retorna True

validaClave:

- La contraseña debe contener un mínimo de 8 caracteres
- Una contraseña debe contener letras minúsculas, mayúsculas, números y al menos 1 carácter no alfanumérico
- La contraseña no puede contener espacios en blanco
- Contraseña válida, retorna True
- Contraseña no válida, retorna el mensaje: *“La contraseña elegida no es segura”*

10. Como ejercicio, escriba una función que use la funcion *tomaNumero* para leer un número del teclado y que maneje la excepción *ErrorNumeroMalo*.