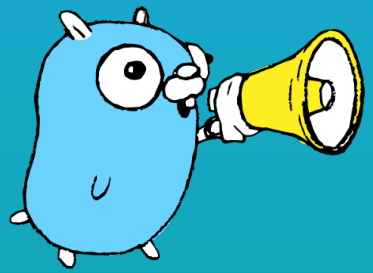
Mastertech, AUG 23 2019

Goroutines + Channels

GolFz

Mastertech

Engineer, R&D



Hi, Let’s talk about

01

Goroutines

Goroutines — Example

```
package main

import "fmt"

func main() {
        go fmt.Println("Hello")
        fmt.Println("I am Gopher")
}
```

https://play.golang.org/p/ArjKRc-iT_3

Exercise : Goroutines (1)

ทดลองเขียน Goroutine ด้วย

- Function แบบปกติ go sayHello()
- Anonymous function go func(){...}()

Exercise : Goroutines (2)

ทดลองแก้ไขโค้ดตัวอย่าง

- ใช้ Goroutines เพื่อให้งานทั้งหมดเสร็จภายใน 2 วินาที

https://play.golang.org/p/9FaIzW_pnSE

02

Channels

สร้าง Channel

```
ch := make(chan string)
```

Channel operator

<-

ส่งค่าเข้าไปใน Channel

```
ch <- "some message"
```

รับค่าจาก Channel

```
message <- ch
```

Channel — Example (1)

```
package main

import "fmt"

func main() {
	messages := make(chan string)
	go func() { messages <- "ping" }()
	msg := <-messages
	fmt.Println(msg)
}
```

https://play.golang.org/p/09_5ahJ0NEf

Function ที่มี Channel เป็น Parameter

```
func sayHello(ch  chan  string) {
    ...
}
```

Channel — Example (2)

```
func sum(s []int, ch chan int) {
        sum := 0
        for _, v := range s {
                sum += v
        }
        ch <- sum
}
```

https://play.golang.org/p/RpFBjv43LbD

Channel Directions

Read only

```
func hello(ch <- chan int) { ... }
```

Write only

```
func hello(ch chan <- int) { ... }
```

Channel Directions — Example

```
func ping(chPing chan<- string, msg string) {
	chPing <- msg
}

func pong(chPing <-chan string, chPong chan<-
string) {
	msg := <-chPing
	chPong <- fmt.Sprintf("I got: '%s' \n", msg)
}
```

https://play.golang.org/p/p1i9c7mp-5T

Channel Buffer

```
ch := make(chan string, 5)
```

Channel Buffering — Example (1)

```
ch1 := make(chan string)

ch1 <- "hello"

fmt.Println("hello")
```

https://play.golang.org/p/RSfzc8wEOkI

Channel Buffer — Example (2)

```
messages := make(chan string, 2)

messages <- "buffered"
messages <- "channel"

fmt.Println(<-messages)
fmt.Println(<-messages)
```

Channel Synchronisation

Channel ใช้ Sync การทำงาน ระหว่าง Goroutines ได้

03

Select

รอหลาย ๆ Channel พร้อมกัน

Select จะทำงาน 1 case เท่านั้น

จากนั้นจะออกจากการทำงาน

ของ Select

Select — Example

```
select {
case msg1 := <-c1:
        fmt.Println("received", msg1)
case msg2 := <-c2:
        fmt.Println("received", msg2)
}
```

https://play.golang.org/p/C7Tl5IkuqC1

Exercise : Select

ทดลองเขียนโค้ดให้สมบูรณ์

เขียนโค้ดในตัวอย่างนี้ให้สมบูรณ์
และทำงานได้ถูกต้อง

https://play.golang.org/p/DI3_EO_3Me5

04

Default

Non-Blocking Channel Operation

ถ้า Select - Case ไม่พร้อมทำงานทันที

จะทำงานใน Default case

Default — Example

```
select {
...

default:
        fmt.Println("no message received")
}
```

https://play.golang.org/p/dq3CU2sBSHP

05

Close

ป้องกันการเกิด Deadlock

ถ้าไม่ close และไม่ตรวจสอบ

จะเกิด Deadlock ที่ Receiver

Close — Example

```
go func() {
        for {
                v, ok := <-ch
                if ok {
                        fmt.Println("value:", v)
                } else {
                        done <- true
                }
        }
}()

ch <- 1

close(ch)
```

https://play.golang.org/p/xo09eLn5OeY

06

Range

รับค่าจาก Channels ทุกตัว

```
for {
    v, ok := ch
```

→ `for v := range ch`

Range — Example

```
queue <- "one"
queue <- "two"

close(queue)

for elem := range queue {
        fmt.Println(elem)
}
```

https://play.golang.org/p/rqnxxRG3mhM

07

WaitGroup

รอหลาย ๆ Goroutines เสร็จก่อน

แล้วค่อยทำงานต่อ

WaitGroup — Example

```
func worker(wg *sync.WaitGroup) {
        ...
        wg.Done()
}

func main() {
        var wg sync.WaitGroup
        ...
        wg.Wait()
}
```

https://play.golang.org/p/H0DhlUYLUAg

Best practice

defer wg.Done

Q & A