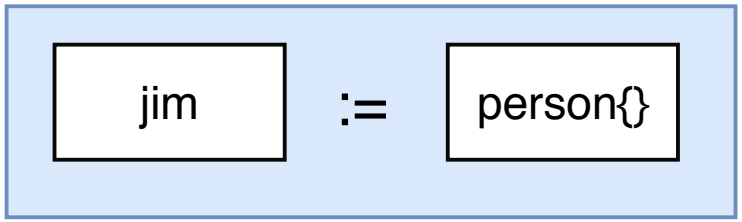
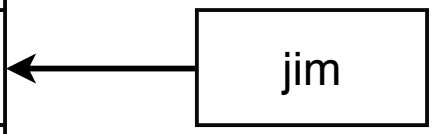
RAM

Address	Value
0000	
0001	
0002	
0003	
0004	



RAM

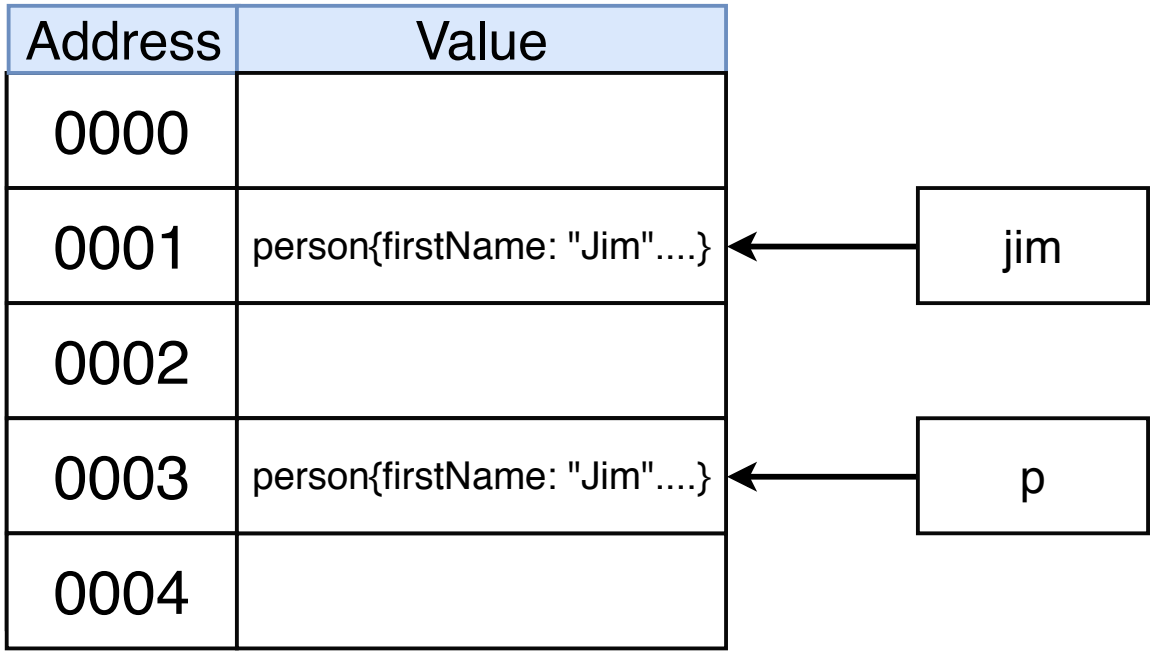
Address	Value
0000	
0001	person{firstName: "Jim"....}
0002	
0003	
0004	

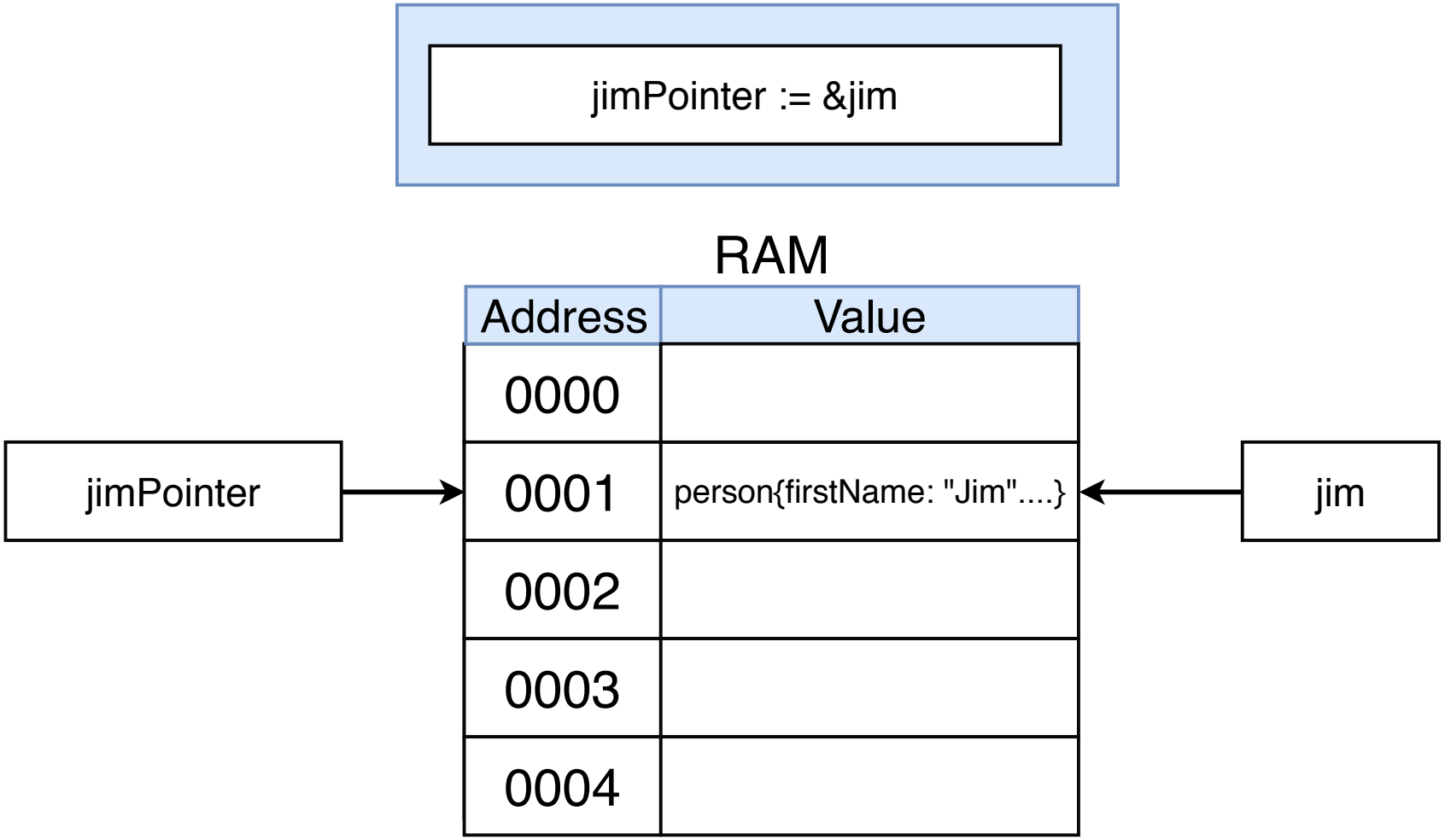


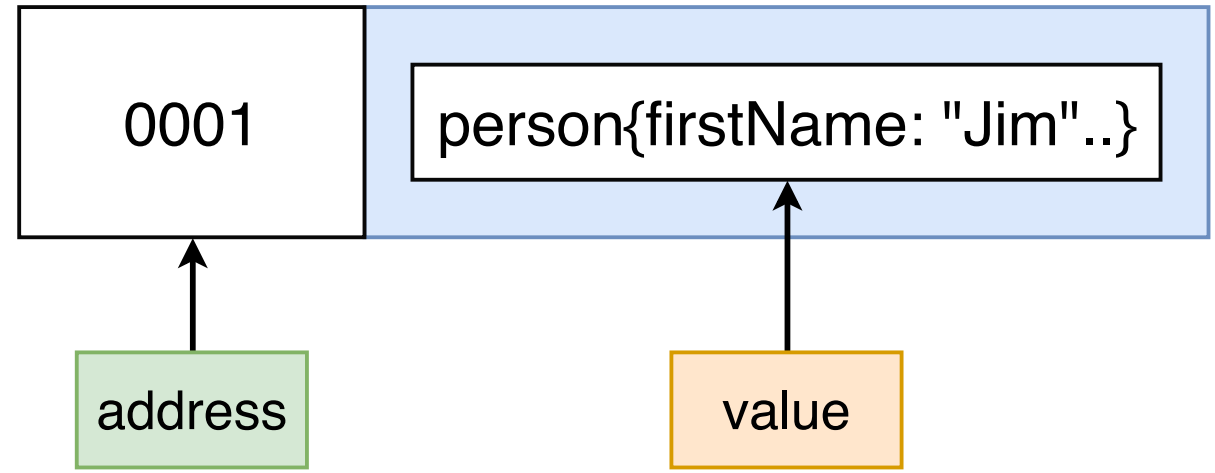
jim.updateName("Jimmy")

func (p person) updateName() {

RAM

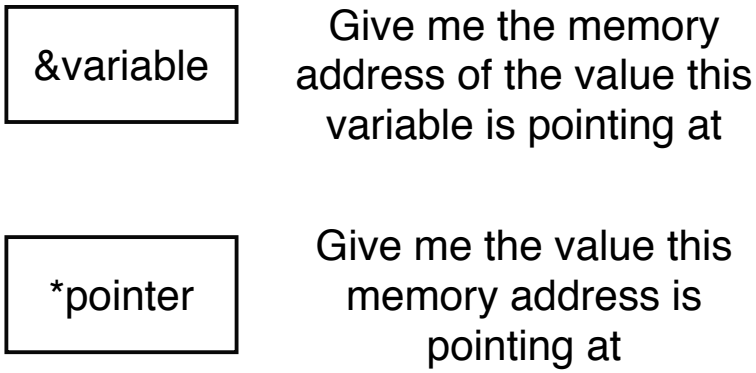


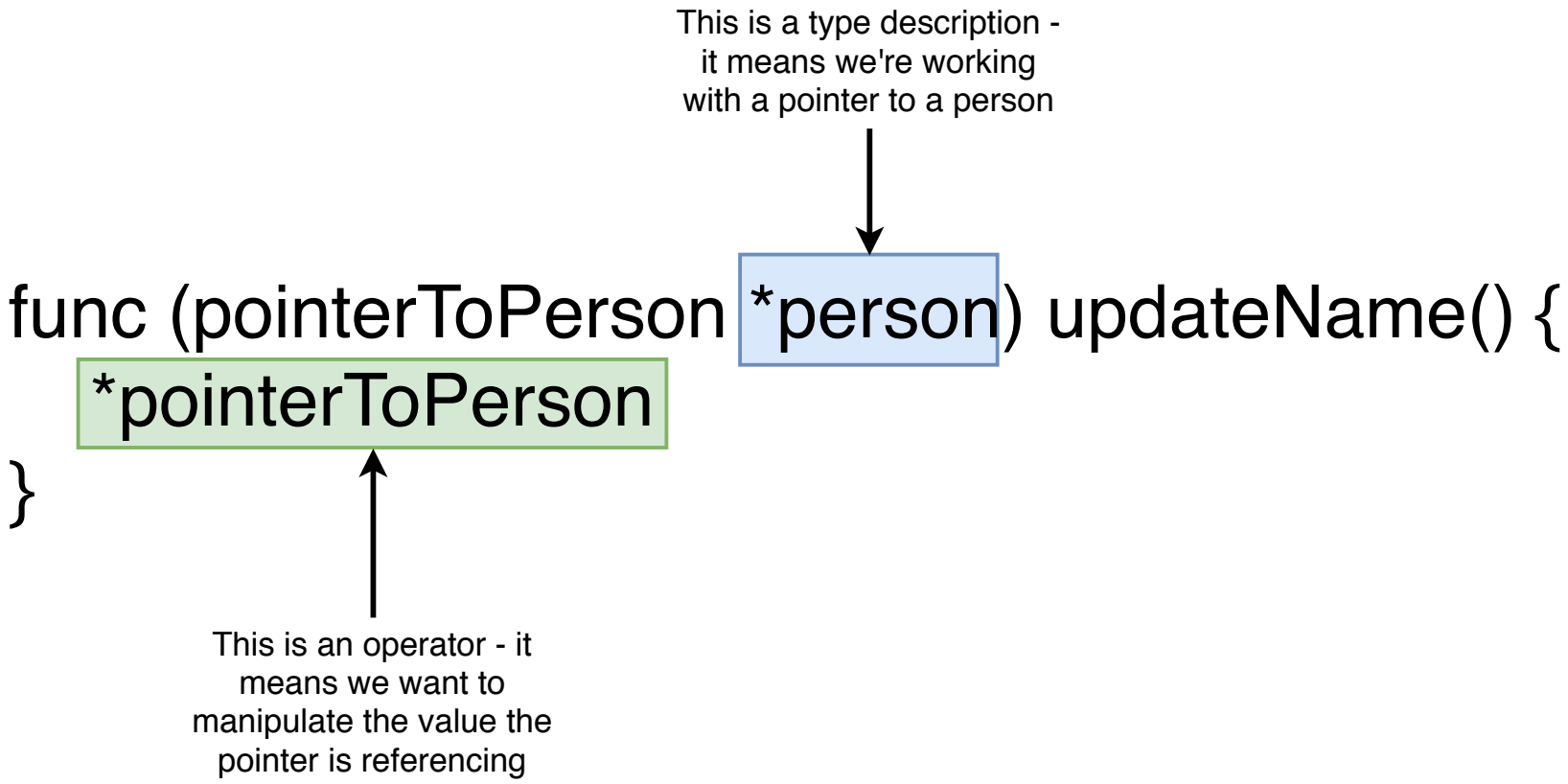

jimPointer := &jim
RAM
Address
Value
0000
0001
0002
0003
0004
jimPointer
person{firstName: "Jim"....}
jim



Turn address into value with *address

Turn value into address with &value


&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


This is a type description - it means we're working with a pointer to a person
func (pointerToPerson *person) updateName() {
*pointerToPerson
}
This is an operator - it means we want to manipulate the value the pointer is referencing

gocurl

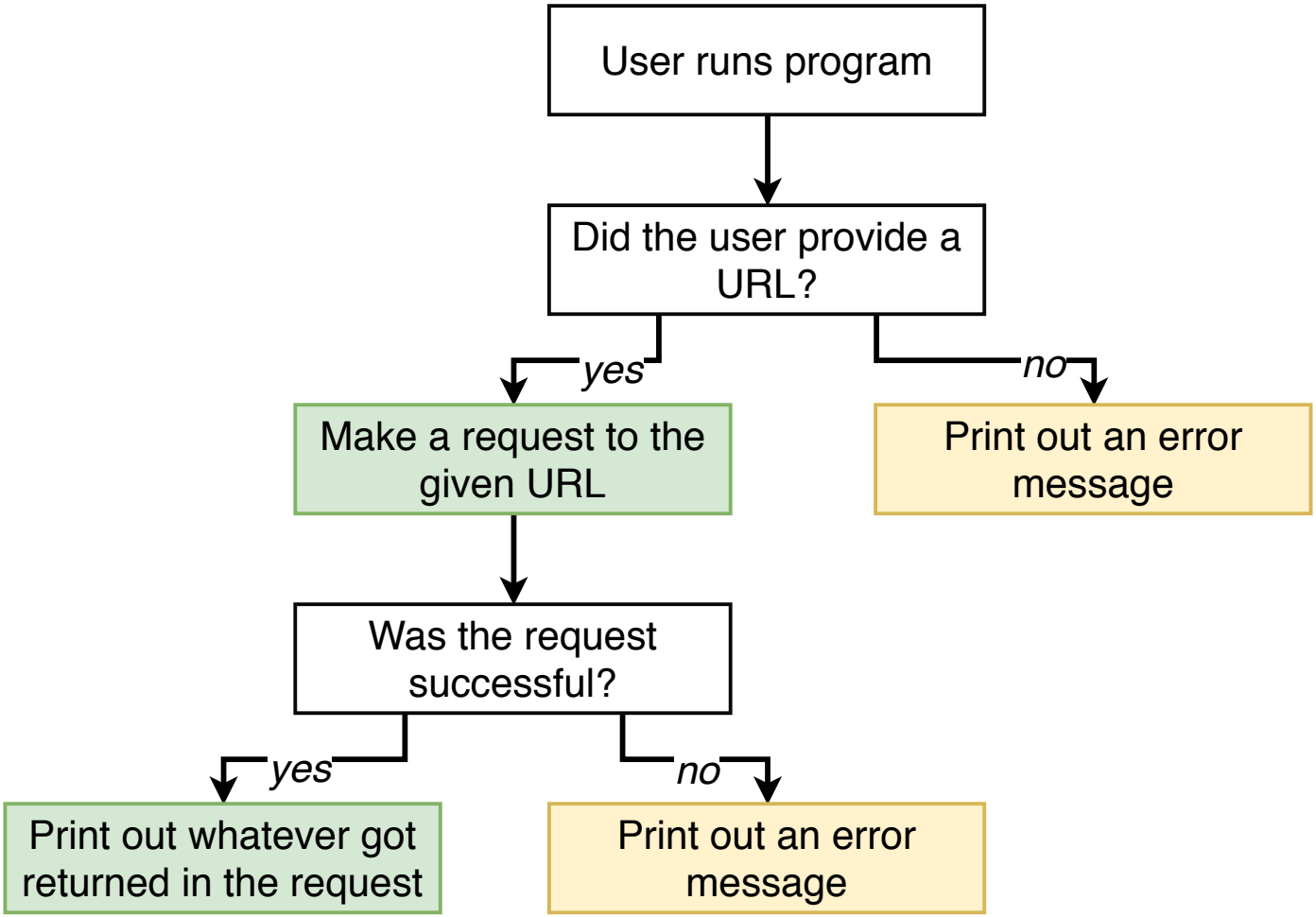
Write a program that can be ran from the command line

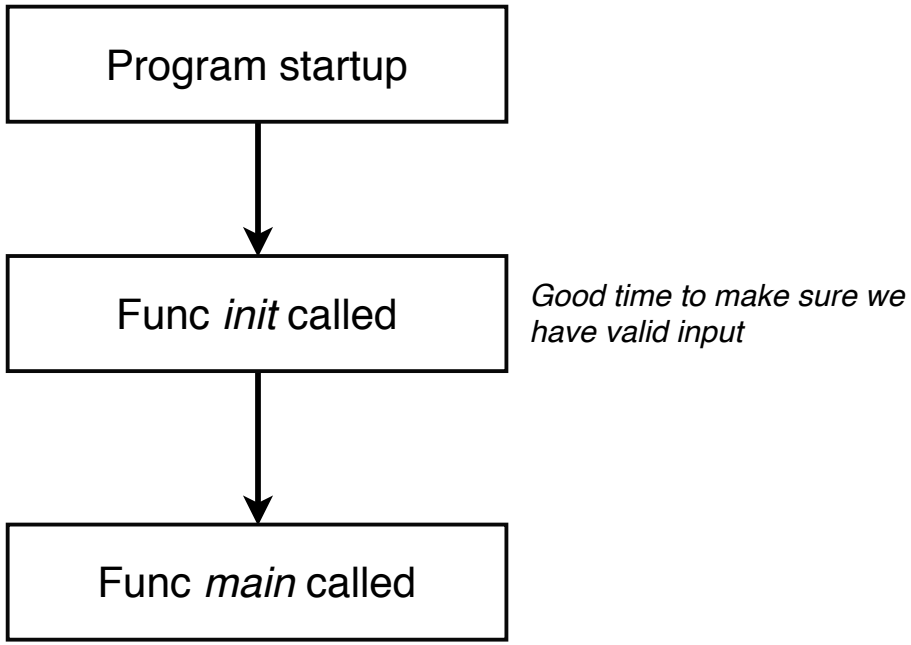
The program should make an HTTP request to a given URL and print out the response

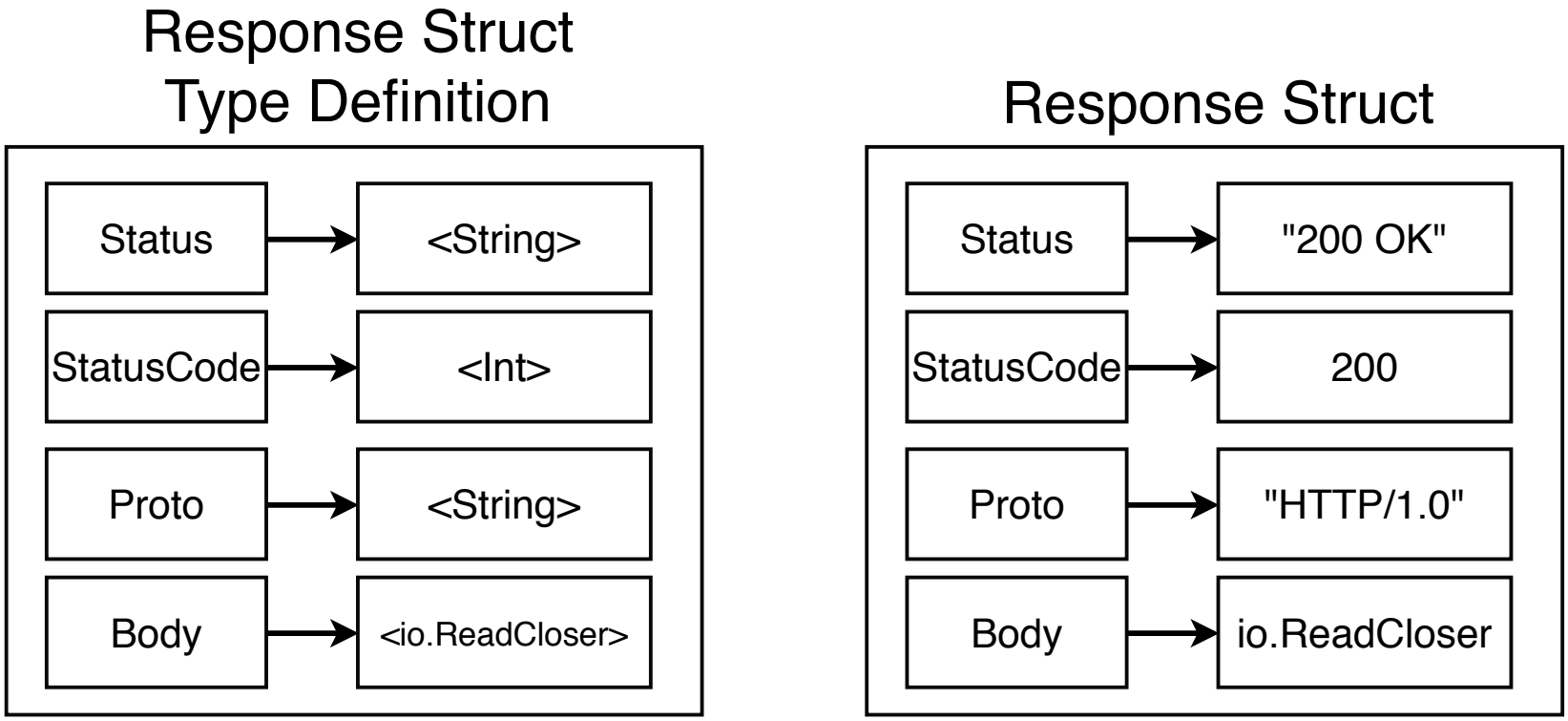
Example usage:

```
> gocurl https://google.com
```

yes


User runs program
Did the user provide a URL?
yes
no
Make a request to the given URL
Print out an error message
Was the request successful?
yes
no
Print out whatever got returned in the request
Print out an error message


Program startup
Func *init* called
Good time to make sure we have valid input
Func *main* called


Response Struct
Type Definition
Status
<String>
StatusCode
<Int>
Proto
<String>
Body
<io.ReadCloser>
Response Struct
Status
"200 OK"
StatusCode
200
Proto
"HTTP/1.0"
Body
io.ReadCloser

Card

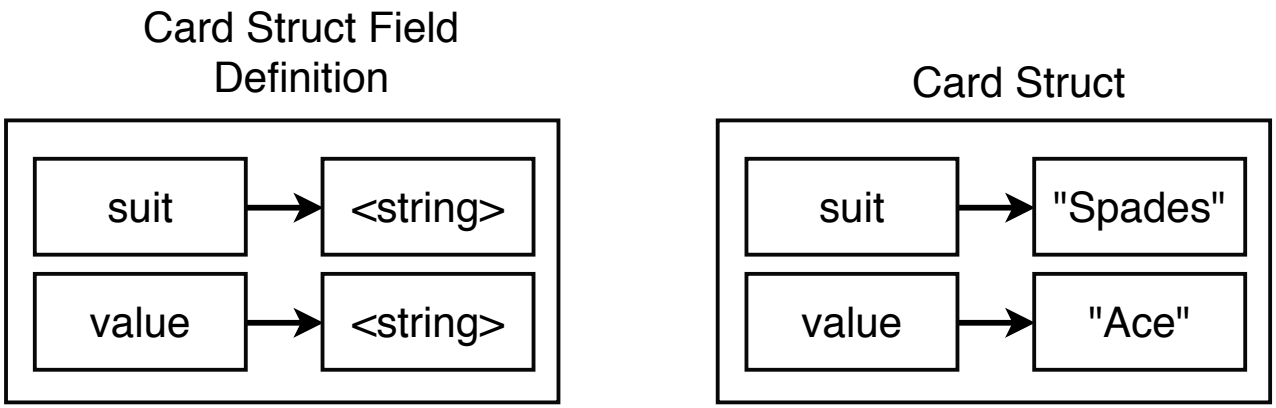
"Ace of Spades"

What's the suit?

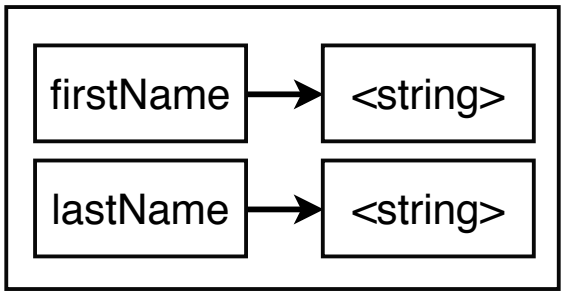
What's the value?

Struct

Data structure. Collection of properties that are related together.

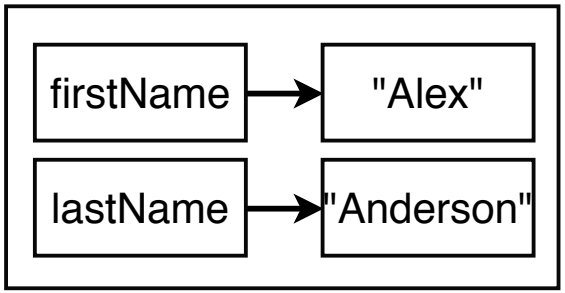

Card Struct Field Definition
suit
<string>
value
<string>
Card Struct
suit
"Spades"
value
"Ace"

Tell go what fields the person struct has

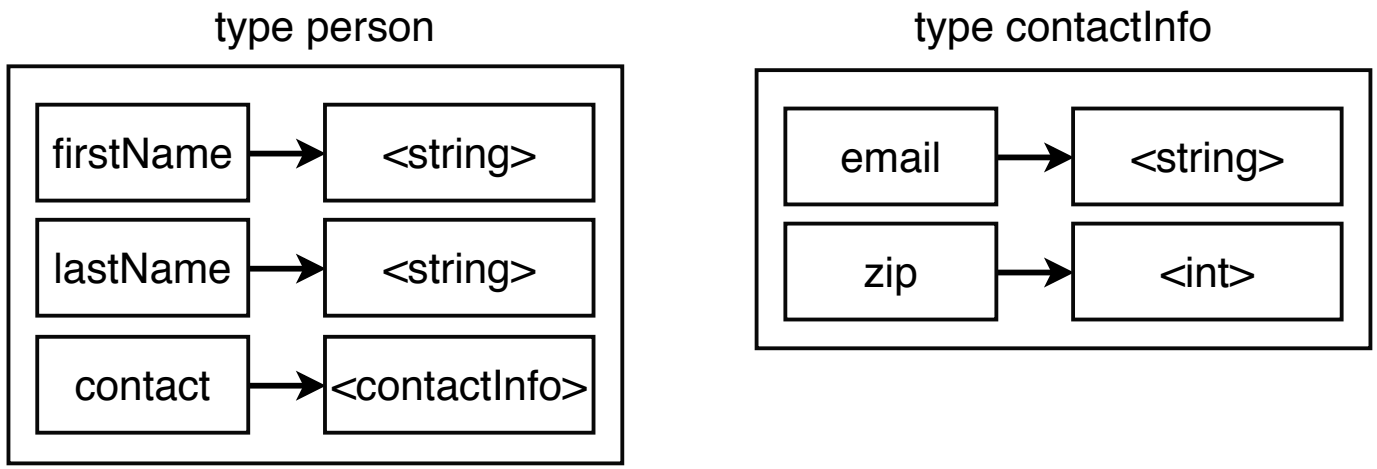


person struct

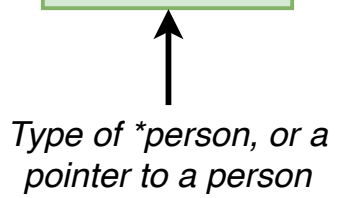
Create a new value of type person



Type	Zero Value
string	""
int	0
float	0
bool	false


type person
firstName
<string>
lastName
<string>
contact
<contactInfo>
type contactInfo
email
<string>
zip
<int>

```
jimPointer := &jim
jimPointer.updateName("jimmy")
```



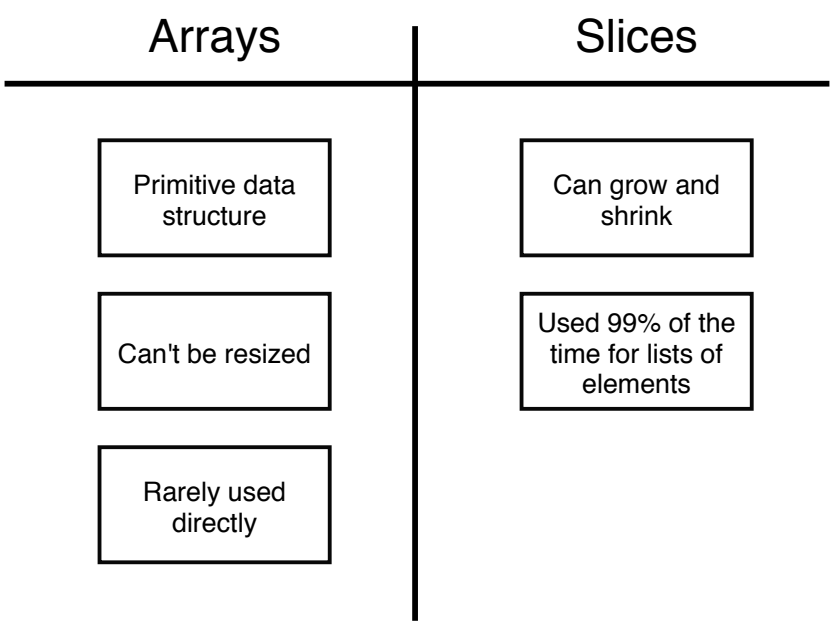
*Type of *person, or a pointer to a person*

```
jim.updateName("jimmy")
```

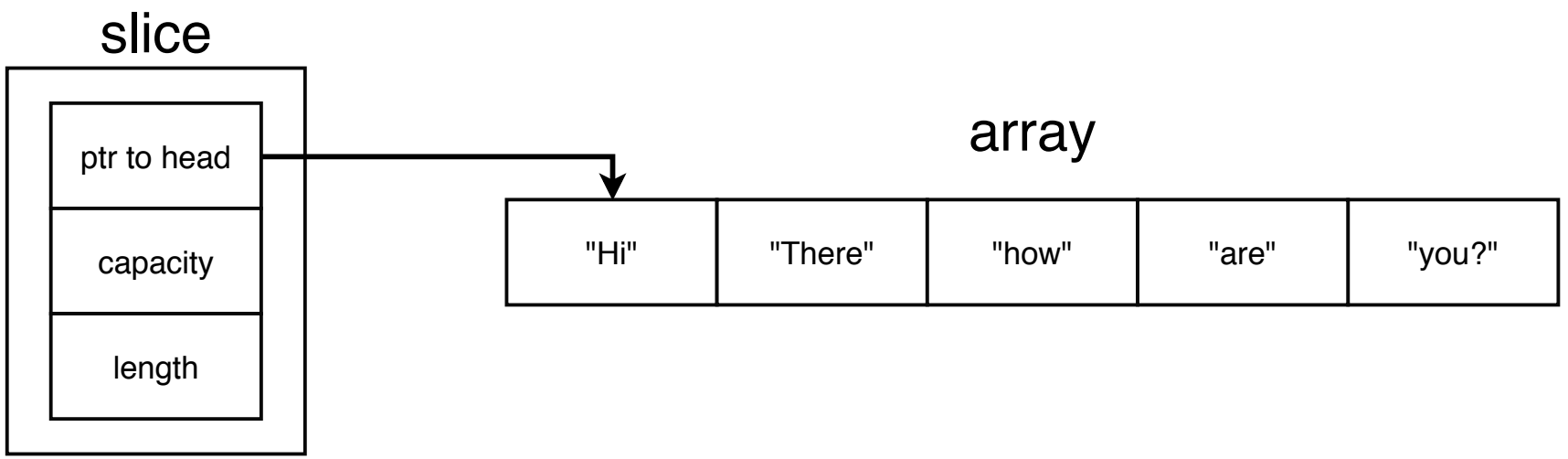
Type of person

```
func (pointerToPerson *person) updateName(
```

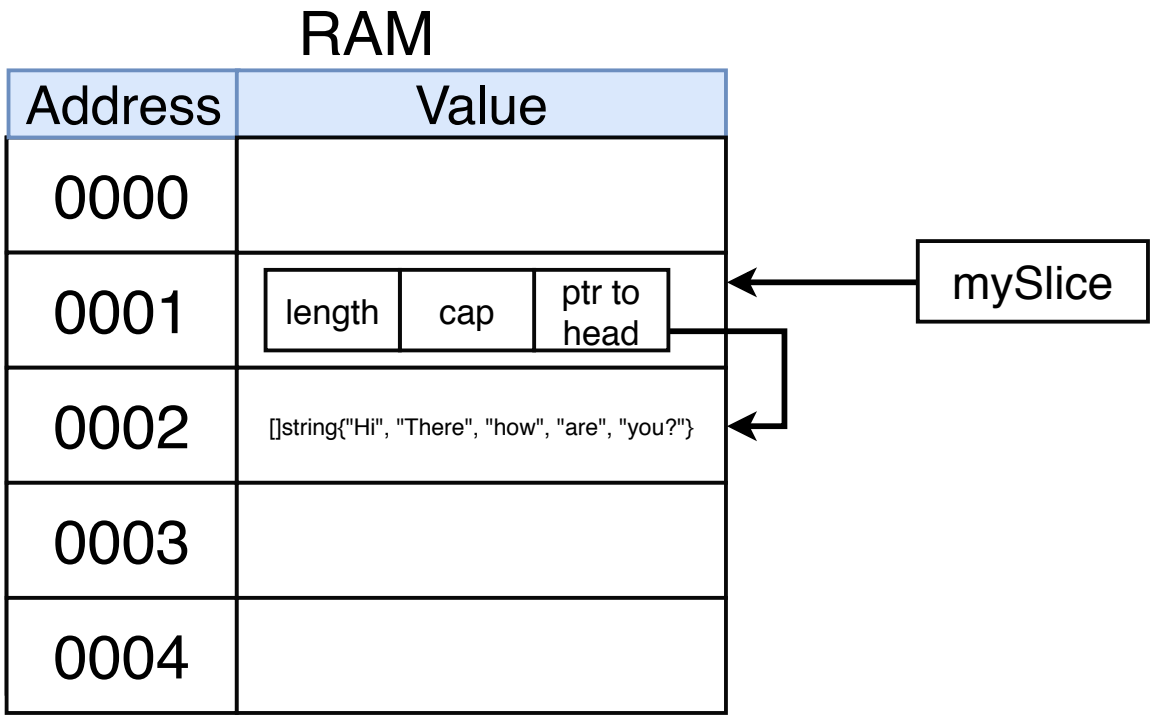
*Type of *person, or a pointer to a person*


Arrays
Slices
Primitive data structure
Can't be resized
Rarely used directly
Can grow and shrink
Used 99% of the time for lists of elements

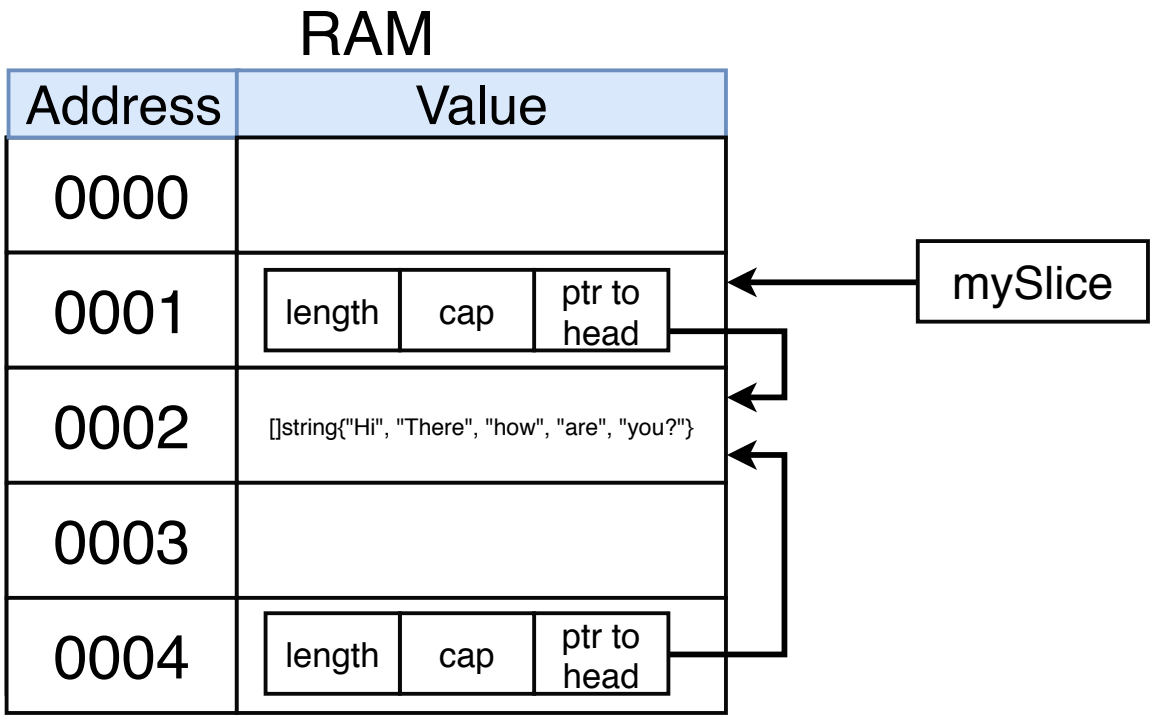
```
mySlice := []string{"Hi", "There", "how", "are", "you?"}
```

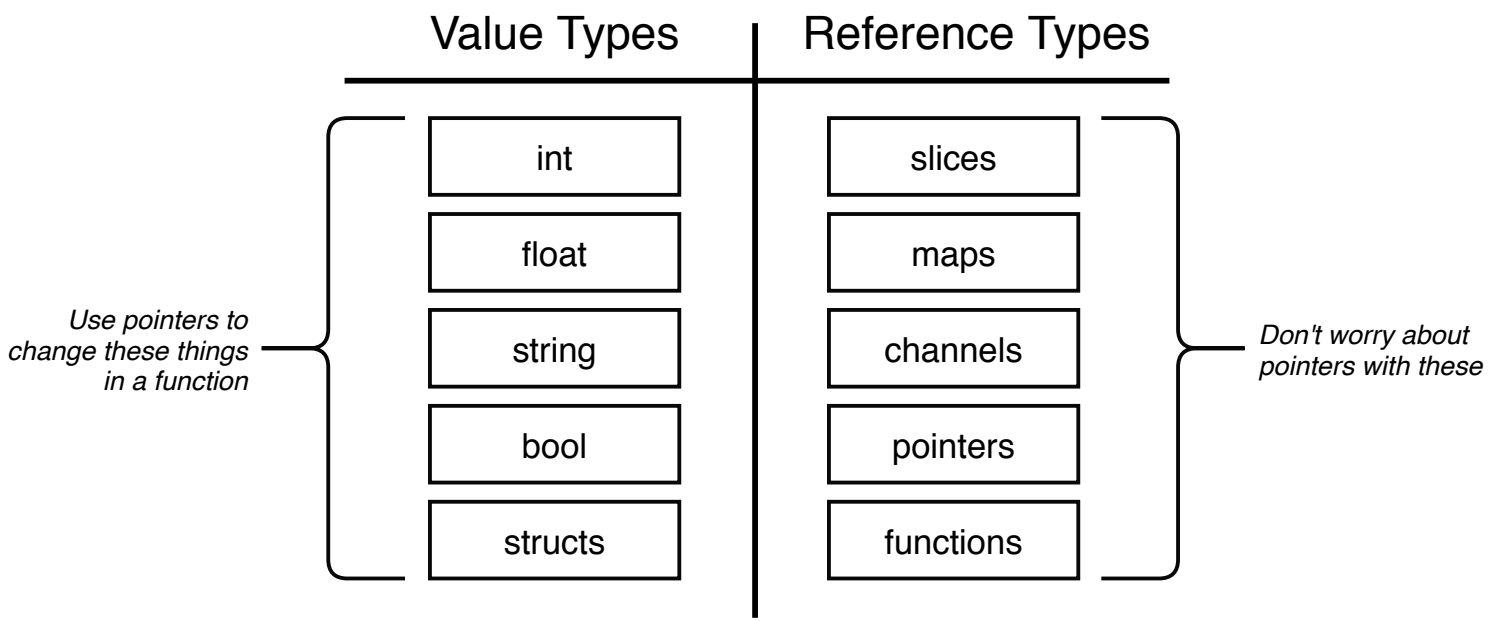


```
mySlice := []string{"Hi", "There", "how", "are", "you?"}
```



func updateSlice(s []string)




Value Types
Reference Types
int
float
string
bool
structs
slices
maps
channels
pointers
functions
Use pointers to change these things in a function
Don't worry about pointers with these