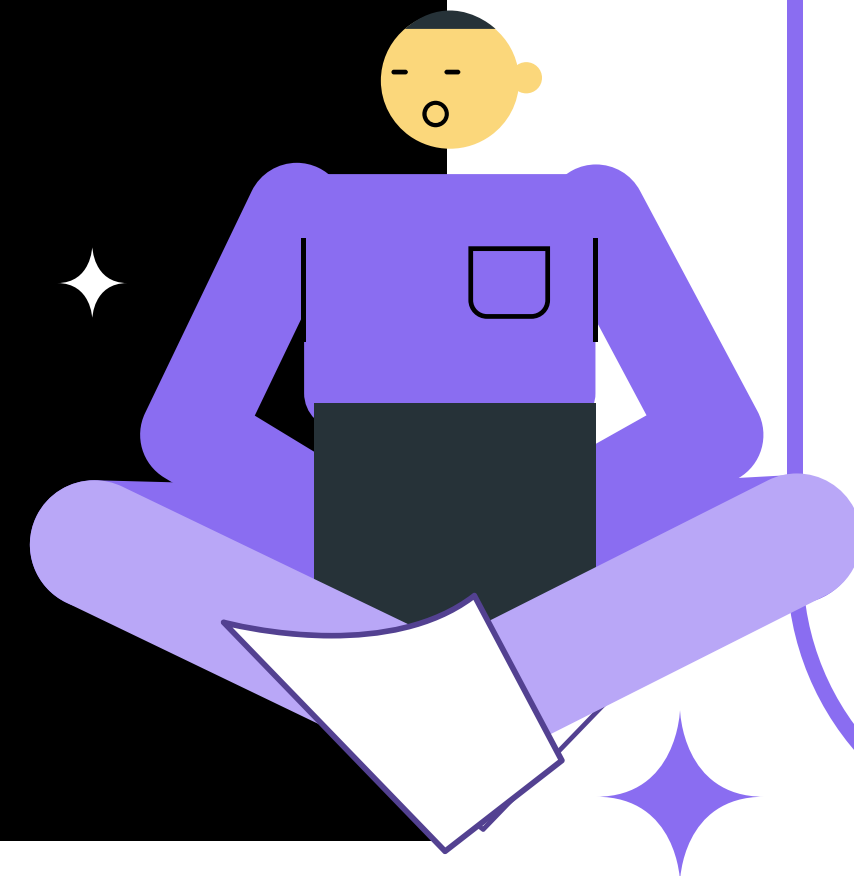


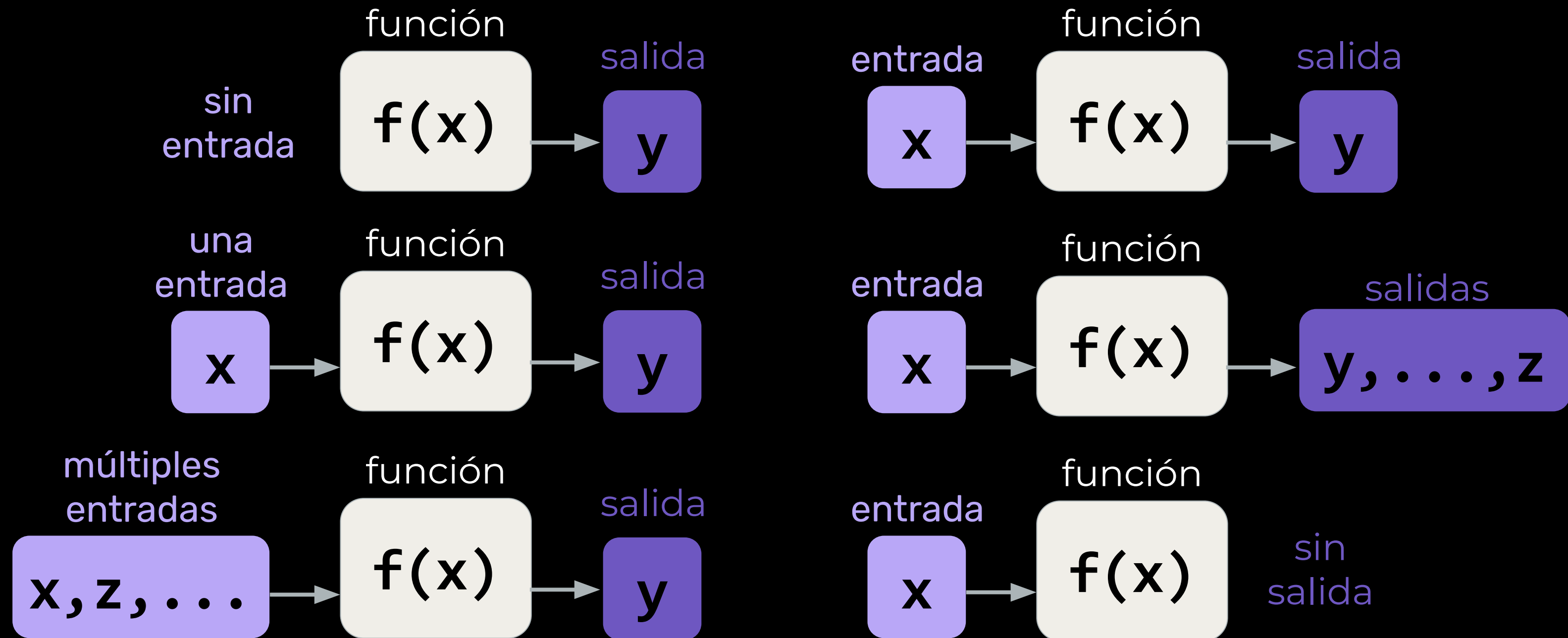
Funciones en Go

¿Qué es una **función**?

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



Una función, en general, puede recibir **ninguna**, **una** o **muchas entradas** y puede **retornar uno**, **muchos** o **ningún valor**.



Composición de una función

Cuando hablamos de declarar una función, nos referimos a cómo se construye una función:

```
func sum(a int, b int) int {  
    sum := a + b  
    return sum  
}
```

Analicemos en detalle cada **componente**.

→ **func**

Palabra reservada

→ **sum**

Nombre de la función

→ **(a int, b int)**

Lista de parámetros

→ **int**

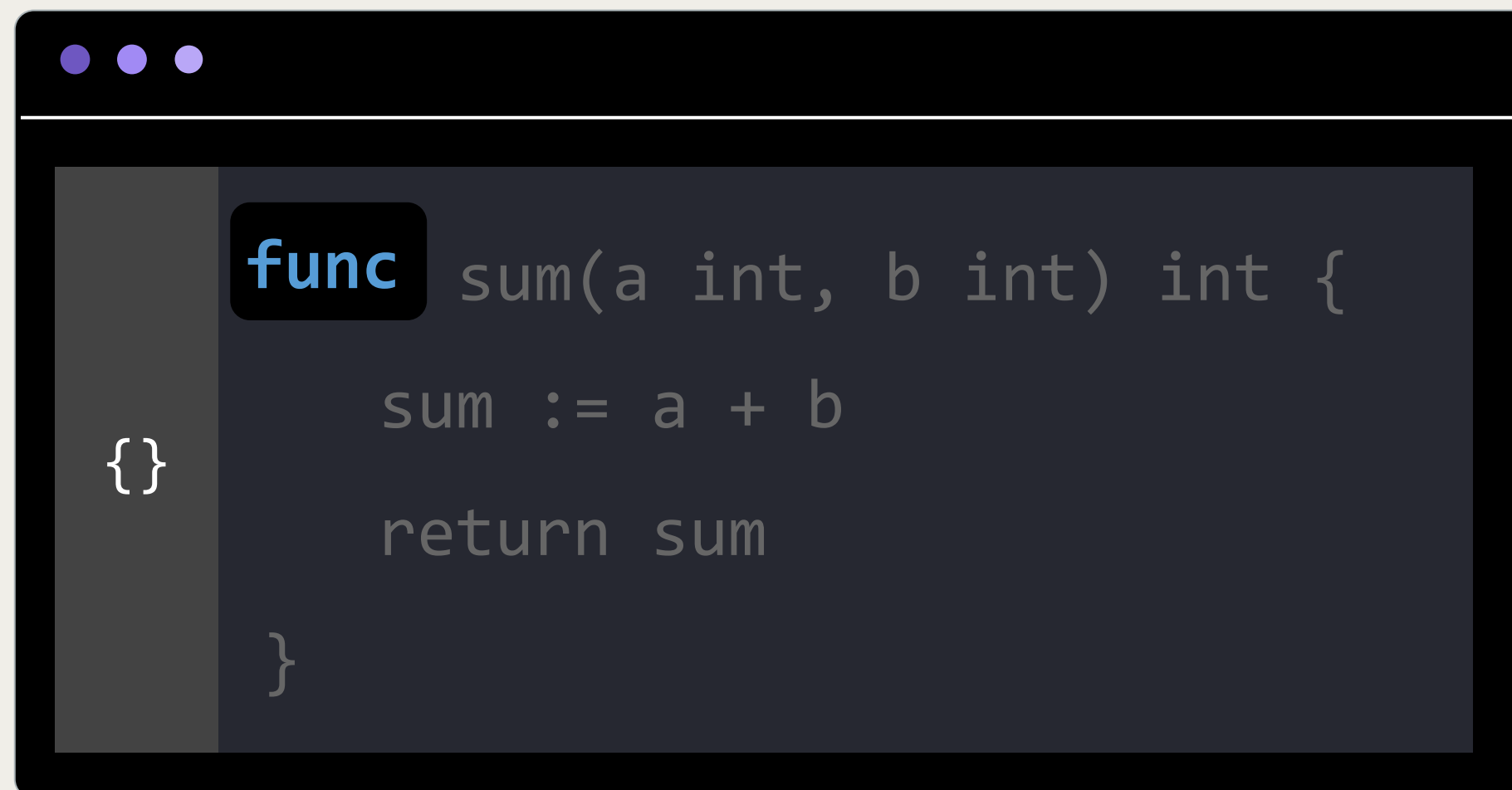
Tipo de retorno

→ **sum := a + b**
return sum

Cuerpo

Composición de una función

Cuando hablamos de declarar una función, nos referimos a cómo se construye una función:



```
func sum(a int, b int) int {  
    sum := a + b  
    return sum  
}
```

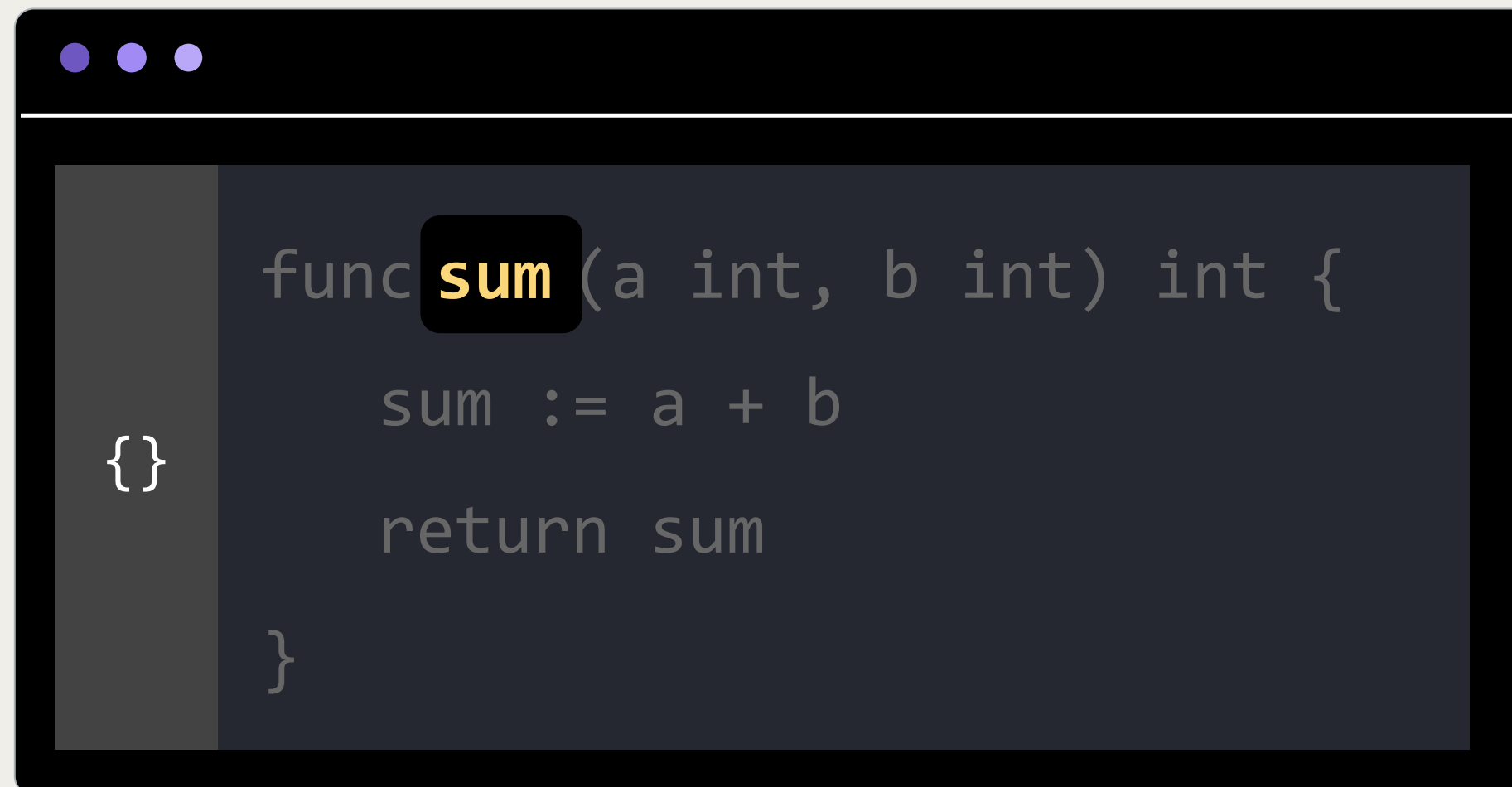
→ **func**

Palabra reservada

La palabra **func** indica el inicio de una declaración de función.

Composición de una función

Cuando hablamos de declarar una función, nos referimos a cómo se construye una función:



```
func sum (a int, b int) int {  
    sum := a + b  
    return sum  
}
```

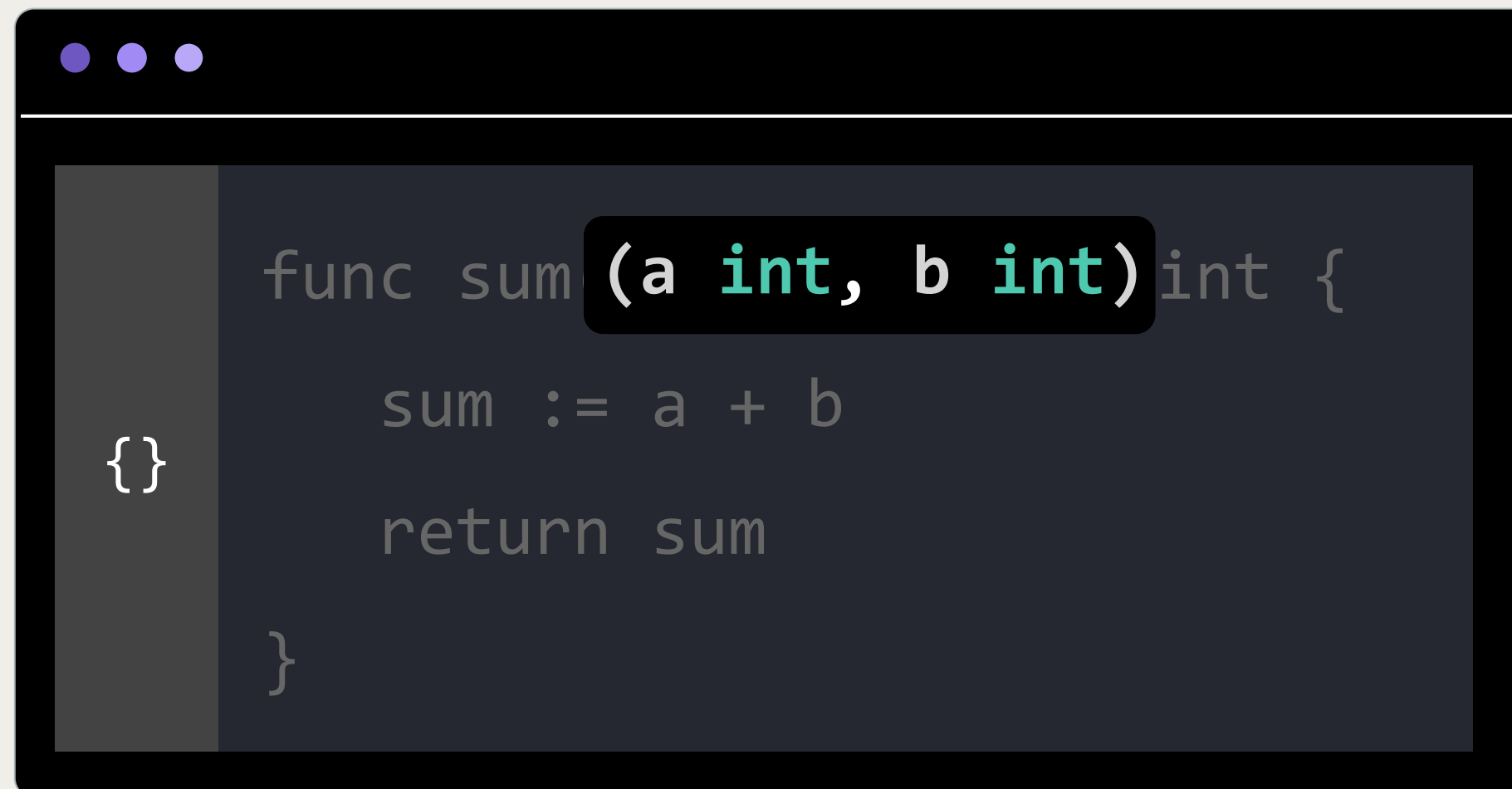


Nombre de la función

Nombre que identifica a la función. Si inicia con minúscula, solo estará disponible en el paquete donde se definió.

Composición de una función

Cuando hablamos de declarar una función, nos referimos a cómo se construye una función:



```
func sum (a int, b int) int {  
    sum := a + b  
    return sum  
}
```

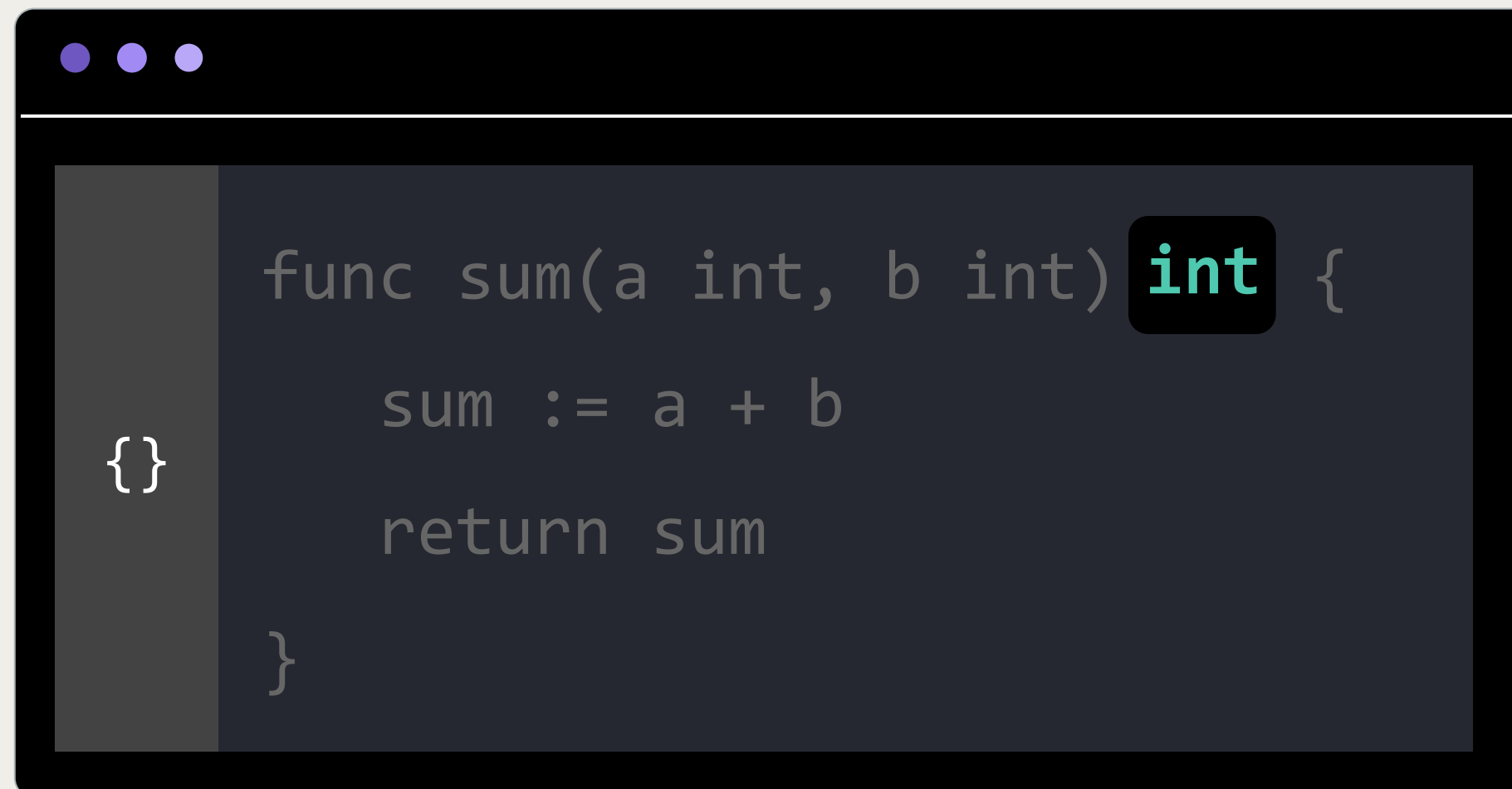
→ (a int, b int)

Lista de parámetros

Estos son los parámetros de entrada de nuestra función. Primero se coloca el nombre del parámetro **(a)** seguido del tipo de variable **(int)**. Esta lista se encierra en paréntesis **()**. Si los colocamos vacíos, indicamos que la función no recibe entradas.

Composición de una función

Cuando hablamos de declarar una función, nos referimos a cómo se construye una función:



```
func sum(a int, b int) int {  
    sum := a + b  
    return sum  
}
```

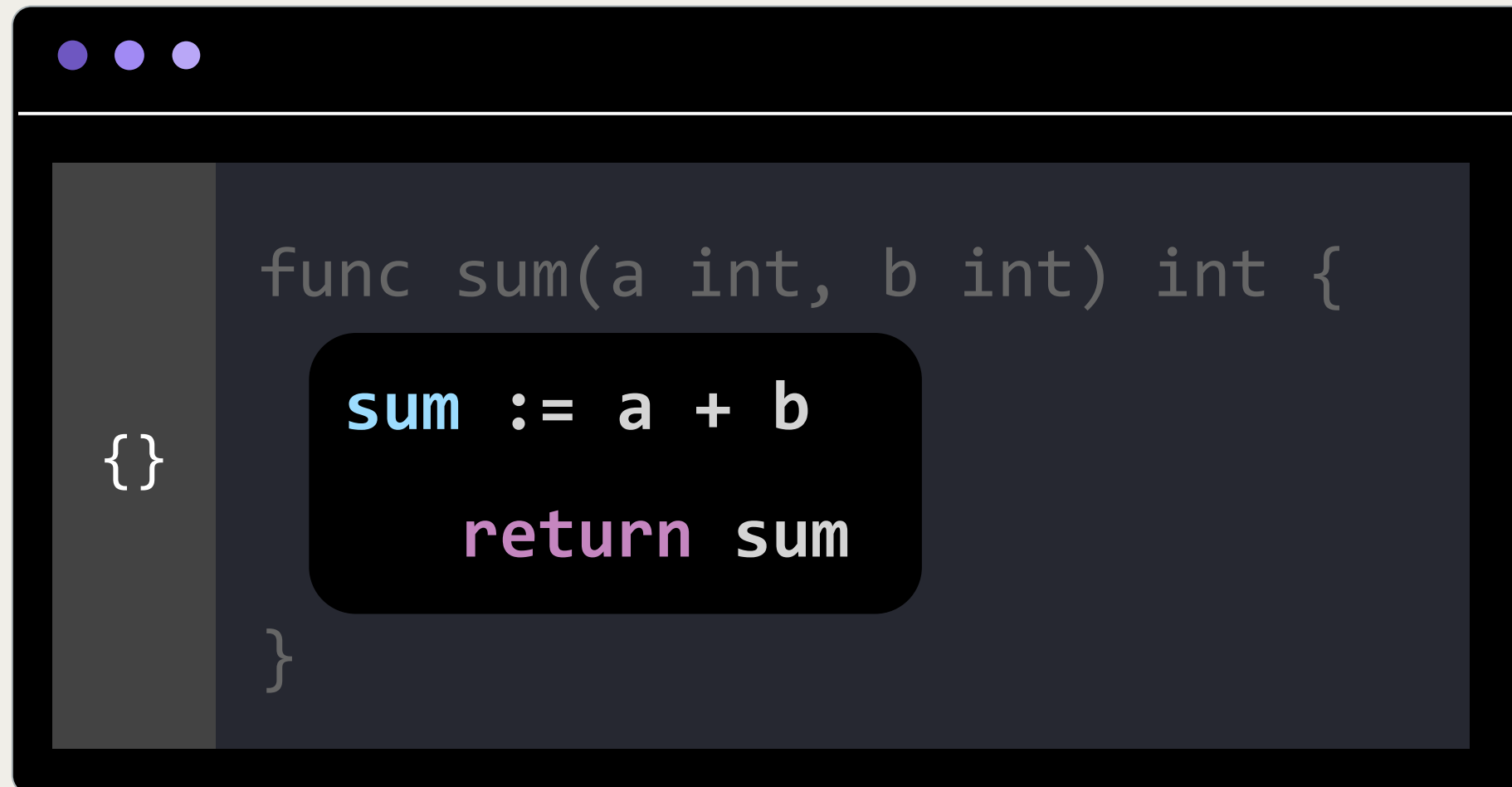
→ **int**

Tipo de retorno

Es el tipo de variable **(int)** que tiene como salida nuestra función. Si está presente, es necesario colocar un **return**. Si no colocamos el tipo de retorno, indicamos que la función no tiene valores de salida.

Composición de una función

Cuando hablamos de declarar una función, nos referimos a cómo se construye una función:



→ `sum := a + b`
`return sum`

Cuerpo

Es el bloque de código encargado de realizar la operación que cumple la función. Este bloque se encierra entre llaves `{}`. Esto define el **scope** de la función que determina el contexto de ejecución de nuestra función. Es decir, todas las variables que se definan dentro de este espacio solo estarán disponibles dentro de él.

Llamado de función

```
func main() {  
    c := sum(5, 5)  
    fmt.Println(c)  
}  
  
func sum(a int, b int) int {  
    sum := a + b  
    return sum  
}
```

→ **sum(5, 5)**

Para llamar una función basta con poner su nombre seguido de paréntesis **()**. En caso de que la función reciba parámetros, deben ir entre ellos. Estos serán asignados en el orden definido en la declaración. Si la función retorna un valor al momento de llamarla, debemos guardar su resultado en una variable.

¡Muchas gracias!