

Todd McLeod - Learn To Code Golang

Hands-On Exercises #01

composite literal; slice literal

<https://play.golang.org/p/nGekFavi0S>

for range loop & index position

<https://play.golang.org/p/1fzbabuaWD>

composite literal; map literal

https://play.golang.org/p/kHmQmHwK_c

print map value without using second return in for range

<https://play.golang.org/p/HvzecFdZPG>

answer

<https://play.golang.org/p/WE8V3YoXu8>

https://play.golang.org/p/_qPtpHASLS

using “make” to make a map

<https://play.golang.org/p/3FDj-UtWru>

composite literal; struct literal

<https://play.golang.org/p/cgc0Na2zhS>

func

<https://play.golang.org/p/vsblP3PmTc>

func (receiver) identifier(parameters) (returns) { <code> }

methods - receivers make methods

<https://play.golang.org/p/XezUp4hdE6>

trip out

<https://play.golang.org/p/3nd8m4oHNR>

interfaces

<https://play.golang.org/p/4kN4P9C2AW>

// HANDS ON 1

// create a type square

// create a type circle

// attach a method to each that calculates area and returns it

// create a type shape which defines an interface as anything which has the area method

// create a func info which takes type shape and then prints the area

// create a value of type square

// create a value of type circle

// use func info to print the area of square

// use func info to print the area of circle

<https://play.golang.org/p/1enChb7Kg5>

// HANDS ON 2

// create a struct that holds person fields

// create a struct that holds secret agent fields and embeds person type

// attach a method to person: pSpeak

// attach a method to secret agent: saSpeak

```
// create a variable of type person
// create a variable of type secret agent
// print a field from person
// run pSpeak attached to the variable of type person
// print a field from secret agent
// run saSpeak attached to the variable of type secret agent
// run pSpeak attached to the variable of type secret agent
SOLUTION: https://play.golang.org/p/RxrkJw9Cd
```

```
// HANDS ON 3
create an interface type that both person and secretAgent implement
declare a func with a parameter of the interface's type
call that func in main and pass in a value of type person
call that func in main and pass in a value of type secretAgent
https://play.golang.org/p/-Ux0gHf4SF
```

solution & optional additional info not necessary to know: assertions
https://play.golang.org/p/0TX4o-u-_B

```
package main
```

```
import (  
    "io"  
    "net/http"  
)
```

```
func foo(res http.ResponseWriter, req *http.Request) {  
    io.WriteString(res, "foo ran")  
}
```

```
func bar(res http.ResponseWriter, req *http.Request) {  
    io.WriteString(res, "bar ran")  
}
```

```
func main() {  
    http.HandleFunc("/", foo)  
    http.HandleFunc("/dog/", bar)  
    http.ListenAndServe(":8080", nil)  
}
```

string / slice of byte []byte

<https://play.golang.org/p/nl3morlwoO>