Why do we need pointer?

To understand the need of pointer let’s look at this example:

package main

import (

    "fmt"

)

func heroName(x string) {

    x = "Batman"

}

func main() {

    x := "Bruce Wayne"

    fmt.Println("value of x after function call ->", x)

    heroName(x)

    fmt.Println("value of x after function call ->", x)

}

output:



In this case, even though we have defined the function call to update the name, but there was no modification to the original value of x in the main function. Now why is this happening?

To understand it we will slightly recap concept of variables.

Variables are the names given to a memory location where the actual data is stored. To access the stored data, we need the address of that memory location.

In our case when we pass variable x to the function heroName, we are passing the value. Now in this method, the value of each variable in calling function is copied into corresponding dummy variables of the called function. And the changes made to dummy variable in the called function have no affect on the original value in the calling function.

Thus, we cannot alter the values of actual variables through function calls.

So, we need to find out the way to access the memory address of variable x and then change the value inside it. That is what Pointer do. Pointer point to the address to a variable, so using that address we can easily update what is stored in it.

So, the above code becomes:

package main

import (

    "fmt"

)

func heroName(x \*string) {

    \*x = "Batman"

}

func main() {

    x := "Bruce Wayne"

    fmt.Println("value of x before function call ->", x)

    heroName(&x)

    fmt.Println("value of x after function call ->", x)

}

Output:



So, if you see the code, we are accessing the address of variable “x” using “&” operator and instead of passing the value we are now passing the address to the function call in the main function.

And inside heroName function we have declared a string type pointer in the parameter, using the address received we would have access to the actual variables and hence we would be able to manipulate it and we the same we are doing by dereferencing the value of this “x” pointer with a new value. Thus, the actual value gets changed from “Bruce Wayne” to “Batman”.

And this technique is also called Pass by reference.