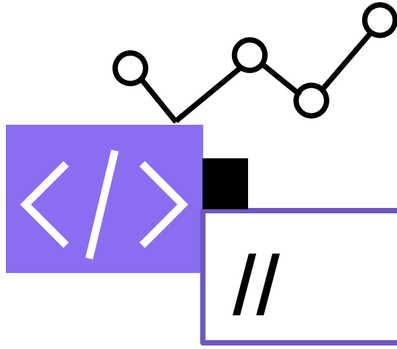
DIGITALHOUSE

Punteros

Certified Tech Developer

The Ultimate Degree

Un puntero es un tipo de dato cuyo valor es una dirección de memoria que se refiere o "apunta a" otra variable

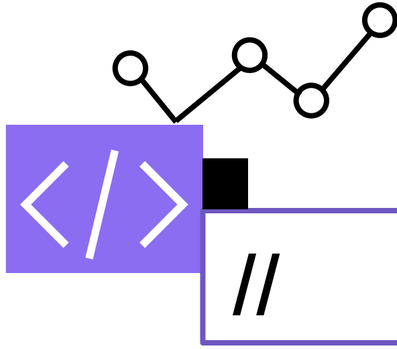


¿Cómo creamos punteros?

Una de las ventajas de Go es que nos permite trabajar con punteros. Para crearlos, colocamos el operador “*” antes de definir el tipo de dato que necesitamos almacenar en esa dirección de memoria.

```
{}
var p *int
```

En la variable **p** tendremos un puntero valor de tipo de dato int.



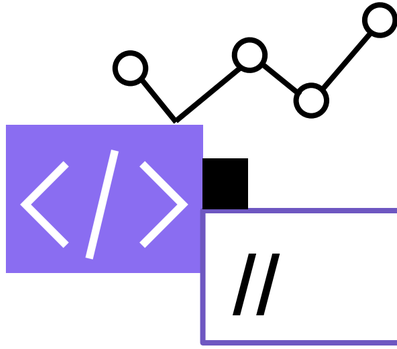
Otras formas de crear punteros

- Utilizando la función **new()** la cual recibe como argumento un tipo de dato.
- A través del shorthand de declaración de variables “**:=**”.

```
{}
```

```
var p1 *float64
var p2 = new(float64)
var v float64
p3 := &v
```

El tipo de datos de las variables **p1**, **p2** y **p3** es un puntero de tipo flotante.

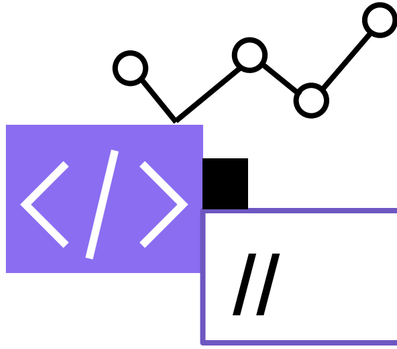


Operador de dirección

Para obtener la referencia o dirección de memoria de una variable debemos anteponer, a la variable, el operador de dirección “**&**”.

```
var v int = 50
fmt.Println("La dirección de memoria de la variable v es: ", &v)
```

Podemos ver que la variable **v** de tipo entero tiene el valor “50” y se almacena en una dirección de memoria que tendrá un formato del estilo **0x70158811**.

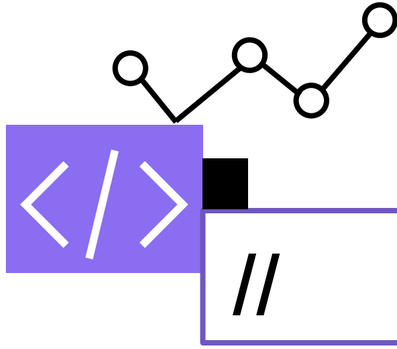


Operador de desreferenciación

Desreferenciar un puntero es obtener el valor que está almacenado en la dirección de memoria a donde hace referencia el puntero. Para hacerlo debemos anteponer el operador “*” a la variable puntero.

{}

```
var v int = 50
var p *int
// Hacemos que el puntero p, almacene (apunte a) la dirección de memoria de la variable v.
p = &v
fmt.Printf("El puntero p apunta a la dirección de memoria: %v \n",p)
fmt.Printf("Al desreferenciar el puntero p obtengo el valor de la variable a la cual apunta, es decir, el valor de v: %d \n",*p)
```



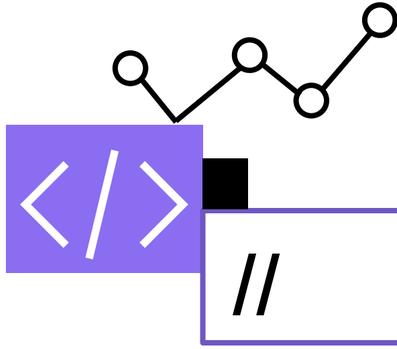
El código anterior produce el siguiente resultado:

```
TERMINAL

 go run main.go

El puntero p referencia a la dirección de memoria: 0x70158811
Al desreferenciar el puntero p, obtengo el valor: 50
```

Podemos ver que el puntero **p** referencia a la dirección de memoria **0x70158811**, y para obtener el valor “50” almacenado en esta dirección, hacemos uso del operador de desreferenciación ***p**.

Veamos un ejemplo:

```go
package main
import "fmt"
// La función incrementar recibe un puntero de tipo entero
func Incrementar(v *int) {
    // Desreferenciamos la variable v con el operador “*”
    // para obtener su valor e incrementarlo en 1
    *v++
}
func main() {
    var v int = 50
    // La función Incrementar recibe un puntero
    // utilizamos el operador de dirección “&”
    // para obtener la dirección de memoria de v
    Incrementar(&v)
    fmt.Println("El valor de v ahora vale: ", v)
}
```



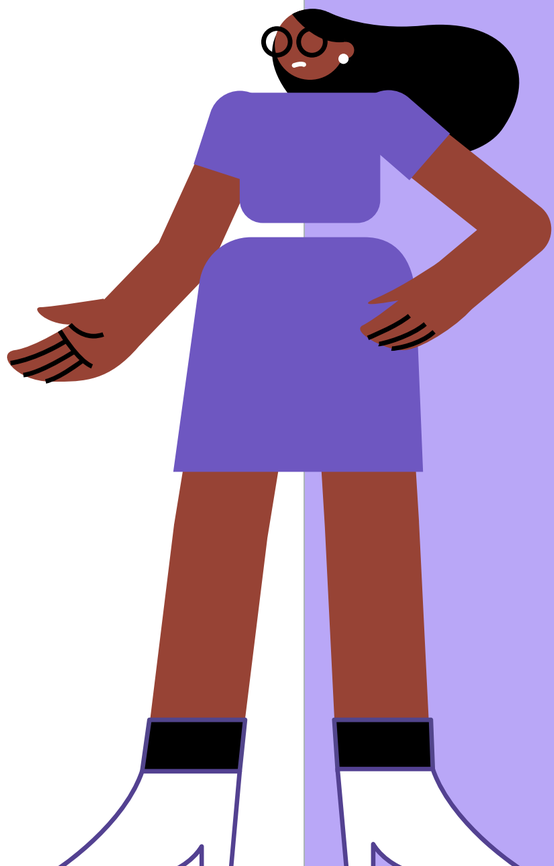
En el ejemplo podemos ver que se pueden pasar punteros como parámetros a una función utilizando los operadores “*” y “**&**”. También vemos el comportamiento de estos al incrementar el valor de la variable **v**. El código produce el siguiente resultado:

```
TERMINAL

go run main.go

El valor de v ahora vale: 51
```

Conclusiones

Como vimos, un puntero es una dirección de memoria que hace referencia a otro valor. También aprendimos el uso de los operadores para obtener la dirección y la desreferenciación de un puntero.

Es importante recordar que en Go los argumentos de las funciones se pasan por valor, por eso el uso de punteros es fundamental.



DIGITALHOUSE

¡Muchas gracias!