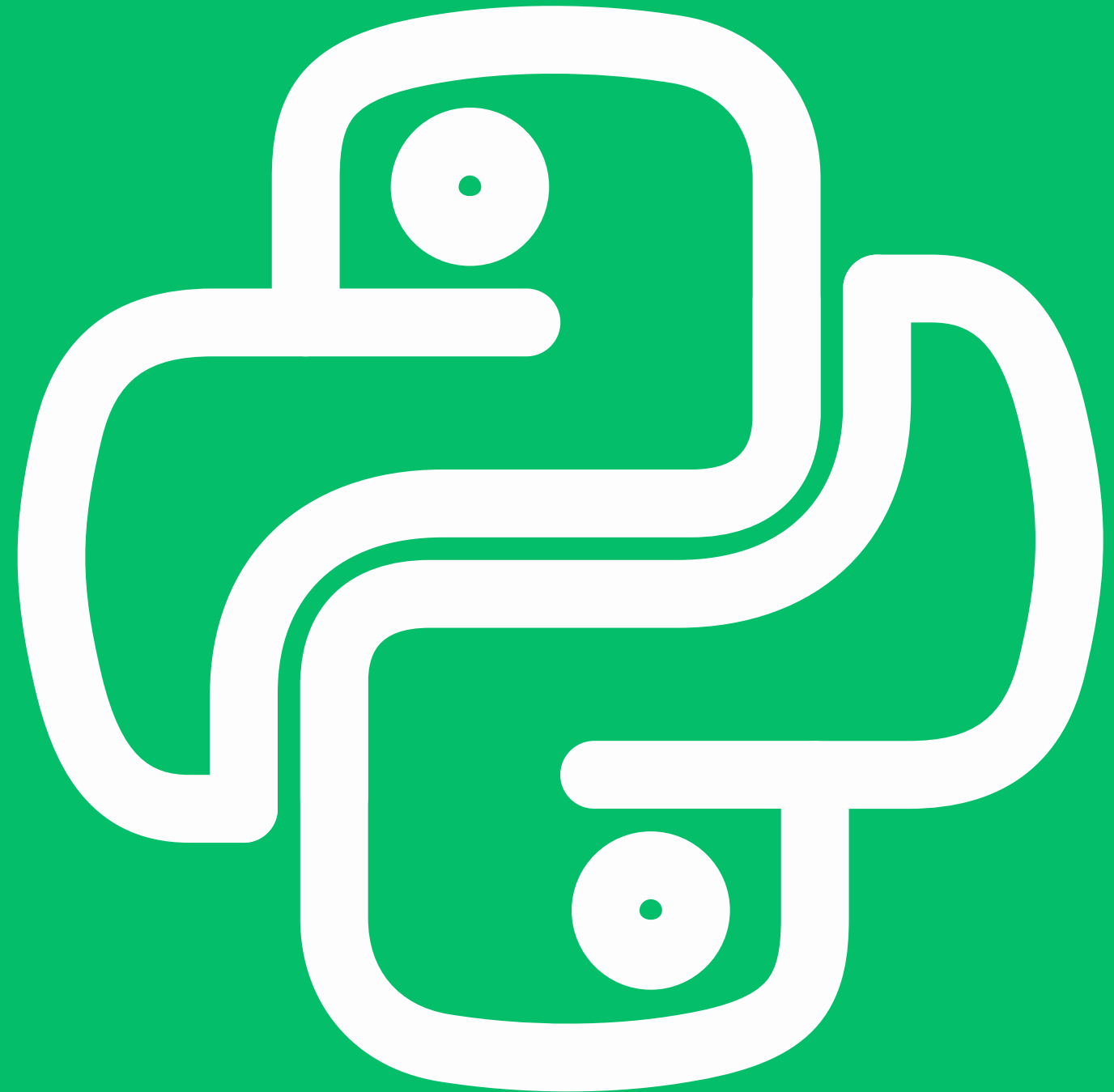


# Modulos: Funciones y Procedimientos



INSTITUTO DE INNOVACIÓN Y  
TECNOLOGÍA APLICADA

# Contenido

**01** Introducción

**02** Ámbitos de variables

**03** Parametros

**04** Funciones y procedimientos



P Y T H O N

Pregunta Inicial:

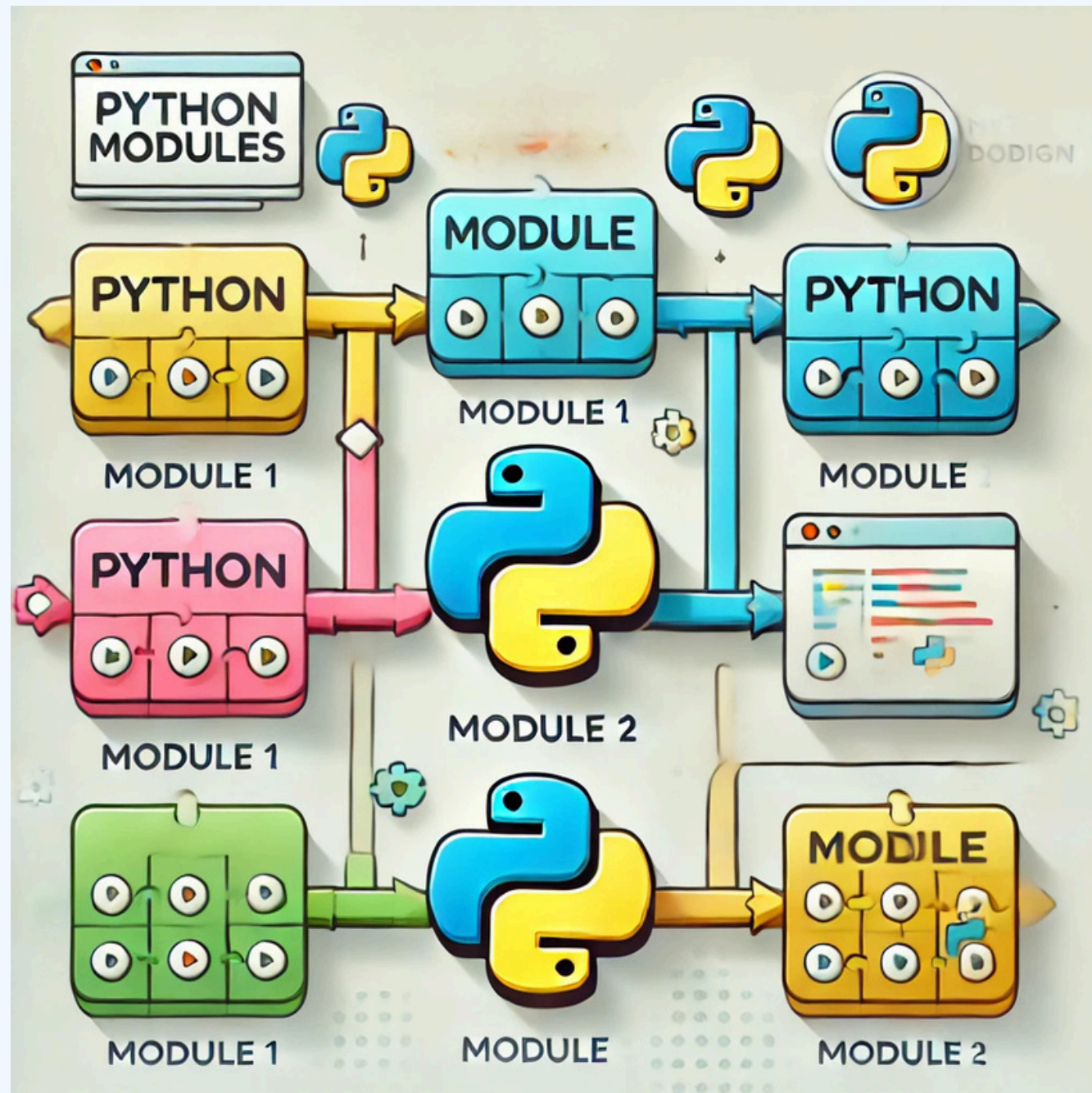
**¿Qué Funciones  
conocen hasta  
ahora?**

## INTRODUCCIÓN

# Modulos

Es un segmento, rutina, subrutina, subalgoritmo o procedimiento que puede ser definido dentro de un algoritmo mayor con el propósito de ejecutar una tarea específica, pudiendo ser llamado o invocado desde el módulo principal cuando se requiera.

Los módulos se declaran una sola vez pero pueden ser utilizadas, mediante llamadas, todas las veces que se quiera en un programa.





## Modulos en Python

La definición de módulos se realiza mediante la instrucción `def` más un nombre descriptivo (para el cuál, aplican las mismas reglas que para el nombre de las variables) seguido de paréntesis de apertura y cierre.

Como toda estructura de control en Python, la definición finaliza con dos puntos (:), y el algoritmo que la compone irá indentado (con sangría).

```
def saludar():  
    print("Hola mundo")  
  
saludar()  
saludar()  
saludar()
```

Un modulo solo se ejecuta cuando se lo llama en el cuerpo principal del programa

## Ambito de las variables

Cuando definimos módulos, estamos “separando” nuestro programa en programas mas pequeños. En cada módulo las variables que cree dentro de ellos no existen fuera de estos.

Las variables LOCALES o de ámbito LOCAL son aquellas que defino dentro de módulos, y solo pueden ser usadas en el mismo.

```
def modulo():  
    print(variable)  
  
variable="hola"
```

```
def modulo():  
    variable="hola"  
  
print(variable)
```

Las variables de ámbito GLOBAL son aquellas que defino fuera de módulos, en el cuerpo principal, y pueden ser usadas en cualquier lugar

## Parametros

Un parámetro es un valor que la función espera recibir cuando sea llamada.

Una función puede esperar 0,1 o más parámetros.

## Importante

Los parámetros que una función espera, serán utilizados por ésta dentro de su algoritmo, a modo de variables de ámbito local

```
def sumar(numero1,numero2):  
    print(numero1+numero2)
```

```
sumar(1,2)  
sumar(7,8)  
sumar(10,2)
```



```
3  
15  
12
```



## A PRACTICAR

01

Solicitar al usuario que ingrese su dirección email. crear un modulo que Imprima un mensaje indicando si la dirección es válida o no. Una dirección es válida si contiene el símbolo "@".

02

Definir una función que muestre el factorial de un número.

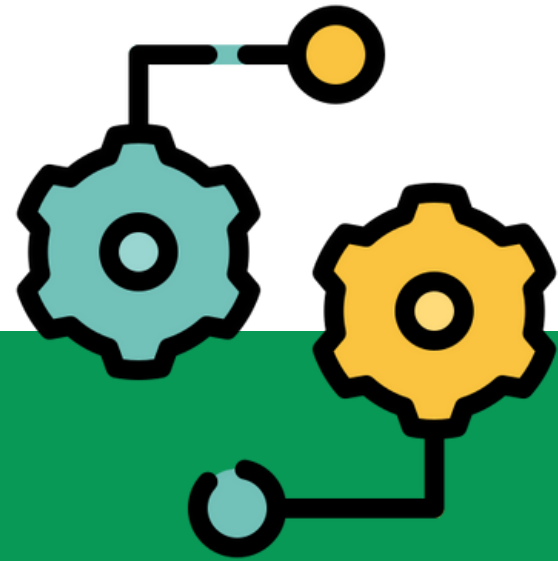
## Ayuda Memoria

```
#crear el modulo
def nombreModulo(parametro1, parametro2):
    #lista de instrucciones
    pass

#llamar al modulo
nombreModulo(valor1,valor2)
```



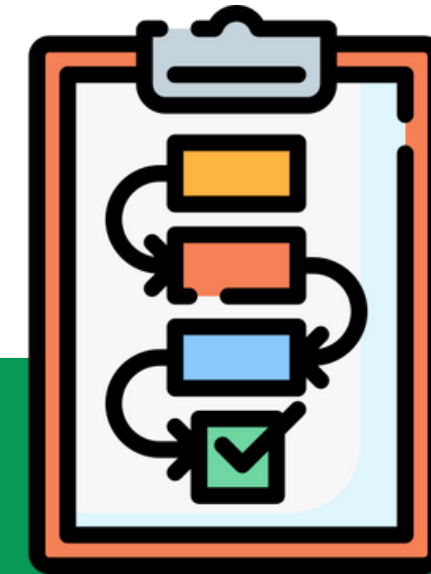
## Tipos de módulos



### Funciones

Aquellos módulos que **SI** retornan valores .

`input()`, `pop()`, `upper()`,  
`lower()`



### Procedimientos

Aquellos módulos que Se limitan a llevar a cabo tareas sin devolver valores.

`print()`, `append()`, `remove()`

## Retornar Valores

Un modulo puede o no retornar datos. Retornar un valor en un modulo sirve para pasar valores de un ámbito local a un ámbito global.(Es importante almacenar estos datos en el ámbito global)

Para retornar un valor de un modulo se utiliza la palabra return, y una vez el programa lee esa palabra el modulo en el que esta está finaliza. (no se ejecuta lo que esta a continuacion)

```
def sumar(numero1,numero2):  
    return numero1+numero2  
  
edad=sumar(10,8)  
print(f"tu edad es {edad}")
```



tu edad es 18

## Cuestiones a considerar

Python maneja funciones y los procedimientos de la misma forma, pero NO son lo mismo.

En la teoría:

Función: devuelve un valor

Procedimiento: solo ejecuta comandos

Al llamar a un Modulo, siempre se le deben pasar sus argumentos en el mismo orden en el que los espera

Un Modulo, no se ejecuta hasta que sea invocado. Para invocar un Modulo, simplemente se la llama por su nombre

## Funciones utiles

```
lista=[]  
len(lista)
```

devuelve el largo de una lista

busca y devuelve la posición de un carácter, o conjunto de caracteres, en un string

```
String="Hola mundo"  
String.find("o")
```

```
String="Hola mundo"  
nuevoString=String.upper()
```

devuelve una copia de el string pero con todos sus caracteres remplazados por su equivalente en mayúsculas.

devuelve una copia de el string pero con todos sus caracteres remplazados por su equivalente en mayúsculas.

```
String="Hola mundo"  
nuevoString=String.lower()
```

01

Escriba un modulo que muestre todos los números primos entre 1 y un número n que es ingresado por parámetro

02

cree una función que reciba una lista de números y devuelva el mayor

## Ayuda Memoria

```
#crear el modulo
def nombreFuncion(parametro1,parametro2):
    #codigo de la funcion
    return #valor a devolver

#llamar a la funcion
variable=nombreFuncion(valor1,valor2)
```

## cosas útiles: parametros

Se pueden Usar "Parámetros por posición":

```
def mensaje(nombre,apellido):  
    print(f"nombre: {nombre}, apellido: {apellido}")  
  
mensaje("Nicolas","Martinelli")
```



```
nombre: Nicolas, apellido: Martinelli
```

## cosas útiles: parametros

Se pueden Usar "Parámetros por nombre":

```
def mensaje(nombre,apellido):  
    print(f"nombre: {nombre}, apellido: {apellido}")  
  
mensaje(apellido="Martinelli",nombre="Nicolas")
```



```
nombre: Nicolas, apellido: Martinelli
```




## cosas útiles: parametros

Se pueden Usar "Parámetros por omisión":

```
def mensaje(nombre="Nicolas", apellido="Martinelli"):
    print(f"nombre: {nombre}, apellido: {apellido}")

mensaje()
mensaje("Pablo", "Sandoval")
```



```
nombre: Nicolas, apellido: Martinelli
nombre: Pablo, apellido: Sandoval
```

I I T A

# ¡Muchas Gracias!

Por su atención

