

Retorno de un valor

Sintaxis de una función con retorno de valor

Para definir a una función que esperamos que nos devuelva un valor, tenemos que indicarle el tipo de dato que esperamos al final de la función. En este caso, creamos una función que reciba dos parámetros y nos devuelva la sumatoria de ellos.

Parámetros de la función.

{}

```
func sum(value1, value2 float64) float64 {  
    return value1 + value2  
}
```

Tipo de dato que va a retornar la función.

Asignación del valor a una variable

Luego de declarar nuestra función con retorno de un valor, a ese valor se lo asignamos a una variable. En este caso, el resultado de la suma se lo asignamos a la variable **result** para luego mostrarla por consola.

```
{}  
func main() {  
    result := sum(4, 5)  
    fmt.Println(result)  
}
```

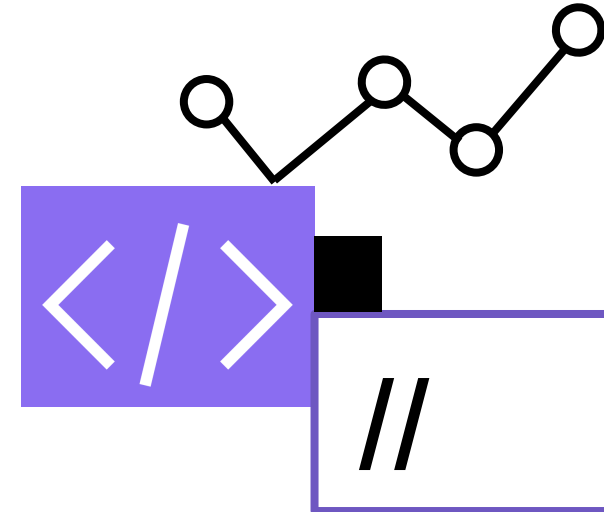


A codear...

Vamos a realizar un ejemplo un poquito más complejo: una función a la cual le pasaremos dos valores y le indicaremos qué operación queremos realizar.

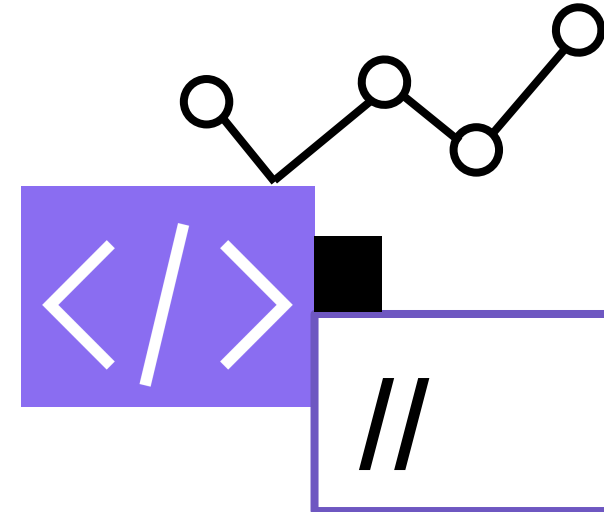
Para ello, definiremos cuatro constantes con las operaciones que queremos realizar, que son las cuatro variables matemáticas principales.

```
{ }  
const (  
    Suma    = "+"  
    Resta   = "-"  
    Multip  = "*"  
    Divis   = "/"  
)
```



Generamos una función que se encargará de orquestar el tipo de operación que le definiremos por parámetro.

```
func operacionAritmetica(valor1, valor2 float64, operador string) float64 {  
    switch operador {  
    case Suma:  
        return valor1 + valor2  
    case Resta:  
        return valor1 - valor2  
    case Multip:  
        return valor1 * valor2  
    case Divis:  
        if valor2 != 0 {  
            return valor1 / valor2  
        }  
    }  
    return 0  
}
```



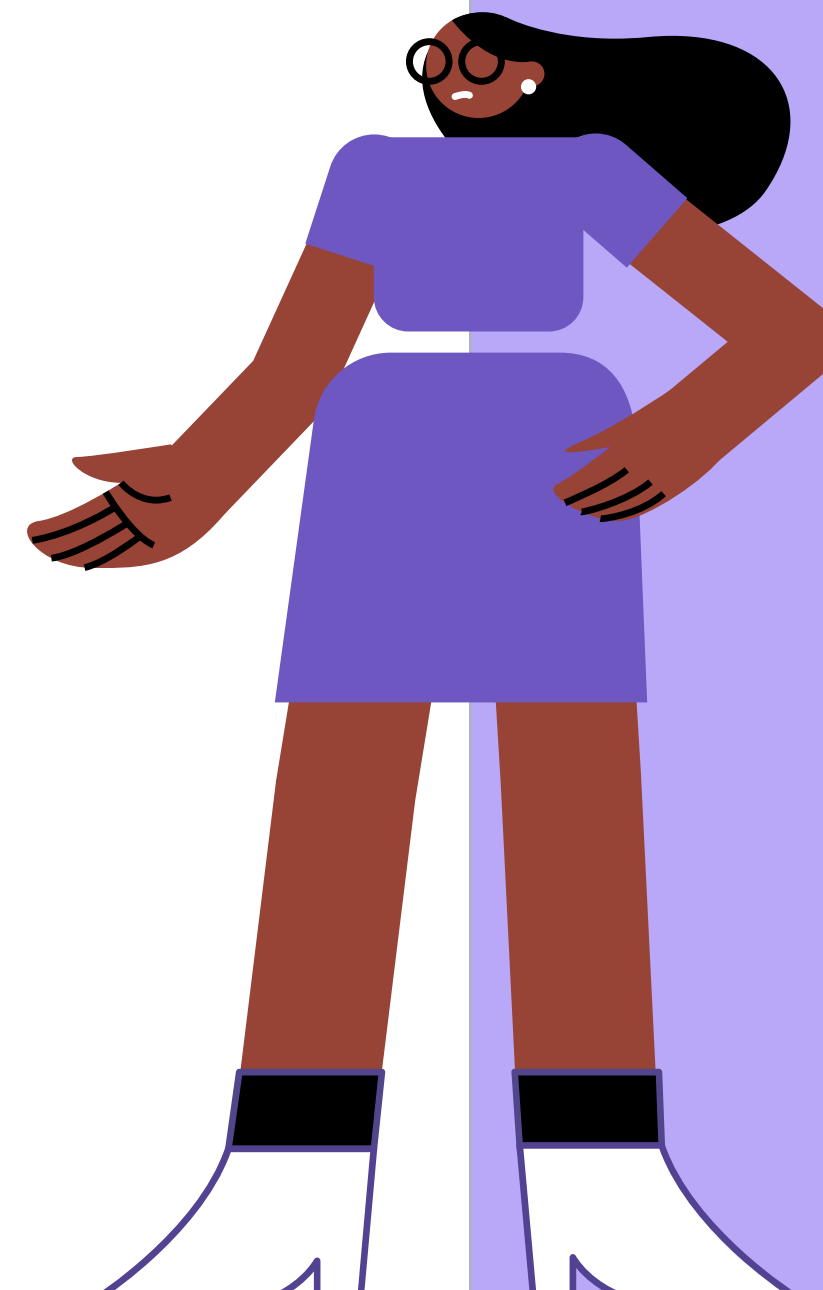
Y, por último, llamaremos a esa función y le indicaremos —con las constantes que definimos— qué operación queremos realizar.

```
{  
func main() {  
    fmt.Println(operacionAritmetica(6, 2, Suma))  
    fmt.Println(operacionAritmetica(6, 2, Resta))  
    fmt.Println(operacionAritmetica(6, 2, Multip))  
    fmt.Println(operacionAritmetica(6, 2, Divis))  
}
```

Conclusiones

A diferencia de una función tradicional, en Go podemos especificar explícitamente qué tipo de valor va a retornar la función, para luego asignar dicho valor retornado a una variable.

Más adelante veremos que podemos retornar más de un valor.



¡Muchas gracias!