

# Looping

---



**Michael VanSickle**

@vansimke



# Introduction



**Infinite Loops**

**Loop till Condition**

**Counter-based Loops**


**Looping Over Collections**



# Looping

// sequential execution


```
func main() {  
    statement 1  
    statement 2  
    statement 3  
    ...  
    statement n  
}
```



# Looping

// looping execution

```
func main() {  
    statement 1  
    loop {  
        statement 2  
        statement 3  
    }  
    ...  
    statement n  
}
```



# Loops

```
for { ... }                                // infinite loop  
for condition { ... }                     // loop till condition  
for initializer; test; post clause { ... } // counter-based loop
```



# Infinite Loops

```
→ i := 1
→ for {           // infinite loop
→     fmt.Println(i)
→     i += 1
→ }
```

1  
2  
3  
4  
...



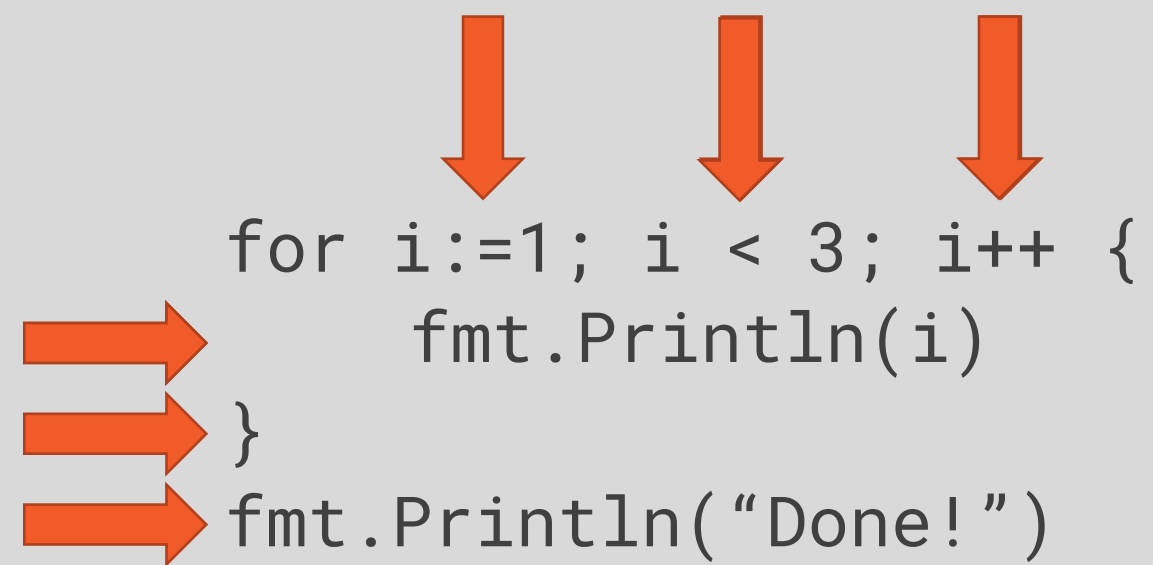
# Loop till Condition

```
→ i := 1
→ for i < 3 {
→     fmt.Println(i)
→     i += 1
→ }
→ fmt.Println("Done!")
```

1  
2  
Done!



# Counter-based Loops



```
for i:=1; i < 3; i++ {  
    fmt.Println(i)  
}  
fmt.Println("Done!")
```

1  
2  
Done!





# Demo



## Infinite loop

- break

## Loop till condition

- continue

## Loop with counter

- Print out menu items for one category

**Start by showing loops in lang spec and effective Go**



# Looping with Collections

```
for key, value := range collection { ... }  
for key := range collection { ... }  
for _, value := range collection { ... }
```

## Collections

array

slice

map



# Looping with Collections

```
➡ arr := [3]int{101, 102, 103}
➡ for i, v := range arr {
➡     fmt.Println(i, v)
➡ }
➡ fmt.Println("Done!")
```

```
0 101
1 102
2 103
Done!
```



# Demo



## Loop over collection

- Update course demo to iterate through entire menu, including each category



# Summary



**Infinite Loops**

**Loop till Condition**

**Counter-based Loops**

**Looping Over Collections**

