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

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Week 0 - Getting Started](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Getting Started](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Blockchain 101](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Blockchain History](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Public and Managed Blockchains](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Consensus in Distributed Networks](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Cryptography](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Self-Assessment Quiz](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Go Introduction - First Steps](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Go Basics](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Go Interfaces](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Control Structures in Go](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Arrays and Slices in Go](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Standard Packages in Go](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Concurrency in Go](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Docker Introduction](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Introduction to the Interchain](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[The Interchain Ecosystem](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Getting ATOM and Staking It](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[A Blockchain App Architecture](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Accounts](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Transactions](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Messages](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Modules](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Protobuf](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Multistore and Keepers](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[BaseApp](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Queries](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Events](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Context](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Testing](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Relaying with IBC](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Interchain Security](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Bridges](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Migrations](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Week 1 Quiz](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Week 2 - First Steps](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[First Steps](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Setup Your Work Environment](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Run a Node, API, and CLI](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Ignite CLI](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Store Object](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Create Custom Messages](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Create and Save a Game Properly](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Add a Way to Make a Move](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Emit Game Information](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Record the Game Winner](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Week 2 Exercise](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[What is IBC?](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[IBC Token Transfer](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[IBC Tooling](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[What is CosmJS?](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Your First CosmJS Actions](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Compose Complex Transactions](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Learn to Integrate Keplr](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Put Your Games in Order](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Auto-Expiring Games](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Let Players Set a Wager](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Handle wager payments](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Integration tests](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Incentivize Players](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Help Find a Correct Move](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Understand IBC Denoms](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Go Relayer](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Hermes Relayer](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

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

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

[Create Custom Objects](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Create Custom Messages](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Get an External GUI](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Backend Script for Game Indexing](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[IBC Deep Dive](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[IBC Application Developer Introduction](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Create a Leaderboard Chain](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Run in Production](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Prepare the Software to Run](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Prepare a Validator and Keys](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Prepare Where the Node Starts](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Prepare and Do Migrations](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Simulate Production in Docker](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Tally Player Info After Production](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Simulate a Migration in Docker](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Final Exam](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[What's Next?](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

[Continue Your Interchain Journey](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html)

Docs Version Switcher

On this page

[Arrays](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#arrays)

[Slices](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#slices)

[Maps](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#maps)

[#Copy link](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#arrays-and-slices-in-go) **Arrays and Slices in Go**

In this section, arrays and slices are introduced.

[#Copy link](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#arrays) Arrays

In Go, the size of an array is a part of the type. Therefore, arrays have a fixed size. The declaration has the following syntax:



Copy

var array [size]type

You can access the data with array[index]. You can see this with a cross product:



Copy

package main

import (

"fmt"

)

func main() {

v1 := [3]float64{7, 5, 4}

var v2 [3]float64

v2 = [3]float64{2, 4, 6}

for v3,i := [...]float64{0, 0, 0}, 0; i < len(v3); i++ {

v3[i] = v1[(i + 1) % 3] \* v2[(i + 2) % 3] - v1[(i + 2) % 3] \* v2[(i + 1) % 3]

defer fmt.Printf("%t\n", v3)

}

}



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

The compiler fits the array depending on the number of elements.



The previous example code is not well-written, but it demonstrates different aspects of arrays.   
  
len(array) is a built-in function that gives the size of an array.   
  
defer is used to defer the execution of last-in-first-out order until surrounding functions return.

[#Copy link](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#slices) Slices

In Go, a *slice* is an abstraction built on top of arrays. Slices are more flexible than arrays and are used more often than arrays because of this flexibility.

A slice does not have a fixed size. To declare a slice, use the following:



Copy

var slice []type

A slice has a length (len(slice)) and a capacity (cap(slice)).

You can also use a built-in function to declare a slice: func make([]type, length, capacity) []type. This returns a slice with the given length, capacity, and type. It allocates an array, which is referred to by the returned slice.

Now create a simple slice with three vectors, and then add a vector with the built-in func append(s []T, vs ...T) [] T function:



Copy

package main

import "fmt"

func main() {

vectors := []struct {

x,y,z float64

} {

{ 1, 2, 3 },

{ 3.2, 4, 6 },

{ 4, 3, 1},

}

fmt.Printf("type %#T and value %v\n", vectors, vectors)

vectors = append(vectors, struct{ x, y, z float64 }{ 7, 7, 7 })

fmt.Printf("type %#T and value %v\n", vectors[3:], vectors[3:])

fmt.Printf("type %#T and value %v\n", vectors[3], vectors[3])

for i, v := range vectors {

fmt.Println(i, " : ", v)

}

numbers := make([]int, 10, 10) // create a slice with an underlying array

fmt.Println(numbers)

for i := range numbers {

numbers[i] = i

}

fmt.Println(numbers)

}



[Test it online (opens new window)↗](https://go.dev/play/p/T8Ppscz5YjO).

You can use range to iterate over an array, a slice, or a map. i is the index, and v is the value of that index.

There is also a built-in func copy(dst, src []T) int to copy one slice into another and return the number of copied elements.

[#Copy link](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#maps) Maps

Maps are stored key/value pairs. The declaration is as follows:



Copy

var m map[keyType]valueType

However, this creates a nil map, which is not so useful. You can read such a map but not write to it. You use make to initialize a map so you can write to it. The following is more useful:



Copy

m := make(map[keyType]valueType)

Now you can work with maps:



Copy

package main

import "fmt"

func main() {

age := map[string]int {"max": 24, "tom": 28}

fmt.Println("map:", age)

m := make(map[string]float64)

m["E"] = 2.7182818284

m["Pi"] = 3.1415926535

m["Phi"]= 1.6180339887

for key, v := range m {

fmt.Printf("Key: %v, Value: %v, Value: %v \n", key, v, m[key])

}

delete(m, "E") // does not return anything. It does nothing, if the key does not exist.

fmt.Println("len:", len(m))

fmt.Println("map:", m)

\_, ok := m["E"] // does the key exists?

fmt.Println("ok:", ok)

}



[Test it online (opens new window)↗](https://go.dev/play/p/1Ny9l13nHUg).

The built-in function func delete(m map[Type]Type1, key Type) deletes the element with the key from the map.



When iterating over maps, the order is not deterministic.



**Further reading:**

* [Go slices (opens new window)↗](https://blog.golang.org/go-slices-usage-and-internals)

synopsis

To summarize, this section has explored:

* How the size of an array is part of the type, therefor arrays have a fixed size.
* How slices are flexible abstractions built on top of arrays.
* How maps are stored key/value pairs.

previous

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

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

up next

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

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

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

[Arrays](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#arrays)

[Slices](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#slices)

[Maps](https://ida.interchain.io/tutorials/4-golang-intro/5-arrays.html#maps)

#### **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)