[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 0 - Getting Started](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Getting Started](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Blockchain 101](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Blockchain History](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Public and Managed Blockchains](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Consensus in Distributed Networks](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Cryptography](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Self-Assessment Quiz](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Go Introduction - First Steps](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Go Basics](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Go Interfaces](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Control Structures in Go](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Arrays and Slices in Go](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Standard Packages in Go](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Concurrency in Go](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Docker Introduction](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Introduction to the Interchain](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[The Interchain Ecosystem](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Getting ATOM and Staking It](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[A Blockchain App Architecture](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Accounts](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Transactions](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Messages](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Modules](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Protobuf](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Multistore and Keepers](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[BaseApp](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Queries](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Events](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Context](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Testing](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Relaying with IBC](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Interchain Security](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Bridges](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Migrations](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 1 Quiz](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 2 - First Steps](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[First Steps](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Setup Your Work Environment](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Run a Node, API, and CLI](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Ignite CLI](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Store Object](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Create Custom Messages](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Create and Save a Game Properly](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Add a Way to Make a Move](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Emit Game Information](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Record the Game Winner](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 2 Exercise](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[What is IBC?](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[IBC Token Transfer](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[IBC Tooling](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[What is CosmJS?](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Your First CosmJS Actions](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Compose Complex Transactions](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Learn to Integrate Keplr](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Put Your Games in Order](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Auto-Expiring Games](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Let Players Set a Wager](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Handle wager payments](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Integration tests](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Incentivize Players](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Help Find a Correct Move](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Understand IBC Denoms](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Go Relayer](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Hermes Relayer](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[CosmJS Advanced](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Create Custom Objects](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Create Custom Messages](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Get an External GUI](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Backend Script for Game Indexing](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[IBC Deep Dive](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[IBC Application Developer Introduction](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Create a Leaderboard Chain](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Run in Production](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Prepare the Software to Run](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Prepare a Validator and Keys](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Prepare Where the Node Starts](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Prepare and Do Migrations](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Simulate Production in Docker](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Tally Player Info After Production](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Simulate a Migration in Docker](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Final Exam](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[What's Next?](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

[Continue Your Interchain Journey](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html)

Docs Version Switcher

On this page

[Goroutines](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html#goroutines)

[Channels](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html#channels)

[#Copy link](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html#concurrency-in-go) **Concurrency in Go**

Go has built-in concurrency. Concurrency and parallel execution are different things: you need concurrency (*program structure*) to enable parallel execution; actual parallelism during execution depends on the hardware.

For concurrency, Go offers so-called *Goroutines* and *Channels*.

[#Copy link](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html#goroutines) Goroutines

A *Goroutine* is a concurrent thread managed by the Go runtime.

To call a goroutine use the following:



Copy

package main

import (

"fmt"

"time"

)

func doSomething(size int) {

for i := 0; i < size; i++ {

fmt.Println(i)

time.Sleep(time.Second)

}

}

func main() {

go doSomething(10) // go statement before a function creates a goroutine

go doSomething(10)

time.Sleep(10\*time.Second)

}



[Test it online (opens new window)↗](https://go.dev/play/p/6c1vJ2Xz9WB).

If you run this program, you will see that both doSomething(10) functions work concurrently. You can wait with time.Sleep(10\*time.Second) to see this in action.

[#Copy link](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html#channels) Channels

Go offers *channels* for communication between goroutines. Channels may be buffered or unbuffered. You can create an *unbuffered* channel with the following:



Copy

ch:= make(chan type)

You can use this channel to send and receive messages with the <- operator.

**Send to** channel ch as follows:



Copy

ch <- v

**Read from** channel ch as follows:



Copy

v := <-ch

Now write an example using channels:



Copy

package main

import (

"fmt"

"time"

)

func doSomething(size int, c chan int) {

for i := 0; i < size; i++ {

time.Sleep(100 \* time.Millisecond)

}

c <- size

}

func main() {

c := make(chan int)

go doSomething(10, c)

go doSomething(20, c)

go doSomething(30, c)

x, y, z := <-c, <-c, <-c

fmt.Println(x, y, z)

}



[Test it online (opens new window)↗](https://go.dev/play/p/MYdZRhyG36y)

Run this program. What happened?

In this case, you do not need to use time.Sleep anymore, because sends and receives are blocked until the other side is ready.

To avoid blocking, you can create *buffered* channels:



Copy

c:= make(chan int, 100)

When a buffered channel is full, sends to it are blocked. When one is empty, receives from it are blocked.

You can iterate over the values of a channel if it is closed:



Copy

package main

import (

"fmt"

"time"

)

func doSomething(size int, c chan int) {

for i := 0; i < size; i++ {

time.Sleep(100 \* time.Millisecond)

}

c <- size

}

func doAll(c chan int) {

d:= make(chan int)

go doSomething(10, d)

go doSomething(20, d)

go doSomething(30, d)

c <- (<-d)

c <- (<-d)

c <- (<-d)

close(c)

}

func main() {

c := make(chan int)

go doAll(c)

for i := range c {

fmt.Println(i)

}

}



[Test it online (opens new window)↗](https://go.dev/play/p/uYYXtXOO-72).

Always close the channel (c) before you iterate over it. If you want to wait for multiple communication operations, Go offers select. This works similar to switch:



Copy

package main

import (

"fmt"

"time"

)

func doSomething(size int, c chan int) {

for i := 0; i < size; i++ {

time.Sleep(100 \* time.Millisecond)

}

c <- size

}

func main() {

c, q := make(chan int), make(chan int)

jobs := 5

go func() {

for i := 1; i <= jobs; i++ {

doSomething(i\*10, c)

}

q <- 0 // done

}()

for {

select {

case x := <-c: // if we have a result

fmt.Println(x)

case <-q: // if we are done

fmt.Println("Finished")

return

default: // if we are waiting

fmt.Print("...")

time.Sleep(time.Second)

}

}

}



[Test it online (opens new window)↗](https://go.dev/play/p/BExHhvrWp5Z).

The default case will run if no other channel is ready.

synopsis

To summarize, this section has explored:

* How Go has built-in concurrency (program structure) which enables parallel execution, which in turn is dependent on the hardware.
* **Goroutines**, which are concurrent threads managed by the Go runtime.
* **Channels**, which permit communication between goroutines and can be buffered or unbuffered.



**Further readings**

Look into Mutexes, which we did not talk about here. This can be important for managing concurrency:

* [Dancing with Go's Mutexes (opens new window)↗](https://hackernoon.com/dancing-with-go-s-mutexes-92407ae927bf)
* [Go by example (opens new window)↗](https://gobyexample.com/mutexes)

previous

[](https://ida.interchain.io/tutorials/4-golang-intro/6-packages.html)

**[Standard Packages in Go](https://ida.interchain.io/tutorials/4-golang-intro/6-packages.html)**

up next

**[Good-To-Know Dev Terms](https://ida.interchain.io/tutorials/1-tech-terms/)**

[[](https://ida.interchain.io/tutorials/1-tech-terms/)](https://ida.interchain.io/tutorials/1-tech-terms/)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

[](https://ida.interchain.io/ida-course/discord-info.html)

On this page

[Goroutines](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html#goroutines)

[Channels](https://ida.interchain.io/tutorials/4-golang-intro/7-concurrency.html#channels)

#### **Get Cosmos updates**

Unsubscribe at any time. [Privacy Policy↗](https://v1.cosmos.network/privacy)

     Next

Documentation

[Cosmos SDK](https://docs.cosmos.network/)[Cosmos Hub](https://hub.cosmos.network/)[CometBFT](https://docs.cometbft.com/)[IBC Protocol](https://ibc.cosmos.network/)

Community

[Interchain blog](https://blog.cosmos.network/)[Forum](https://forum.cosmos.network/)[Discord](https://discord.gg/cosmosnetwork)

Contributing

[Source code on GitHub](https://github.com/cosmos/sdk-tutorials)

[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)

**[](https://blog.cosmos.network/)[](https://twitter.com/cosmos)[](https://discord.gg/cosmosnetwork)[](https://www.linkedin.com/company/interchain-foundation/about/)[](https://reddit.com/r/cosmosnetwork)[](https://t.me/cosmosproject)[](https://www.youtube.com/c/CosmosProject)**



Dark mode

† This website is maintained by the Interchain Foundation (ICF). The contents and opinions of this website are those of the ICF. The ICF provides links to cryptocurrency exchanges as a service to the public. The ICF does not warrant that the information provided by these websites is correct, complete, and up-to-date. The ICF is not responsible for their content and expressly rejects any liability for damages of any kind resulting from the use, reference to, or reliance on any information contained within these websites.

Cosmos is a registered trademark of the [Interchain Foundation.](https://interchain.io/)[Privacy](https://v1.cosmos.network/privacy)