



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



ALGORITMOS Y ESTRUCTURAS DE DATOS (TSDS)

ASIGNATURA:	ALGORITMOS Y ESTRUCTURAS DE DATOS
PROFESOR:	Ing. Lorena Chulde
FECHA:	15-11-2024
PERÍODO ACADÉMICO:	2024-B

Taller

TÍTULO:

```
int sumar(int numero1, int numero2)
{
    int suma = numero1 + numero2;
    return suma;
}
```

Diagram illustrating the function definition and its components:

- Parámetros de entrada**: `int numero1, int numero2`
- Nombre de la función**: `sumar`
- Instrucciones**: `int suma = numero1 + numero2;`
- Valor de retorno**: `return suma;`

Estudiante
Felipe Zapata

Parte 1: Funciones. Paso de parámetro por valor y por referencia

Paso por valor y referencia

Dependiendo del tipo de dato que enviemos a la función, podemos diferenciar dos comportamientos:

- **Paso por valor:** Se crea una copia local de la variable dentro de la función.
- **Paso por referencia:** Se maneja directamente la variable, los cambios realizados dentro de la función le afectarán también fuera.

Tradicionalmente:

- **Los tipos simples se pasan por valor:** Enteros, flotantes, cadenas, lógicos...
- **Los tipos compuestos se pasan por referencia:** Listas, diccionarios, conjuntos...

Paso de parámetro por valor

Los números se pasan por valor y crean una copia dentro de la función, por eso no les afecta externamente lo que hagamos con ellos:

Ejemplo: crear una función doblar valor que multiplique a un número por dos

```
1 def dob():
2     x=float(input("Ingrese un número: "))
3     print(x*2)
4
5 dob()
```

```
Ingrese un número: 3
6.0
```

```
9 def doblar(numeros):
10     for i,n in enumerate(numeros):
11         numeros[i]*=2
12
13 lista=[10,50,100]
14 doblar(lista)
15 print(lista)
```

```
[20, 100, 200]
```

Paso de parámetro por referencia

Sin embargo las listas u otras colecciones, al ser tipos compuestos se pasan por referencia, y si las modificamos dentro de la función estaremos modificándolas también fuera:

Ejemplo: crear una función doblar valor que multiplique los elementos de una lista

```
lista=[10,20,25]
def agregar(numeros):
    numeros.append(40)

agregar(lista)
print(lista)
```

```
PS C:\Users\ALGORI
[10, 20, 25, 40]
```

Uso de la función `id()`.

```
x = 10
print(id(x)) # 4349704528
def funcion(entrada):
    entrada = 0
    print(id(entrada)) # 4349704208

funcion(x)
```

El ejemplo anterior nos podría llevar a pensar que si en vez de añadir un elemento a `x`, hacemos `x=[]`, estaríamos destruyendo la lista original. Sin embargo esto no es cierto.

```
x = [10, 20, 30]
def funcion(entrada):
    entrada = []

funcion(x)
print(x)
# [10, 20, 30]
```

```
x = [10, 20, 30]
print(id(x)) # 4422423560
def funcion(entrada):
    entrada.append(40)
    print(id(entrada)) # 4422423560

funcion(x)
```

Para modificar los tipos simples podemos devolverlos modificados y reasignarlos:

```
def doblar_valor(numero):
    return numero * 2

n = 10
n = doblar_valor(n)
print(n)
```

20

Y en el caso de los tipos compuestos, podemos evitar la modificación enviando una copia:

```
def doblar_valores(numeros):
    for i,n in enumerate(numeros):
        numeros[i] *= 2

ns = [10,50,100]
doblar_valores(ns[:]) # Una copia al vuelo de una lista con [:]
print(ns)
```

[10, 50, 100]

Parte 2: Ejercicios con Funciones

1. Solicitar al usuario que ingrese su dirección email. Imprimir un mensaje indicando si la dirección es válida o no, valiéndose de una función para decidirlo.

Nota: El correo se considerará válido si tiene el símbolo "@".

```
correo=input("Ingrese su correo: ")

def verC(email):
    emailValido=False
    for c in correo:
        if c == '@':
            return True
    return False

if verC(correo):
    print("Correo válido.")
else:
    print("Correo inválido")
```

```
Ingrese su correo: asdaf
Correo inválido
```

2. Pedir que el usuario ingrese valores hasta que ingrese el 0. Por cada valor, mostrar la suma de sus dígitos. Use una función que sume.

<https://github.com/FelipeZapata137/Algoritmos-/blob/main/Parcial%202/D7E3.py>

```
Ingresar un número (0 para salir): 234
La suma de los dígitos de 234 es: 9
Ingresar un número (0 para salir): 98
La suma de los dígitos de 98 es: 17
Ingresar un número (0 para salir): 21
La suma de los dígitos de 21 es: 3
Ingresar un número (0 para salir): 0
¡Adiós!
```

3. Solicitar al usuario que ingrese valores hasta que ingrese el cero. Por cada valor ingresado sumar sus dígitos y mostrarlos en pantalla. Al finalizar, calcular la suma total de todos los valores ingresados y la suma de sus dígitos. (TAREA)

<https://github.com/FelipeZapata137/Algoritmos-/blob/main/Parcial%202/D7E4.py>

```
Ingresar un número. Para terminar, ingresa 0.
Ingresar un número: 23
La suma de los dígitos de 23 es: 5
Ingresar un número: 43
La suma de los dígitos de 43 es: 7
Ingresar un número: 25
La suma de los dígitos de 25 es: 7
Ingresar un número: 0
Resultados finales:
La suma total de los números ingresados es: 91
La suma total de los dígitos de todos los números es: 19
```

```

1  def sumar(numero):
2      suma=0
3      while numero!=0:
4          digito=numero%10
5          suma=suma+digito
6          numero=numero//10
7      return suma
8
9  numero=int(input("Ingrese el número: "))
10 while numero !=0:
11     print("La suma es: ",sumar(numero))
12     numero=int(input("Ingrese el número: "))

```

Ingrese el número: 123
La suma es: 6

4. Solicitar al usuario que ingrese un número entero e imprimir en pantalla si es primo o no. Utilice una función booleana que lo decida.

119 2	119 7	NO
19 59	49 17	
1	0	

```

def primo(numero):
    for x in range(2,numero):
        if numero%x == 0:
            return False
    return True

name=input("Ingrese su nombre: ")
numero=int(input("Ingrese un número: "))

if primo(numero):
    print(name," , el numero",numero,"es primo." )
else:
    print(name," , el numero",numero,"no es primo." )

```

Ingrese su nombre: Felipe
Ingrese un número: 9
Felipe , el numero 9 es primo.

5. Imprimir en pantalla la cantidad de ocurrencias de un dígito que se encuentra en un número entero ingresado por el usuario.

Nota: El usuario digitará tanto el número entero como el dígito.

Utilice una función que calcule la frecuencia.

```
1 def contarOcurrencia(numero,digito):
2     cantidad=0
3     while numero!=0:
4         digitoCont=numero%10
5         if digitoCont==digito:
6             cantidad+=1
7         numero=numero//10
8     return cantidad
9
10 numero=int(input("Ingrese el numero: "))
11 digito=int(input("Ingrese el dígito a contar: "))
12 print(contarOcurrencia(numero,digito))
```

```
Ingrese el numero: 525
Ingrese el dígito a contar: 5
2
```

6. Calcular el factorial de un numero ingresado por el usuario.

Nota: Utilice una o más funciones, según sea necesario.

<https://github.com/FelipeZapata137/Algoritmos/blob/main/Parcial%202/D7E5.py>

```
Calculadora de factorial
Ingresa un número entero positivo: 5
El factorial de 5 es 120
```

7. Escribir un programa que pida números al usuario, mostrar el factorial de cada uno y, al finalizar, la cantidad total de números leídos en total.

Nota: Utilice una o más funciones, si es necesario. **Tarea**

<https://github.com/FelipeZapata137/Algoritmos/blob/main/Parcial%202/D7E6.py>

```
Ingresa números para calcular su factorial. Para terminar, ingresa -1.
Ingresa un número entero positivo: 3
El factorial de 3 es 6
Ingresa un número entero positivo: 5
El factorial de 5 es 120
Ingresa un número entero positivo: 4
El factorial de 4 es 24
Ingresa un número entero positivo: -1

Has ingresado un total de 3 números.
```

Parte 3: Ejercicios con Listas

1. Programa que imprime las asignaturas definidas en una lista.

```
1 asignaturas=['Programación','Algoritmos','Ecología','TICS']
2
3 for a in asignaturas:
4     print(a)
```

```
Programación
Algoritmos
Ecología
TICS
```

2. Imprimir el tamaño de una lista dada

```
1 asignaturas=['Programación','TICS','Ecología','Algoritmos']
2
3 print(len(asignaturas))
```

```
PS C:\
4
```

3. Programa que imprime las asignaturas con for (elemento por elemento)

```
1 asignaturas=['Programación','Algoritmos','Ecología','TICS']
2
3 for a in asignaturas:
4     print(a)
```

```
Programación
Algoritmos
Ecología
TICS
```

4. Unir dos listas y mostrar en una tercera lista

```
1 lista1=['1','2','3']
2 lista2=['4','5','6']
3 lista3=lista1+lista2
4
5 print(lista3)
```

```
['1', '2', '3', '4', '5', '6']
```

5. Agregar un elemento quemado a una lista usando append()

<https://github.com/FelipeZapata137/Algoritmos/blob/main/Parcial%20D7E7.py>

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


6. Agregar un elemento pidiendo al usuario a una lista usando append()

```

2  asignatura=input("Ingrese la asignatura: ")
3
4
5  while True:
6      asignatura=input("Ingrese la asignatura: ")
7      asignaturas.append(asignatura)
8      if asignatura== "salir":
9          break
10
11 print(asignaturas)

```

Ingrese la asignatura: Programación
 Ingrese la asignatura: Algoritmos
 Ingrese la asignatura: Inglés
 Ingrese la asignatura: TICS
 Ingrese la asignatura: salir
 ['Algoritmos', 'Inglés', 'TICS', 'salir']

7. Insertar las asignaturas en la lista e imprimirlas.

```

2  asignatura=input("Ingrese la asignatura: ")
3
4
5  while True:
6      asignatura=input("Ingrese la asignatura: ")
7      asignaturas.append(asignatura)
8      if asignatura== "salir":
9          break
10
11 print(asignaturas)

```

Ingrese la asignatura: Programación
 Ingrese la asignatura: Algoritmos
 Ingrese la asignatura: Inglés
 Ingrese la asignatura: TICS
 Ingrese la asignatura: salir
 ['Algoritmos', 'Inglés', 'TICS', 'salir']

8. Crear una lista e inicialízala con 5 cadenas de caracteres. Copia los elementos de la lista en otra lista, pero en orden inverso, y muestra sus elementos por la pantalla.

<https://github.com/FelipeZapata137/Algoritmos-/blob/main/Parcial%20D7E8.py>

```

Veinte
González
Zapata
Javier
Felipe

```

9. Con función **sort**: ordenar un arreglo de elementos de menor a mayor

```
1 numeros=[8,5,19,1,7,12]
2
3 numeros.sort()
4 print(numeros)
```

[1, 5, 7, 8, 12, 19]

10. Crear una fn para recorrer una lista.

<https://github.com/FelipeZapata137/Algoritmos-/blob/main/Parcial%20D7E9.py>

```
1
2
3
Felipe
Zapata
```

Método burbuja:

Crear una función “ordenar ()” para ordenar elementos de un array de menor a mayor.

```
def ordenar(array):
    tamaño = len(array)
    #for para recorrer las posiciones de 0 a 5
    for i in range(0,tamaño-1):
        #con este bucle comparamos
        for j in range(0,tamaño-1):
            if array[j] > array[j+1]:
                aux = array[j]
                array[j] = array[j+1]
                array[j+1] = aux
    return array

numeros = [6,3,8,2,7]
print("la lista original es: ", numeros)
#ordenar(numeros)
print("la lista ordenada es: ", ordenar(numeros))
```

Explicar el algoritmo paso a paso

1. Definición de la función ordenar
 2. Entrada: Se recibe una lista llamada 'array'.
 3. Variable 'tamaño': Se calcula la longitud de la lista con len(array).
 4. 2. Primer bucle for (índice i)
 - Función: Recorre la lista un número de veces igual a tamaño - 1.
 - Rango: range(0, tamaño - 1) (es decir, desde el índice 0 hasta el penúltimo índice).
- Este bucle externo asegura que todas las posiciones se comparen las veces necesarias.
5. Segundo bucle for (índice j)
 - Función: En cada iteración del primer bucle, este segundo bucle recorre la lista desde el índice 0 hasta el penúltimo elemento no ordenado (por eso también usa tamaño - 1 como límite).

- Propósito: Compara pares consecutivos de elementos.
 6. Comparación en el if
- Condición: `if array[j] > array[j + 1]`
 - Si el elemento actual es mayor que el siguiente, entonces hay que intercambiarlos.
- Intercambio:
 1. Se guarda temporalmente el valor de `array[j]` en `aux`.
 2. `array[j]` toma el valor de `array[j + 1]`.
 3. `array[j + 1]` recibe el valor almacenado en `aux`.
 4. Retorno

Al finalizar todos los bucles, la lista estará ordenada y se devuelve con `return array`.

Ejecución del código

1. Lista original: `numeros = [6, 3, 8, 2, 7]`.
2. Salida intermedia: El mensaje 'la lista original es: ' imprime la lista original.
3. Llamada a `ordenar(numeros)`: Ordena la lista usando el algoritmo explicado.
4. Resultado final: El mensaje 'la lista ordenada es: ' imprime la lista ordenada, que será `[2, 3, 6, 7, 8]`.