



Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

定时器

目录



PART 01
time.Timer



PART 02
time.Ticker



time.Timer

1.time.Timer



time.Timer

Timer是一个定时器，代表未来的一个单一事件，你可以告诉timer你要等待多长时间。

```
type Timer struct {  
    C <-chan Time  
    r runtimeTimer  
}
```

它提供一个channel，在定时时间到达之前，没有数据写入timer.C会一直阻塞，直到定时时间到，系统会自动向timer.C 这个channel中写入当前时间，阻塞即被解除。



定时器的常用操作：

1、实现延迟功能

```
1) <-time.After(2 * time.Second) //定时2s, 阻塞2s,2s后产生一个事件, 往channel写内容
    fmt.Println("时间到")
2) time.Sleep(2 * time.Second)
    fmt.Println("时间到")
3) 延时2s后打印一句话
    timer := time.NewTimer(2 * time.Second)
    <- timer.C
    fmt.Println("时间到")
```

2、定时器停止

```
timer := time.NewTimer(3 * time.Second)
go func() {
    <-timer.C
    fmt.Println("子协程可以打印了, 因为定时器的时间到")
}()
timer.Stop() //停止定时器
for {
}
```

3、定时器重置

```
timer := time.NewTimer(3 * time.Second)
ok := timer.Reset(1 * time.Second) //重新设置为1s
fmt.Println("ok = ", ok)
<-timer.C
fmt.Println("时间到")
```



time.Ticker

1.time.Ticker

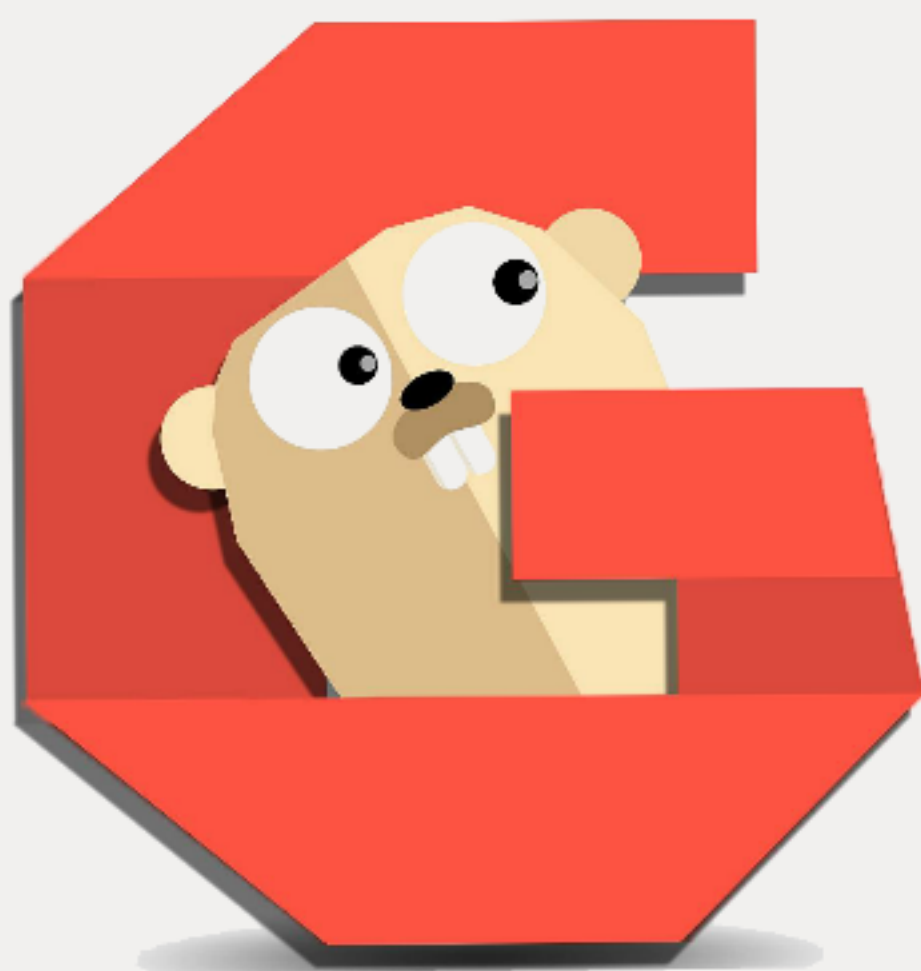
Ticker是一个周期触发定时的计时器，它会按照一个时间间隔往channel发送系统当前时间，而channel的接收者可以以固定的时间间隔从channel中读取事件。

```
type Ticker struct {  
    C <-chan Time           // The channel on which the ticks are delivered.  
    r runtimeTimer  
}
```



实例演示





江洲老师云课堂

—— 主讲：江洲老师 ——

感谢您的聆听和观看