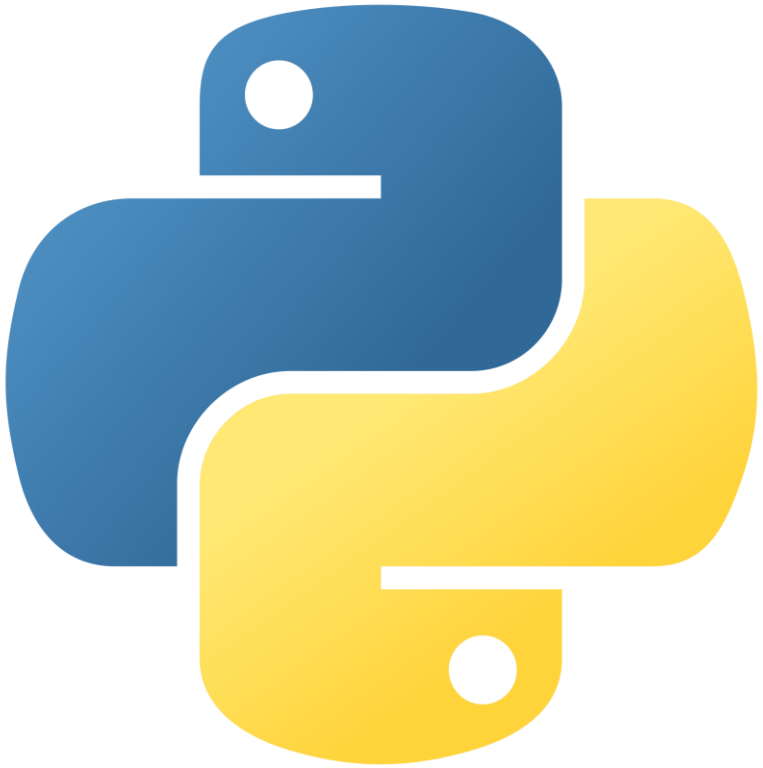


## MODULO 2

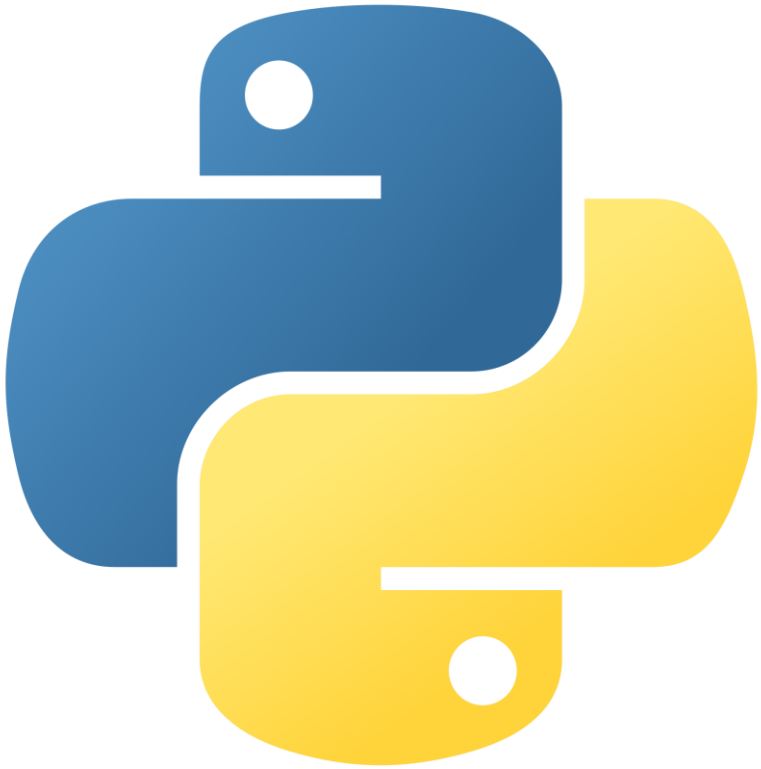
# FUNCIONES

## LOGRO DE LA SESION

Al finalizar la sesión, comprenderá cómo crear bloques de código reutilizables que realizan tareas específicas. Aprenderá la sintaxis para definir funciones, incluyendo el nombramiento y la transferencia de argumentos. Además, comprenderá la importancia de los parámetros y su aplicación. También se familiarizará con el uso de valores de retorno para obtener resultados y cómo la modularidad de las funciones ayuda a organizar el código de manera más eficiente.

GUSTAVO CORONEL gcoronelc@gmail.com

# Contenido



- Introducción
- Definición
- Sintaxis
- Programando funciones
- Conclusiones
- Evaluación Continua 3

# Introducción

# Definición

```python
def sumar( num_1, num_2 ):
    suma = num_1 + num_2
    return suma

print( sumar(34,67) )
```

Una función es un bloque de código reutilizable que realiza una tarea específica.

GUSTAVO CORONEL gcoronelc@gmail.com

# Sintaxis

```
1    def nombre_función (parámetros):
2        bloque de código
3        return resultado
```

# Programando funciones

En este ejemplo se define una función que no realiza nada.

```
def no_hace_nada():
    pass

no_hace_nada()
```

# Programando funciones

En este ejemplo se define una función que muestra un mensaje.

```
def saludar():
    print("Hola todos!!")

saludar()
```

# Programando funciones

La llamada a una función se puede asignar a una variable.

```
def saludar():
    print("Hola todos!!")

llamada = saludar()
llamada
```

# Programando funciones

## Pasando parámetros a una función.

```
def saludar( nombre ):
    print("Hola",nombre)

saludar( "Gustavo Coronel" )
```

# Programando funciones

## Función que retorna un valor.

```python
def media(num_1, num_2, num_3):
    suma = num_1 + num_2 + num_3
    prom = suma / 3
    return prom

promedio = media(13,19,16)
print("Promedio:", promedio)
```

# Programando funciones

El orden de los argumentos es importante.

```
def restar(num_1, num_2):
    resta = num_1 - num_2
    return resta

print(restar(7,3))
print(restar(3,7))
```

# Programando funciones

Aplicando los **KEYWORD ARGUMENTS**.

```
def restar(num_1, num_2):
    resta = num_1 - num_2
    return resta

print(restar(num_1=7,num_2=3))
print(restar(num_2=3,num_1=7))
```

# Programando funciones

## Valores por defecto para los argumentos.

```
def restar(num_1=5, num_2=2):
    resta = num_1 - num_2
    return resta

print(restar())
```

# Programando funciones

## Ámbito de las variables

Las variables creadas dentro de una función son locales, no se pueden usar fuera de la función.

```
def restar(num_1=5, num_2=2):
    resta = num_1 - num_2
    return resta

print(restar())
print(resta)
```

```
NameError: name 'resta' is not defined. Did you mean: 'restar'?
```

# Programando funciones

## Ámbito de las variables

Las variables creadas dentro de una función son locales, no se pueden usar fuera de la función.

Fuera de la función se tiene la variable global resta y dentro de la función se tiene la variable resta local.

```
resta = 20

def restar(num_1=5, num_2=2):
    resta = num_1 - num_2
    return resta

print(restar())
print(resta)
```

# Programando funciones

## Ámbito de las variables

Desde una función se puede acceder a variables globales utilizando la instrucción **global**.

```
resta = 20

def restar(num_1=5, num_2=2):
    global resta
    resta = num_1 - num_2
    return resta

print(restar())
print(resta)
```

GUSTAVO CORONEL gcoronelc@gmail.com

# Programando funciones

## Procesando listas

Las funciones pueden recibir listas en sus argumentos.

```
def sumar( lista ):
    suma = 0
    for n in lista:
        suma += n
    return suma

print( sumar([4,7,1]))
print( sumar([40,70,10,20]))
```

# Ejercicios

Debes aplicar los conceptos que se bien utilizando: Entrada de datos ⇨ Proceso ⇨ Reporte

GUSTAVO CORONEL gcoronelc@gmail.com

# Ejercicio 1

Generar dos listas de números enteros de 10 elementos cada uno.

Generar una tercera lista que sume los elementos de los dos arreglos anteriores.

No es suma de matrices.

| | Lista1 | Lista2 | Lista3 |
|---|---|---|---|
| 0 | 30 | 20 | 50 |
| 1 | 15 | 30 | 45 |
| 2 | 16 | 45 | 61 |
| 3 | 40 | 15 | 55 |
| 4 | 30 | 36 | 66 |
| 5 | 25 | 22 | 47 |
| 6 | 18 | 10 | 28 |
| 7 | 34 | 41 | 75 |
| 8 | 41 | 38 | 79 |
| 9 | 40 | 21 | 61 |

GUSTAVO CORONEL gcoronelc@gmail.com

# Ejercicio 2

Desarrollar un programa que genere una lista de 10 números enteros entre 20 y 50.

Luego debe contar cuantos números son pares y cuantos son impares.

| | |
|---|---|
| 0 | 28 |
| 1 | 37 |
| 2 | 36 |
| 3 | 42 |
| 4 | 29 |
| 5 | 46 |
| 6 | 21 |
| 7 | 24 |
| 8 | 39 |
| 9 | 48 |

**Pares: 6**

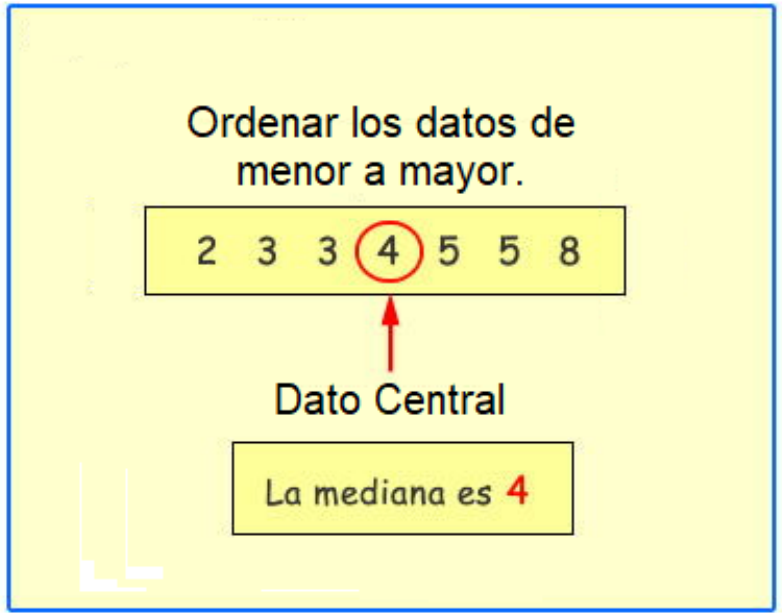
**Impares: 4**

GUSTAVO CORONEL gcoronelc@gmail.com

# Ejercicio 3

Desarrollar un programa que permita el ingreso de "N" números enteros positivos.

Luego debe encontrar la media y la mediana.




GUSTAVO CORONEL gcoronelc@gmail.com

# Ejercicio 4

Desarrollar un programa que genere una matriz de 4x3 de números aleatorios comprendidos entre 10 y 30.

Luego debe calcular la suma de cada columna.

Finalmente, el programa debe mostrar la matriz y la suma de sus columnas.

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 15 | 19 | 29 |
| 1 | 22 | 18 | 13 |
| 2 | 25 | 25 | 25 |
| 3 | 17 | 28 | 18 |
| **Suma** | **79** | **90** | **85** |

GUSTAVO CORONEL gcoronelc@gmail.com

# Ejercicio 5

Desarrollar un programa que permita a un profesor generar la nota de sus alumnos, las notas se deben encontrar entre 0 y 20.

Luego el programa debe mostrar los siguientes datos estadísticos:

- Cantidad de aprobados
- Cantidad de desaprobados
- La nota mayor
- La nota menor
- La nota promedio

**REPORTE**

- Cantidad de alumnos: 10
- Aprobados: 6
- Desaprobado: 4
- Nota mayor: 18
- Nota menor: 10
- Nota promedio: 15

GUSTAVO CORONEL gcoronelc@gmail.com

# Ejercicio 6

**Desarrollar una función que determine si un numero es primo o no.**

GUSTAVO CORONEL gcoronelc@gmail.com

# Ejercicio 7

**Desarrollar una función para determinar si 2 números son amigos.**

GUSTAVO CORONEL gcoronelc@gmail.com

# Ejercicio 8

**Desarrollar una función para calcular el valor de la siguiente sumatoria.**

$$serie = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + \frac{x^{2n-1}}{(2n-1)!}$$

GUSTAVO CORONEL gcoronelc@gmail.com

# Conclusiones

- Uno de los beneficios de las funciones es la reutilización de código.
- La abstracción es otro beneficio importante con el uso de funciones, permite ocultar los detalles de la implementación.
- Permite aplicar de manera muy eficiente el principio "Divide y Vencerás".
- En general, el l uso de funciones permite la escritura de código más eficiente, legible y fácil de mantener.


GRACIAS
TOTALES