

## Taller de funciones en Python – Funciones integradas.

### ClaseLab 2.5 - Ejercicios de Funciones

#### Objetivos:

- El estudiante estará en capacidad aplicar los conceptos estudiados en clase sobre el uso de las funciones para resolver problemas con el lenguaje Python.
- El estudiante participará activamente en clase en la solución de los problemas planteados, compartiendo código, explicaciones y opiniones sobre las soluciones propuestas en la clase.
- El estudiante trabajará organizadamente todo su código fuente en su repositorio de GitHub y comentará este código fuente.
- Los pasos estipulados en el planteamiento descrito en la siguiente sección **deben ir acompañados de pruebas gráficas de su ejecución**. Documentar claramente el código, escribiendo comentarios para su facilidad de comprensión. Cada función generada deberá poseer información de cuáles parámetros recibe y de qué tipo, así como el tipo de retorno (si lo tiene).

#### Ejercicios propuestos:

1. Pedir al usuario el ingreso de números primos. La lectura finalizará cuando ingrese un número que no sea primo. Por cada número, mostrar la suma de sus dígitos. También solicitar al usuario un dígito e informar la cantidad de veces que aparece en el número (frecuencia). Al finalizar el programa, mostrar el factorial del mayor número ingresado.

Lee el ejercicio, plantea tu diagrama de flujo o pseudocódigo con la solución que se te ocurre y luego sí codifica en Python el código fuente de dicha solución.

En programación, el estudio, la buena comprensión lectora y la práctica, hacen al maestro. La lógica se fortalece la práctica continua.

A continuación, se propone una solución que se puede mejorar, impleméntala en tu entorno y procede a ejecutarla para evaluar mejoras y realizarlas.

Solución propuesta:

```
def primo(num):
    for i in range(2,num):
        if num%i==0:
            return False
    return True

def frecuencia(numero,digito):
    cantidad=0
    while numero!=0:
        ultDigito=numero%10
        if ultDigito==digito:
            cantidad+=1
        numero=numero//10
    return cantidad

def factorial(numero):
    f=1
    if numero!=0:
        for i in range(1,numero+1):
            f=f*i
    return f

def sumaDigitos(numero):
    suma=0
    while numero!=0:
        digito=numero%10
        suma=suma+digito
        numero=numero//10
    return suma

mayor=0
numero=int(input("Número primo: "))
while primo(numero):
    print("Suma de los dígitos:",sumaDigitos(numero))
    digito=int(input("Dígito: "))
    print("El",digito,"aparece",frecuencia(numero,digito),"veces")
    if numero > mayor:
        mayor=numero
    numero=int(input("Número primo: "))
print("Factorial de",mayor,":",factorial(mayor))
```

```

mayor = 0
numero = int(input("Introduzca un número primo: "))
while primo(numero):
    print("Suma de los dígitos:", sumaDigitos(numero))
    digito = int(input("Introduzca un dígito (0-9): "))
    if digito > 9:
        print('\nNo se escribió un dígito...') # Una explicación en caso que falle
    else:
        print(f"El {digito} aparece {frecuencia(numero, digito)} veces")
    if numero > mayor:
        mayor = numero
    numero = int(input("Número primo: "))
print(f"\nFactorial del mayor ({mayor}): {factorial(mayor)}")

```

```

Introduzca un número primo: 11
Suma de los dígitos: 2
Introduzca un dígito (0-9): 1
El 1 aparece 2 veces
Número primo: 5
Suma de los dígitos: 5
Introduzca un dígito (0-9): 12

No se escribió un dígito...

Número primo: 4

Factorial del mayor (11): 39916800

```

La interfaz se modifica para ser mas amigable con el usuario, dándoles instrucciones claras y explicándoles en caso de que se equivoquen y detengan el código por accidente. También se simplificaron los formatos, remplazándolos por f-strings.

2. Sin ejecutar el siguiente programa, determina cuál es la salida en pantalla si se ingresan los valores  $x=6$ ,  $y=7$ .  
Adjunta el traceo que realizaste para obtener la respuesta.  
¿Qué hace este programa? Mejora la interacción con el usuario de acuerdo con tu interpretación del código. Adjunta captura de pantalla del nuevo programa mejorado y de su salida.

```

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)
```

X = 6

Y = 7

Por i en rango (0,2)

0:

Z = 38

X = 7

Y = 8

1:

Z = 41

X = 8

Y = 9

2:

Z = 44

X = 9

Y = 10

El programa no imprime la variable Z, entonces habría que agregarla al código o eliminarla.

3. El siguiente programa debería imprimir el número 2 si se le ingresan como valores x=5, y=1 pero en su lugar imprime 5. ¿Qué hay que corregir? Realiza la corrección en tu consola de programación y adjunta captura de la prueba.

```
def maximo(a,b):
    if x>y:
        return x
    else:
        return y

def minimo(a,b):
    if x<y:
        return x
    else:
        return y
```

```
#programa principal
x=int(input("Un número: "))
y=int(input("Otro número: "))
print(maximo(x-3, minimo(x+2, y-5)))
```

```
def maximo(x, y):
    if x > y:
        return x
    else:
        return y

def minimo(x, y):
    if x < y:
        return x
    else:
        return y

# programa principal
x = int(input("Un número: "))
y = int(input("Otro número: "))
print(maximo(x - 3, minimo(x + 2, y - 5)))
```

```
Un número: 5
Otro número: 1
2
```

Código original realizado por Edwin Nicolás Tarazona

Las variables dentro de la función máximo() y mínimo() estaban mal implementadas, por lo que detectaba que 5 > 1 y no realizaba las operaciones necesarias, solo imprimía x sin cambios.

4. Lee sobre Funciones Integradas en Python (built in) en [5.6. Funciones integradas — Materiales del entrenamiento de programación en Python - Nivel básico \(entrenamiento-python-basico.readthedocs.io\)](#)

Escoge dos y explícalas con un ejemplo que debes compartir con tus compañeros en clase, anexa captura del entorno de programación con la prueba de las funciones.

#### 4.1. Función round()

La función round se encarga de redondear el número digitado a n dígitos.

##### Argumentos

round() se compone de dos argumentos, el número a redondear y el aproximado que se quiere. Pueden ser tanto integer como float.

```
round(6, 10)  
round(0.8, 1)  
round(2.6)
```

##### Valor devuelto

round() devuelve el número mas cercano a la potencia menos n dígitos. Si n dígitos se omite, se redondea al valor más cercano. Si dos valores están igual de cerca, se redondea al valor par.

```
print(round(4.73654, 3))  
4.737
```

#### 4.2. Función bin()

Transforma números enteros en una cadena binaria

##### Argumentos

Cualquier integer, puede ser transformado en binario. Si se quiere utilizar un float, se debe especificar al programa que lo transforme en integer primero

```
bin(15)  
bin(32)  
bin(int(3.5))
```

##### Valor devuelto

El programa transforma el numero digitado en cadena binaria y lo devuelve con el prefijo "0b"

```
print(bin(48))  
0b110000
```