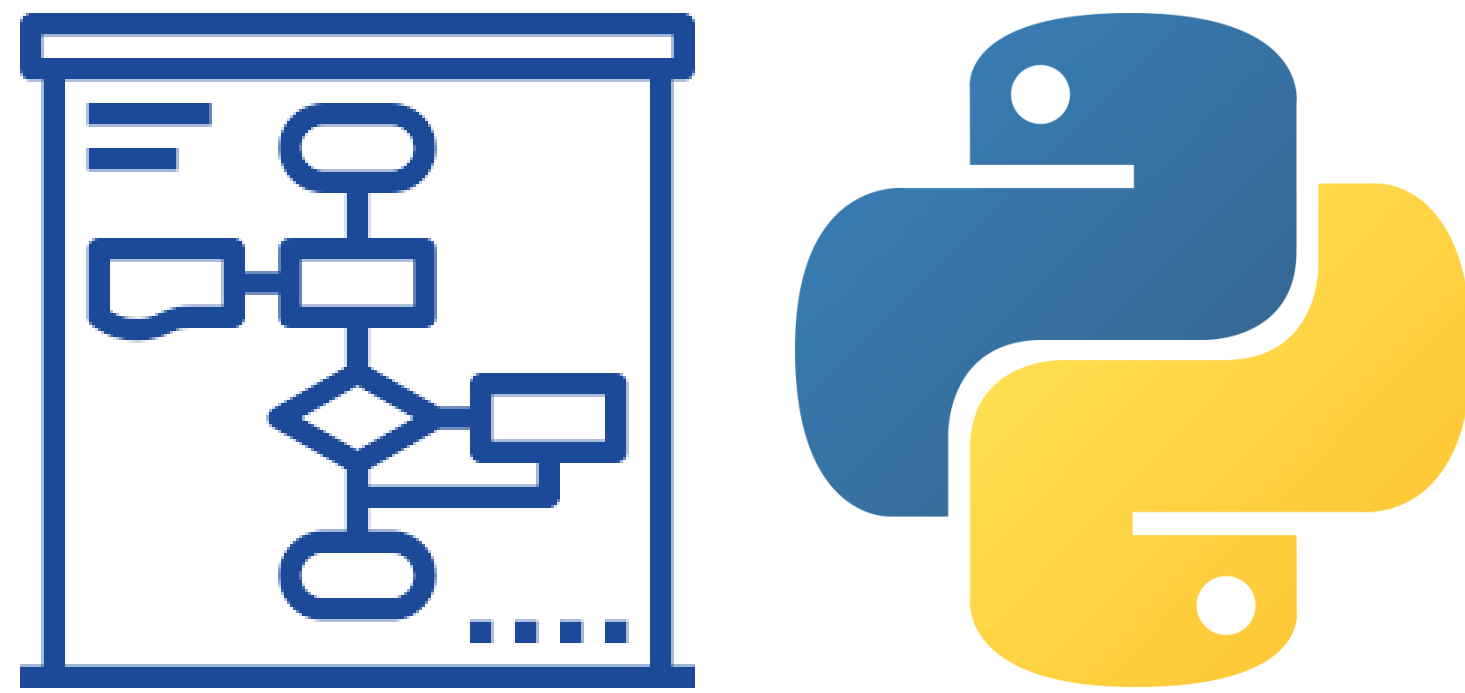




INTRODUCCIÓN A LA PROGRAMACIÓN



OTOÑO, 2022

UNIVERSIDAD TECNOLÓGICA DE CHILE
INSTITUTO PROFESIONAL
CENTRO DE FORMACIÓN TÉCNICA





UNIDAD II

Lenguajes Altamente Dinámicos/ Débilmente Tipados

Funciones: definiciones y llamada

¿Qué es una función en Python?

- Una función es un bloque de código que solo se ejecuta cuando se llama.
- Puede pasar datos, conocidos como parámetros, a una función.
- Una función puede devolver datos como resultado.

```
def saludar():
```

¿Por qué necesitamos funciones?



- Utilizamos funciones cuando queremos reutilizar varias veces una o más líneas de nuestro código.
- Las utilizamos como herramientas, con el fin de hacer la vida más fácil, y para simplificar tareas tediosas y repetitivas.
- Muy a menudo ocurre que un cierto fragmento de código **se repite muchas veces en un programa**. Se repite de manera literal o, con algunas modificaciones menores, empleando algunas otras variables dentro del programa.

Si un fragmento de código comienza a aparecer en más de una ocasión, **considera la posibilidad de aislarlo en la forma de una función**, invocando la función desde el lugar en el que originalmente se encontraba.

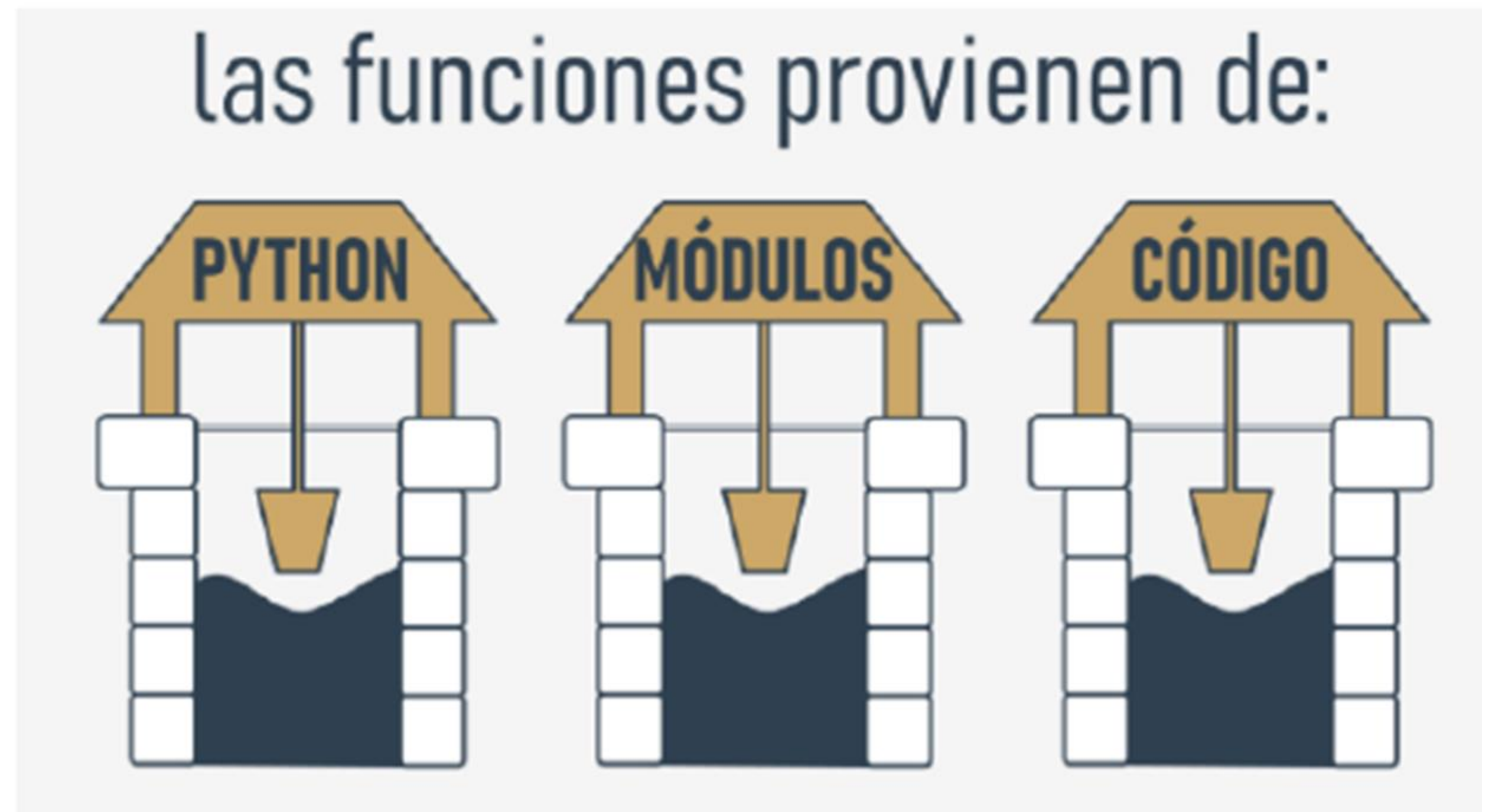
¿De dónde provienen las funciones?

En general, las funciones provienen de al menos tres lugares:

De Python mismo: varias funciones (como `print()`) son una parte integral de Python, y siempre están disponibles, a estas funciones se les conoce como “funciones integradas”.

De los **módulos preinstalados** de Python: muchas de las funciones, las cuales comúnmente son menos utilizadas que las integradas, están disponibles en módulos instalados juntamente con Python; para poder utilizar estas funciones el programador debe realizar algunos pasos adicionales

Directamente del código: puedes escribir tus propias funciones, colocarlas dentro del código, y usarlas libremente.



Definición y llamada

Se definen con la palabra reservada **def**, seguido el nombre de la función, luego los argumentos y unos paréntesis, que también se utilizan para hacer la llamada. Se debe considerar que se pueden definir con y sin retorno de datos y con y sin parámetros de entrada.

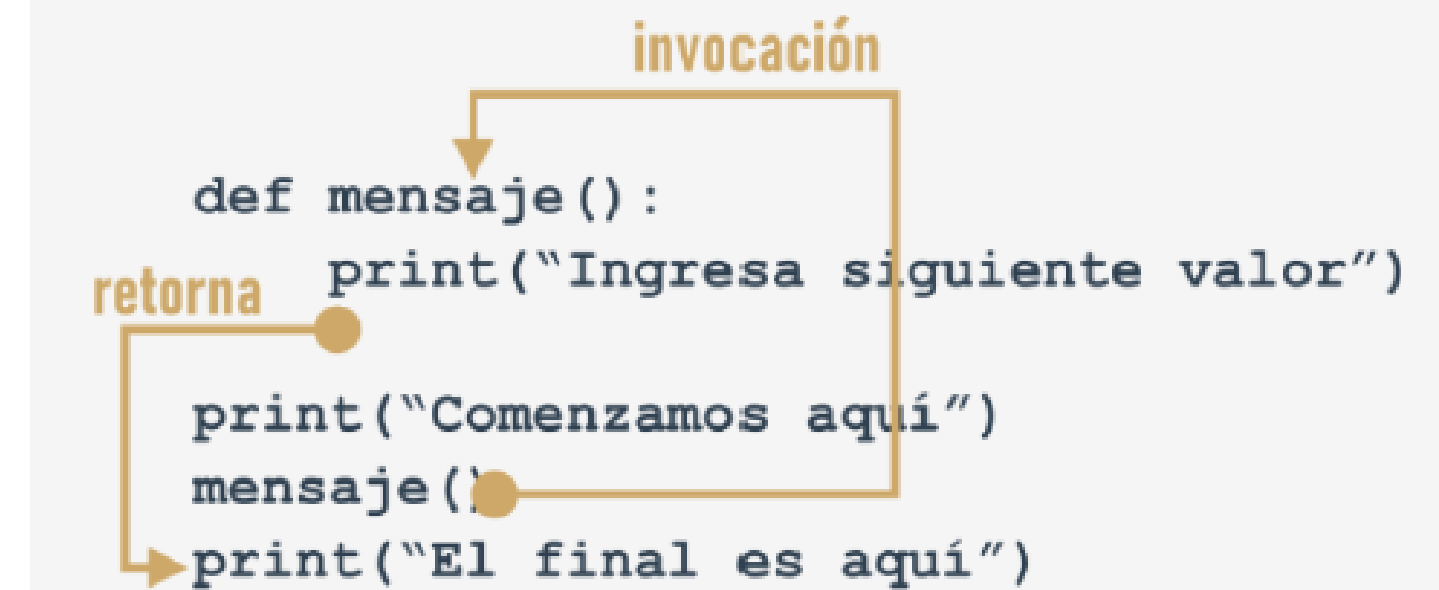
```
def NombreFunción(argumentos):  
    cuerpoFunción
```

```
def test():  
    print("Este print se llama desde la función saludar()")  
test()
```



Para llamar a una función, se usa el nombre de la función seguido de paréntesis:

```
def mensaje():  
    print("Ingresa un valor: ")  
print("Se comienza aquí.")  
mensaje()  
print("Se termina aquí.")
```



La imagen intenta mostrar el proceso completo:

- Cuando se **invoca** una función, Python recuerda el lugar donde esto ocurre y *salta* hacia dentro de la función invocada.
- El cuerpo de la función es entonces **ejecutado**.
- Al llegar al final de la función, Python **regresa** al lugar inmediato después de donde ocurrió la invocación.



Puntos clave de ejecución

No invocar una función antes de ser definida.

No indicar argumento si la función no los tiene.

Dentro de una función podemos utilizar variables y sentencias de control.

```
def listado():  
    for i in range(1,11,1):  
        print(i,end=" ")  
listado()
```

Retorno de valores

Para comunicarse con el exterior, las funciones pueden devolver valores al proceso principal gracias a la instrucción **return**.

En el momento de devolver un valor, la ejecución de la función finalizará:

```
def test():  
    return "Una cadena retornada"  
test()
```

▶ Los valores devueltos se tratan como valores literales directos del tipo de dato retornado:

```
def test():  
    return 123456  
print(test())
```

▶ También se puede operar directamente en el return, como se puede ver en este ejemplo.

```
def suma_de_dos_numeros(num1, num2):  
    return num1 + num2
```

Funciones con parámetros/argumentos

- La información se puede pasar a funciones como argumentos.
- Los argumentos se especifican después del nombre de la función, entre paréntesis. Puede agregar tantos argumentos como desee, solo sepárelos con una coma.
- Si la función espera 2 argumentos, debe llamar a la función con 2 argumentos, ni más ni menos.

```
def hola(nombre): # definiendo una función
    print("Hola,", nombre) # cuerpo de la función

nombre = input("Ingresa tu nombre: ")

hola(nombre) # invocación de la función
```

```
def suma(a, b, c):
    print(a, "+", b, "+", c, "=", a + b + c)
Suma(2,3,3)
#Salida consola: 2 + 3 + 3 = 8
```

```
def mi_funcion(f_nombre, l_nombre):
    print(f_nombre + " " + l_nombre)

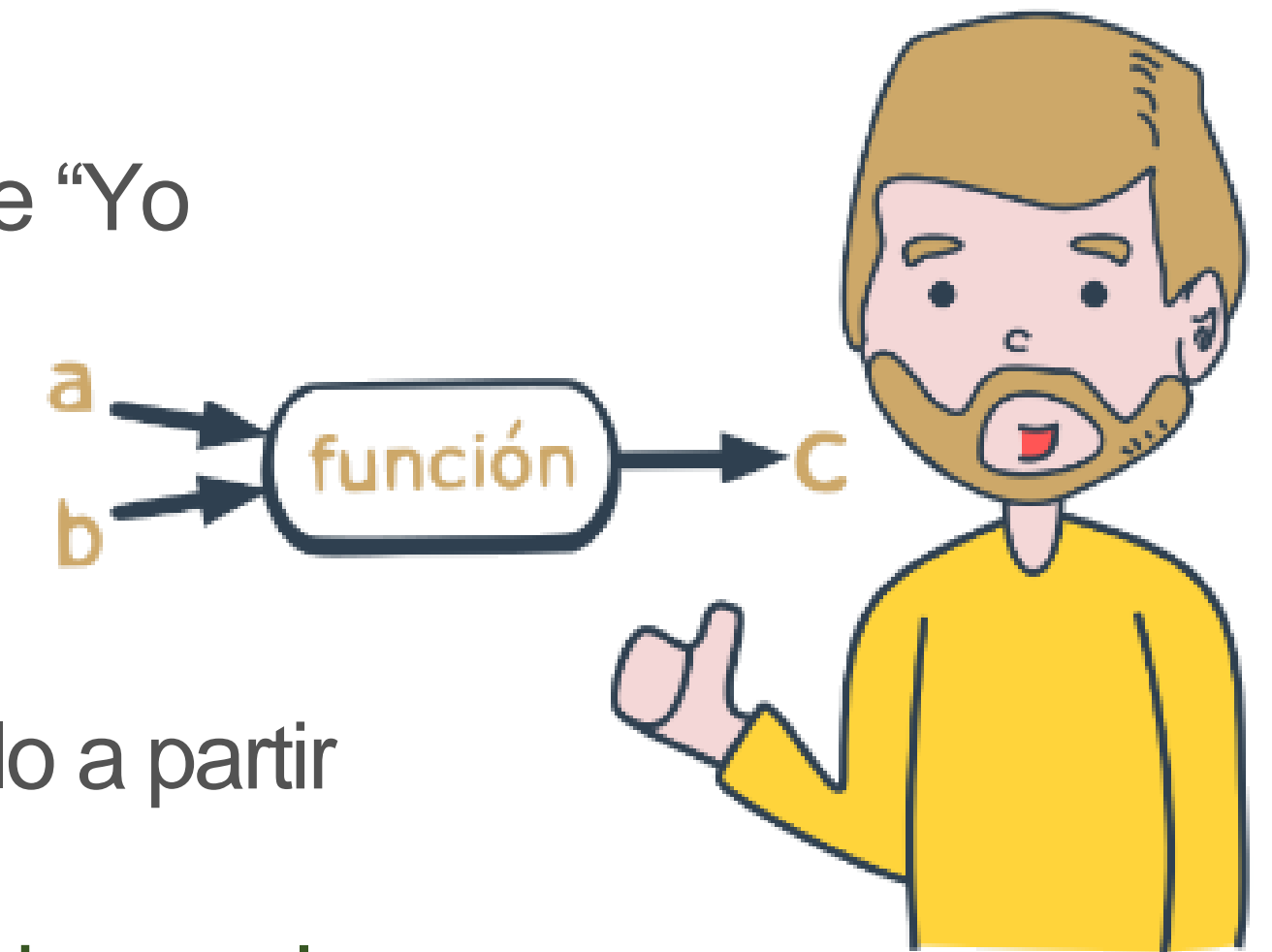
mi_funcion("Alumno", "Programador")
```

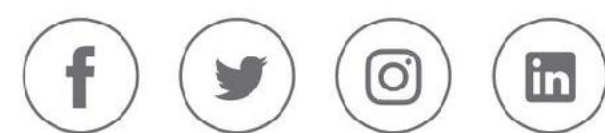


```
mi_funcion(l_nombre="Alumno", f_nombre="Programador")
#argumentos por nombre
```


Ejemplos/Ejercicios

- Crear una función que pida la ciudad de residencia y que devuelva como mensaje “Yo vivo en + la ciudad ingresada”.
- Crear una función que indique si un número ingresado por teclado es par.
- Realiza una función llamada `area_circulo(radio)` que devuelva el área de un círculo a partir de un radio. Calcula el área de un círculo de 5 de radio.
#El área de un círculo se obtiene al elevar el radio a dos y multiplicando el resultado por el número pi. Puedes utilizar el valor 3.14159 como pi o importarlo del módulo math.
- Crear una función que reciba una frase por teclado y retorne la última letra de la frase ingresada.





inacap.cl