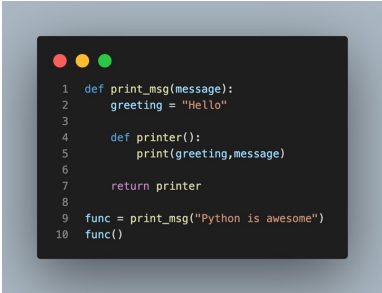


# Funciones Originales

**"El arte de la simplicidad es un rompecabezas complejo."**

-Las funciones son "áreas de código separadas" que se ejecutan independientemente del resto del código.



```
1 def print_msg(message):
2     greeting = "Hello"
3
4     def printer():
5         print(greeting,message)
6
7     return printer
8
9 func = print_msg("Python is awesome")
10 func()
```

Figura 3.1. Ejemplo de una función.

# Funciones Originales

## Sintaxis de funciones

```
def NombreFuncion(Listaparametros) :  
    codigo normal  
    ...  
    return Resultados
```

La lista de parámetros es una lista de nombres de variables separados por comas. Estos nombres de variables se utilizan para "canalizar información hacia la función desde el exterior"(transferencia de valores). Entiéndase la lista de parámetros como un punto en el que se crean variables.

# Ejemplo

## Función exponencial

```
def Funcionexp(y0) :  
    y = 1.0  
    denominador = 1.0  
    resultado = 1.0  
    iteraciones = 10  
    for k in range(1, iteraciones) :  
        y *= y0  
        denominador *= k  
        resultado += y / denominador  
    return resultado  
  
NroEuler = Funcionexp(1.0)  
print(f"el numero de euler es {NroEuler} ")  
print(f" $e^3 = \{Funcionexp(3.0)\}$ ")
```

# Asignar variables por valor o por referencia

Nos hemos dado cuenta de que las variables que utilizamos en una función están separadas del resto del código. ¿Pero qué pasa con los parámetros?

ejemplo:

```
def tempovar(nd) :  
    nd.append(1)  
    nd = nd + [2]  
    print("in tempovar: nd =", nd)  
nd = []  
tempovar(nd)  
print("a nivel de modulo: nd =", nd)
```

# Variables Locales

Nota: Siempre verifique la indentación de las líneas de comandos.

## Locales vs Globales

Digamos que definimos un string `¡Me encanta Panama en verano!`, antes de llamar a `f()`. El cuerpo de `f()` consiste únicamente en la línea `print(s)`. Como no hay variable local `s`, es decir, no hay asignación a `s`, se utilizará el valor de la variable global `s`. La salida será el string: *"¡Me encanta Panama en verano!"*. Preguntas, ¿qué ocurrirá si cambia el valor de `s` dentro de `f()`? ¿Afectará también a la variable global? :

```
def f():  
    s = "Me encanta Miami!"  
    print(s)  
s = "Me encanta Panama!"  
f()  
print(s)
```

# Argumentos opcionales: Parámetros Keywords

El scope permite utilizar parámetros posicionales, parámetros de palabra clave, parámetros variádicos y parámetros de **palabra clave** variados en cualquier combinación.

ejemplo:

```
def introperso(**kwargs) :  
    print("algunos detalles acerca de mí:")  
    for (key, value) in kwargs.items():  
        print(key, ": ", value)  
  
introperso(  
    apodo="El bicho",  
    talentos="tomar cerveza",  
    Nrodecel="pregunta amablemente"  
)
```