

Pointers

Why Pointers?

In many programming tasks, we need to **share or modify data without creating unnecessary copies**. Go, like C and C++, provides **pointers** to achieve this

```
fmt.Println("Enter the PIN: ")  
fmt.Scanln(enteredPin)
```



Copy

```
fmt.Println("Enter the PIN: ")  
fmt.Scanln(&enteredPin)
```

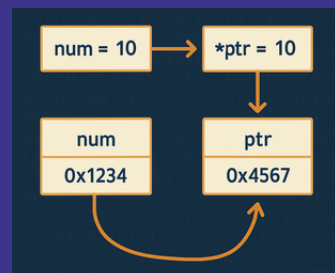


Pointer (reference to the original variable)

They allow us to reference a memory location directly

This enables functions to modify values outside their scope

They improve performance by avoiding deep copies of large data

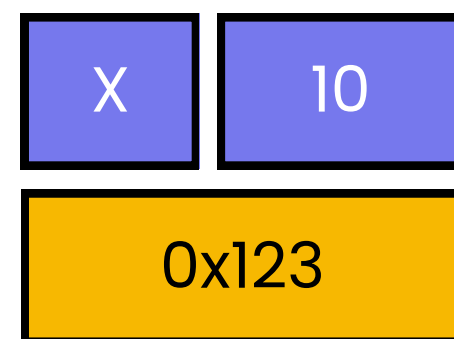


Pointers

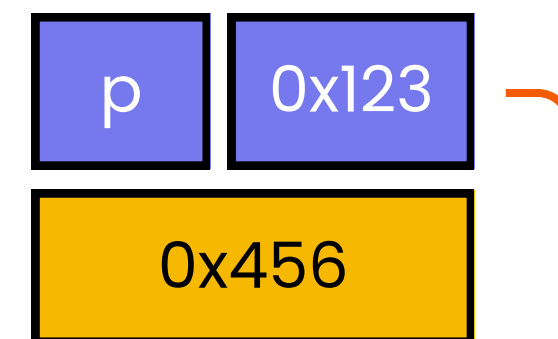
What are Pointers?

A pointer is a variable that **stores the memory address** of another variable.

$x := 10$



$p := \&x$



```
fmt.Println("Enter the PIN: ")  
fmt.Scanln(enteredPin)
```

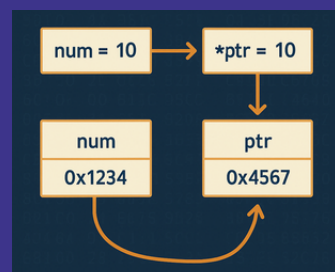
Copy



```
fmt.Println("Enter the PIN: ")  
fmt.Scanln(&enteredPin)
```

Pointer (reference to the original variable)





Pointers

Creating Pointers

```
var ptr *int // pointer to an int
```

```
x := 42
```

```
ptr = &x // assign address of x to ptr
```