



# Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

## 无缓冲channel

# 目录



PART 01  
**channel**



PART 02  
**无缓冲channel**



# channel

1.channel

2.定义channel变量



## channel

channel是Go语言中的一个核心类型，可以把它看成管道。

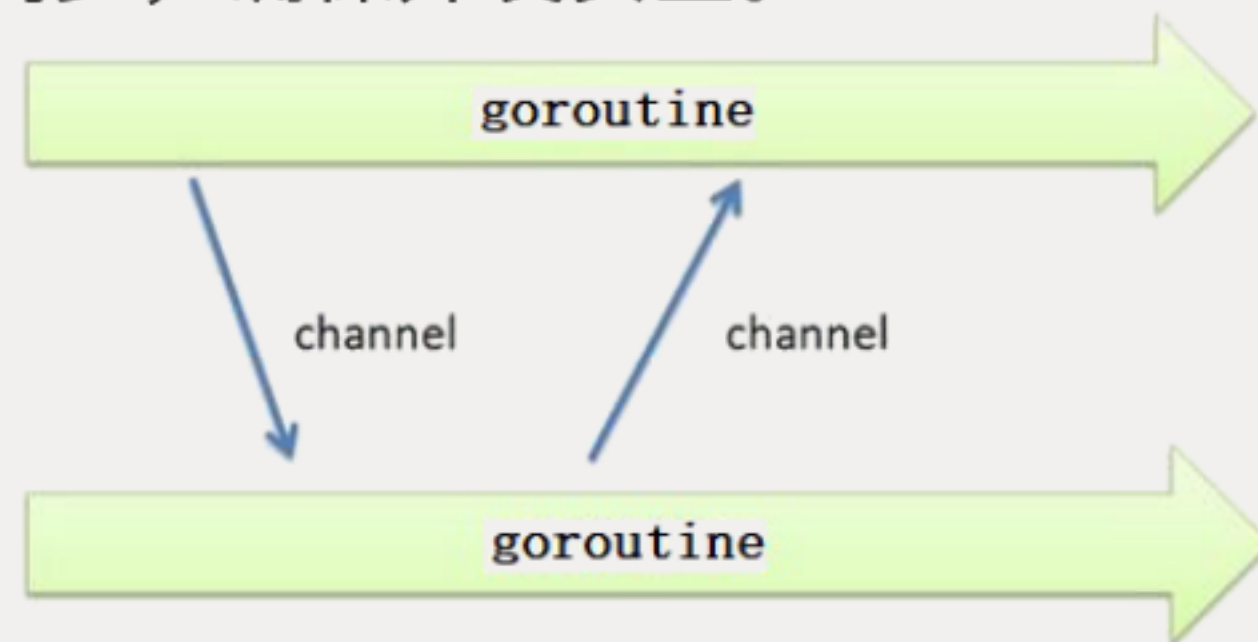
并发核心单元通过它就可以发送或者接收数据进行通讯，这在一定程度上又进一步降低了编程的难度。

channel是一个数据类型，主要用来解决go程的同步问题以及go程之间数据共享（数据传递）的问题。

goroutine运行在相同的地址空间，因此访问共享内存必须做好同步。

goroutine 奉行**通过通信来共享内存，而不是共享内存来通信。**

引用类型 channel可用于多个 goroutine 通讯，其内部实现了同步，确保并发安全。





## 定义channel变量

---

和map类似，channel也是一个对应make创建的底层数据结构的引用。

当我们复制一个channel或用于函数参数传递时，我们只是拷贝了一个channel引用，因此调用者和被调用者将引用同一个channel对象。和其它的引用类型一样，channel的零值也是nil。

定义一个channel时，也需要定义发送到channel的值的类型。channel可以使用内置的make()函数来创建：

chan是创建channel所需使用的关键字。Type 代表指定channel收发数据的类型。

```
make(chan Type) //等价于make(chan Type, 0)
make(chan Type, capacity)
```

channel非常像生活中的管道，一边可以存放东西，另一边可以取出东西。

channel通过操作符 <- 来接收和发送数据，发送和接收数据语法：

```
channel <- value    //发送value到channel
<-channel          //接收并将其丢弃
x := <-channel      //channel中接收数据，并赋值给x
x, ok := <-channel  //功能同上，同时检查通道是否已关闭或者是否为空
```







# 无缓冲的channel

## 1.无缓冲的channel



## 无缓冲的channel

---

无缓冲的通道（unbuffered channel）是指在接收前没有能力保存任何值的通道。

这种类型的通道要求发送goroutine和接收goroutine同时准备好，才能完成发送和接收操作。否则，通道会导致先执行发送或接收操作的 goroutine 阻塞等待。

这种对通道进行发送和接收的交互行为本身就是同步的。  
其中任意一个操作都无法离开另一个操作单独存在。

阻塞：由于某种原因数据没有到达，当前协程（线程）持续处于等待状态，直到条件满足，才解除阻塞。

同步：在两个或多个协程（线程）间，保持数据内容一致性的机制。

无缓冲的channel创建格式：

`make(chan Type)` //等价于`make(chan Type, 0)`

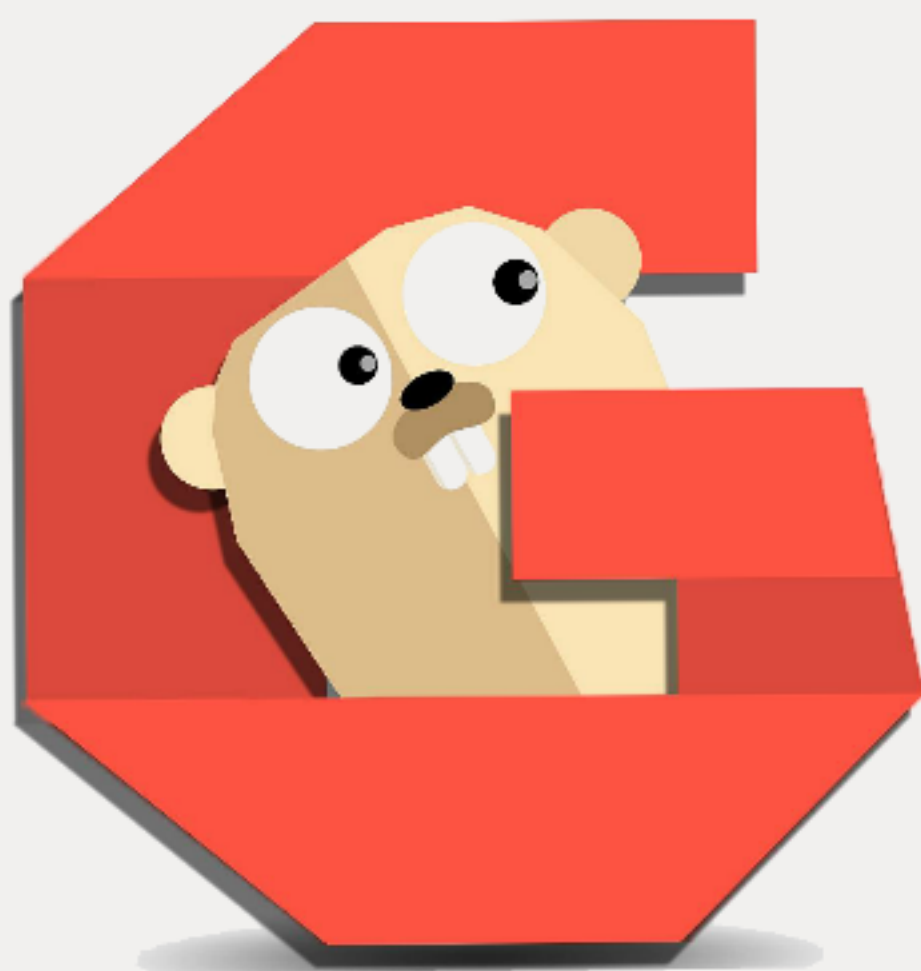
如果没有指定缓冲区容量，那么该通道就是同步的，因此会阻塞到发送者准备好发送和接收者准备好接收。



## 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