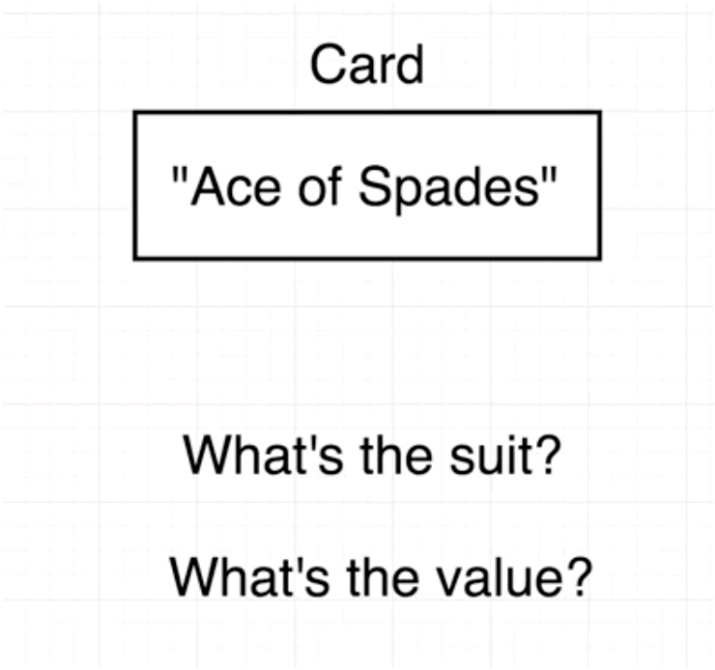
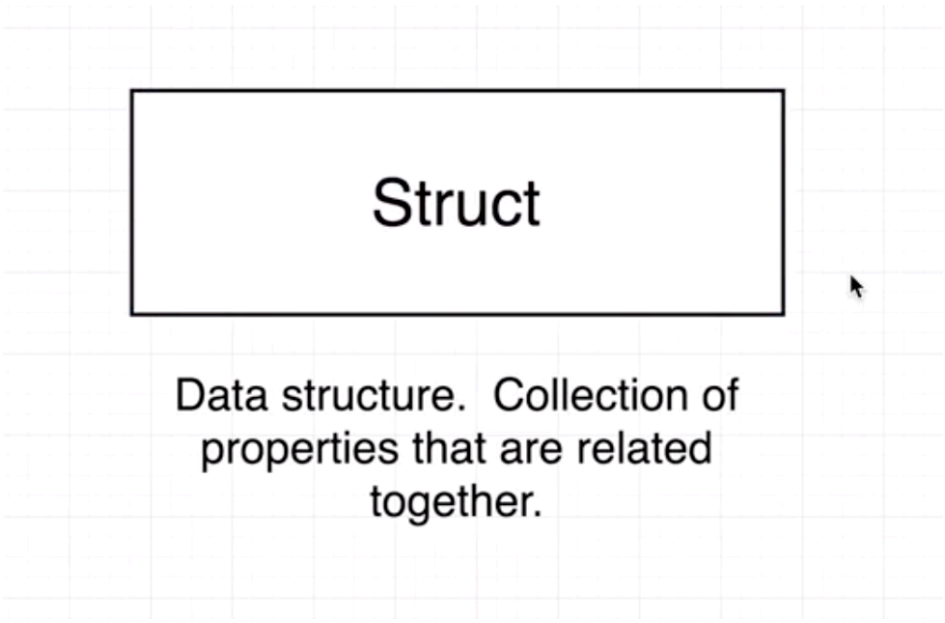
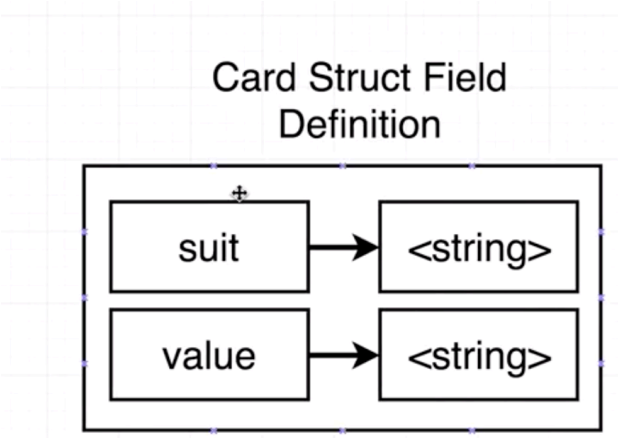
4 - Organizing Data with Structs

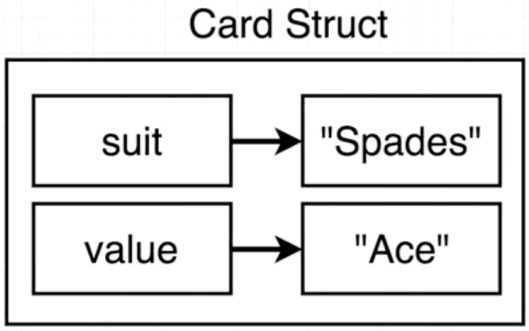
Structs in Go



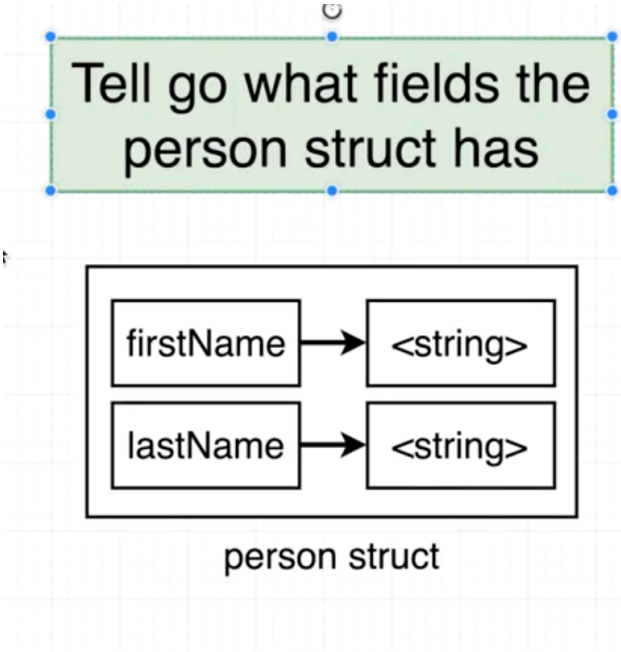
Struct

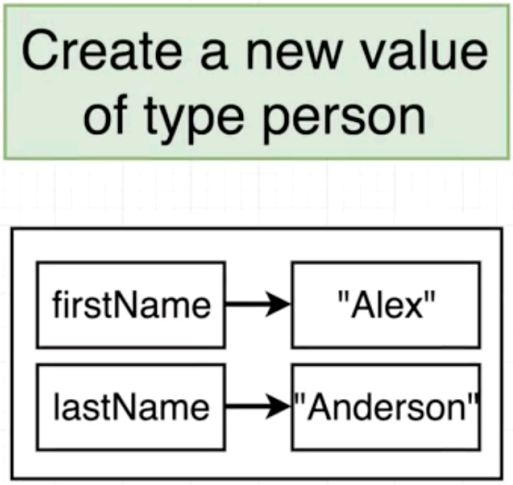






Defining Structs





/src/struct/main.go

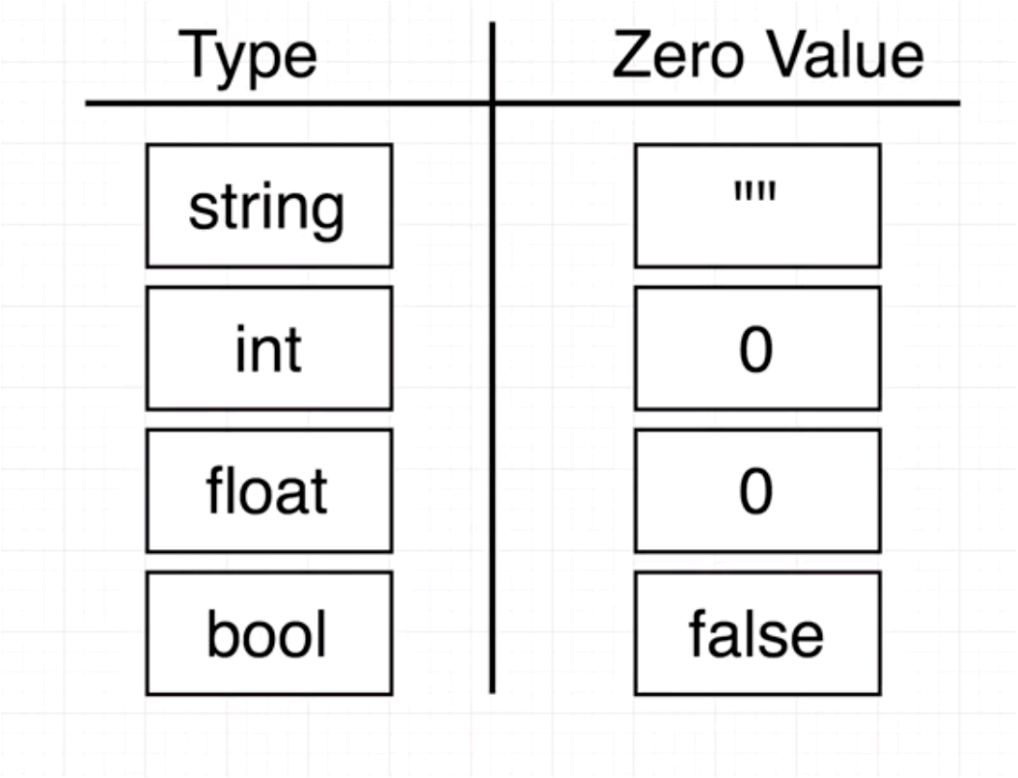
```
type person struct {
    firstName string
    lastName  string
}
```

Declaring Structs

/src/struct/main.go

```
func main() {
    alex := person{firstName: "Alex", lastName: "Anderson"}
    fmt.Println(alex)
}
```

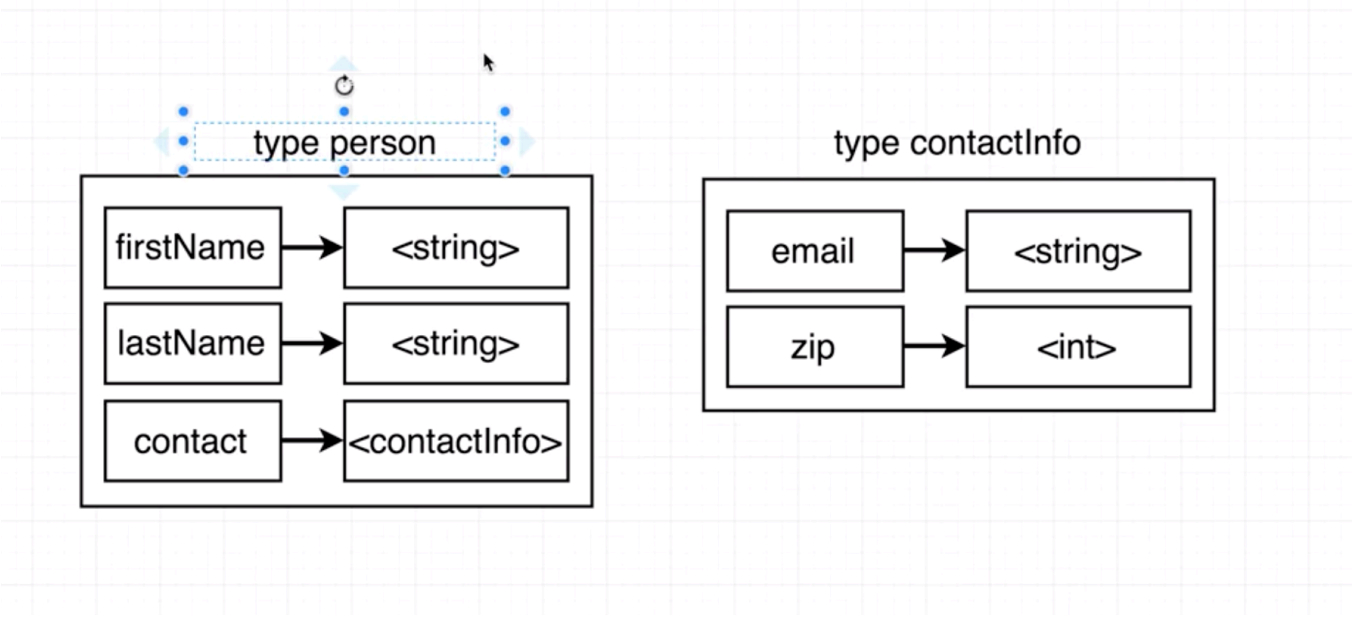
Updating Struct Values



/src/struct/main.go

```
1 func main() {
2     var alex person
3 
4     alex.firstName = "Alex"
5     alex.lastName = "Anderson"
6 
7     fmt.Println(alex)
8     fmt.Printf("%+v", alex)
9 }
```

Embedding Structs



/src/struct/main.go

```
type contactInfo struct {
    email   string
    zipCode int
}

type person struct {
    firstName string
    lastName  string
    contact   contactInfo
}

func main() {
    mer := person{
        firstName: "Mer",
        lastName:  "JQ",
        contact: contactInfo{
            email:   "mer@gmail.com",
            zipCode: 94000,
        },
    }

    fmt.Println(mer)
}
```

Structs with Receiver Functions

src/struct/main.go

```
type contactInfo struct {
    email   string
    zipCode int
}

type person struct {
    firstName string
    lastName  string
    contactInfo
}

func main() {
    jim := person{
        firstName: "Jim",
        lastName:  "Party",
        contactInfo: contactInfo{
            email:   "jim@gmail.com",
            zipCode: 94000,
        },
    }

    fmt.Println(jim)
}
```

Add receiver functions

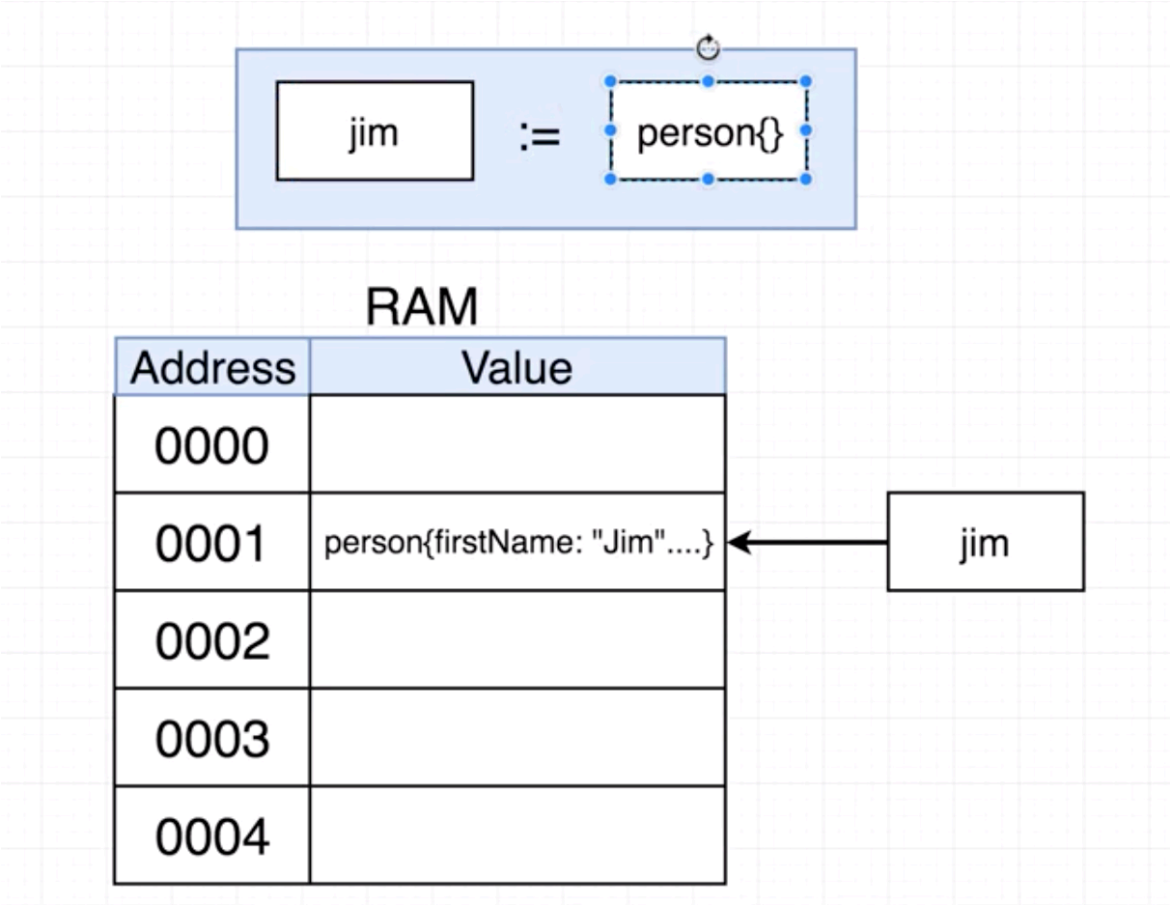
/src/struct/main.go

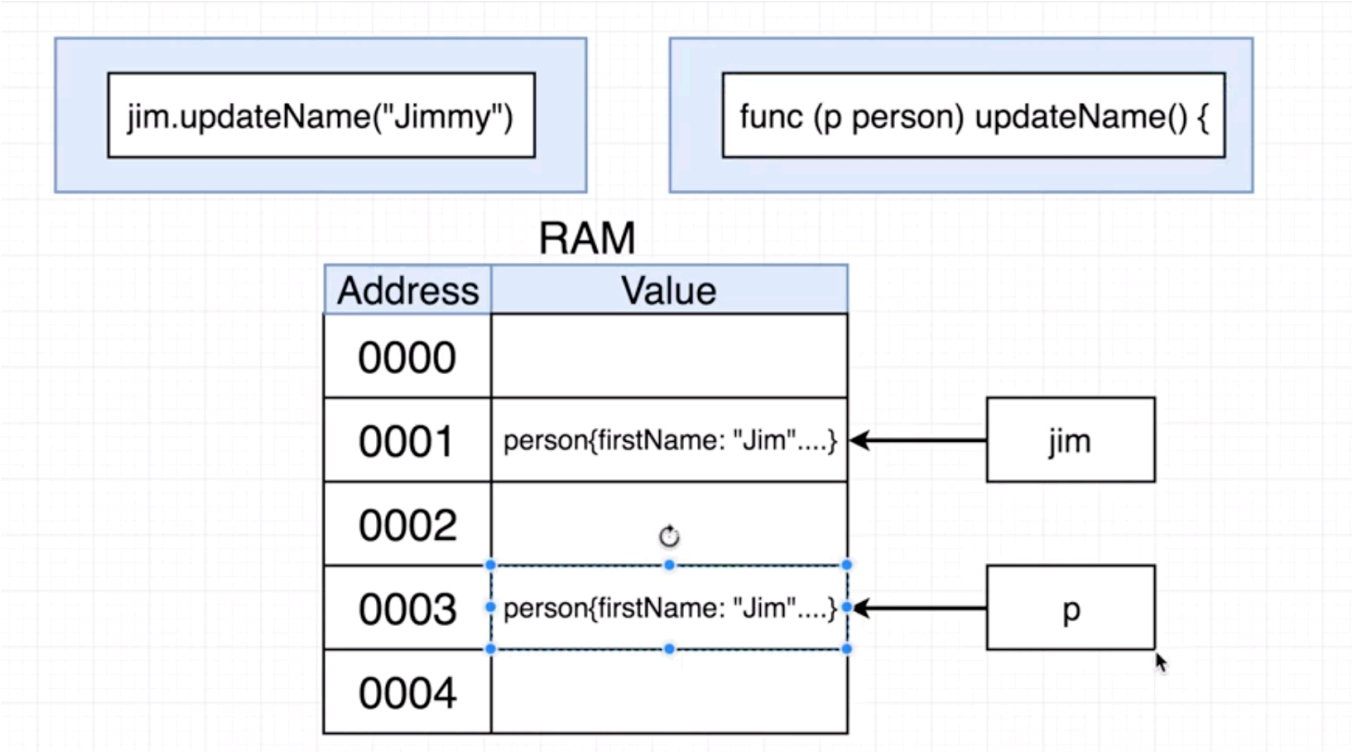
```
func main() {
  mer.updateName("Jieqiong")
  mer.print()
}

func (p person) print() {
    fmt.Println(p)
}

func (p person) updateName(newFirstName string) {
    p.firstName = newFirstName
}
```

Pass By Value





Structs with Pointers

/src/struct/main.go

```
func main() {
  merPointer := &mer
    merPointer.updateName("Jieqiong")
    mer.print()
}

func (p *person) updateName(newFirstName string) {
    (*p).firstName = newFirstName
}
```

Pointer Operations

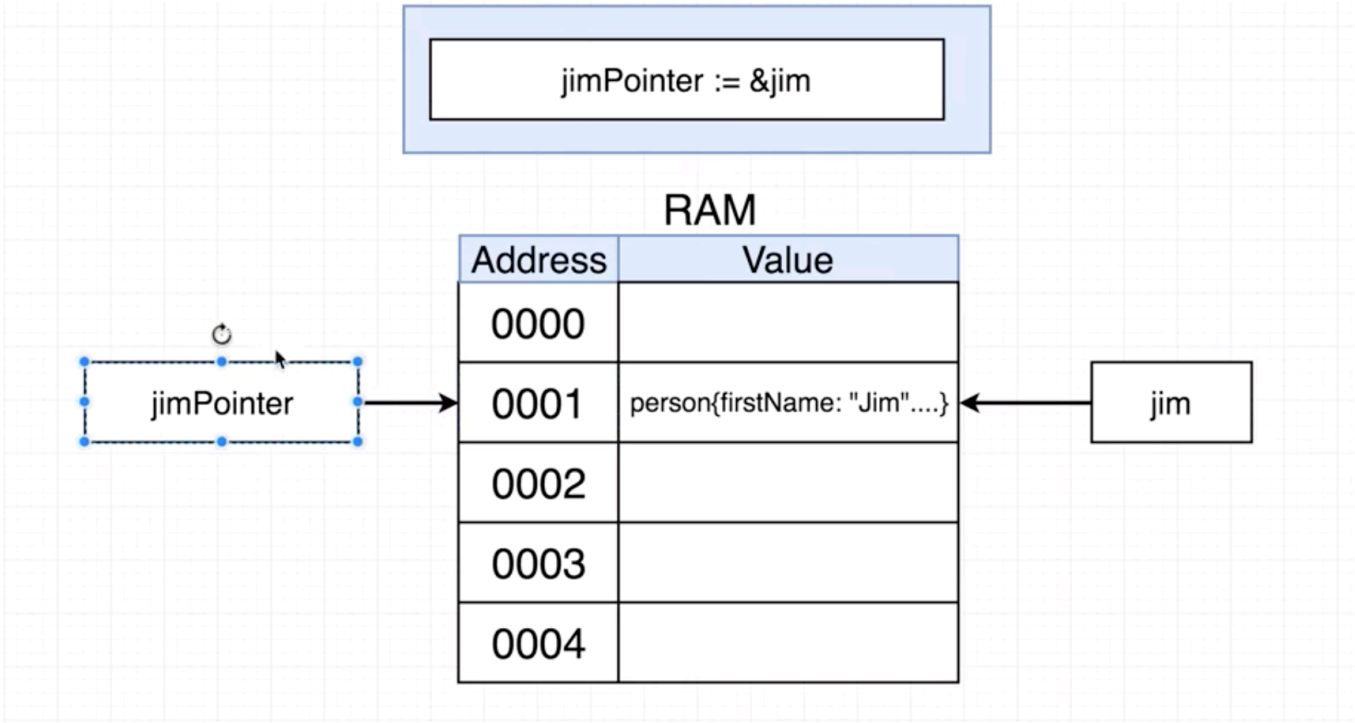
&variable

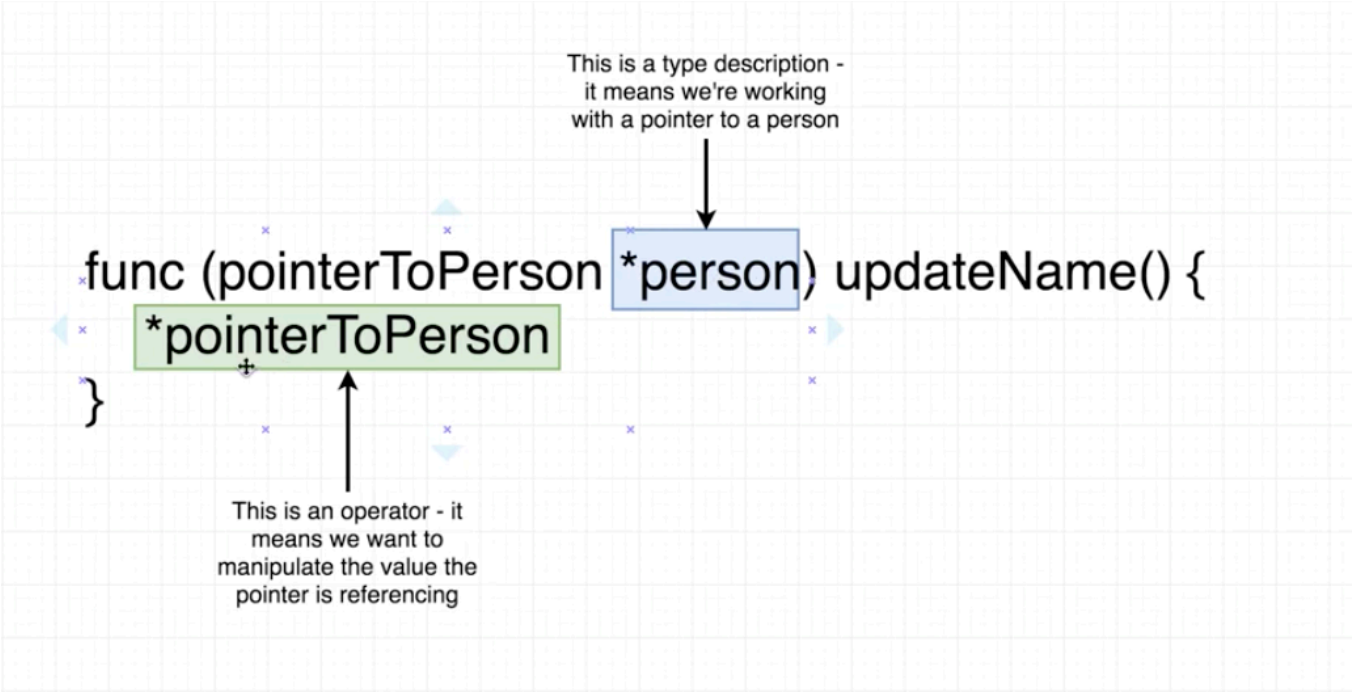
Give me the memory address of the value this variable is pointing at

*pointer

Give me the value this memory address is pointing at

&variable	Give me the memory address of the value this variable is pointing at
*pointer	Give me the value this memory address is pointing at



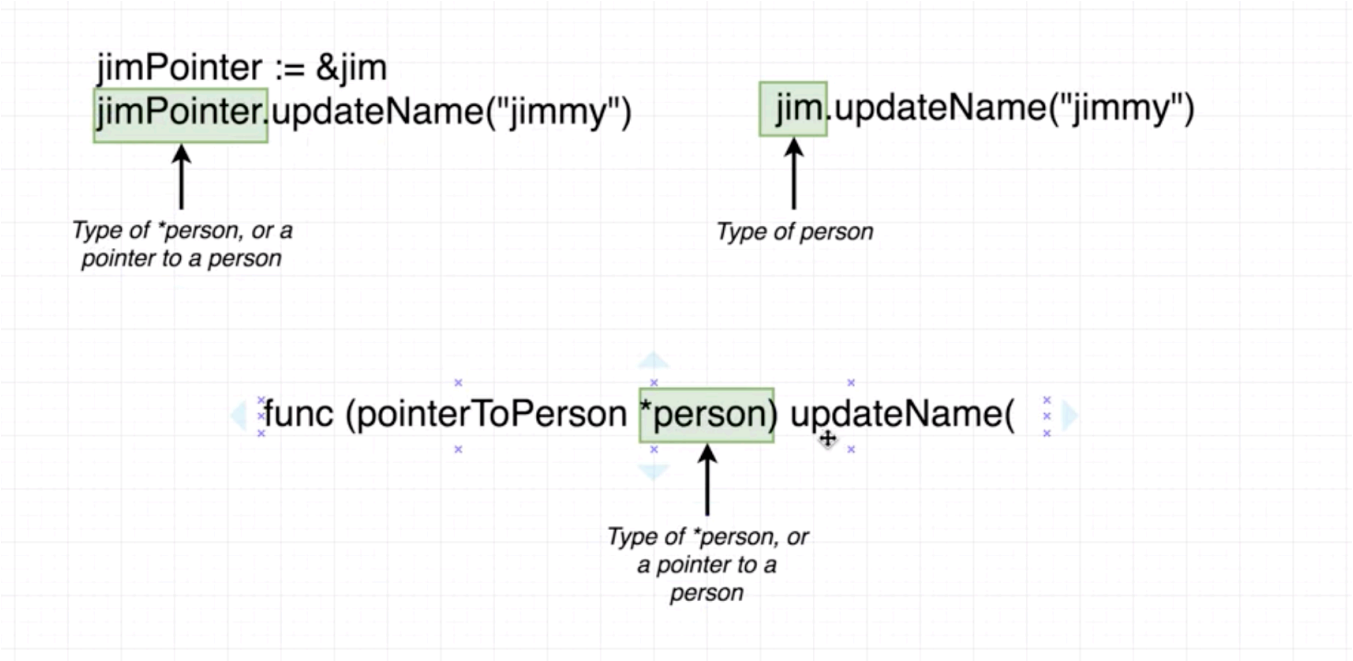


Pointer Shortcut

main.go

```
func main() {
    mer.updateName("Jieqiong")
    mer.print()
}

func (pointerToPerson *person) updateName(newFirstName string) {
    (*pointerToPerson).firstName = newFirstName
}
```

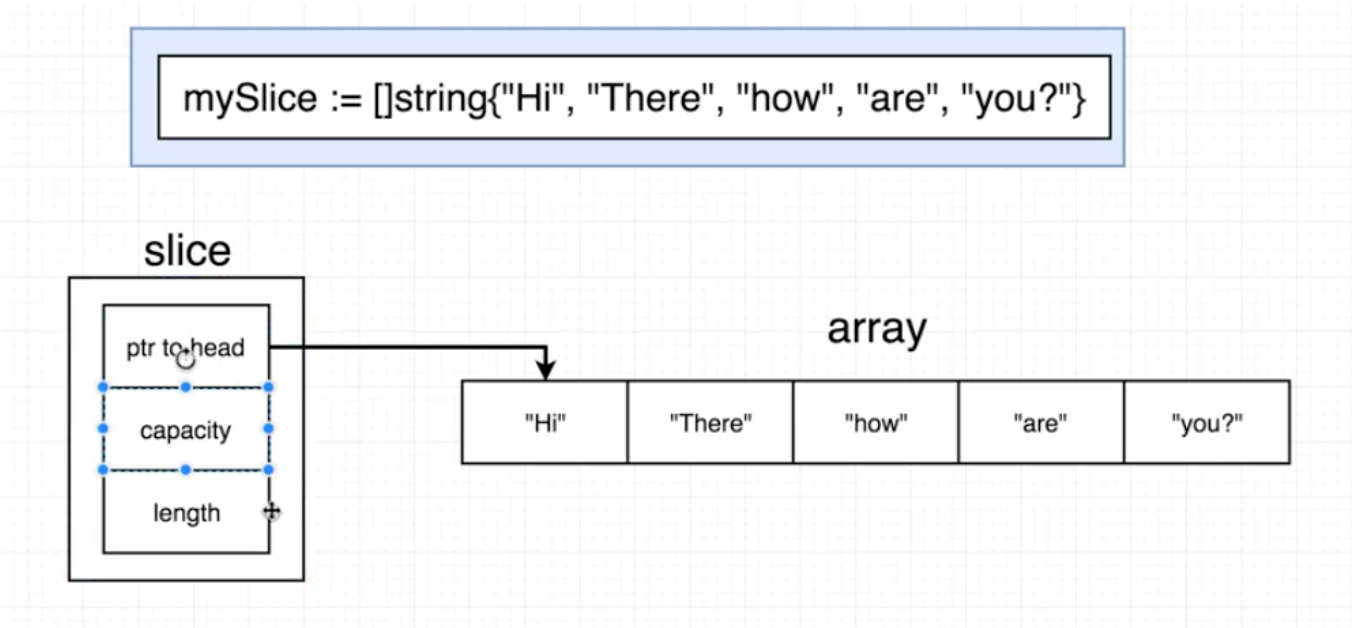


Gotchas With Pointers

/src/struct/main.go

```
1 func main() {
2     mySlice := []string{"Hi", "There", "How", "Are", "You"}
3     updateSlice(mySlice)
4     fmt.Println(mySlice)
5 }
6
7 func updateSlice(s []string) {
8     s[0] = "Bye"
9 }
```

Reference vs Value Types



```
mySlice := []string{"Hi", "There", "how", "are", "you?"}
```