



Argentina Programa 4.0

Universidad Nacional de San Luis

DESARROLLADOR PYTHON

Práctico Nro. 4.4

*Lenguaje de Programación Python: Estructuras de
Control y Funciones - Funciones*

Autor:

Dr. Mario Marcelo Berón

Argentina Programa 4.0

Universidad Nacional de San Luis

Práctico Nro. 4.4: *Estructuras de Control y Funciones - Funciones*

Ejercicio 1: Defina la función `sumaPar` la cual recibe como parámetro una lista `l` de números enteros y produce como resultado la suma de los números pares que se encuentran en `l`. Luego construya un programa principal que:

- Permita que el usuario ingrese una lista de números enteros.
- Imprima la suma de los números pares que se encuentran en la lista.

Ejercicio 2: Realice las siguientes actividades:

1. Defina la función `pares` la cual recibe como parámetro una lista de números enteros `l` y retorna como resultado una lista que contiene los números pares `l`.
2. Defina la función `impares` la cual recibe como parámetro una lista de números enteros `l` y retorna como resultado una lista de números impares.
3. Defina la función `mayoría` la cual recibe como parámetro dos listas de números enteros, una que contiene números pares y otra que contiene números impares. La función informa si hay mayoría de números pares o mayoría de números impares o no hay igualdad de números pares e impares.
4. Construya un programa principal que:
 - a) Permita que el usuario ingrese una lista de números enteros.
 - b) Informe si la lista tiene mayoría de pares o mayoría de impares o tiene la misma cantidad de pares que impares.

Ejercicio 3: Escriba una función que reciba como parámetros una lista de strings. La función crea un diccionario cuya clave son las letras del string y cuyo valor la cantidad de veces que aparece la clave en la lista de string. Luego escriba un programa principal que permita que el usuario ingrese una lista de string e imprima el diccionario resultante de aplicar la función a la lista ingresada por el usuario.

Ejercicio 4: Realice las siguientes actividades:

1. Defina la función `invertir string` la cual recibe como parámetro un string `s` y retorna como resultado otro string cuyo contenido es el de `s` pero en orden inverso.
2. Defina la función `invertirTodo` la cual recibe como parámetro la lista `l` de strings y retorna como resultado la lista `l` con los strings invertidos.
3. Construya un programa principal que:
 - a) Permita que el usuario ingrese una lista de strings `ls`.
 - b) Invierta los strings de `ls`.
 - c) Imprima `ls` y `ls` invertida.

Ejercicio 5: Implemente una calculadora de cuatro funciones. Para llevar adelante esta tarea ud debe:

1. Definir una función suma.
2. Definir una función resta.
3. Definir una función multiplicación.
4. Definir una función división.

El programa debe mostrar un menú de opciones que le de la posibilidad al usuario de elegir la operación deseada y ejecutarla. El programa debe finalizar cuando el usuario ingresa la opción -1. Si el usuario ingresa una opción no válida el programa debe informar la situación y volver a mostrar el menú de opciones.



Ejercicio 6: Dado el siguiente programa:

```
def coordenadaZ(x,y):  
    x=x+10  
    y=y+15  
    return x+y  
  
#programa principal  
x=int(input("Coordenada_eje_x:_"))  
y=int(input("Coordenada_eje_y:_"))  
for i in range(3):  
    z=coordenadaZ(x,y)  
    x=x+1  
    y=y+1  
    print(x,"_._",y,"_._",z)
```

Se pide:

1. De un ejemplo de ejecución del programa.
2. Diga que hace el programa.

Ejercicio 7: Dado el siguiente programa:

```
def maximo(x,y):  
    return x if x>y else: y  
  
def minimo(x,y):  
    return x if x<y else: y  
  
#programa principal  
x=int(input("Un_número:_"))  
y=int(input("Otro_número:_"))  
print(maximo(x-3, minimo(x+2, y-5)))
```

Se pide:

1. De un ejemplo de ejecución del programa.
2. Diga que hace el programa.
3. Critique la organización del código.

Ejercicio 8: Escriba una función que, dado un número de DNI, retorne True si el número es válido y False si no lo es. Para que un número de DNI sea válido debe tener entre 7 y 8 dígitos en base 10. Luego escriba un programa que pruebe la función.

Ejercicio 9: Implemente un padrón de personas. Por cada persona se almacena el nombre, dni y domicilio. El programa debe permitir que el usuario pueda:

1. Incorporar personas al padrón.
2. Eliminar personas del padrón.
3. Modificar los datos de una persona en el padrón.
4. Imprimir por pantalla los datos de una persona específica.

El programa debe mostrar por pantalla un menú de opciones con los ítems descritos anteriormente. El programa finalizará si el usuario ingresa como opción un 0.

Ejercicio 10 Construya una función que reciba como parámetro dos valores si esos valores son enteros la función retorna como resultado la suma de dichos valores. Si los valores son strings la función retorna como resultado la concatenación de los strings. En otro caso la función dispara una excepción ValueError.

Ejercicio 11: Defina la función porcentaje la cual recibe como parámetro una lista de números flotantes l y un número entero p. La función retorna como resultado el porcentaje p de la suma de los elementos de l. La función recibe como parámetro por defecto la lista vacía.



Ejercicio 12: Implemente las siguientes funciones:

- $\text{Factorial}(n) = 1 \times 2 \times 3 \times \dots \times n - 1 \times n$
- $\text{Fibonacci}(n)$ la cual se define como sigue:
 - $\text{Fibonacci}(0) = 0$
 - $\text{Fibonacci}(1) = 1$
 - $\text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2)$.
- $\text{Potencia}(n,m) = n^m$

Luego construya un programa principal que permita probar las funciones. Esto es invocarlas con argumentos correctos e incorrectos. Maneje los errores que surgen por la invocación de argumentos incorrectos con excepciones.