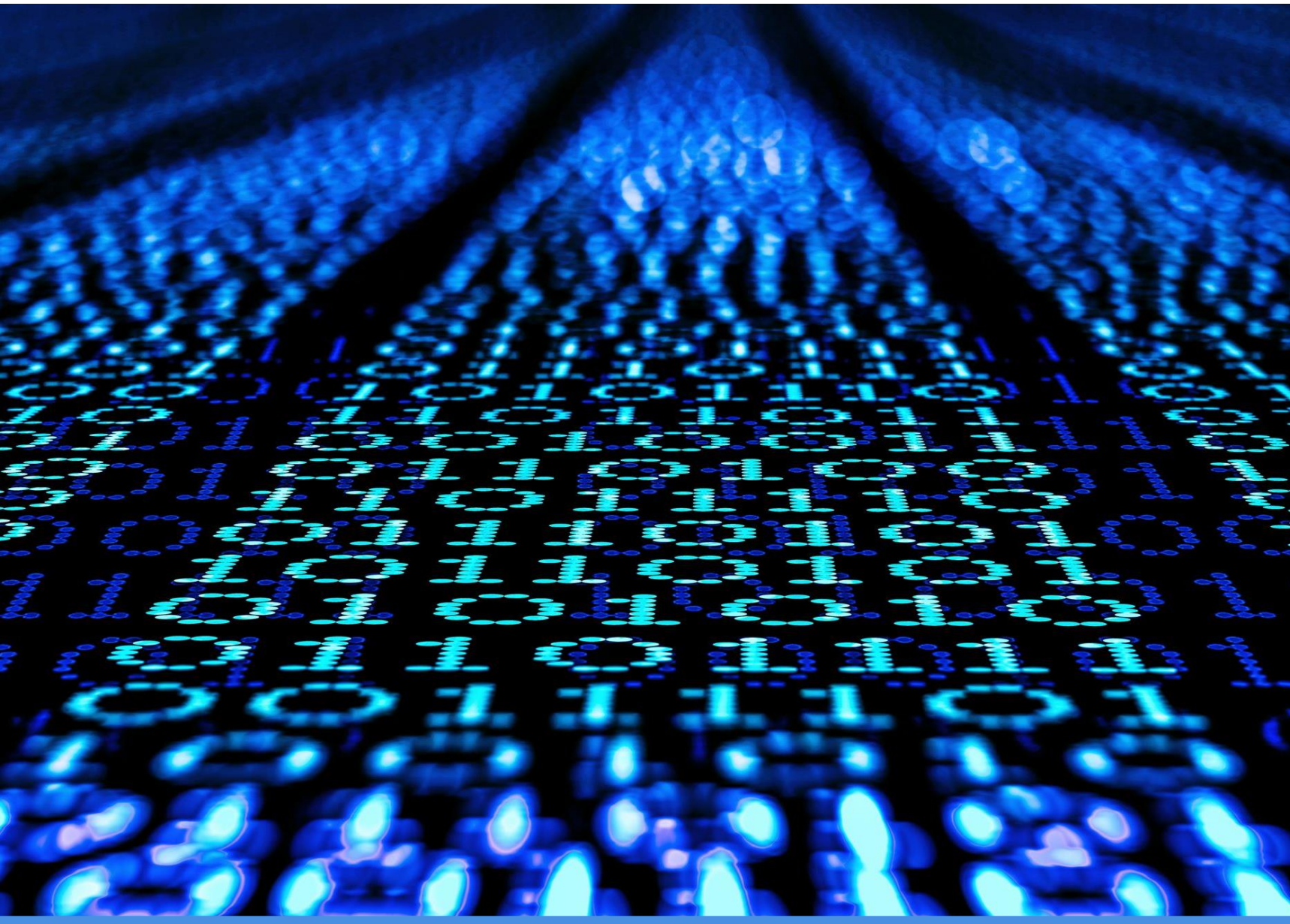


BLOQUES DE CÓDIGO  
REUTILIZABLE PARA  
UN MEJOR  
RENDIMIENTO

132  
 133  
 134  
 135  
 136  
 137  
 138  
 139  
 140  
 141  
 142  
 143  
 144  
 145  
 146  
 147  
 148  
 149  
 150  
 151  
 152  
 153  
 154  
 155  
 156  
 157  
 158  
 159  
 160  
 161  
 162  
 163  
 164  
 165  
 166  
 167  
 168  
 169  
 170  
 171  
 172  
 173  
 174  
 175  
 176  
 177  
 178  
 179  
 180  
 181  
 182  
 183  
 184  
 185  
 186  
 187  
 188  
 189  
 190  
 191  
 192  
 193  
 194  
 195  
 196  
 197  
 198  
 199  
 200

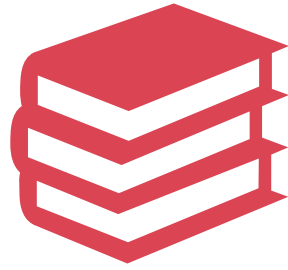


# ¿Qué son las funciones?

Las funciones son bloques de código reutilizables que realizan una tarea específica, lo que ayuda a mejorar la estructura de nuestro código y a ahorrar tiempo en la escritura de código repetitivo.

---

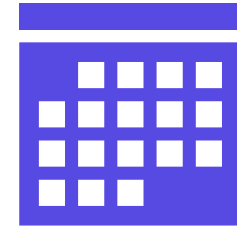
# Importancia de las Funciones



## División de Tareas Grandes

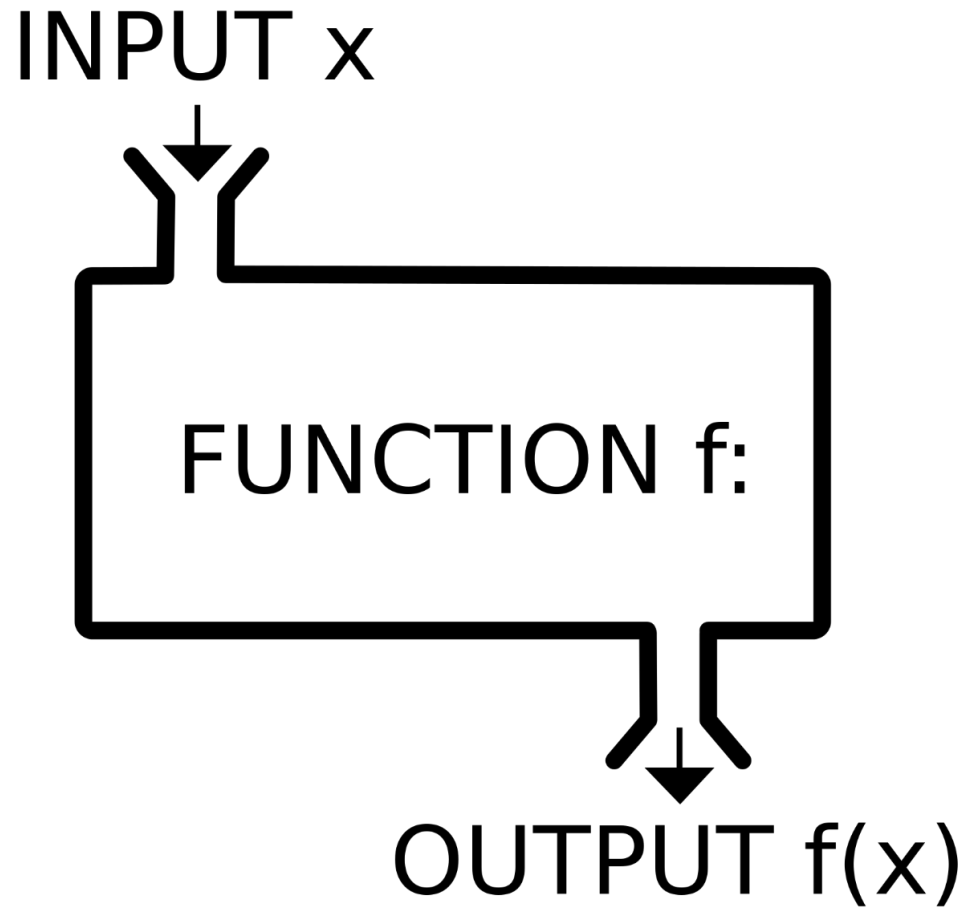
Recomendación de dividir en tareas más pequeñas

Facilita la gestión y el enfoque



## Concepto de Funciones en Programación

Procesan entradas para generar salidas  
Elemento esencial para la estructura del código



Funciones  
en Python

---

---

# Definición y Sintaxis de Funciones



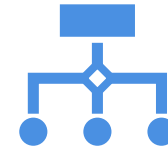
## Sintaxis de Funciones

Definir nombre y parámetros

Incluir documentación descriptiva

Agregar instrucciones necesarias

Opcional: retornar un valor



## Llamado de Funciones

Invocar con parámetros específicos

Asignar resultado a una variable (opcional)





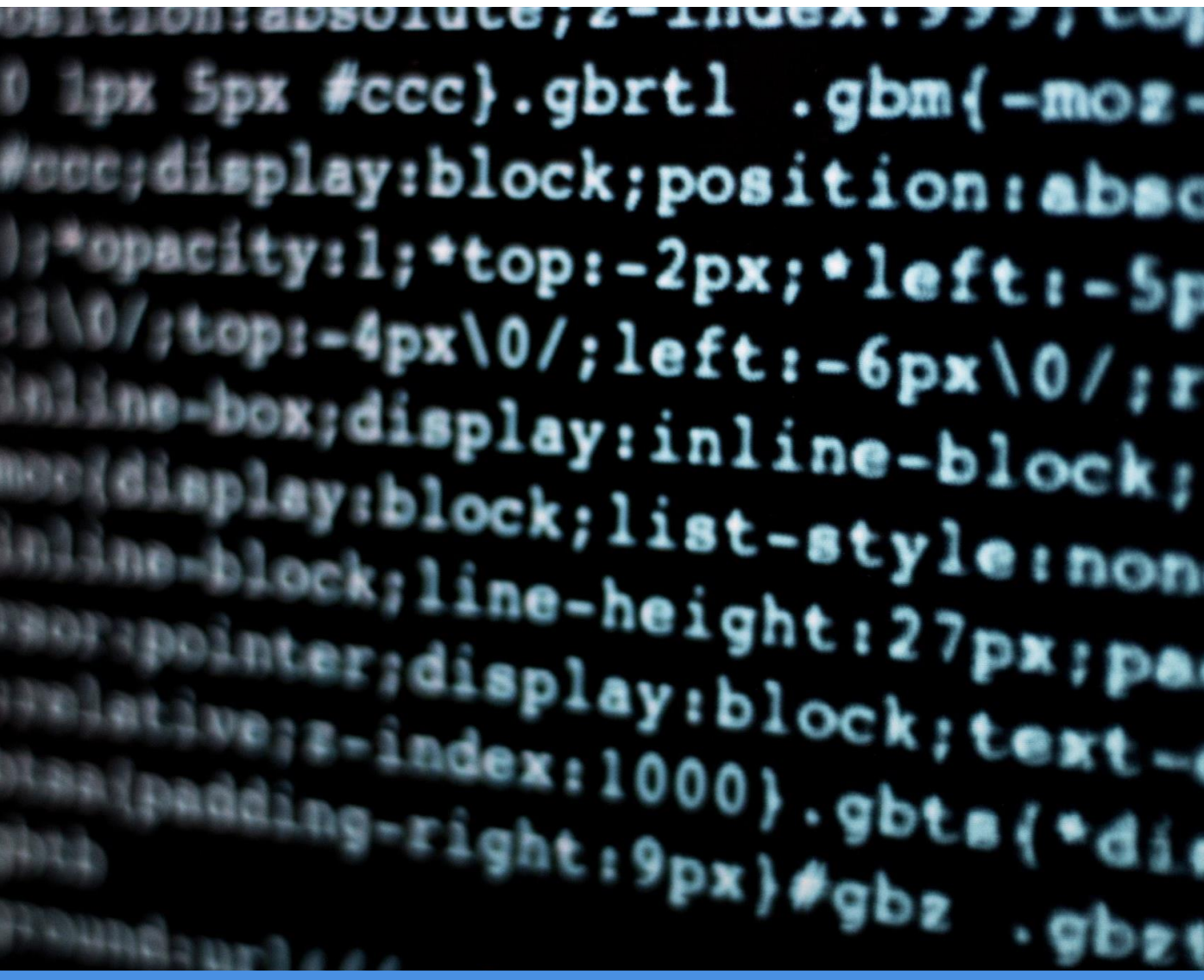
# Argumentos

## Argumentos de Posición

Los argumentos de posición son una forma de pasar argumentos a una función en Python. Se pasan en el orden en que se necesitan para realizar la tarea específica.

## Argumentos por Nombre

Los argumentos por nombre son una forma de pasar argumentos a una función en Python. Se pasan con el nombre del argumento y su valor, lo que permite una mayor flexibilidad y claridad en el código.



# Valores de retorno

## 'return' en Python

'Return' es una palabra clave en Python que se utiliza para especificar los valores que devuelve una función después de realizar una tarea específica. Es importante incluir una declaración de retorno en una función para que sepa qué valor o valores desea devolver.

## Valores de retorno en Python

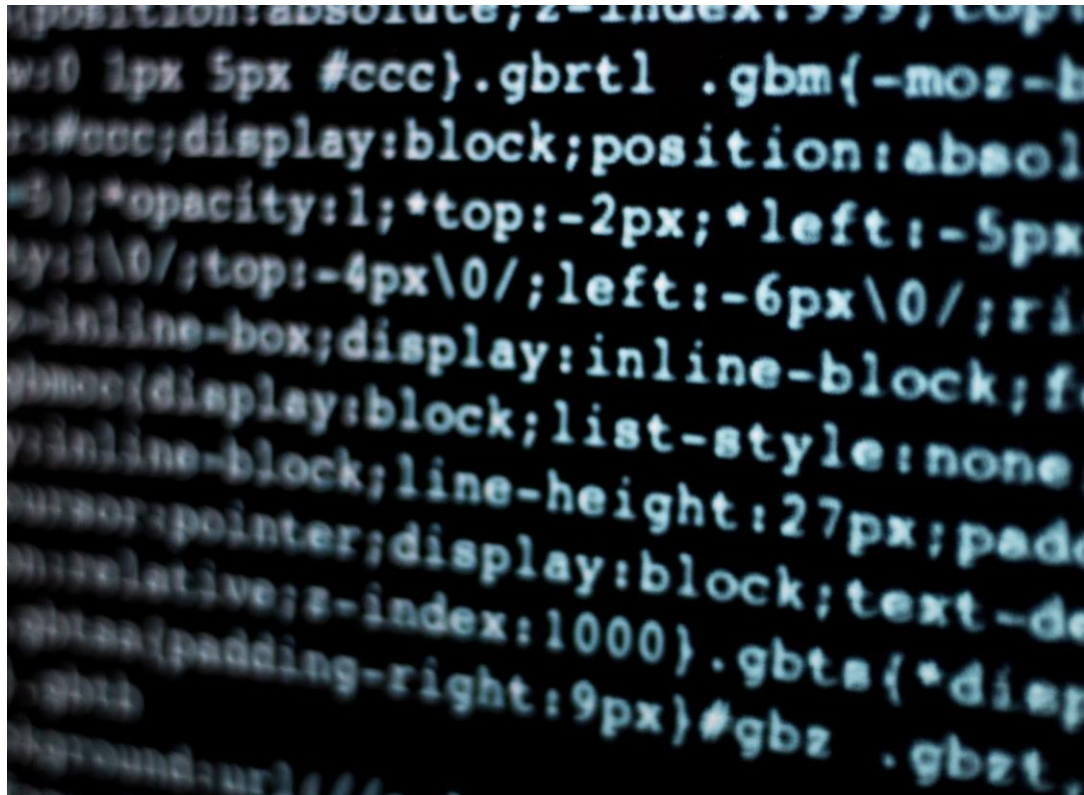
Los valores de retorno pueden ser cualquier tipo de dato en Python, como una cadena, un número, una lista o un diccionario. Se pueden devolver múltiples valores utilizando tuplas.

```
def imprime_lista(lista):  
    """Imprime el parametro lista"""  
    if isinstance(lista, list):  
        for i in range(0, len(lista)):  
            print(lista[i])  
miLista=["mango", "papaya", "piña", "sandía"]  
imprime_lista(miLista)
```

EJEMPLO DE  
DEFINICIÓN DE  
FUNCIONES:  
IMPRIME\_LISTA



# EJEMPLO DE DEFINICIÓN DE FUNCIONES: IMPRIME LISTA



- Definición de la Función
  - La función 'imprime\_lista' toma un parámetro 'lista'.
  - Utiliza un bucle 'for' para imprimir cada elemento.
- Comprobación del Tipo de Dato
  - Verifica si el parámetro es una lista con 'isinstance'.
- Uso de la Función
  - Se llama a la función con 'miLista' que contiene frutas.

# Ejemplos de Definición de Funciones: suma\_numeros

---

```
def suma_numeros(num1,num2):  
    """Suma dos numeros y regresa el resultado"""  
    suma=num1+num2  
    return suma  
  
x=suma_numeros(8,10)  
print(x)
```

# Ejemplos de Definición de Funciones: Suma\_NUMeros



## Definición de Función

La función 'suma\_numeros' toma dos argumentos: num1 y num2.



## Documentación de la Función

La cadena de documentación explica que la función suma dos números.



## Proceso de Suma

La variable 'suma' almacena la suma de num1 y num2.



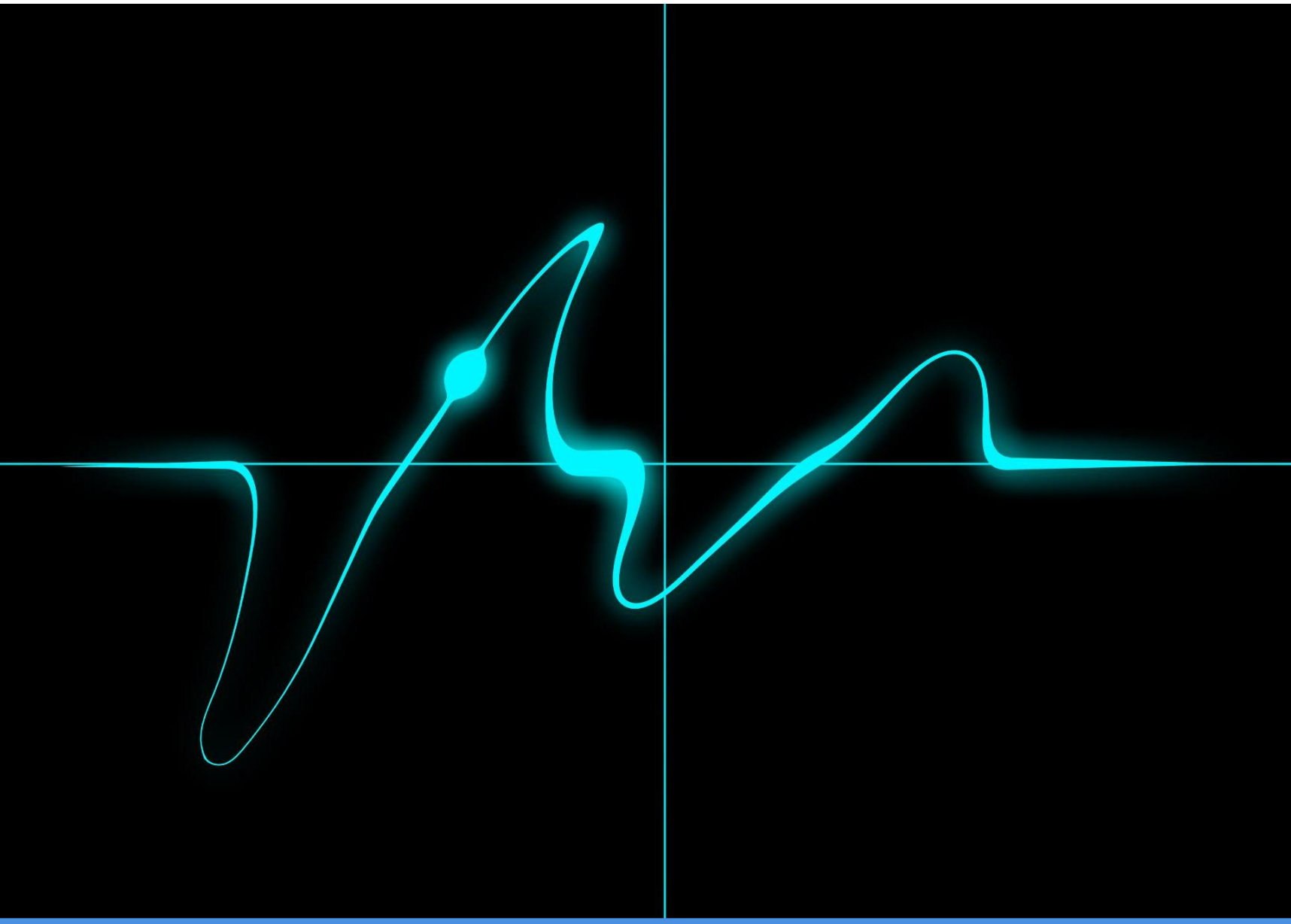
## Retorno de Resultados

La función devuelve el valor de 'suma'.



## Uso de la Función

La función se llama con 8 y 10 como argumentos y el resultado se imprime.



# Funciones Lambda

Las funciones lambda se conocen como funciones anónimas porque no necesitan un nombre para definirse y ejecutarse. Son útiles para definir funciones simples y rápidas que solo se necesitan una vez.

# Funciones Lambda o Anónimas



## Sintaxis de Funciones Lambda

Se define con la palabra clave 'lambda'  
Seguida de parámetros y una instrucción



## Uso de Funciones Lambda

Asignación a una o más variables  
Invocación con parámetros  
Una función anónima de una sola línea en Python  
Usada para operaciones simples y rápidas





# Características de las Funciones Lambda

## Definición Simplificada

- Son funciones con una forma más corta y concisa

## Restricción de Instrucción

- Limitadas a una sola instrucción por función

## Flexibilidad de Parámetros

- Capacidad para manejar múltiples parámetros

## Optimización de la Programación

- Facilitan y agilizan el proceso de codificación
-



# Valores por omisión

Los valores por omisión son aquellos que se utilizan cuando un valor de argumento no se especifica en la definición de la función. En Python, los valores por omisión se pueden definir en la definición de la función.





# Número indeterminado de parámetros

## Definición de funciones en Python

En Python, se pueden definir funciones con la sintaxis '\*' antes del nombre del parámetro, lo que permite una cantidad indeterminada de argumentos. Esto facilita la escritura de funciones para realizar tareas de procesamiento de datos complejas.

## Versatilidad en la definición de funciones

El uso de argumentos con un número indeterminado de parámetros en Python permite a los desarrolladores definir funciones que pueden manejar una amplia variedad de entradas, mejorando la versatilidad y escalabilidad del código.



# Recursividad

Las funciones recursivas son funciones que se llaman a sí mismas para resolver un problema en lugar de utilizar un bucle. En Python, las funciones pueden ser recursivas y pueden ser útiles en situaciones donde la solución requiere que se aplique la misma operación a un subconjunto más pequeño de datos hasta que se cumpla una condición de parada.



# Conclusión

## Funciones en Python

Las funciones en Python son una herramienta poderosa que permiten la reutilización de código y la creación de programas complejos y estructurados.

## Bibliotecas y Módulos

Python también admite una gran cantidad de bibliotecas y módulos que amplían la funcionalidad de las funciones.