## [Course Overview](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd)

### [Course Overview](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd)

[Hi,](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=4.34) [I'm Nigel. I'm a longtime technology geek, and I've written best‑selling](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=6.43) [books on things like containers in Kubernetes, and I've got a ton of](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=10.65) [courses here at Pluralsight. But let me tell you,](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=14.66) [I am excited to bring you this course, Go Fundamentals. Because you](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=18.21) [know what? Coding is a massively useful skill, but as well, Go is](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=23.03) [easily one of the most useful languages you could ever learn like it](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=28.53) [is powering just about everything these days. Anyway, as you progress](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=32.9) [through the course, you'll do everything from the, well, mandatory Hello World,](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=38.55) [but all the way through decision making with conditionals,](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=43.42) [the basics of functions,](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=46.84) [working with data using variables, and things like slices and maps. And](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=48.51) [you know what? You'll even learn how to create your own custom data types](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=53.16) [with structs. Plus, and this is a big one, you'll get your head around](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=57.56) [concurrent programming with goroutines.](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=63.28) [So, yeah, a really exciting agenda.](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=67.34) [And you know what,](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=70.65) [whether you've coded in other languages before or you're an utter nube,](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=71.3) [this course has got so much for you. And then, when you're](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=75.16) [done, I promise you'll have everything you need to press on at](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=79.73) [your own pace. And you know what,](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=84.23) [maybe even start using Go in some of your own or your work projects.](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=85.66) [So, join me on this journey as we add programming in Go to your expanding list of skills.](https://app.pluralsight.com/course-player?clipId=6d082c27-e6a4-40b3-ad77-462ec18730cd&startTime=90.02)

## [Course Introduction](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f)

### [Course Introduction](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f)

[Right then, I'm Nigel, and welcome to this course, Go Fundamentals.](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=1.54) [And, I am stoked that you've chosen it, because honestly, Go is an](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=5.79) [epic language to learn. It's modern,](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=12.24) [it's relatively simple, and it is playing a huge role in](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=15.54) [shaping the future. Now, on that, just a few examples so you](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=20.15) [know I'm not just making stuff up,](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=25.23) [but, the who's who of major cloud infrastructure projects are](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=27.22) [all written in Go. So, Kubernetes, heck yeah,](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=32.14) [that's written in Go, Docker, Terraform, CoreDNS, Hugo,](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=36.97) [CockroachDB, Dropbox, SoundCloud; honestly, Go is the engine behind all of them.](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=41.53) [But you know what, outside of big projects and open source stuff like that,](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=50.84) [real companies hiring real people are using Go. So, we're not messing](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=54.75) [about, Go is a proper language, and it is one of the best tech skills](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=61.28) [to have in today's world. Magic? Well, let's go and see what we've got on the agenda.](https://app.pluralsight.com/course-player?clipId=ad74995f-4307-4469-b583-5fcd0cfbef1f&startTime=65.61)

### [Course Agenda](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f)

[So, first and foremost, this is an entry‑level course or entry‑ish.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=1.64) [And that's probably as clear as mud, so let me explain.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=8.04) [I'm obviously not going to be telling you what a keyboard or a mouse is,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=12.24) [and I'm not going to be showing you how to save files.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=15.7) [But,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=18.28) [what I will be doing is explaining Go stuff as clearly as](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=19.13) [is humanly possible, or at least for me.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=23.86) [So, I'm assuming you know your way around a computer,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=26.94) [but I'm also assuming that you've got zero clue when](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=31) [it comes to programming in Go.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=35.08) [So, yeah, go on, a quick example.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=37.74) [Let's say I'm explaining arrays and slices.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=40.32) [Okay,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=44.84) [well, I'll explain the lists of items that are all the same](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=45.27) [type, so maybe all integers or maybe all strings or](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=48.51) [something, and I'll compare it to, I don't know,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=51.69) [a class register with student names or something like that.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=55.73) [But I'll also be saying, hey,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=58.49) [if you're familiar with other languages, it's pretty much the same as X,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=60.58) [Y, Zed or X, Y, Z.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=64.06) [Anyway, I'm going to be as clear and concise as possible.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=67.14) [So, if you're new to it all, you get the fundamentals.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=70.23) [But if you've got some experience,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=73.37) [I'm not going to be boring you with the uber basics.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=75.94) [Well,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=79.74) [you know what, for everything we learn, we'll cover the](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=80.01) [theory and we'll get our hands on.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=82.16) [But look, I'm waffling.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=84.34) [This is what we'll cover.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=85.61) [We'll learn the history of Go and things like why it](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=88.34) [was created and how to install it.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=92.17) [Then, we'll train in the footsteps of all the programming greats](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=95.54) [and we'll write our very own Hello World program.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=99.38) [After that, we'll start getting into the weeds.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=103.84) [We'll see how to work with variables and constants.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=106.1) [We'll get our heads around functions. We'll do a bit of decision](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=109.34) [making with conditionals. We'll see how to loop.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=112.68) [We'll look at lists with arrays and slices. I guess more lists with maps.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=115.32) [And then, we'll round things out by looking at creating our own custom](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=120.7) [variable types with structs, and then the icing on the cake, with an](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=125.65) [intro into concurrent programming in Go.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=129.94) [And honestly, when we're done,](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=134.04) [well, on the downside, you will be sick of my voice, but on the](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=136.17) [plus side, you'll have a proper idea of how Go works and you'll be](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=140.59) [ready to power on with further learning.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=144.16) [Okay, well a quick mention of who I am and a bit about the courses repo on GitHub.](https://app.pluralsight.com/course-player?clipId=239ad4b1-f10c-4349-ad95-0eea5070577f&startTime=148.44)

### [About Me and About the GitHub Repo](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d)

[All right then, two quick things, who am I, and where](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=1.34) [can you get the code for the course?](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=5.02) [So, I'm Nigel,](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=8.04) [I'm a proper techaholic, and I'm from the UK, though really, I](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=9.44) [live on the internet, only, not in a creepy way.](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=13.32) [Anyway,](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=18.54) [this is me just about everywhere on the net, and I am more than happy to](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=19.25) [connect. And, really quick, I've worked in tech since my late teens, and](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=22.91) [I feel like I've done everything pretty much from writing code to](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=29.38) [Windows and Linux admin, networking, storage, and these days I mostly](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=32.8) [work with containers in Kubernetes, but that's enough about me.](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=37.09) [Throughout the course, we'll be using sample code files, and they're all](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=41.64) [on GitHub in this repo. Now, honestly, the stuff we'll be doing is so](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=45.88) [easy,](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=50.95) [you can type everything in by hand if that's your thing, but, if you want](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=51.55) [to, you can download it all with the following commands. Now then, you](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=55.85) [will need to install Git, but it's dead easy.](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=60.68) [Just Google, how do I install Git? And then, once you've got it,](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=63.68) [just clone the repo like this, and then switch into the directory,](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=68.48) [and from there, you've got all the code in the folders, just like on](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=74.94) [GitHub. Great, but seriously, don't stress if you don't know Git or](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=78.36) [GitHub, you don't need to, I'll be keeping things so simple to follow along with.](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=84.79) [But yeah, I reckon that's it. Come on, let's crack on with the course.](https://app.pluralsight.com/course-player?clipId=5ffaed65-7a62-4d76-ad9d-2bd0fb05838d&startTime=90.84)

## [Introducing Go](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a)

[Right, then.](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=1.74) [The plan for this module, I think, is to talk a bit about Go as a language,](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=2.94) [like what do we mean when we say things like it's a compiled,](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=8.2) [statically‑typed systems language that does its own garbage](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=12.55) [collection, it infers, it supports concurrency,](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=15.82) [all of that jargon, what does it mean?](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=18.87) [Now,](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=22.74) [look, if you already know all of that, magic. This course will](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=23.43) [be a walk in the park. But if that actually sounded like I was](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=27.45) [speaking Klingon or something,](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=31.64) [that's alright as well. We'll explain everything as we go.](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=33.76) [And you'll be putting it all into action,](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=37.86) [so you'll pick it up. And you know what? By the end of the course,](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=40.24) [you'll be like, wow, I actually know what all that means.](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=43.1) [Anyway, after that, we'll see how to install Go, which is super easy. Alright then, let's try and explain Go.](https://app.pluralsight.com/course-player?clipId=d4a3d77a-b45d-45ba-80cb-c955f5cd179a&startTime=47.24)

### [Go Overview](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e)

[Okay,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=1.84) [so a quick 5 or so minutes getting our head around what](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=2.55) [Go is and maybe its major features.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=6.31) [Well, I reckon first up we need to address the gopher in the room.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=10.24) [This is the Go gopher, and it's the official mascot of Go. And no,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=14.54) [it doesn't have a name.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=19.55) [It is simply the Go gopher, but it is a big part of the](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=21.14) [community, and you're going to see it everywhere.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=25.21) [Now interestingly, it was designed and drawn by Renee French.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=29.1) [She's a well‑known writer and illustrator, and she's of](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=33.73) [interest to us because she's married to Rob Pike, one of the](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=36.65) [three original creators of Go, seriously.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=40.11) [So look, Go was created as a language at Google by Robert Griesemer,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=44.44) [Ken Thompson, and Rob Pike.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=49.1) [So, Rob Pike, one of the Go founders,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=51.86) [his wife designed the gopher. Anyway, work on the](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=54.08) [language started in something like 2007.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=57.85) [And as the legend goes, while the three founders were waiting](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=60.95) [an eternity for the C++ program to compile,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=63.99) [they decided to sketch out what C or C++ would look like if they designed it,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=67.27) [or do you know what? Maybe what it would look like if it was designed then in](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=71.97) [2007, rather than like 30 or 40 years earlier.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=76.17) [Well,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=81.04) [fast forward a couple of years to 2009 when that C++ program had almost finished](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=81.34) [compiling, and they released the first publicly available version of Go, and](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=87.04) [they made it open source under a BSD style license.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=92.4) [So, at the time that I'm producing this course, Go](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=97.44) [is well and truly established.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=100.75) [I mean, okay, it's properly young compared to the likes of C and C++,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=102.48) [both of which are 70s and 80s babies.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=107.39) [But you know what?](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=109.99) [It's even young compared to the likes of Python and Java.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=111.05) [Now you can look at it being young in two ways.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=116.14) [Yes, It's obviously not as mature as some of those other languages,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=118.78) [but hey, they were all written way back in the last century.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=123.14) [So no, Go is not as mature as some of those,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=127.57) [but at least it was written in this century.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=130.77) [And when you consider how much the world has changed with the advent](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=133.94) [of cloud and the likes, then Go as a pretty modern language, it's](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=136.68) [well suited to this century's demands.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=140.99) [In fact,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=144.14) [you know what? The who's who of major cloud](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=144.5) [infrastructure projects are all written in Go.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=147.62) [So Docker and Kubernetes,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=150.94) [that's right, both written in Go. Etcd and](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=152.9) [Terraform, you've got it, written in Go.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=155.83) [So yeah,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=159.64) [it's young, but it is widely used and very much production‑ready. Now, I](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=160.42) [think it's fair to say that Go was designed as a systems language. I'm](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=167.99) [using air quotes there. So, it was designed for things like building](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=171.97) [operating systems and infrastructure projects like Docker and Kubernetes,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=176.05) [yeah? But do you know what?](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=179.99) [It's evolved over the years to be a solid choice for things](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=183.34) [like web services and other high‑level apps.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=186.69) [So today, Go is a solid all‑rounder, and it is a cracking language for](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=189.55) [you to learn. Now then, as a language, it is a bit like C and C++,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=194.82) [only fortunately, it is way simpler. Like as a language, it's cleaner](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=201.72) [and more concise than C, plus,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=207.74) [it does its own garbage collection, which C doesn't. Now, I'm](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=209.87) [aware that I'm throwing around the odd buzzword. And if you're](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=214.43) [not sure about what some of those mean,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=216.89) [don't sweat. We'll be explaining them all later on, and](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=218.83) [you'll even see most of them in action.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=221.82) [But for now, garbage collection is just the language itself](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=224.24) [taking care of memory management. You know,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=228.16) [like deallocating memory and the likes automatically](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=230.96) [so that you as a developer, well,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=233.86) [I guess you can just think about other stuff, yeah? Anyway, Go only](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=235.73) [has 25 keywords, and that really isn't many. It's definitely less](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=241.23) [than C and Python, and it is way less than Java, so technically](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=246.49) [easier to learn. Obviously as well, it's cross‑platform,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=251.38) [so you're good to go on Linux, Windows,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=256.13) [macOS, BSD, and even good old Solaris. It is a compiled language. That means](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=258.67) [the nice, easy‑to‑read code that we're going to write,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=266.4) [that gets compiled down to machine code that runs](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=268.88) [without an interpreter. Buzzwords again.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=272.02) [Okay, tell you what. My first programming language,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=275.51) [FoxPro, way back in the day,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=279.05) [that was an interpreted language. Jargon that meant the](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=281.33) [programs I wrote in FoxPro needed a separate piece of](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=285.24) [software called an interpreter to run them.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=288.28) [So I'd write programs for customers, and then for the customers to run them,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=290.83) [they'd basically need to install the FoxPro interpreter to](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=294.82) [run my programs, and I was never a fan.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=298.06) [Well, it's not like that for Go programs. They compile](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=302.24) [down to native machine code that run on their own without](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=306.05) [the need for an interpreter.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=309.94) [And not only is that a whole lot simpler in my view,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=311.26) [it also makes them a whole lot faster.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=314.09) [Go is a strongly‑typed language.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=318.24) [This means it can be picky when it comes to variables and the likes.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=322.34) [So, maybe, I don't know, if you try and add an integer with a](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=326.35) [floating point number, that's not going to work.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=331.26) [The compiler is going to throw a type mismatch and your code just won't run.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=333.91) [And I will say, if you're not used to this kind of behavior,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=337.97) [I'm not going to lie, it can take a bit of getting used to.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=342.27) [In fact,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=345.74) [you know what? Strong typing is a bit like someone](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=346.27) [that's really picky with grammar, you know,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=349.38) [the self‑appointed grammar police. I'm sure most of us know at least](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=352.28) [one person like that, and, of course, we love them.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=355.65) [Well, the strongly‑typed nature of Go means it's really strict about](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=358.74) [assigning the right types to variables. And yeah,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=362.78) [this means we're probably more likely to get errors at compile time,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=367.01) [but it does make us write good, clean, predictable code. Oh yeah,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=370.51) [sticking with types, Go supports type inference.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=376.55) [So, we can write something like this on the screen, and Go's going to look](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=381.14) [at it and say, okay, 5.5, that looks like a floating point number to me, so](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=384.75) [it will assign the variable type as a float.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=390.13) [But then maybe if we assign 10 to a,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=393.54) [well, this time it's going to assign it as an integer.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=397.74) [Now,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=401.34) [if you compare this type inference to something like C](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=401.97) [where we have to explicitly state the type,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=404.99) [you might be able to see how Go can be simpler and more concise.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=407.41) [Well, do you know what? I reckon that'll do for now. Go is a modern, clean,](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=412.64) [simple, and concise language.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=416.55) [It's great for systems programming, but it's also a really good all‑rounder.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=418.2) [We said it is strongly typed, and it compiles down to machine code, magic.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=422.15) [Now let's go and see how to install it.](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=428.13) [Sorry, I had to do that at least once. Come on, let's go!](https://app.pluralsight.com/course-player?clipId=9ffa7243-9456-4ec7-8891-8ac79d02853e&startTime=431.55)

### [Installing Go](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e)

[Alright, installing Go. And, I'm just going to](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=1.94) [level with you, for the most part, this is so easy,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=5.29) [it's probably not even worth me recording the lesson,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=8.85) [but I am, so I'll make it quick.](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=11.94) [We already said that Go is cross‑platform, and I actually meant a couple of](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=15.34) [things when I said that. I meant it compiles programs that run natively on](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=19.24) [different operating systems and architectures,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=24.5) [but, I also meant you can install Go itself, so, the language,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=27.44) [the compiler, and the other tools, also on different OS's and](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=32.69) [architectures. Now, installing on Linux, Windows, and Mac are easy.](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=36.37) [It's a clickety‑click wizard for Mac and Windows and a walk in the park](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=42.51) [command line for Linux. For anything more exotic, sorry,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=46.59) [you're probably installing from source, which is basically jargon for harder,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=52.5) [you know, no nice wizard or whatever.](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=57.25) [Anyway, look,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=59.94) [I'm on a Mac right now, and I literally typed install Go into](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=60.77) [my favorite search engine, and I ended up here at the](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=64.51) [official Download and install page.](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=67.14) [So, I'm just going to click this big Download Go button,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=70.04) [wait for that to download, and then I'll fire it up. A few clicks](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=74.64) [and the likes, and check that out, all done.](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=81.57) [So I think a quick check in the terminal, and I am ready to rock and roll,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=86.94) [told you it was easy.](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=93.04) [So you've now got the Go compiler and all the other tools that you need to](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=95.34) [write Go programs and build them into executable apps. Now you've also got the](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=100.76) [standard library packages that you'll get familiar with later in the course.](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=106.83) [Now, I know that was just a Mac install,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=111.44) [but it's practically identical on Windows, and like I said,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=113.88) [it is a simple command line operation on Linux. And, if you are following along,](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=117.26) [bring it on, because you are ready to rock and roll with Go. Speaking of which, next up, we're writing code.](https://app.pluralsight.com/course-player?clipId=4dd433bd-9077-4f3b-9acf-2959bd06511e&startTime=122.8)

## [Hello World](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed)

[All right, this is where the good stuff starts.](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=1.64) [We'll start the ball rolling with an overview of Go modules,](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=7.04) [packages, and source code files.](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=10.47) [Then we'll roll our sleeves up and get our hands on.](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=13.34) [We'll start out by creating a workspace and initializing a Go module, and](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=17.94) [then we'll start writing our first program. And it's okay if some of this](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=23.36) [is jargon. We're about to demystify it.](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=27.24) [Anyway,](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=30.64) [as we're writing that code, we'll dip into the notion of functions;](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=31.34) [we'll circle back to our code and import a package that lets us print](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=35.07) [text to the screen; we'll see how to run,](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=38.66) [build, and install the fully compiled app; and then we'll wrap the module with](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=42.35) [configuring VS Code for Go. Now, it might look a bit complex,](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=47.94) [but believe me, it is not.](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=54.21) [There's basically three things, writing the code, building it as an app and](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=57.44) [running it, and then configuring VS Code. And you know what?](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=62.71) [The VS Code bit's totally optional.](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=66.34) [You don't have to, and you can totally follow the course without it.](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=68.96) [But, I recommend you do, because if you're getting serious about Go,](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=74.34) [it's probably the most popular tool available, and it's free. But you know what, talk is cheap. Let's crack on.](https://app.pluralsight.com/course-player?clipId=606c95c4-d535-4d4e-b00a-80cb2a9e23ed&startTime=78.28)

### [Modules, Packages, and Source Code](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5)

[Okay, so a bit of housekeeping before we crack on.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=1.64) [First up, a version check.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=5.27) [Everything we're going to do is going to work fine in Go version 1.14 and newer.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=7.24) [You know what?](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=14.34) [It should even work as far back as 1.11.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=15.04) [You just need the $Go11MODULE variable set on for anything before 1.14.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=17.67) [And the other thing you need to know is the](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=24.34) [structure of Go packages and modules.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=26.15) [So a Go application consists of one or more source code files.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=30.04) [All files relating to a particular program get grouped into a](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=34.94) [package, and packages get grouped into modules,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=38.6) [which I know is a ton of jargon, so let's look at it from the inside out](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=42.64) [and then from the outside in. From the inside out,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=47.3) [you write your app as a bunch of source code files that](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=51.12) [live in the same folder on your computer.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=54.11) [Well, in Go we call that folder and source code a package,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=56.94) [and that means when you compile the app,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=60.92) [you're actually compiling the entire package, so all the](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=64.35) [source code files in that directory.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=68.84) [Now,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=72.24) [we'll see it in just a second, but because all these different](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=72.76) [files are part of the same application package,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=76.18) [then variables and functions and stuff that are in them are](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=79.14) [accessible to all the other files in the package. Anyway,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=82.55) [a module is then a set of related packages that you release together.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=86.82) [Magic.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=92.14) [But in the hands on,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=93.54) [we'll be doing it the other way, from the outside in,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=94.7) [so we'll create the module first,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=97.69) [then we'll create the packages, and finally the source code files.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=100.24) [So in a second, you'll use the go mod init command to](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=104.64) [initialize a new Go module in an empty directory.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=108.19) [Then you'll create a subfolder for your first package, and finally,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=112.34) [you'll start writing code.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=115.86) [Now,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=118.04) [I totally get that it can look a bit complex and maybe even](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=118.38) [a bit daunting if you're new to this,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=121.26) [but I promise you, you'll set this stuff at once,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=123.07) [and for the rest of the course,](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=126.69) [you'll just be coding and barely even know that the](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=127.75) [package and module stuff's even there.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=130.47) [But you know what, I literally said talk is cheap, and I'm still talking, so let's go and get our hands on.](https://app.pluralsight.com/course-player?clipId=cae237f6-2ade-4a2b-9f35-59948d0911a5&startTime=134.14)

### [Initializing a Go Module](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506)

[Alright,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=1.34) [then. We're about to create and initialize a Go module and then](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=1.7) [write some source code for our first package.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=5.82) [So the first thing you need is an empty folder somewhere on your computer.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=9.94) [I don't know,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=13.94) [it's probably easiest to create a folder called go in your home directory,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=14.6) [but you don't have to.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=19.22) [I mean,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=20.2) [I'm down in the depths of some arcane folder structure that I've](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=20.72) [been using for years for my Pluralsight courses.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=24.38) [So feel free to choose anywhere.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=27.64) [It just needs to be an empty folder on your machine that you'll](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=29.58) [use to store all the programs we'll write. Well,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=33.49) [once you've got that, you need to initialize it as a Go module.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=36.75) [So making sure you're in that module folder you just created,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=41.24) [run the go mod init command like this.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=44.74) [Now then, this bit here can pretty much be whatever you want, but if](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=48.84) [you plan on ever publishing the code in the module,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=54.5) [then it's got to be a path to the repository where](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=57.49) [you'll eventually host the code.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=60.65) [So for me, the code for this course is hosted on GitHub in this repo.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=63.44) [And look, if you're unsure about this or you don't think there is a cat](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=69.34) [in hell's chance of you ever publishing this code,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=73.62) [which there probably isn't, I mean it is just learning code from the](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=76.12) [course, then maybe just use your name or something like this.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=79.68) [It really doesn't matter.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=83.15) [Okay, so that's created this go,mod file here in the root of your module folder.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=85.94) [And if we look at it, yeah,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=91.68) [we can see it lists the name or the path to the module,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=94.67) [and then this is the version of Go that created it.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=97.73) [Now, the whole idea of modules is that how GO tracks dependencies and versions.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=101.84) [So, as you add dependencies to your apps, they'll appear in](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=108.04) [here with the required versions and the likes,](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=112) [but right now, that's your Go module initialized, and we're ready to write some code.](https://app.pluralsight.com/course-player?clipId=867c0bd8-50f3-44a5-a5e5-e16cc61ef506&startTime=114.24)

### [Writing Your First Program](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5)

[Remember we said modules contain packages,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=1.54) [and packages contain source code files.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=3.88) [Well, we've got our module,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=6.74) [so let's create a new folder inside of it for the hello‑world package.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=7.97) [And see how the name of the folder is the name of the package.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=13.45) [Well,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=18.34) [any code you write in this folder is going to be](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=19.01) [part of that hello‑world package.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=21.84) [Well for now, I'm going to write the code in Vi.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=24.74) [If you're on Windows, I recommend you probably go with Notepad,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=26.79) [and for sure we're going to get more professional in a second.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=30.23) [I just want the first program we write to be the simplest](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=33.6) [thing in the world with no fancy tools distracting us from](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=36.59) [what's actually going on.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=39.6) [Anyway, I'm calling the file hello.go.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=42.04) [Right, the first thing every Go program needs is a package declaration.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=45.2) [Now calling a package main makes it special.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=51.84) [This tells Go to compile this code that we're going to write](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=55) [as an application and not a shared library.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=58.43) [So, actually,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=61.07) [an application's something that maybe you would](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=62.18) [double‑click to run and it does something,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=64.76) [whereas a shared library generally only runs when another application calls it.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=66.62) [Anyway, look, next up, we want a function, and we're calling this one func main.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=72.54) [Be careful how you pronounce that. But func main's special,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=78.81) [just like package main, because at runtime,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=83.43) [so when we eventually run this as an application,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=85.99) [the first thing that'll run will be func main inside of package main.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=88.62) [And sometimes we call this combo the application's entry point.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=93.58) [Basically, without these two, the application won't run.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=97.06) [Now, you know what,](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=100.95) [functions are pretty much at the center of](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=103.32) [everything you're going to do with Go.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=105.89) [So I want to give you a quick heads up on function basics before we go any further.](https://app.pluralsight.com/course-player?clipId=0a4a84d1-1c5b-4d09-bd26-7551a1c486c5&startTime=107.73)

### [Functions Primer](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b)

[Right, then. Functions are at the heart of modularizing your code, and pretty](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=1.54) [much all the code you'll ever write in Go is going to be inside a function. Like](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=6.4) [just about every Go program has loads of functions.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=12.32) [In fact, it's pretty much one function per, well,](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=16.6) [function or maybe feature. Like if you program takes some input from the user,](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=20.79) [maybe it performs a conversion on that input and then returns the converted](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=26.1) [output to the user, you're probably looking at three functions, so one for](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=29.93) [each feature or function that the program executes. Now, we'll see it as we](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=35.21) [go, but functions and packages are the two major ways that we modularize and](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=40.41) [reuse code in Go. Now, at a high level, every function starts with the func](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=45.81) [keyword, followed by a name. And for the most part, we can name them](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=51.99) [anything we want that is except for name, which we just said is special.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=56.41) [But, I highly recommend you give your functions meaningful names.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=61.74) [Well, we can pass and return values to a function in parentheses, and then the](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=67.74) [actual code of it goes in between curly braces. And again,](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=73.09) [I'm really sorry about this, but if it's jargon right now,](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=77.6) [honestly it's fine.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=80.38) [We'll be doing it really soon.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=81.47) [But that's the basics of functions. But you know what? For our simple](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=83.36) [app, we're only interested in this main function here.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=87.6) [So we already said that func main is special.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=91.94) [It's the first block of code that'll execute in any Go application, so we](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=94.61) [call it the entry point. And in fact, actually, if you're writing a shared](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=98.65) [library and not an executable, then it doesn't get a main function. But we](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=103.88) [are writing an executable, so func main is our entry point, and it's](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=108.86) [special again because it doesn't take any arguments, and it doesn't return](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=113.1) [any values.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=117.67) [Well actually, you know what? When func main exits,](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=118.4) [the entire program exits, and it returns an exit code,](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=121.76) [which, of course, follows the standard practice of 0 indicating success of the](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=124.91) [application and a non‑0 code indicating some kind of error.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=129.72) [But for us right now, that's the very basics of Go functions and func main.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=134.64) [But you know what? Right now our function's empty, so let's go and add some code.](https://app.pluralsight.com/course-player?clipId=1368d3dc-0cf7-4e10-b4ba-d5f57094322b&startTime=141.14)

### [Importing Packages](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2)

[Okay.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=1.44) [The stuff we've got so far is pretty much the structure of a Go program.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=2.12) [All that's missing is the actual program itself.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=5.94) [Well, as this is our Hello World package,](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=9.24) [we're going to be printing some text to the screen. And we do this in Go](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=11.8) [with the Println function from the fmt or the FMT package.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=15.76) [So, up at the top here outside of our function,](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=20.44) [we need to import the fmt package. Now, well actually two things, feel free](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=23.05) [to pronounce this however you want, fmt, FMT, even format,](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=29.09) [everyone knows what you mean.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=33.51) [But the main thing to learn is we're importing a package of](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=36.14) [functions to be able to use them in our app.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=39.84) [So to be clear,](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=43.24) [fmt here is a package in the standard Go library that gets](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=44.45) [installed in your machine when you install Go. And like any](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=48.35) [package, it is a collection of functions.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=51.45) [In fact, give me a second here.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=54.59) [Okay, yeah,](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=57.34) [this is the official fmt package hosted on GitHub. And](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=58.45) [then we can see if I highlight here, the Println function.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=62.65) [So,](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=67.54) [by importing this into our package, we'll be able to](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=68.18) [call on this Println function here.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=72) [So I'll tell you what,](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=75.34) [let's jump back to our function, and we'll add this line here. And all](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=76.1) [it's saying is from that fmt package that we just imported, use this](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=80.69) [Println function, which we just saw, to print this text to the screen.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=85.62) [And that's our first program right there. Make sure you save it, and we're ready to see it run.](https://app.pluralsight.com/course-player?clipId=1e122783-52d6-4990-9f5d-89b5f11fcff2&startTime=90.45)

### [Running and Compiling Go Apps](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a)

[Okay.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=1.54) [So there is a few ways that we can run our app. We can just run it, we can build](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=2.22) [it, or we can install it, and they're all dead simple so we'll do them all. Now,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=7.53) [probably the one you'll do most is just run it.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=14.52) [So in a terminal making sure we're in a directory where the app](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=17.44) [is and as long as you've got Go installed,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=20.7) [just go run and then the name of the file. And there we](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=24.32) [go, Hello Pluralsight. And if that's your first Go](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=28.82) [program, seriously, congratulations.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=32.59) [Now what happened there was that Go compiled the app](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=36.84) [into a temp directory somewhere.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=39.77) [It ran it and then it cleaned up.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=41.67) [And honestly, you'll do this a lot,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=45.04) [especially while you're developing and testing your apps,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=47.54) [but when your programs are ready to ship,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=51.44) [you'll either build them or install them.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=53.75) [So go build here takes the same file name, or if](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=56.44) [you're in your packages directory,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=60.53) [you can just go with go build on its own or you can add a period](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=62.13) [and it'll compile the app as a binary executable file for your](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=66.04) [operating system that you can just run.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=70.35) [So you know what, this time I'll run it without the name of the file](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=74.14) [because I'm in its directory. Now, if we list the directory,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=76.78) [see how we've got a new file called hello‑world.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=81.34) [Well, that's the compiled application ready to run.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=84.21) [Oh,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=87.19) [and it's called hello‑world because we're in the](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=87.64) [package folder called hello‑world.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=90.69) [So maybe if we had multiple source code files here making](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=94.24) [up the app and we ran the build command,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=97.26) [they'd all get compiled into an executable binary with the name of the package.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=99.22) [Well,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=105.54) [I'll tell you what, if we run that executable, there we](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=105.8) [go again, Hello Pluralsight. In fact, you know what?](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=108.84) [You can even run it from Finder or Windows Explorer like a normal app](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=112.55) [because it is. And like I think I said, this is useful for building apps](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=117.37) [that you're ready to ship to your customers.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=124.91) [Well,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=127.84) [the final thing we mentioned was go install and this does the](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=128.28) [same as go build, only it places the compiled binary in a folder](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=131.22) [called go/bin in your home directory.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=135.64) [So I should have a file here, which I do, and like we said,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=139.84) [it's called hello‑world in the go/bin folder of my profile.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=146.08) [Now you can change the location where it drops the file by](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=152.34) [setting the GOBIN and the GOPATH variables.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=155.09) [I'm showing the commands here,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=158.27) [but actually I'm cool with the default location. And seriously,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=159.7) [that's our first program working and compiled as a](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=163.82) [self‑executing application. Now, yeah, go on then.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=167.47) [I want to just clarify something in case you're new to coding.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=171.49) [When I say compile,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=174.27) [I mean Go taking the programs that we write so our source code files and](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=176.13) [building them into a machine code app that'll run like we just showed. Okay.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=180.81) [Well everything we've done so far is well and good,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=188.04) [but I'd say probably all of you are going to write and run your](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=191.06) [code in an integrated development environment like Visual Studio](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=194.48) [Code. Yeah? Other IDs do exist and you can totally follow the](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=198.02) [course with other environments,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=203.08) [but I'll be rocking for the rest of the course with VS Code,](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=205.84) [and unless you've got a reason not to, do you know what, I recommend you probably do as well.](https://app.pluralsight.com/course-player?clipId=9108bfea-4a7b-436c-aa8d-de58bf00560a&startTime=209.13)

### [Visual Studio Code](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f)

[Right, then.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=1.44) [Let's see how to get VS Code and how to configure it for Go.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=2.2) [Now, a couple of things right off the bat.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=6.13) [VS Code's just a tool to help us with Go, so I'm not going to dwell on it.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=8.59) [We'll get it configured, and we'll crack on with the course.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=13.05) [The other thing,](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=16.84) [what I'm about to show you might change slightly in the future,](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=17.58) [and there's not a lot I can do about that,](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=21.33) [but for the most part,](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=23.28) [any changes will be minor, and what you see here is still going to](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=24.77) [apply and help you configure VS Code. Alright then,](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=28.56) [VS Code is a free IDE, that's integrated development environment.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=32.57) [And do you know what? The download and install is so simple, I'm not even](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=38.14) [going to show you. Just download the version for your OS and crack on with the](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=41.88) [installer. When you're done with that, fire it up.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=46.3) [Well,](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=51.04) [it looks pretty professional, yeah? But all we've got right now is a framework.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=51.58) [First order of business is to install the Go extension.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=56.68) [So I'll hit the extension's icon here and type in go at the top.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=61.14) [This is the one we want, so we'll install it. And if](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=65.54) [it recommends any updates or tools, install them.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=69.51) [And then when that's done,](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=73.44) [I think the last thing to do is add a workspace over here, and that's](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=75.49) [just the root folder of the module you created before.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=79.13) [Oh, yeah, you might be asked to trust the workspace and maybe install](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=84.54) [additional tools like gopls or whatever. I'm going to say yes.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=88.15) [This might take a second as well. But when it's done,](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=93.24) [this is a module with the hello‑world package and then your source code file.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=97.84) [In fact, see how the code's colored.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=102.62) [That's because the Go extension recognized this as a Go program.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=105.53) [Magic.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=110.24) [Well, that's VS Code ready to go. Time, I think, for a quick recap before we crack on digging deeper.](https://app.pluralsight.com/course-player?clipId=9c719286-bf58-4eda-a2c9-545dd03f192f&startTime=111.07)

### [Recap](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9)

[Magic. You've just written a Go program.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=1.54) [So you initialized a brand new Go module, and we've not](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=4) [really got into modules yet, but I'd say for now, that Go's](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=8.43) [built‑in dependency tracking system.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=11.9) [But once they're installed, they'll pretty much just sit in the background,](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=15.44) [for most of this stuff we'll do. Anyway, within that module, we created a](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=18.99) [package called hello‑world, and we wrote a program.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=24.49) [So, as we can see,](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=28.14) [source code belongs to packages, and packages belonged to modules.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=30.03) [And it'll sink in later if it hasn't already.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=33.98) [But looking at our code,](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=37.84) [we started by saying every Go program that we want to compile as an](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=39.44) [executable application needs a package main and a func main.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=43.77) [These are what we call the applications entry point, and therefore, the](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=49.04) [first bits of code to run when the program starts. This line here](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=52.88) [imports the func package and makes all of its functions available to our](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=57.95) [code so that down here in the main function, we can call on the Println](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=62.24) [function from the fmt package and use it to display text to the screen.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=66.78) [With the code all written, we then use the go run command to run the app.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=71.72) [And we said that's probably how you're going to test](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=76.26) [your code while you're developing it.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=78.28) [And I'm sure we'll see it later, but if we make typos or whatever,](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=80.19) [they'll get flagged and we'll have an opportunity to fix them. Anyway,](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=83.16) [when you're happy with your program and you think it's ready to ship, go](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=87.48) [build is the command that you'll need. This compiles your code into a](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=91.18) [binary executable that runs on its own on other computers without the](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=95.87) [need for the Go compiler.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=100.91) [And I think that was pretty much it.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=103.54) [So with these basics ticked off, let's start digging deeper with variables and constants.](https://app.pluralsight.com/course-player?clipId=9bd62478-6056-4b04-b0ad-52aa1e751be9&startTime=106.74)

## [Working with Variables and Constants](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d)

[Okay, check that out for a module title.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=1.64) [Total opposites, yeah?](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=4.71) [So variables are designed to, well, vary, yeah?](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=6.02) [Whereas constants, not so much.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=10.89) [Anyway, here's the plan.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=14.44) [We'll look at the basics of variables and how we create them.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=15.68) [We'll look at some of the different types of variables Go](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=19.94) [supports, and then we'll do a bit of a sidestep, showing the](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=22.74) [common short assignment method.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=26.13) [Now that's buzzwords at the moment, but you'll get it when we get there.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=28.26) [After that, though,](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=32.14) [we'll look at pointer variables, and we'll see how they let us](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=33.09) [pass variables by either value or reference.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=36.57) [And of course,](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=39.77) [again, that's more buzzwords, but we'll demystify everything as we go.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=40.39) [Anyway, look,](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=44.74) [we'll round things out with a look at constants, then how we](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=45.44) [can access environment variables on your system, and, of](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=49.04) [course, we'll do a quick recap.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=52.37) [Now,](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=55.34) [look, to help with the demos, we'll use a simple program that](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=55.71) [shows info on a Pluralsight subscriber, so maybe the name,](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=58.7) [which course they're watching, and whereabouts in that](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=62.8) [course they're up to, I don't know, like, maybe module 4,](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=64.78) [clip 2 or something.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=67.74) [But the thing is right. It's just a daft, oversimplified program whose](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=69.24) [only purpose in life is to introduce you to variables, o it is not](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=73.41) [supposed to be production‑grade code. Anyway,](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=78.89) [look, we know by now that talk is cheap, so let's get coding.](https://app.pluralsight.com/course-player?clipId=8281a605-e546-4621-ad0e-441c658c5f9d&startTime=81.46)

### [Declaring Variables](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c)

[Okay.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=1.34) [We already said that Go is a statically‑typed language and we said that's](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=2.14) [jargon for, well you know what, it means a few things actually,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=7.23) [but for now, it means when we declare a variable,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=11.16) [we've got to tell Go what type it is like is it a text string or](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=14.55) [maybe a floating point number or something.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=21.4) [Anyway, look, we've got the framework of a program here,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=24.94) [it's actually in the variables folder of the course's GitHub](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=28.05) [repo and I think it's called var‑framework.go.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=31.04) [Maybe if you want to clone it locally and work from that,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=34.32) [and if you do, the commands are here and you'll need git](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=37.15) [installed to be able to do that.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=39.85) [But you know what,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=41.57) [you don't have to do that so maybe feel free to just type the](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=42.27) [code into your editor, or you know what,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=45.9) [even copy and paste it from the file on GitHub. It's all good as](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=47.67) [long as you've got that code in your editor.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=51.8) [Well, if we declare variables up here at the package level,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=54.74) [so outside of any function, we've got to do it with the var keyword.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=58.38) [So no, Go doesn't let you use non‑declaration statements at the package level.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=64.04) [Okay well,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=70.74) [for now we go var and then I like to list them out on their own lines like this.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=71.44) [So each line here is starting with the name of the variable and then](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=76.71) [the type. Now, on the topic of names actually,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=80.7) [I'm not sure if I said this already, maybe I did,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=85.13) [maybe I didn't,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=87.73) [but variable names have got to start with a letter and they can't be any of](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=88.47) [the 25 Go keywords that were shown here on the screen.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=93.94) [So I mean you can't call a variable var or func, or to be honest,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=97.64) [any of these 25 words on the screen. As well as that though,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=102.51) [they can't contain spaces and no special characters](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=106.04) [like math operators and the likes.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=108.99) [Oh, do you know what, and of course look,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=110.72) [do yourself and the universe a favor and use common sense like for](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=113.74) [the love of all things good, give variables meaningful names and](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=118.33) [preferably short names. And as well,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=122.29) [you know what, if the variable name is long and it contains more than one word,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=125.53) [then capitalize the first letter of each word just to make](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=130.04) [things more readable, camelCase here.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=133.27) [Do you know what?](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=136.64) [And on the topic of this, for now right,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=137.81) [always make the first letter of a variable lowercase because](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=140.23) [and I don't really want to get here now,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=145.14) [but we'll find out later in the course that making the first letter of a](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=146.7) [variable a capital letter, that makes that variable visible outside of the](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=150.27) [package it's defined in, but you know what,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=155.31) [that's a bit advanced for now, we'll get to it later.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=157.27) [Well look,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=159.94) [it is possible to list variables of the same type on the same line like this.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=160.64) [So name and course are strings and then module and clip are](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=166.15) [integers. And then actually declaring variables like this where](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=169.96) [we're not assigning them any values, they'll be initialized with 0 values.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=174.54) [So the integers will get actual zeros here and then the](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=179.38) [strings will be assigned an empty string.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=182.85) [In fact, do you know what?](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=185.94) [Let's add some code to our main function here to see if that's actually true.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=186.91) [Now, before we run this right,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=192.04) [these are just a couple of Println functions like we've already](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=193.68) [seen, only this time within the parentheses,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=197.41) [but outside of the double quotes, we're referencing the variables we created.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=200.87) [Well,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=206.74) [if you're in VS Code, you can hit Run Without Debugging up here and it'll](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=207.23) [show you the results at the bottom of your screen, and you know what, feel](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=211.08) [free to do this with all of the examples if you're using VS Code. But if](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=214.44) [you're using something different,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=220.24) [just open a terminal in the directory where you create it and save the file](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=221.63) [and you will need to save the file, but then just do a go run like this. I've](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=225.91) [called mine vars.go, yours might be different.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=231.59) [But look, that's our program and actually,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=234.54) [yeah,](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=238.86) [it's not so easy to tell with the strings because](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=239.63) [they're just empty spaces here, but the ends are clearly initialized as zeros.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=241.87) [So 4 variables, each assigned a name, and of course, a location in](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=247.84) [memory, but importantly, initialized with 0 values, oh, and each one](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=252.34) [has got a type. And actually speaking of types, that's where we're going next.](https://app.pluralsight.com/course-player?clipId=d294eef8-1911-46a3-aa61-85ebabfe6a1c&startTime=258.32)

### [Variable Types and Type Conversion](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8)

[So we can check the types of variables and stuff in Go through something](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=1.54) [called runtime reflection using the reflect package.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=6.14) [So, let's go and import that up here, and then let's](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=9.74) [add these two statements down here.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=14.38) [Now, again, that just prints ln functions, only this time, we're calling](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=17.74) [the type of function from the reflect package as well.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=22.62) [So this one here is going to print the type of the name](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=27.34) [variable and then this one the type of module.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=30.28) [Well, tell you what, save that, and we'll give it a run.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=34.54) [Okay, name is of type string, and module is of type int.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=38.64) [But yeah, I mean we knew that, right? We called it out when we declared them.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=44.04) [Well, Go actually supports something called type inference, and it's](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=48.94) [pretty cool because it lets us write shorter code. So, if we change the](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=53.58) [variable declarations up here to be this,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=58.56) [see how we're no longer declaring the types,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=62.64) [but we are assigning values.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=65.5) [Well,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=69.24) [Go was clever enough to figure out that Nigel Poulton and](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=69.89) [Getting Started with Kubernetes are both strings and then](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=73.2) [that 4 and 2 are integers.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=76.09) [Although, you know what? I'm saying it's clever enough.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=78.84) [Let's save that and run it to see if it actually is. Magic.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=81.54) [Take a second to look at that.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=87.75) [Okay, now look,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=90.94) [I'm not a fan of initializing multiple variables on the same line.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=92.5) [I just think it's easier to read if we list them](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=96.14) [out on their own lines like this.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=98.68) [Now then, a couple of things. These double slashes here are how we](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=102.44) [add line comments in Go. Basically, anything we put after them are](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=106.59) [comments and get ignored by the compiler.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=111.61) [But the thing is, they're really important for good code,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=115.24) [especially if you work on a team where, of course, it's important for your](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=118.47) [teammates to understand the code that you've written. Well, as well though, Go](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=122.1) [is really picky about types. Like you can't add or subtract a string from an](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=127.83) [integer, and that's a bit of a daft example.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=133.99) [I mean,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=136.47) [how would you do that, right? It makes sense that you couldn't.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=136.79) [But, you can't even mix and match integers and floating point](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=140.04) [variables when you're doing math operations.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=144.35) [Well, look. To show this,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=148.14) [I'm going to change the variable declaration here so that module is](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=149.29) [actually a string. Now, I know it looks like an integer,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=153.12) [but because I'm putting it inside of double quotes,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=156.81) [I'm telling Go, no, store it as a string, and you'll see why in a second.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=159.4) [Well,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=166.24) [this code here will try and sum that module variable with the clip variable.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=166.67) [Let's give it a try.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=170.66) [Okay, no joy.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=174.24) [And that's expected, right?](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=175.76) [Even though module looks like an integer, it's a string, which](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=177.29) [is where conversion comes to the rescue.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=182.21) [So the strconv package has a bunch of functions that do type](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=185.84) [conversion, and we know the crack by now, right?](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=189.99) [We import the package up here.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=192.83) [We'll add this code here, which, yeah, okay.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=196.24) [It might look a bit complex if you're new to this.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=200.11) [But believe me, it's not. And you know what?](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=202.45) [It's a good way to get you thinking about some of the stuff that](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=204.69) [we're going to cover in more detail later.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=207.2) [Well, the main part of the code is here.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=210.84) [No, yeah, okay.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=213.11) [We've imported the strconv package, and we're using this Atoi](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=216.58) [function that converts ASCII strings to integers.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=221.09) [So, we're feeding it our module string, and it will return the equivalent](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=224.92) [integer. Only, it gives a return code as well, and we have to do something](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=230.97) [with that return code. So, we're saying this new iModule variable here,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=236.19) [assign that the integer that the Atoi function is going to return. And](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=243.43) [then this err variable, assign that the return code that the Atoi function](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=247.9) [will return.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=252.86) [Well, then we're doing some basic error checking and saying, so long as the](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=255.04) [return code's a 0, which indicates a success here, well,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=259.97) [as long as we get a success, do this calc here, and print the](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=263.75) [answer. And yeah, if it seems a lot, don't stress.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=267.25) [The point is, we're using a function from an external package to](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=272.74) [perform a type conversion, but as well, we're just handling it with](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=277.47) [some good old‑fashioned error checking. And we'll see a bunch more](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=282.13) [of this as we crack on, so it will all fall into place as we go](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=286.07) [through the course.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=289.44) [Well, I guess we should save that and check if it works, which it does.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=291.24) [So the Atoi function from the strconv package converted the ASCII](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=296.71) [string 4 to the integer 4. Then after that,](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=302) [the calculation worked because, obviously, we can](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=306.98) [sum two integers. Marvelous! Well, I've dropped this little gem in here, so I should probably explain it.](https://app.pluralsight.com/course-player?clipId=47b43643-d66b-431a-9e6f-d1cb835e3ce8&startTime=309.49)

### [Short Declaration](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75)

[Okay, quickly, back to variable declaration. So far,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=1.34) [we've only talked about variable declaration outside](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=5.11) [the functions at the package level.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=8.12) [And you know what,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=10.49) [declaring them here makes them available to all functions in the entire package,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=11.17) [so global in scope.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=16.23) [But if we declare them within a function,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=18.54) [which we'll do a lot, we can use a shorthand,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=22.12) [more concise method of declaration, like we did here.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=25.24) [Now, declaring variables within a function makes them local,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=29.44) [so only available inside the function they're declared in.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=32.96) [But that's okay for us because right now, we've only got one function.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=36.51) [Anyway, if we declare them here, like this,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=40.93) [then we can get rid of these up here.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=44.54) [Now, this might seem a small thing declaring them like this,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=47.74) [but it's the most common way, and you'll see it everywhere.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=50.69) [In fact, if we jump over to the kubernetes project on GitHub here,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=54.13) [Kubernetes is written in Go, yeah, well,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=57.81) [in this package, if we search for the var keyword,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=60.86) [so the long way of declaring, nothing, not a single occurrence.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=63.87) [But if we look for the short assignment statement,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=69) [there's loads, so this is the really common method.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=72.74) [But back here, like we just said, it only works inside of functions. Oh,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=76.95) [and obviously,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=81.54) [if you're not in a position to be assigning values to your variables yet,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=83.09) [you'll have to use the long form of the var keyword.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=87.02) [Well, I'll tell you what,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=91.04) [let's throw in another variable to indicate whether](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=91.86) [the viewer has completed the course.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=94.67) [So, this one is a true or false Boolean, and we're assigning it false for now.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=97.64) [Well, let's save that and try to run it, the emphasis being on try.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=105.04) [Okay, check that out,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=110.24) [no joy; Go is not liking the fact that we've](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=111.68) [created the courseComplete variable, but then not used it.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=115.07) [I guess, it doesn't like waste. Only there's a but, or a](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=119.14) [caveat, you can totally declare a variable at the package](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=123.57) [level up here and then never use it.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=128.08) [But if you're declaring it inside of a function,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=131.24) [sorry, guys, you've got to use it.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=134.01) [Well, I'll tell you what, for now, we'll comment that out and try it again.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=137.04) [Okay, it works this time.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=143.34) [So we've switched our variable declarations to inside the function,](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=145.4) [and we've adopted the short assignment method. Next up, we'll learn about pointer variables.](https://app.pluralsight.com/course-player?clipId=56db7803-d922-4fa2-8f6b-d2f6c075ad75&startTime=149.74)

### [Values and Pointers](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff)

[Right, come on, let's start with some jargon.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=1.24) [Go passes arguments by value, not by reference.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=5.04) [And what, pray tell, does that mean, Nigel?](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=10.24) [Well, on this line here,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=16.34) [we're passing the module variable to the Atoi function from the strconv package,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=17.34) [only we're not.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=22.35) [What actually happens behind the scenes is that Go makes a copy](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=25.34) [of Module and passes the copy to the function.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=29.52) [So real quick, when we create a variable,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=33.84) [Go allocates it in area and memory,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=36.83) [and let's say the address of that memory is 0xaa,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=39.49) [and it's the course variable with the value Getting Started with Kubernetes.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=42.63) [Well, when passing this as an argument to a function,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=47.84) [Go makes a copy of it and passes the copy to the function.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=50.8) [See how it's got a different address but the same value?](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=54.16) [Well, this means we've got two copies,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=58.24) [and I suppose we could say it is a form of immutability, like any changes](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=60.76) [made by the function only affect the copy at 0xbb. But this is only the](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=65.52) [default behavior, and we can absolutely change it.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=71.56) [Well, no,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=74.78) [actually, I should say, we can get around it, and we do that with pointers.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=75.57) [So, as an example,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=82.14) [if we add this line here to our code, by putting the ampersand here in](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=84.04) [front of the course variable, when we run the code, we'll see the memory](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=88.75) [address of the variable instead of its value.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=93.16) [So we'll give that a save, and let's try it.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=97.04) [All right, there we go, hex whatever, yeah? That's](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=100.94) [the memory address of the variable.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=103.51) [Anyway, point is, as the name suggests,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=106.64) [are special variables that point to other variables.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=108.65) [You know what, it's probably easiest if I just show you.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=112.54) [So, we've got an existing variable called course, and we know that it is](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=116.64) [storing the value Getting Started with Kubernetes.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=120.05) [It's a great course, by the way, and you should most definitely take it. Anyway,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=122.93) [this line here creates a new variable called ptr, and then the](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=127.44) [asterisk before the type makes it a pointer variable.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=131.69) [Then the &course bit on the end says point it to the](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=135.81) [memory address of the course variable.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=139.18) [So, if we add this line here, obviously, it's printing text and variables](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=142.24) [to the screen, but you know what, let's step through it.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=148.13) [We are printing this text here that says "Pointing to course variable](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=152.34) [at address," and then, of course, we want it to print the memory](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=156.55) [address. So we do that by printing the contents of the ptr variable,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=159.83) [which, remember, because it's a pointer variable, is the hex address](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=164.66) [of the course variable.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=169.1) [Stick with me. We're also then saying, which holds this value, and](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=172.24) [then we're wanting to print the value of the variable, so we do that](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=177.69) [by referencing the ptr variable again, only this time we actually](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=181.47) [dereference it with the asterisk. So this time it'll print the value](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=186.68) [stored in &course,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=191.33) [which should be Getting Started with Kubernetes. Oh,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=193.15) [tell you what, let's give it a try.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=197.21) [Oh, and remember to save your changes first.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=200.24) [All right, magic, that looks good.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=205.04) [However, I totally get it that it can be confusing, especially if it's new.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=208.04) [So, for the sake of those who are confused, and it's totally all](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=212.63) [right to be confused, most people are when they're first](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=216.66) [learning pointers, but winding it back,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=219.63) [variables are addresses in memory that store data.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=223.47) [Usually, we store a useful value, but pointer variables store](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=226.73) [the memory address of other variables, so they're useful for](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=232.81) [referencing other variables.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=236.94) [Well, as well as that, in code, the asterisk performs two functions.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=240.04) [First up, if we declare a variable with an asterisk like we did here,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=245.54) [then that asterisk makes it a pointer variable.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=250.82) [But when referencing pointers in code, slapping an asterisk in front of](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=254.84) [it dereferences it, which is jargon talk for, instead of telling us the](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=260.27) [memory address of the variable you're pointing to, actually tell us the](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=265.6) [value inside the variable you're pointing to. Woo, so this gives this,](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=269.53) [and this gives this.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=276.2) [Well, finally, putting an ampersand in front of any variable will](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=279.64) [return that variable's memory address. And you know what, we'll see it](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=284.32) [really soon, but pointers are really important when we start using functions, and actually, you know what, let's quickly set the scene for that.](https://app.pluralsight.com/course-player?clipId=a68dcd46-1060-4f45-9ee4-2004140838ff&startTime=289.34)

### [Passing by Value](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb)

[Right then, I think this code's getting a bit cluttered,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=1.24) [so just a quick bit of tidying up.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=3.6) [We're done with the reflect and strconv packages, so we'll lose them.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=6.08) [We are going to use the name and the course variables, but you know what?](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=11.84) [We'll lose the rest because remember,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=16.07) [Go doesn't like it when we create a variable inside of a function at](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=17.68) [least and we don't then use it. And you know what,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=20.95) [we'll just lose all of this here.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=25.14) [So, basically, we're at the bare bones of an app.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=27.33) [So, what we're going to do here is write a simple function to change the](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=32.44) [value of the course variable. And we'll get properly into functions in the](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=37.06) [next module, so don't overthink things right now, we're just looking at how](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=42.47) [variables get passed around. Anyway, in our main function here, let's print](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=46.53) [out the current value of course.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=51.63) [And look, I'm just adding a line feed character on the end here to](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=54.01) [make the output on the screen easier to read.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=56.85) [Well, next, let's call to a new function called updateCourse](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=60.54) [and we'll pass it the course variable.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=63.93) [Now then, we've just done this, Go doesn't actually](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=66.74) [pass the course variable, yeah,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=69.85) [it makes a copy and it passes that. And this is crucial.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=71.48) [So keep that front and center of your thinking.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=75.01) [Well, now to write the function.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=78.84) [So in the function signature here, and that's just this top line](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=81.74) [that we're writing, we give it an input, that's the course variable](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=84.83) [or copy, yeah. And we tell it the type, which is a string. And](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=87.98) [then, every function returns an output, which for us, that will be](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=92.89) [an unnamed string.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=96.75) [Basically, we're putting a string in and we're getting a string out. But](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=98.18) [remember, don't overthink this yet, we'll come properly to functions next.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=101.71) [Let's add some statements to do some work.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=106.34) [So this line here is assigning a new value to course.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=109.44) [This line's printing that to the screen,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=113.62) [then we're telling it to return the update to value, and](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=116.43) [that's our function. Take some input,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=119.34) [change it and print it to the screen, and return a value.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=121.61) [So I'll tell you what, back up here we'll add another Println to](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=126.34) [check whether or not the change we made in the function actually](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=129.38) [stuck. Now a couple of things.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=133.12) [First up, program flow is like this, we start at package main,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=135.42) [import a bunch of other packages, and jump into func main.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=139.64) [Remember from right at the beginning,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=144.24) [package main and func main are the programs entry point, so they're always the](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=145.78) [first things to run. Next, we initialize a couple of variables and print them](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=150.27) [to the screen. And it's all pretty orderly so far,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=155.48) [basically starting at the top and stepping through line](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=158.91) [after line; however, this call here to the function switches](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=161.27) [flow to the function down here.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=166.43) [We then step through that one line at a time, and when it's done,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=168.94) [flow returns back to here, and we run this line before the program exits.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=172.1) [Cool.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=177.34) [Well, the other thing I wanted to mention is this line](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=179.34) [here. Notice how we're just using the equals operator and](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=182) [not the full short assignment.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=186.66) [That's because we're not initializing a new variable, we're](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=188.39) [assigning a new value to an existing one.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=191.6) [Well, let's see if it runs.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=195.94) [Okay.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=200.54) [So, current course is Getting Started with Kubernetes,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=201.1) [we're changing it to Getting Started with Docker,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=203.96) [but then it's back to getting started with Kubernetes. But that's expected,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=206.16) [right? The updateCourse function made changes to a copy of the course variable,](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=211.53) [not the original.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=217.11) [In fact, the original was unchanged, and we can see that here.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=218.15) [Okay, fair enough.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=223.74) [But what if we actually want to change the original of the actual variable?](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=225.23) [I'll tell you what, no sweat. We're going to do that next.](https://app.pluralsight.com/course-player?clipId=245f7315-13e4-4d97-9435-fb1bd5142dfb&startTime=229.5)

### [Passing by Reference](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8)

[Okay, to get this updateCourse function to change the actual](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=1.44) [value of the course variable, so instead of a copy, we need a](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=5.45) [pointer, which, if you remember from before, is going to be a](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=9.5) [combo of ampersands and asterisks.](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=13.86) [So, we're initializing course here, and we're assigning it](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=18.24) [the value, Getting started with Kubernetes. Then, we're](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=21.78) [printing it to standard out.](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=24.91) [But then, as things stand,](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=27.14) [this call to updateCourse is creating a copy and working off of that.](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=28.88) [Well, if we lash an ampersand on the front here,](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=34.94) [instead of sticking a copy of the value,](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=38.54) [Getting started with Kubernetes onto the stack,](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=41) [it'll stick a pointer or a reference to its location in memory.](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=43.71) [Well then, down here in the function, we stick an asterisks here](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=49.14) [before the string. Remember, doing that makes the variable a](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=53.33) [pointer. Then we need to stick an asterisks here when we're changing](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=57.82) [it to Getting Started with Docker.](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=63.19) [So what this is doing is telling Go to assign Getting Started with](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=65.74) [Docker, which is another epic course, by the way, but assign this](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=69.87) [new value to the location in memory that the course point of](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=74.28) [variable is referencing. Oh, yeah, another one here, if we don't,](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=78.98) [we'll get the hex memory address again, and another one down here on the](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=86.2) [return, and then I reckon, yeah I reckon that should work.](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=91.83) [So this time around, we should actually change the original value.](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=97.94) [Well, moment of truth I guess. Like we ever doubted it.](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=103.54) [So, current course is Getting started with Kubernetes. We're updating it](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=113.66) [to Getting Started with Docker, and the changes stuck. You know what? That](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=117.82) [folks, is pointers and passing by reference, which can be very useful. Well, next on the cards, constants.](https://app.pluralsight.com/course-player?clipId=979fc5d3-1a03-4b1d-a66d-8640f4a525b8&startTime=123.41)

### [Constants](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f)

[So this will be nice and short because constants are pretty](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=1.54) [much the same as variables, except, I guess, for the](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=4.93) [obvious. Constants are immutable,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=7.94) [so they never change, whereas variables, well, these](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=9.96) [most definitely can and do change.](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=13.08) [Anyway,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=17.04) [I'm cleaning this up so that we've got pretty much a new program, and](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=17.59) [I'm going to say that as consts.go. Now then, if your package imports](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=21.9) [and things go away like mine just did there,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=29.85) [it's probably because you're using VS Code with the Go](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=32.64) [plugins and you've imported a package that you're not](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=35.31) [actually using in your code yet.](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=38.29) [Well, we actually are going to use func, so I'll put it back.](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=40.94) [Okay,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=45.44) [well, constants are declared in Go with the const keyword, and it's pretty much](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=45.96) [the same as the long form variable definitions that we saw earlier. Though](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=50.67) [actually, this is the only way to declare a constant,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=55.36) [so there's no shorthand way like we have inside of functions for variables.](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=58.91) [Well anyway, look,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=63.84) [this one is defining probably the world's most famous constant,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=65.07) [the speed of light, which if you didn't know,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=69.06) [travels at pretty much 186,000 miles per second, or I guess 300,000](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=71.04) [kilometers a second, if you prefer kilometers. Kilometers,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=77.69) [kilometers, I don't know.](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=81.49) [Well look, this will print it to the screen, and I think we'll give](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=83.4) [that a save and run. And we're good. Simple stuff, yeah? Now look,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=87.32) [really quick, just to prove that constants cannot be mutated, this](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=95) [line is trying to change it. So speed it up a tad, yeah, actually to](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=100.07) [Warp 9 in miles per hour. And I know I'm mixing miles per hour and](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=104.42) [miles per second, but you know what, who cares?](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=111.85) [But I'll tell you what, let's try it. Now, no joy.](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=116.74) [As we said before, constants can't be changed.](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=122.09) [Okay, well,](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=126.04) [you know what? Time for a quick look at how we interact with environment variables on your system before we wrap the module with a recap.](https://app.pluralsight.com/course-player?clipId=571f25c3-e47b-4a5d-9bb3-936aa7c2361f&startTime=126.88)

### [Accessing Environment Variables](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37)

[Right.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=1.64) [A super quick look at environment variables with Go.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=2.18) [Now first up, we're in a new program file here and well,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=6.24) [environment variables are well,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=10.6) [they're variables here or they're settings on your host system.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=14.05) [So for me right now,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=18.37) [that's my office Mac because that's where I'm running the demos from.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=19.96) [For you,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=23.11) [it'll be whatever machine you're following along on and it's](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=23.74) [pretty much the same whether it's Linux,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=26.44) [Windows, or Mac.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=28.63) [Now to get access to environment variables,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=30.94) [we're going to leverage functions from the OS package.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=33.44) [Actually, it gives access to a ton of OS stuff,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=36.85) [so not just environment variables, but we know the deal by now.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=39.58) [It gets imported up here.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=43.4) [Anyway,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=46.34) [the environment function here lists all our environment variables on our system,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=47.15) [so let's save that and see.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=52.36) [Now this time the file is called env.go.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=55.14) [Okay, well here, it's a proper mess,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=58.84) [but it is a list of all of the environment variables](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=62.57) [on your system as key value pairs, but you know what,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=66.12) [I'll tell you what a simple loop here like this that'll iterate through](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=69.79) [them and then put each one on its own line and look,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=74.09) [we'll get into loops later so don't worry about it.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=77.54) [It's just going to go through that list of environment](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=80.18) [variables with each one on its own line.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=82.48) [So we'll give that a save and go again.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=85.64) [Oh, miles better.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=90.14) [But I'll tell you what,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=93.34) [let's jump back to the variables program that we've been working on.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=94.36) [Mine was saved as vars.go,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=98.64) [but there is a fresh copy in the variables folder on the](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=100.95) [GitHub repo called envvars.go and you can work from that if](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=104.19) [you didn't save it from earlier.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=108.33) [Anyway, look, first things first, we've got to import the OS package.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=110.44) [Then down here, let's change the name variable so that,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=115.14) [instead of assigning it this string literal here where it's my](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=118.99) [name with a capital letter at the beginning of each of my I](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=122.44) [guess first name and last name, instead of that,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=125.48) [let's assign it the value of the currently logged on user.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=128.23) [Well, to do that, it'll be os.Getenv.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=133.64) [Oh, actually before we do that, see all of the options that we've got here,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=137.66) [so env, pagesize, pid, a bunch of host system stuff,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=143.14) [yeah.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=148.07) [Well we want Getenv, and then because I'm on a Mac, I want USER.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=148.68) [On Windows, it's USERNAME.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=153.28) [But do you know what?](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=155.8) [It's all simple stuff,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=156.93) [it's just an environment variable on your system holding](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=158.66) [the name of the currently logged on user.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=161.65) [Oh, and of course we're assigning the output to the name variable here.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=164.94) [Well, I'll tell you what, give that a save,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=169.34) [and then in the terminal here, well, you know what,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=172.04) [first up, I'm going to show you the value of the USER environment variable.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=174.51) [Okay, nigelpoulton with no capitals or spaces.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=179.44) [Well, let's see if the program pulls that and it did,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=184.24) [nigelpoulton like that instead of with capitals and a space](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=189.69) [because it grabbed the value from the environment.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=193.29) [Spot on.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=196.86) [Well, of course, we're only just getting our feet wet,](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=199.04) [totally feel free to have yourself a play around with some of the](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=201.84) [other stuff available through the OS package.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=205.45) [But do you know what, as far as this set of lessons on variables goes, we're done and it's time for a quick recap.](https://app.pluralsight.com/course-player?clipId=9aa4fd81-761e-421a-b723-8a7a1c6b5b37&startTime=208.44)

### [Recap](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e)

[Okay, I feel like we've covered an absolute ton,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=1.54) [so I'll keep this as sharp as I can.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=4.49) [We learned that we can declare variables with the var keyword, and we can](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=8.14) [even do that without initializing them with a value.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=12.91) [But, if we do that, Go automatically assigns them a default value.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=16.84) [We saw that strings defaulted to an empty string,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=21.24) [integers to a 0, and I think we saw that Booleans get a](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=24.41) [false. Well, we said that variables declared at the](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=27.52) [package level are global in scope, so, available to all functions.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=31.78) [And also though, when declaring at the package](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=37.54) [level, so outside of any function, the only way to do it is with the var keyword.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=40.19) [However,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=46.44) [if we're inside of a function, then the more idiomatic way is](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=47.74) [with the short form operator, and to be fair,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=52.09) [that's the way we'll do it most of the time.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=54.83) [We also saw that Go supports type inference.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=58.14) [So, if we do initialize a variable with a value,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=61.81) [well, Go's clever enough to figure out what type it should be.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=65.27) [We also said that Go passes variables to functions](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=69.74) [by value and not by reference.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=73.22) [So, instead of passing the actual variable itself, Go creates a copy with the](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=75.63) [exact same value as the original, and it passes the copy to the function. But](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=81.36) [do you know what? That's not always what we want, so we looked at pointer](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=87.27) [variables, and, these are how we can force a function to work on the original](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=91.15) [variable instead of a copy, and we call that passing by reference this time,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=96.98) [rather than by value.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=101.58) [Then we switch to constants,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=104.14) [which, honestly, are pretty much the immutable sibling of variables.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=105.67) [They look, smell, and feel a lot like variables,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=110.35) [only once you create a constant, you can't change it.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=114.06) [Well,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=119.04) [then we wrapped the module looking at how Go can access](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=119.82) [environment variables from a host system and use those values in](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=123.05) [programs. And that was about it, though,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=127.59) [of course,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=131.06) [we've only covered the basics, and you should feel free](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=131.8) [to take some time to play around,](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=134.49) [make a bunch of mistakes yeah, and generally try things out. Well, next on the card, and this is a good one, functions.](https://app.pluralsight.com/course-player?clipId=b1af4875-f1fd-4595-ae92-19c53c619d8e&startTime=137.23)

## [Working with Functions](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f)

[Right then, functions.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=1.54) [And if you've been following along, I've kind of bigged them up already,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=3.31) [and rightly so, because they are a big deal.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=6.95) [As well though,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=10.89) [if you're new to coding and maybe it's your first encounter with functions,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=12.73) [I recommend you buckle up, because this is gonna be,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=17.46) [yeah, one heck of a ride.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=20.57) [So we'll start the ball rolling with why we have functions,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=23.94) [kind of set the scene.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=27.33) [Then we'll get into a bit of detail.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=28.39) [We'll start out at looking at how we define functions in Go,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=30.31) [so, like, what does the syntax look like?](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=34.06) [Then we'll get our hands on,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=36.84) [and we'll probably start out with func main that we've said every program has,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=38.7) [but we'll also create our own. And then we'll round things](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=42.76) [out with some slightly more advanced stuff.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=47.06) [Now, I don't mean anything hard here.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=50.04) [We're a fundamentals course, yeah,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=52.54) [so we're not going to be rocking it with things like callback functions and](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=54.21) [closures. But we might give them a mention later in the course.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=57.62) [The thing is, right,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=61.64) [the aim of this module is to get you comfortable with the basics of](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=63.09) [functions and able to start writing your own. And then this will obviously](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=67.3) [have you set up for the more advanced stuff later.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=71.63) [Now, one last thing, functions.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=74.84) [They are a massive topic, and there's no way we can do them proper](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=77.94) [justice in a course like this. But don't worry,](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=82.88) [we will be giving you more than enough to get started. So, come on, let's do this.](https://app.pluralsight.com/course-player?clipId=1334adf8-4605-4595-ad5c-7471f549482f&startTime=85.35)

### [Why Functions](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435)

[Okay.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=1.64) [So why do we have functions? Basically, they let us](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=2.18) [write clean, tidy, reusable code.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=6.55) [So let's say that this is 150 lines of code here.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=9.94) [I mean obviously, I know it's not, but you get where I'm going.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=13.19) [Well anyway, let's say that at three separate points were doing,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=16.84) [I don't know,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=20.54) [let's say some text conversion, maybe converting to title case. And](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=20.93) [we'll assume that the code to do that is 10 lines long.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=25.12) [Again,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=28.07) [I know it's not showing us 10 lines on the screen, but you](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=28.56) [get my point, we're repeating the same code over and over](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=30.72) [again in different places.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=34.17) [Now, not only is that wasteful, it's also prone to error.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=37.34) [A much better option is to just write it out once and then every time](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=41.74) [we need it we make a call to it or a reference.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=46.49) [Well, that's the essence of functions. Take a block of](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=50.54) [code that does a job, in our case, convert some text into title case,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=53.75) [write it out once as a function, and then every time we need](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=57.94) [it, we call out to it. And you know what,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=61.08) [as simple as that, we go from writing out 10 lines of code 3](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=65.54) [times, to writing it and testing it only once. And look, again,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=68.61) [it's a totally naughty example, but it does demo the point.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=76.33) [Functions exist to let us reduce,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=80.1) [not only the number of lines of code we write but also the amount of testing](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=82.44) [we do and the amount of mistakes we'd potentially make. And you know what, it](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=86.32) [is slap bang right at the heart of coding in Go. And you'll see this all the](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=91.89) [time as you progress, but Go programs,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=96.96) [they're basically a collection of smaller modular functions.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=99.85) [Okay, well you know what, that's the mega high level,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=105.14) [let's add a little bit of detail.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=107.65) [Architecturally speaking, we'd give a function an input, the function](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=110.74) [does something with that, and it returns an output.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=115.18) [So for our title case example,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=119.24) [we'd have a function, and maybe we'd call it title case, but you](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=121.06) [know what, you can call it pretty much anything you want, but we](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=124.04) [pass it a text string, the function performs its magic, and it](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=127.98) [returns the same text as an output.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=131.79) [Only this time, formatted as title case. And actually,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=134.04) [yes,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=137.86) [that is proper title case propositions like with don't actually get](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=138.34) [converted. But you get the point right? It takes an input,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=142.15) [performs a task,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=145.92) [returns an output. And you know what, that kind of job,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=146.93) [so changing text into title case,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=152.01) [it's exactly the kind of thing that functions are designed for.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=154.33) [So performing a discreet operation that can be reused throughout the program.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=157.33) [So instead of writing out the same block of code,](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=162.09) [time and time again with the risk of typos and all the other mistakes](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=164.67) [we can make, just write it out once as a function, and any time you](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=168.19) [need it, make a call to it. Sweet. Well, coming up next, we'll have a look at how to write functions.](https://app.pluralsight.com/course-player?clipId=8b4fa83c-555d-473e-a82d-bee553268435&startTime=172.55)

### [Function Syntax](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e)

[Right, we start defining functions with the func keyword, followed by](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=1.44) [whatever we want to call it and a set of parens. And if it looks familiar,](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=5.82) [it's because we've already seen it with the main function. However though,](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=11.66) [any code we want the function to execute,](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=15.61) [which in the examples we've been citing is converting text to](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=18.53) [titlecase, well, that goes inside a pair of curly braces.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=22) [Okay,](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=27.74) [well that's kind of the basic structure, but we're not done](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=28.64) [yet. In the little picture we just saw, we gave the function an](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=30.74) [input, and it turned out an output.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=34.6) [So, the parameters we passed to a function go here, and](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=38.64) [that's what we actually call them, parameters.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=42.54) [So, I guess that means somewhere in our code we have got a](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=45.84) [variable holding the text we're going to convert.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=49.12) [And then, when we make the call to the function, we pass that variable](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=52.94) [as an argument. Here, we're calling our function as part of a Println](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=56.71) [statement, and then the argument that we're passing gets represented](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=61.05) [here in the function signature,](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=65.51) [which actually is what we call this line here as well, the function signature.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=67.53) [Anyway, look, starting at the top,](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=73.34) [we've got a string variable holding the value containers on aws wavelength.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=75.64) [Then, we call to the Println function from the fmt package, which](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=81.04) [in turn makes a call to our function called titlecase. And as part](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=85.43) [of that, it passes it the text variable, or actually, it passes a](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=90.11) [copier. Remember, Go passes arguments to functions by value rather](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=94.36) [than by reference.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=99.14) [Okay, well next up is our actual titlecase function.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=102.14) [So we've declared it with the func keyword, and we've given](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=105.24) [it a meaningful name. Then, inside the parens, we'll list](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=107.91) [any parameters passed to it.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=112.23) [So, we passed it the text variable, which is a string.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=115.14) [Now,](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=119.74) [it's possible to pass multiple parameters to functions, and we'll see](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=120.74) [all of that later as we crack on with the course.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=124.6) [But, however many we pass, we have to tell the function what type they](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=127.24) [are, so for us, a string. Well, then we tell it the type of the value](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=131.6) [we'll return. That will be a string as well.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=137.87) [Remember, we're feeding in a string, we're going to convert](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=140.03) [it to titlecase, and spit out a string.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=142.97) [Well,](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=146.74) [I guess that takes us to the body of the function, which contains the](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=147.14) [code to convert the contents of the text variable into titlecase. And](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=150.52) [then last, but not least, we use the return keyword to end the function](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=156.07) [and return the converted value back to the caller, so back to Println.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=160.48) [I'll tell you what, well, was there anything else?](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=165.72) [Oh yeah, do you know what?](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=171.8) [Be careful where you put these curly braces.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=173.02) [So they've got to go exactly where I'm showing here.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=175.49) [Like, I don't know if you put the opening curly on its own line.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=179.14) [Believe me, the compiler is going to introduce you to the pain function.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=183.41) [Never mind.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=190.34) [Oh, do you know what?](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=191.5) [Actually, we can also return multiple values to the caller,](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=192.34) [but to do that, we need to enclose them in a set of parens as well.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=196.22) [Oh gosh, you know what?](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=201.24) [We can name returns and do loads more.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=202.36) [We are literally only scratching the surface.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=205.28) [But do you know what? I do think for us right now, that is enough on the theory. Let's go and get our hands on.](https://app.pluralsight.com/course-player?clipId=3f561603-5d7a-4bcd-9689-ef91e35b906e&startTime=208.94)

### [Writing Your Own Functions](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f)

[So, let's ease ourselves into this.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=1.54) [So we've got a shiny new program and we're](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=4.2) [importing the fmt and strings packages.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=6.45) [And you know what, as we're learning functions, let's write out our own main.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=8.67) [So it's the func keyword, obviously, it's called main,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=13.64) [and it gets a set of parens.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=16.95) [Now the main function's a bit special, yeah.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=20.04) [I mean we already know it's the program's entry point,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=22.07) [so it gets called automagically.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=24.82) [But for us right now, it doesn't even accept any inputs,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=26.88) [and it doesn't return anything either, so its signature's super simple.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=31.24) [Obviously, though, we do give it some curly braces,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=37.14) [and then the code we write goes in here.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=40.08) [Well, for this demo, and look, it is just a demo,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=43.94) [remember,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=46.91) [but let's implement a quick mod of the title case](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=47.63) [example we just hashed out on the slides.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=50.32) [So we'll declare and initialize a couple of variables here.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=53.1) [Feel free to use your own, of course, but let's print them to the screen.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=57.19) [Only, let's do that after we've modified them through a function.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=62.25) [So, we're calling the name of the function converter,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=68.44) [and we're passing it the two variables.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=71.53) [And, yeah, we're calling it from within Println.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=73.68) [Magic.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=78.84) [Well, if we move down here, and notice, we're outside the main function.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=79.44) [Well, this is where we'll define ours.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=83.73) [Okay,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=87.84) [so we've got to call it converter because that's what we called it up there.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=88.18) [It's receiving two strings, and it is returning two.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=92.34) [Now,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=98.96) [you might find that it's a good idea to write your](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=100.58) [return line now so you don't forget it.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=102.78) [Okay, well, to actually do the converting,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=106.64) [I'm going to use a couple of functions from the strings package.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=108.86) [So, we're converting the author to uppercase and the course to title case.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=112.76) [And you know what, let's go from the top.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=119.07) [We're declaring and initializing two string variables,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=121.33) [but importantly for us, both of them are holding lowercase values.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=124.32) [Well, we're passing them to our custom function called converter,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=131.84) [and then we've rigged things up in the function signature to receive them,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=136.08) [and we're also going to return two strings,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=140.19) [so input and output.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=142.67) [Then, of course, within the function, we actually do the conversions.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=145.44) [Well, let's see if it works.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=148.71) [Magic, uppercase author and title case course.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=154.34) [Now, a couple of quick things.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=159.84) [When calling the function, back here,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=162.04) [we're passing it the author and course variables.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=163.85) [But, down here in the signature,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=166.23) [we can totally assign them different names if we want.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=168.58) [So, I don't know, let's say, well, s1 and s2.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=171.73) [Now, look, because of the order we pass them in, author](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=177.44) [will be assigned to s1 and course to s2.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=181.05) [And remember, we pass by value rather than by reference. So we're](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=184.34) [not passing the variables themselves; we're passing the values in](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=189.04) [them, and then, of course, we're assigning them to two new](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=192.59) [variables with these names.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=194.94) [So you know what, even if we had kept the same names,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=196.39) [they wouldn't be the same variables; they're copies. Anyway, if we do this, we](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=199.37) [need to change these lines here and the return. And, okay,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=204.2) [go on, you know what, seeing as how I'm feeling adventurous,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=214.61) [let's give the returns different names as well.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=217.58) [Now, look, honestly, in no way am I saying you should do it like this.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=222.74) [I mean, to me, it is messy and a recipe for confusion.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=228.93) [And as well, your teammates are probably going to hate you for this.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=232.47) [I mean, look, we're passing them in this author and course,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=235.64) [referencing them inside the function as s1 and s2 and returning them as](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=238.3) [str1 in str2. I mean, it's horrible and it's unreadable,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=242.83) [but it can be done, or, you know what, at least I'm saying it can.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=247.84) [Let's give it another try.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=251.82) [Yeah, okay, it does work, but like I say, I'm not saying it's recommended.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=256.14) [Well look, up next,](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=261.44) [we're going to say how to write functions when we have no idea how many arguments are getting passed.](https://app.pluralsight.com/course-player?clipId=9de5000f-c17c-4502-b5f1-edf9e5f4250f&startTime=262.54)

### [Variadic Functions](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb)

[Quick question.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=1.74) [What do you do when you don't know the exact number of arguments that will be](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=3.04) [passed to a function? Answer, create a variadic function.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=6.61) [Sorry, what's that, Nigel? A vari‑what‑ic function? Yeah,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=13.04) [I know, a proper buzzword, but the clue is in the vari bit of the](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=17.77) [name. Variadic functions can be called with a varying number of](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=21.76) [trailing arguments, and you know what, if that still sounds](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=26.79) [complicated, it really isn't.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=30.1) [So, look, we've got some code here.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=31.94) [It's actually in the functions folder of the](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=34.38) [course's GitHub repo called vary.go.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=36.38) [Anyway, we've got a package declaration,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=39.94) [an import statement, and a main function.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=41.94) [Now what we're going to do is create our own function to calculate Max](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=44.64) [Verstappen's highest Formula One championship finish since 2015.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=48.4) [So, to do that, we'll create a new variable here, and assign it the](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=55.04) [return value of a function that we're about to write. And, look, if](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=60.02) [you think about it, that's pretty cool, yeah? Using a function to](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=64.13) [declare a variable. Anyway,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=67.82) [we'll pass that Max's six previous championship results, and we're doing it as](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=69.83) [six ints, and that's so we can show how functions can deal with an unknown](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=74.47) [number of input parameters, because, I guess next year it'll be seven, yeah?](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=79.85) [Well, then a call to Println to print his best finish. Now, for a custom](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=85.74) [function. So, it's func and then the name, and then for us the crucial part](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=93.13) [when telling it what to expect as input parameters, we put ellipses before the](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=99.57) [type, and it's these ellipses that tell the function to expect any number of](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=105) [ints.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=111.88) [Okay,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=114.14) [well, we'll also return an int that's going to be](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=114.72) [his highest finish, but you know,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=117.78) [I'm getting ahead of myself. For now, the way this works under the](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=119.49) [hood is that the input parameters get stored as a slice of ints.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=123.36) [Now, I do know we haven't covered slices yet, so, just think of them](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=130.4) [as a list of ints, so, for us,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=135.61) [a list of the six championship finishes that we're passing in,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=138.02) [and then, because it's a list, we'll use this loop to range over](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=142.24) [it and determine the best finish.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=148.01) [And again, I know, we haven't covered loops yet either,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=150.04) [but honestly, the detail doesn't matter, it's just some code](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=154.85) [that reads the values in the list or the slice here, and it](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=159.46) [figures out which value is the lowest.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=162.59) [So, just for clarity, we create the best variable here, best, it's](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=166.64) [called best, yeah, I'm not saying it is the best. Anyway,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=172.76) [we create that, and we give it the value of the first item in the list.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=175.88) [Remember, computers start counting at 0. Then, we pull in](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=181.05) [every other value one at a time, and if it's lower than](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=187.23) [the value currently in best, we update it. And, you know what?](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=190.59) [Honestly,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=195.58) [if loops and the likes are a bit new to you, no sweat, we'll](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=196.08) [cover all of this in more detail later. Anyway, after we're](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=198.91) [done ranging over the slice,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=203.56) [we return the value in best to the caller that assigns it to our best finish](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=204.91) [variable back here in main, and we print it to the screen.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=210.12) [Shall we try it?](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=215.14) [Oh, look at that.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=220.74) [Best championship finish at the time of recording](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=222.64) [is third place. Though honestly,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=225.19) [right now he's looking like this season he'll either win it or come](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=227.85) [second, but the thing is, for us in this course,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=230.78) [I reckon that's enough on functions. Though, honestly,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=234.07) [like I've been saying,](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=236.96) [we are only tickling the surface, there's so much more.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=237.83) [But for us, as a fundamentals course, that is more than enough to keep us going. Let's do a quick recap.](https://app.pluralsight.com/course-player?clipId=5350c503-63c1-4679-b202-5e2420f1bafb&startTime=241.34)

### [Recap](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432)

[All right, that was a quick tour of functions in Go,](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=1.24) [the emphasis being on quick.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=4.71) [But, I need to stress a couple of important things.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=6.91) [First up,](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=10.04) [you really do need to understand the basics of functions because they are](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=10.85) [absolutely at the center of everything you're going to do in Go.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=14.37) [So, if you've completed the module and you're still a bit unsure, I](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=18.08) [reckon you need to have a play around, maybe even watch the module](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=24.08) [again because it really is important you're comfortable with them. As](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=26.93) [well though, we've honestly only scraped the surface, there's like](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=31.53) [higher‑order functions, closures, user‑defined functions,](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=35.42) [loads more.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=38.45) [But it's all good, they all build on the](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=40.44) [fundamentals you've learned here. Anyway,](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=42.76) [functions are brilliant for writing modular reusable code.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=46.1) [And we said that all Go programs are lots of small functions,](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=49.94) [each one doing a specific job but working together to create a useful program.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=54.42) [We also looked at the syntax of writing functions.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=60.94) [They start with the func keyword,](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=63.91) [they need a name, and we list the input parameters and outputs](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=65.66) [inside a pair of parentheses, including types.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=69.69) [Then, the actual function code itself goes inside a set of curly braces.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=73.64) [We also looked at variadic functions that use ellipses to](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=78.34) [handle situations where we don't know exactly how many input](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=81.9) [parameters are going to be passed.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=85.39) [Okay, sweet.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=88.44) [Well. if you think you understand most of that, and I hope you](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=89.34) [followed along with the examples, then up next, we're getting into conditionals, so ifs, whats, and buts, kind of.](https://app.pluralsight.com/course-player?clipId=7c882283-2dec-49b1-b1ab-f5fb217bb432&startTime=93.3)

## [Working with Conditionals](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110)

[Right, then. Working with Conditionals. And I get it](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=1.24) [if for some of you, that's jargon.](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=4.53) [However,](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=6.19) [if you've programmed in other languages, I know](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=6.79) [you'll know what we're talking about.](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=9.28) [It's like if/else, else/if, all of that stuff.](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=10.67) [However, if you're brand spanking new to programming,](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=14.74) [it's about making decisions in code.](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=18.38) [So, for example, maybe your program asked the user for input.](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=21.04) [Maybe, I don't know, what their age is. And then your program does](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=24.87) [one thing if the user is under 18 years old,](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=28.77) [but it does another if they're 18 or over.](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=32.09) [Well, conditionals,](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=36.04) [like we'll see, let your programs do exactly that. If the user's](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=37.25) [over 18, branch out onto this set of code,](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=41.05) [if they're younger, branch out somewhere else.](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=43.62) [Well,](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=47.04) [for us, we're going to be learning if and switch in Go. We'll start](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=47.64) [out by looking at the basic syntax of if statements,](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=52.43) [then we'll get our hands on with some basics, and we'll dig a bit deeper. Once](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=55.24) [we're good with if, we'll look at switch case. And again,](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=60.15) [we'll cover the syntax, get our hands on, and dig a bit deeper.](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=63.6) [Finally,](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=67.94) [we'll round out the module with how to use if to handle](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=68.67) [errors, and then we'll do a recap. Well, let's get rocking and rolling!](https://app.pluralsight.com/course-player?clipId=6516a078-4cb2-4b40-80f0-cf89fa972110&startTime=72.15)

### ["If" Syntax](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250)

[The humble, but immensely powerful if.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=1.44) [And let's just bust some jargon here.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=8.04) [If statements let those evaluate conditions,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=10.41) [and based on whether or not that condition is true or false,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=14.13) [branch out and execute specific code.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=17.75) [So, yeah, like we just said a second ago actually,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=22.34) [if a person is 18 or over, execute, whatever,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=24.79) [code A, yeah, whereas if they're younger,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=28.1) [execute code B.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=30.59) [Now then, on the jargon front, we are evaluating conditions,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=32.15) [and that's where the term conditionals comes from.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=38.67) [As well though, you might hear the term branching.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=42.04) [Well, like I just said, if a user is 18 or over,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=45.32) [branch out onto the orange code here on the slide.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=48.88) [If they're younger, we branch to the blue.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=51.54) [The if statement evaluates true or false conditions.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=55.94) [And if you like your jargon, I'm talking Boolean true or false,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=60.84) [so that means using Boolean operators.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=66.75) [So we're not talking just a string here with the words true or false in it or,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=70.15) [I don't know, even an integer of 0 or 1 representing true or false.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=74.63) [No, remember, Go is really strict when it comes to types,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=78.71) [so we'll be dealing with a Boolean true and Boolean false,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=82.81) [and, well, we'll see it in a second, so don't stress.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=87.13) [Well, for the syntax, first up, we've got the if keyword.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=91.64) [And yes, it is case sensitive,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=95.47) [so always lowercase. Well, then after that comes the expression to evaluate,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=97.42) [and like we said, this can be anything that evaluates to either true or false.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=103.45) [So like we said, again, is a user 18 years or older?](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=107.54) [If they are, do this; if they're not, do that.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=112.06) [Other examples.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=115.84) [Oh, is free disk space more than 10 GB?](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=117.71) [If it is, carry on; if it's not, throw an error.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=121.12) [Hopefully you get the picture.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=125.05) [Then we need a set of curlies, and like I just said with functions,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=127.84) [they've got to go here; do not be going and putting](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=131.79) [this first one on its own line.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=135.69) [If you do, you'll just make the compiler grumpy.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=137.74) [And actually, on that note, now, this doesn't really matter,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=141.35) [so don't let it mess with your mind or anything,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=144.67) [but the reason behind all the strictness with the placement of curlies,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=147.01) [well,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=152.54) [you know how in some of the languages, they make you](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=153.54) [terminate lines with semicolons?](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=155.71) [Well, guess what?](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=158.01) [Go actually does as well,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=159.07) [only Go is really nice and the compiler does the semicolons for us.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=160.8) [So, if you throw your opening curly on its own line by accident,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=165.94) [the compiler is going to throw a semicolon up here and](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=170.42) [it's going to break the program.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=173.16) [But who cares, right?](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=174.81) [I don't know why I even said that.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=176.76) [Anyway, look, we put the code that we want to execute inside the curlies,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=179.54) [and that's a basic if,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=183.94) [that block of code will only execute if the expression here evaluates to true.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=185.24) [But you can do else if and else branches,](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=191.7) [and we're about to see it all in action.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=195.45) [But that is the basics of if syntax in Go.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=197.4) [Whew! Let's go see it in practice.](https://app.pluralsight.com/course-player?clipId=4dc65c04-4514-443f-812e-4829aa694250&startTime=200.84)

### ["If" in Practice](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7)

[Alright, we've got this framework of a program with a couple](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=1.54) [of variables declared and initialized.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=4.36) [Actually,](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=7.27) [the code I'm using is in the conditionals folder of the course's GitHub](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=7.87) [repo, and it's called if.go. I know, really creative.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=11.1) [Anyway,](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=17.04) [this first variable here is showing the length of my Docker Deep](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=17.9) [Dive course in minutes, and then this one is showing the same for](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=21.89) [my Containers on AWS Wavelength course.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=26.39) [So, Docker Deep Dive's, like, I don't know, what is](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=28.92) [that, about 4 and a bit hours long,](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=32.1) [but Containers on AWS Wavelength, only half an](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=34.03) [hour, so entirely different beasts.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=36.98) [Okay, oh actually, remember, Go can infer types, so it](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=40.44) [knows both of these are integers.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=44.55) [Anyway, let's throw in a simple if block.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=47.54) [Alright, magic.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=51.14) [Like we just saw, we start with the if keyword, and we're evaluating](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=51.98) [whether Docker Deep Dive length in minutes is greater than Containers on](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=56.08) [AWS Wavelength length in minutes. Wavelength, length. Anyhow, like I said](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=60.64) [before, we use Boolean operators like these I'm showing on the screen to](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=67.55) [do the actual evaluations.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=71.12) [Well obviously, this time we're asking is the length of Docker](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=74.04) [Deep Dive greater than the length of Containers on AWS](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=77) [Wavelength? If it is, execute this code here.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=79.96) [Okay, well, magic.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=85.84) [But what if it's not, or actually, what if they're the same?](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=87.55) [Well, this else if here uses the equals operator in case they are](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=91.94) [the same, and we use double equals here so that Go doesn't think](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=96.4) [we're assigning a variable a value.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=99.68) [Anyway,](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=102.34) [it executes this line here, and then we've gotten else on its](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=102.86) [own that basically says if none of the above are true, run this](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=106.95) [code here. Now, it's a no brainer.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=110.43) [I get it. Docker Deep Dive's obviously longer. But now I'm seeing it in](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=113.76) [action, and remember, save any changes you've made.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=117.47) [Okay, mine's called if.go, and there we go. Docker Deep Dive is](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=121.94) [longer than Containers on AWS Wavelength.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=127.23) [Okay, I'll tell you what. Let's jump back here, and we'll have a step through.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=131.04) [We've got the usual blah, blah, blah at the top that I'm hoping we know by](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=135.94) [now. We're declaring a couple of integers as variables here,](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=139.74) [one's way bigger than the other, and then we're running this line that says,](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=144.21) [hey, if Docker Deep Dive's longer than Containers on AWS](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=148.11) [Wavelength, which it is, run this code here.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=151.45) [And because that expression was true and its code executed,](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=155.84) [program flow then skips right to the end here, and it doesn't](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=160.08) [even bother checking any of the rest.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=163.52) [And then because we've got no more code in the program,](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=167.04) [the program exits with a 0 return code, indicating a success.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=169.92) [Fabuloso! But, what if Docker Deep Dive wasn't longer, like](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=175.44) [what if the variables were like this?](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=180.33) [Well, looking at the code, it's pretty simple.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=184.34) [We come into the if block, is Docker Deep Dive length in](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=186.77) [minutes more than Containers on AWS Wavelength? That will be](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=190.15) [a negative this time, Houston.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=193.69) [So, we skip here.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=195.32) [Are they the same? No, that's also a negative.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=197.24) [So, we skip to this catch‑all else statement, and we execute this block of code.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=200.51) [Well, I'm saying that's what will happen.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=207.54) [Let's give it a save and a try. Containers on AWS](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=209.29) [Wavelength must be longer than Docker Deep Dive.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=214.64) [That's the fundamentals of if in Go,](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=217.9) [but there's more. So, up next, we're going to dig a tiny bit deeper with simple initialization statements.](https://app.pluralsight.com/course-player?clipId=4b29f8a5-720b-4176-beca-e62f1239bca7&startTime=221.74)

### [Simple Initialization](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33)

[Okay,](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=1.54) [sticking with the same code, if blocks also allow simple initialization](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=2) [statements that are executed right before the expression gets evaluated. So](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=7.91) [as an example, instead of declaring and initializing our variables up here,](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=14.14) [we can actually do it right here in the if block. And a reason we might do it](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=18.55) [this way is that any variables we declare here are scoped within the](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=24.46) [boundaries of the if block.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=29.47) [So, they only exist while we're executing this small block of code, then](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=31.1) [they get garbage collected when program flow moves on.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=36.66) [Okay, so let's get rid of these here, and we'll declare them here instead.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=40.44) [Anyway, look, when execution reaches here,](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=46.74) [we read it left to right, top to bottom.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=49.3) [So, we're declaring and initializing dddLengtMins as](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=52.54) [275, and cawLengthMins is just 30.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=57.22) [Then, we're evaluating the expression is Docker Deep Dive longer than](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=64.84) [Containers on Wavelength? And then, everything's as it was before.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=68.74) [So, yes, it is longer.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=72.97) [And we execute this code here and jump immediately to the closing curly. And I](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=75.1) [think, I don't know actually, well, I think I hinted at it earlier, but once](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=81.55) [any expression evaluates to true, and remember, we go from top to bottom, but](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=86.26) [any time one of them evaluates to true,](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=91.27) [we execute its code, but then we jump straight to the end](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=93.64) [and execute any code right below here.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=97.24) [So it's not like we evaluate this one.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=101.54) [Oh yeah, it's true,](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=103.6) [we execute its code, then we evaluate the next one, and the next one.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=104.75) [No. As soon as any of them is true, execute its code, and jump to the end.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=109.05) [Also though, we can have as many else ifs as we want,](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=117.24) [but only one final catch‑all else.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=121.87) [And then, I think last, but not least, we can nest ifs](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=127.14) [within ifs, within ifs, within ifs.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=131.19) [So just as a quick example, and this code's in the ifnest.go](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=134.74) [file in the conditionals folder of the courses GitHub repo,](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=140.23) [but we're evaluating this here.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=143.94) [It's true, so we execute its code. But part of that code is another if saying](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=145.83) [is the course over 4 hours long? Again, that's true, so execute this code](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=151.7) [here. And go on, just so you know I'm not lying, save any changes, and let's](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=156.94) [give it another run. And there we go.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=162.93) [Yes, Docker Deep Dive is longer, but oh my goodness, it is](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=167.94) [borderline put you to sleep territory.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=172.03) [But that's if in Go, dead easy once you have a play around, but super powerful. Well, next up, switch and case.](https://app.pluralsight.com/course-player?clipId=50a4263d-80d5-4756-a9ea-6626edb37d33&startTime=175.94)

### [Switch and Case Syntax](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa)

[Right then, switch case in Go is a bit like if,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=1.54) [so this should be nice and familiar.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=4.87) [However, there are some important differences,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=7.74) [and we'll see them as we go, but first up,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=9.83) [we'll nail the syntax.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=12.02) [We start out with the switch keyword, and then just like with if,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=13.32) [you can have a simple statement before the expression.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=17.43) [And also, like we saw with if,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=20.45) [any variables we declare here in the simple statement are](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=22.72) [scoped only within the switch block.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=26.88) [So, they get garbage collected once execution leaves the block,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=30.22) [but as well, and I reckon you're getting used to this by now,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=34.86) [but the placement of the curlies is once again and always vital.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=37.95) [So the first one abso‑freaking‑lutely has to go in the](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=43.19) [same line as the switch keyword.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=46.74) [Whoo!](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=51.34) [Anyway, between the curly braces, we stick one or more case statements like this,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=52.16) [and an optional default block, and you probably guessed it,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=57.47) [the default block is the catch‑all that executes if](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=61.6) [none of the case expressions are true.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=64.34) [Okay, well, to keep it simple for now, we won't bother with a simple statement,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=67.37) [and we'll go straight with a simple, simple,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=72.58) [simple expression, Kubernetes Deep Dive.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=75.3) [Right, well, go put that value in memory and then for each case statement,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=78.91) [it looks for a match.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=84.48) [Well, in this example, we match on the first case,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=86.74) [and we execute this block of code here.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=89.96) [And then, importantly, and this differs from some of the languages,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=92.59) [but once the first case block executes,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=96.21) [flow jumps immediately to the closing curly,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=99.07) [and it picks up on the following line.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=102.15) [Now, like I say, some other languages have an implicit fall‑through behavior,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=105.02) [which is where once any case matches,](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=111.43) [all of the ones below it will execute as well.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=114.34) [But that's not the case with Go, but I'm getting ahead of myself. Let's go see it in practice.](https://app.pluralsight.com/course-player?clipId=432b6f35-4539-454a-8481-f7c801cd39fa&startTime=117.74)

### [Switch in Practice](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823)

[Here we are then in a new code file called switch.go,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=1.54) [and if you're following along,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=4.31) [it's in the conditionals folder of the course's repo,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=5.74) [and look at it, it's got a simple framework for a switch case.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=9.34) [So, in the expression here,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=13.44) [we'll start out epically simple with just a string literal,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=14.65) [Kubernetes.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=18.81) [Now, obviously, this can be a variable or a call to a function,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=20.54) [or whatever, but for now,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=24.79) [we're keeping it really simple so the focus is on what](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=27.01) [we're doing with switch and case.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=29.96) [Anyway, Go's going to put that in memory.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=33.14) [Then, it's going to step through each of the cases looking for a match,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=35.79) [so, let's add these.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=39.36) [Now, I'm being daft here, I know, but I just want to be clear,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=41.64) [when it's looking for matches, it's got to be exact.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=45.68) [So, even though some of these might look similar,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=49.59) [they're not, only this one here is going to match.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=52.51) [But as well, and this should be obvious as well, the types have to match.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=56.53) [So we're comparing against a string, that means an integer or a float,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=61.83) [or a Boolean, or anything else is never ever going to match.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=66.22) [Well, tell you what,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=71.24) [let's add some trusty Printlns here to tell us which case matched.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=72.41) [And by the way, it's totally okay to put the code blocks on new lines,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=77.54) [in fact, I recommend it.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=81.1) [Nice readable code should be a top, top priority in all of our lives.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=83.4) [Well,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=89.64) [program execution is going to come in here and it's going to store](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=90.44) [Kubernetes (with a capital K) in memory, then, it's going to step](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=93.72) [through each of the case statements looking for a match. For me,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=97.97) [this one's not going to match,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=101) [so, we should not see this text. But this one here, that is a match,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=102.52) [so when we run it in a second,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=108.35) [we should see this text, Case 2. Kubernetes, with a capital](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=110.13) [kicking K. But, then Go has no implicit fall through](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=114.51) [behavior, so, when it's done with that,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=120.35) [it's not even going to bother looking at these below it,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=122.94) [it'll jump straight down to here and the program will exit.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=125.34) [Well, let's take it for a run.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=129.84) [Oh, and don't forget to save any changes, and there we go, Case 2. Kubernetes](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=131.45) [with a capital K, and look, it's just a super simple example.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=138.9) [You should definitely have a play around and try stuff out. Once you've done](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=146.15) [that though and you feel like you know what you're doing,](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=151.04) [join me in the next lesson, where we'll take a closer look at breaking and fall through.](https://app.pluralsight.com/course-player?clipId=38c096a2-0df2-4198-8983-1d7535bff823&startTime=153.35)

### [Breaking and Fallthrough](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15)

[Okay,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=1.54) [so we've said there's no implicit fallthrough in](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=2.03) [Go's implementation of switch case.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=4.8) [And the reason for that is that each case statement](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=7.22) [comes with an implicit break.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=10.43) [So in some languages, if you don't want fallthrough,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=13.54) [you have to manually insert an explicit break.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=16.65) [Well, with Go, it's the opposite, breaks are in by default.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=19.89) [So, if you want fallthrough,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=23.74) [you've got to manually insert one as the last line before the next case block.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=25.76) [Probably best if we just see it.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=32.14) [So we're obviously the same code here, matching on this one here,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=34.94) [the print Case 2, Kubernetes with a capital \"K\"." concur.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=38.15) [But, if we put a fallthrough here, well actually, what do you think will happen?](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=41.84) [Do you reckon it'll run just the next one or do you think all of them will run?](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=48.05) [Oh, and what about the default block?](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=52.53) [Well, I'll tell you what. Let's take a look.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=55.44) [Okay, so obviously, it ran Case 2, but it only ran Case 3,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=60.44) [the one immediately following it.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=64.55) [And if we look back here, yeah, look,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=66.31) [we've actually got four cases plus the default block.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=68.76) [So, these didn't run. And you know what? If you think about it, that's right.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=71.86) [Remember, we said every case has an implicit break.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=77.98) [Well,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=81.79) [we overrode the one in this block with the fallthrough, so execution moved down](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=82.2) [to here. But when it finished, it hit the implicit break for this block, and it](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=86.19) [jumped straight down here, and the program exited.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=90.95) [So, fallthrough only applies to the case statement it's part of,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=94.34) [meaning if we want a fallthrough to the next level, and the next, and](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=100.81) [the next, we need a fallthrough each time.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=103.83) [So, right now, this should run cases 2, 3, and 4,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=107.04) [but hopefully not the default block.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=111.52) [Let's have a check.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=113.59) [Okay, yeah.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=117.54) [But, will it fall into the default block if we put a fallthrough down here?](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=119.24) [I guess only one way to find out.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=125.04) [Yeah, there we go.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=128.84) [So putting a fallthrough in the case above the default block](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=130.03) [lets execution drop into the default block.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=134.29) [Okay, magic.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=138.54) [But one last thing. To test the default block on its own,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=141.74) [we'll put whatever up here. So we're no longer going to](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=145.43) [match on any of these cases.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=149.15) [Well,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=151.54) [save that, and there we go. We only got the default block. Now](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=151.92) [then, and please don't hate me for this,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=159.68) [but it's more idiomatic in Go, and that just means more common,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=162.24) [by the way, or more of a good practice, but it is more idiomatic](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=166.26) [not to use fallthrough and instead just match multiple values in](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=169.8) [the same case statement.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=174.11) [So to demo this actually, I'll dial it back to this.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=176.64) [The actual code's in a file called idioswitch for idiomatic switch.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=181.1) [And, of course, it's in the conditionals folder of the GitHub](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=185.19) [repo, if you're following along. Anyway,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=187.55) [I'm importing a couple of packages here to help with](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=190.85) [some random number generation.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=192.94) [But also, we're getting a tad more realistic here by switching on a value](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=194.76) [returned by a function rather than just some daft string literal.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=200.08) [Well,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=205.44) [we're declaring a new variable called tmpNum, and we're initializing it](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=205.71) [with the value returned by a function called random.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=209.08) [So, down here, let's drop this code here in for the random](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=212.74) [function. And if you've been following along,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=216.22) [you'll know it's taken no argument, but it is returning a single integer.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=219.16) [Here, we're setting it based on the current time to help with the](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=225.04) [number generation, and then based off of that seed,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=228.46) [we're returning like a randomish number between 0 and 9.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=231.66) [So, back up here,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=236.84) [tmpNum gets assigned that random number that's going to be between 0 and 9.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=239.16) [So, in our case statements,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=244.34) [let's do this. Now, each line is a comma‑separated list that](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=246.28) [importantly matches multiple values per case statement instead of](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=252.96) [having loads of cases and loads of fallthroughs. Basically, we're](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=257.79) [branching on whether the number is odd or even.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=261.92) [Well, as always,](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=266.84) [let's see if it works. Okay. Yeah look, sometimes generating an odd](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=267.76) [number, sometimes an even. Alright, time for one last look back at if, but in the context of error checking.](https://app.pluralsight.com/course-player?clipId=938295d8-3852-4a20-b937-d9cf91e30e15&startTime=279.57)

### ["If" and Error Handling](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f)

[Right.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=1.14) [Time for a quick look at how the if statement is](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=1.68) [commonly used for error handling.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=4.27) [Now, there are other ways to handle errors,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=7.34) [but what we're about to look at has been pretty much the way for the last,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=9.64) [I don't know, forever.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=14.31) [Basically, you'll see this in just about all code out there.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=16.74) [And you know what?](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=20.75) [Error handling might possibly be the single most common usage of if in Go.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=22.41) [Anyway look, and I'm not sure if we mentioned this yet,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=29.51) [but it's idiomatic in Go for functions to return an](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=32.51) [error value as its last return,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=36.64) [like this function here that's testing connectivity to a remote host.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=39.35) [So we've got the function name and its input parameter and type,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=42.82) [but then two return values, and of interest towards us is the last one,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=46.27) [and yes, it is of type error.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=51.43) [So yes, error is actually a defined type in Go,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=54.61) [just like we've seen with strings and integers and Booleans and the likes.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=58.5) [As well though,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=62.24) [it's normal practice in computer science in general for a](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=63.65) [nil or 0 error code to indicate success,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=67.21) [and then anything other than nil or 0 indicates an error.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=70.68) [So if the function executes as expected,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=74.88) [like it gets the input values and everything it needs,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=78.07) [but more importantly, the function code executes as expected,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=80.41) [then it returns nil value and all is good in the world,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=84.72) [but if something goes wrong, then it returns a non‑0 code,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=87.5) [like a 1, or a 2, or a 3, or whatever, right?](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=91.56) [And of course,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=94.13) [it's normal to define different error codes based on the type of the error,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=94.93) [and look, we'll see it in a second.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=99.56) [Now then, of course,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=101.79) [it's all well and good to write good functions that return error codes,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=104.24) [but the onus is on us as developers to check for](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=107.42) [them and do something with them, and we'll see this in a second,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=110.88) [right?](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=113.82) [But honestly, all good Go code consistently,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=114.23) [like all the all the time, is checking for errors and acting accordingly.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=118.14) [Anyway, what we've got here is a call to the open function in the OS package.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=124.64) [Oh,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=129.35) [and we're in the iferror.go program in the](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=129.89) [Conditionals folder of the Github repo, yes?](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=132.26) [Anyway, we're trying to open a file called test1.txt in my current directory.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=134.24) [Now really quick,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=141.14) [there are more idiomatic ways to open a file in the current working directory,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=142.13) [such as the getwd or the executable functions,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=146.56) [both from the OS package.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=149.93) [But you know what?](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=151.86) [That'll just clutter this code and distract from what I'm showing.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=152.87) [So I'm doing it like this.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=155.6) [Anyway, the os.Open function returns two values.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=158.84) [We're not bothered about the first one,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=162.41) [so we'll just ignore it with the underscore,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=164.33) [but the last value it returns is an error code,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=166.5) [so we'll capture that there with the err variable.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=170.34) [And honestly, oh, you are going to see this here everywhere,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=173.26) [but, and this is the important bit,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=179.14) [we check that error code stored in the err variable,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=181.5) [and if it's not nil, so if there is an error,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=184.71) [we're going to print it to the screen.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=187.75) [Obviously, in the real world, we'd have some code hid to catch it and handle it,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=189.38) [but because I want you to see what's actually in the error code,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=194.22) [we'll print it to screen.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=197.6) [Well, we'll save that and give it a whirl.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=199.74) [I said this one was called iferror.go, and remember,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=202.44) [you have to have saved any changes, but also,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=206.3) [if you cloned the repo locally,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=208.7) [you will need to run it from within the Conditionals folder.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=210.32) [But there we go, and funnily enough, it threw an error.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=213.24) [Who would have thought?](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=217.51) [Well, this is the actual text returned by the Open function in the OS package,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=218.66) [and it looks like the file doesn't actually exist.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=223.42) [Let's have a look.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=226.13) [Ah!](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=228.14) [Right!](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=228.64) [We're looking for test1, but it's actually just test.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=229.48) [So if we jump back here to our code and make this just test,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=232.61) [and as it's obviously going to work this time,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=236.2) [let's put a copy of this println down here,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=239.28) [so we see what the return code looks like when it actually does work.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=241.61) [Give that a Save and Run.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=245.4) [Okay, magic!](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=248.54) [An actual nil value, just we said.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=251.44) [So we've got this here when it errored out, and then a nil value when it worked.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=255.84) [Well,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=262.34) [I have been saying you'll see this kind of error checking all over the place,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=262.58) [so, oops, give me a second, actually.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=266.7) [Right.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=269.16) [This here is the deployment\_controller package from the Kubernetes project,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=269.63) [meaning Kubernetes is obviously written in Go,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=273.88) [and actually, you know what?](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=277) [Most of the major cloud infrastructure projects are.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=278.34) [So Docker, Istio, Linkerd, obviously Kubernetes,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=281.29) [and way more, all of them, written in Go.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=284.99) [Anyway,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=288.44) [the reason we're looking at the Kubernetes code is that it](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=289.12) [is proper weapons‑grade production code.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=291.63) [And if we search up here for if error, well,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=294.79) [look at that; 20‑odd references just in this package.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=298.52) [And look, the second one down is pretty much exactly what we've done.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=303.29) [If err is not nil, okay, and this is saying or,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=308.42) [if the length of the value in the deployments variable is 0,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=312.41) [do this here, basically exit the function and return a nil value.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=316.36) [So I guess yes, a tiny bit more complex than what we've just done,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=320.84) [but the point is,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=324.03) [the code is literally littered with this type of error checking,](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=325.11) [so like we said, very idiomatic.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=329.16) [And you know what?](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=332.11) [That's been a lot. So time for a quick session recap.](https://app.pluralsight.com/course-player?clipId=9116922b-623b-421a-9985-5fac5d8d748f&startTime=333.72)

### [Recap](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf)

[Okay, I'm going to make this short and sweet.](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=1.54) [We started out looking at if statements and how they use Boolean](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=3.56) [logic to evaluate if things are true or false.](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=6.96) [The structure is basically if something is true,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=10.64) [execute the block below it, and, of course, we can have as many else ifs as](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=12.9) [we want, and then, obviously, if any of those are true,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=17.81) [execute their code, and we round things out with a catchall else](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=20.27) [that'll run if none of the stuff above it is true.](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=25.48) [Well we also said that as soon as any of the statements evaluates to true,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=29.34) [and we evaluate them from top to bottom, but, as soon as one fires as true,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=33.51) [we run its code, but then program execution jumps straight to the end, so](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=37.84) [we don't bother checking the rest.](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=41.84) [Well, then we looked at how if is commonly used to check for errors. So, it](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=44.14) [is good practice for Go functions to return error codes as their last return](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=48.5) [value, and then, we normally capture that in a variable called err, and then](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=52.86) [use it to say if err isn't nil, so it's not nil, indicating an error, run](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=57.6) [this code and handle it. Well,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=63.17) [we also looked at switch case in Go, and the logic is similar to if,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=66) [but, it tends to be a shorter way of writing an if block without](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=69.58) [having to have tons of else ifs. Anyway,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=73.79) [we evaluate an expression. The first case to match runs its](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=77.23) [code, and then it breaks to the end here. But if we want to run](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=80.33) [more of the case statements, then we can drop in a fallthrough,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=84.65) [and the next one will run, but only the next one.](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=87.62) [So, if we want them all to run, they all need a fallthrough.](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=91.05) [And I reckon that about wraps it. Again,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=95.84) [practicing is vital,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=98.18) [so, fire up your editor and play around, and see if you](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=99.86) [can break it and figure out what works and what doesn't,](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=103.05) [and when you think you're good, join me in the next module, because we're looking at loops.](https://app.pluralsight.com/course-player?clipId=f05b3c73-fa0a-4f14-aad0-b3755bb9e0bf&startTime=105.3)

## [Working with Loops](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b)

[Right, if you're following along, we are properly building some Go skills.](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=1.54) [Well, in this module, we're looking at loops. And in Go, there's only one loop,](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=7.44) [the for loop. But to be honest,](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=12.93) [that's a bit misleading because it is a really flexible loop, in fact,](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=15.96) [I should probably have said there's only one loop keyword in](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=20.32) [Go, the for keyword. But it can do infinite loops, while loops,](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=24.2) [and it can even range over lists.](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=30.08) [Anyway, we'll hack it like this, we'll look at the basic syntax first,](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=33.04) [then we'll look at infinite and while loops,](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=37.01) [then ranging, and a bit about breaking and continuing,](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=39.34) [and we'll finish with a recap. Well, let's do this.](https://app.pluralsight.com/course-player?clipId=471d07d8-50fb-41d9-871f-6ec0dd1ef43b&startTime=42.48)

### [Syntax](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3)

[So the basic syntax is dead simple, it's the for keyword and an expression.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=1.54) [Though actually,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=6.91) [the way to create an infinite loop is to just leave out the expression.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=8.88) [That way, Go assumes it's Boolean true and it will loop while true.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=13.03) [If it's not blank though,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=18.84) [then it can be either a Boolean expression or a range expression.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=20.65) [And we're about to see them both,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=24.79) [so don't worry if that's a bit confusing right now.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=26.26) [Anyway, as always, placement of the curly braces is vital.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=29.34) [Then between them, we put the code to execute.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=33.64) [So, like we said, an infinite loop looks like this,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=36.54) [just the for keyword followed by the opening curly.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=39.54) [And, also like we just said, it's basically for true,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=42.94) [or while true as you get in some other languages.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=46.27) [The thing is,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=49.26) [it'll loop forever unless we throw a break in the code somewhere here.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=51.14) [Well, to loop on a Boolean expression looks something like this,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=57.04) [this one's saying for i is less than 10,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=60.57) [but you can use any valid Boolean expression,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=62.95) [and it'll basically loop while i is less than 10.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=65.06) [Oh,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=68.36) [and it's idiomatic across just about all programming languages to use a](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=68.86) [lowercase i as your incremental or your index value.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=72.99) [We'll see it soon.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=76.2) [Anyway, finally, there's for range, and this is the bizzo, right?](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=78.24) [It takes a list of some sort, maybe a slice or a map,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=82.43) [and we'll cover these next I think,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=85.71) [but it takes the list and it illiterates or it ranges over it from the](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=87.28) [top to the bottom looping through the whole list.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=91.92) [And then, when it reaches the end,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=95.84) [the loop exits and code picks up after the closing curly.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=97.31) [So, in the example we're looking at,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=102.34) [the list is called courseList and for every iteration of the loop,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=104.82) [it works its way down the list and assigns the current value to the i variable.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=108.53) [Okay, one last thing about syntax before we see it in action.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=114.74) [Just like with if and switch in the previous module,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=119.09) [we can give simple pre and post statements on the opening line.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=122.11) [So, if we want the example back to the Boolean expression,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=127.14) [basically a while loop in a lot of languages,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=130.73) [well, we can do this.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=133.12) [So, it's three statements divided by semicolons.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=136.14) [This is a simple pre statement, initializing i as zero.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=140.54) [Next, is the Boolean expression saying loop while i is less than 10.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=143.69) [And then, this final bit is the post statement,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=147.61) [and it's applying the increment operator,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=149.95) [basically, incrementing the value of i by 1 each time the loop completes.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=152.41) [Now, it's important that this is a post statement, yeah.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=159.14) [So it runs after each iteration of the loop.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=163.19) [So we initialize i with 0 in the pre statement,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=166.94) [the expression evaluates to true because,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=169.78) [well, we've just set it to 0.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=171.92) [The loop then runs and then the post statement increments it from 0 to 1,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=174.54) [and we go again, rinse and repeat 10 times until i actually is 10.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=180.14) [Then the loop exits, i gets garbage collected,](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=184.54) [and execution picks up after the final curly. Well, we know that talk is extremely cheap, so let's see this stuff in action.](https://app.pluralsight.com/course-player?clipId=88be42a7-8876-4f71-9e97-d9a24be5dbe3&startTime=187.16)

### [Infinite and While Loops](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578)

[Right then, we're in a new code file here called self‑destruct.go,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=1.64) [and you'll see why in a second.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=6.6) [But, it's in the loops folder of the GitHub repo, if you're following along.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=8.84) [Now then, we're going to write a super quick countdown timer,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=14.54) [counting from 10 down to 0.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=17.34) [And, because I'm, I don't know, a little bit of a sci‑fi geek,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=19.84) [I'm imagining we're on a spacetime ship that is under attack from a hostile](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=24.87) [alien force intent on stealing some important technology,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=28.92) [of course.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=32.06) [Well, we're also outgunned and we're about to lose the tech.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=34.14) [So, as the captain of the ship,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=37.76) [I've launched the rest of the crew in escape pods and heroically stayed behind](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=39.55) [to initiate the ship's self destruct on a 10 second timer.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=44.07) [So, we're about to write that self‑destruct timer code.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=49.64) [Anybody still here, or have you all left?](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=53.24) [Okay, look,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=56.84) [we're inside the main function and I guess we could](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=57.9) [initialize the time of variable right here,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=60.63) [but we're not going to, it's more idiomatic to declare it](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=62.61) [as part of the loop. And like I said,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=65.56) [we'll start at 10, but, we want it to count down to 0 and](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=68.47) [then stop, so, each time the loop runs, we want to decrement the value by 1.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=73.1) [Okay, well, like all good self‑destruct timers,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=79.74) [we need something telling us how long is left, and we're](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=82.53) [going to need something to insert approximately a 1‑second](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=86.34) [delay each time we iterate.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=89.72) [Okay, I reckon that might do us.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=92.94) [Oh, I forgot to mention, obviously we're leveraging a couple of functions](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=94.57) [from the time package we're importing up here. Anyway,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=99.02) [when the timer reaches 0, we want something a little bit](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=103.54) [spectacular, so let's whack in a quick if statement, so, if](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=106.59) [timer = 0, then run this, boom.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=111.31) [Yeah, not quite so spectacular.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=114.74) [Whatever though,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=117.35) [let's give it a save and a run, literally, run, before](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=118.42) [the ship blows up. Oh dear. Okay,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=121.8) [so, what shall I do in the last few seconds of my life?](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=124.74) [Oh, yeah, obviously, Sunderland are the greatest football team in the world,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=127.59) [ciao, ciao. Okay, it worked, well,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=132.05) [kind of, I've got this sloppy 0 hanging around. Now, to fix that, I](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=136.99) [guess we could loop while greater than or equal to one,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=141.86) [but actually no, it would never trigger the if statement, would it?](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=145.74) [So, tell you what,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=148.43) [we'll slap in a break. Now, we're going to look at break properly](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=149.42) [later. Anyhow, after we execute the boom, and I suppose it wouldn't](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=153.44) [make much of a sound in the near vacuum of space,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=158.33) [but what the heck, then as soon as we hit the break,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=160.87) [we exit the loop and we run any potential code immediately following the](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=164.39) [closing curly. Meaning, we shouldn't see the sloppy 0 anymore.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=168.19) [Well, we'll check if that works.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=174.24) [Let me speed it up this time, it's a bit like Groundhog](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=178.24) [Day or Boss Level, if you've seen that.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=180.8) [All right, way better this time. But, really quick before we move on,](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=183.44) [just to be clear, this pre‑statement here gets executed before the first](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=188.42) [evaluation of the expression. Then, the post statement, that gets executed](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=194.51) [post, or after each iteration of the loop.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=199.57) [But importantly, that is before we reevaluate the loop condition. Okay, well, next up, we'll look at for range loops.](https://app.pluralsight.com/course-player?clipId=2f22fb8a-ce2b-4d32-920f-d7dbc77d6578&startTime=202.64)

### [Range Loops](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412)

[Okay, for‑range loops.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=1.64) [And to demo this,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=3.54) [we're back from outer space. I guess I went back in time or](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=4.41) [something so that we're before the self‑destruct.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=8.07) [Either way, we're back to our good old Pluralsight courses.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=10.36) [Now then, this here is a list of courses in progress. And to make it](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=14.44) [easier to read, I've split it over multiple lines.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=18.65) [Now technically, it's a slice, basically an unordered list of numbered items.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=21.46) [Well actually, under the hood, it's a reference to an array and the likes,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=26.84) [but we'll cover all of that in the next module. For now,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=30.73) [all we need to know is it's a list.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=34.31) [So, we've got a slice of four strings, four courses, here.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=37.74) [And to ease you in gently, let's just jump straight and see for‑range in action.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=41.72) [So, the range command is going to take the list, range over it, so step](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=46.54) [through it, one value at a time until it reaches the end, but it steps](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=52.01) [through one element of the list per iteration of the loop.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=56.8) [So, the first time the loop runs, it'll get one course. The](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=60.84) [next time, it'll get another, and so on.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=64.17) [Now then, with slices, for‑range returns two values each time it](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=68.04) [loops, the index value and the data value.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=73.13) [We've got four items in our list, so that will be indexes 0 through 3.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=76.84) [The first time through the loop,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=82.84) [it's probably going to get 0 and Docker & Kubernetes: The Big Picture.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=84.16) [The second time, 1 and Docker Networking, Then 2 and Getting Started](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=88.12) [with Kubernetes, and finally, 3 and Kubernetes Deep Dive. Now, for](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=92.23) [our for loop, we're not bothered about the index value, so we'll](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=97.83) [ignore that by passing it to the underscore, the blank identifier.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=100.85) [But each data value, yeah,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=104.65) [we'll hang on to that with the i variable. Then obviously, we're](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=108.31) [printing i to the terminal each time we loop.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=112.4) [So, we should end up printing the names of all four courses.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=114.87) [Okay, let's save that and give it a go.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=121.54) [Alright, just as expected.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=126.84) [Now then, we can also nest loops, so loops within loops. So](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=129.24) [to do that, we'll have another slice of strings. This time,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=135.53) [courses marked as completed.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=138.66) [So, just to be clear, we've got two lists now, one listing courses in progress,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=141.74) [the other listing courses completed.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=147.19) [Well, let's add another for‑range here, importantly inside the existing loop.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=149.94) [Well, this one's going to loop through the completed courses list, and](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=156.64) [see how it's using j to hold the values from it.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=160.9) [So, the outer loop is holding values in i, the second inner one in j.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=164.72) [Well, let's add an if here,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=171.94) [comparing i with j and then printing this if they match.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=173.57) [So, looping over courses in progress and also courses completed with an](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=178.18) [if block to spot if a course appears in both the in progress list and the](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=183.79) [completed list. And if you're new to this and it's feeling a bit much,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=189.38) [let's just step through the flow nice and slowly. We have got the outer](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=195.19) [loop stepping through the list of courses in progress, but inside of](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=201.35) [that, we've got a second loop stepping through the list of courses](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=205.41) [completed.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=208.5) [Now then, on the first iteration of the outer loop, it's going to stick](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=210.24) [Docker & Kubernetes: The Big Picture into the i variable.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=214.38) [In fact, let's put that up here.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=217.99) [Okay, anyway. Next up, we hit this line here,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=220.24) [so we start iterating through the inner loop. And we stick the](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=223.5) [first value from the completed list into the j variable. And look,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=226.9) [it's going to be Docker & Kubernetes: The Big Picture as well. That](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=230.9) [means when we do this if comparison here, we'll get a match, and](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=235.51) [we'll print this line.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=239.65) [Okay,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=242.24) [well, we'll hit the closing curly of the if statement here, and](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=243.12) [we'll go back to the top of this loop again,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=246.74) [the inner one, and we'll run through that again. This time,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=249.16) [we'll put this in j.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=252.01) [Does i match j this time? Negativo. And we hit the end of the completed list.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=253.83) [So, we leave the inner loop. We hit the closing curly of the outer loop, but](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=259.94) [we've not reached the end of the list for the outer loop,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=264.8) [so we go again.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=267.49) [Well, this time we'll put the Docker Networking course](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=270.94) [into i and go through it all again.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=272.97) [So back to our inner loop, a couple of iterations,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=274.97) [one for each of the two values here.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=277.96) [Each time again, checking for a match. This time,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=279.7) [we won't get any matches, so we'll come back up here, rinse and repeat.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=282.66) [Shall we see if it works?](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=288.44) [Okay, well, remember to save your changes.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=290.14) [Oh, and we don't need this Println here anymore.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=292.35) [Let's give it a go, though.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=295.54) [Oh, and as if by magic, it works. Now, I know it was](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=301.44) [a bit of a convoluted example,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=307.59) [but we've seen how to nest loops within loops, as](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=309.54) [well as throwing in some if blocks.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=313.15) [Well,](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=316.44) [time for a quick look at break and continue before we recap everything we've learned.](https://app.pluralsight.com/course-player?clipId=66d16939-c874-44d8-bc33-24aa1ec6a412&startTime=316.83)

### [Break and Continue](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee)

[Time for a quick look into how break and continue work.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=1.64) [In fact,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=4.61) [we saw break earlier when the self‑destruct timer was](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=5.14) [flashing a 0 that we didn't want.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=7.79) [Remember, we put a break in here inside the if block,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=10.2) [which in turn is inside the loop.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=14.47) [But, as soon as program flow hits the break,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=17.04) [it literally breaks out of the loop and execution](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=19.62) [picks up after its closing curly.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=22.66) [And you know what, yeah, that is break in its most basic form.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=25.08) [It breaks out of the current loop.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=29.13) [But what about nested loops like we've got here?](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=32.34) [And apologies for the colors.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=35.81) [They're just there trying to help highlight the different loops.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=37.97) [But we've got three nested loops, and apologies if you're color blind.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=41.18) [I hope they are visible and make sense.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=46.85) [Well, if we break here in the innermost loop,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=50.24) [we'll drop out of this loop here, the blue one.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=53.75) [Then we'll hit its terminating curly, go back to the top for further evaluations,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=56.65) [and potentially more iterations.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=61.7) [Magic.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=64.28) [But what if we want to break out, I don't know,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=65.69) [two levels, maybe to the outermost orange loop?](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=68.63) [No sweat.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=71.92) [That's where labels come into play.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=72.8) [So, we can make a label up here.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=74.9) [We'll call this one breakPoint.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=77.37) [But, honestly, you can call it pretty much whatever you want.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=79.94) [Just, obviously, don't call it one of Go's whatever it is, 25 keywords.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=83.6) [Anyway, when we make the call to break here,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=88.66) [if we give it the label, instead of just breaking out of the current loop,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=92.26) [we'll break to here, thanks.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=95.7) [And that really is the crux of break.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=98.74) [In its most default form, it breaks us out of the current loop,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=101.13) [but if we use labels we can break to pretty much anywhere we want.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=104.67) [Though, a bit of a gotcha, Go being Go, yeah,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=110.66) [it is not a fan of you defining labels and they're not using them.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=114.65) [So, if you define one, you've got to use it.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=118.32) [Well, switching gears slightly to continue.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=122.94) [The idea with continue is that whenever Go encounters one in a loop,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=126.22) [it drops whatever it was doing and it jumps straight back to the top for a new](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=131.19) [evaluation and potentially more iterations through the loop.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=135.9) [Oh, and actually, post statements, they are actually](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=140.54) [executed as part of the continue.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=144.73) [So basically, stop what you're doing, run any post statements, and](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=146.53) [then reevaluate the expression, and maybe go again.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=150.66) [Well, we've got a bit of our old self‑destruct timer code here,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=154.54) [so defining a timer as 10 and then looping while it is >= 0 and](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=158.89) [decrementing by 1 for each iteration of the loop.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=164.87) [Well, if we change this if statement to this, okay,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=168.34) [we're using some simple math to determine whether or not the](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=174.98) [value in timer is an even number. If it is,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=178.31) [we'll run the continue here,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=182.47) [which, we just said, interrupts normal flow, so it skips the Println here](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=184.03) [and obviously the injection of time as well, but it jumps us straight back](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=188.62) [to the top where we run the post statement here to decrement the timer by 1,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=193.55) [and then we reevaluate the expression.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=197.46) [Look, basically, we are skipping the Println every time it's an even number,](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=200.34) [meaning when we run it, we should only see odd values.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=205.53) [Well, let's go and see if that actually happens.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=209.97) [Okay, magic, only odd numbers.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=215.04) [And that, folks, is break and continue. Whoo, let's wrap the module with a quick recap.](https://app.pluralsight.com/course-player?clipId=81700c8f-f504-4800-b889-63c96b0825ee&startTime=217.54)

### [Recap](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015)

[Okay, let's make this quick. Go only has one key word for loops, for,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=1.54) [but it is pretty flexible.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=7.34) [It can do infinite loops, traditional while loops, where it](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=9.71) [loops while a Boolean expression is true, and it also does range](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=12.75) [loops, where it steps through a list.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=16.64) [So, the for key word on its own is the equivalent of while true, so an](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=19.64) [infinite loop. If we use it with an expression like this,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=24.47) [it loops while the expression evaluates to true.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=27.89) [Like I said, a lot like a while loop in some other](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=30.88) [languages. But, then range loops,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=33.89) [which look like this, range or iterate over a list, and the loop](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=36.4) [exits when the end of the list is reached.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=40.38) [Well, as well as that, we saw they can have simple](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=44.34) [initialization statements that run before the loop starts, as](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=47.04) [well as post statements that run after each loop.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=50.85) [Okay, we also saw we can nest loops within loops within loops within loops.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=55.44) [And, of course, other code,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=60.2) [like if statements and the likes, can all go inside of loops. But, then last](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=61.5) [but not least, we saw how we can use break on its own to break out of the](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=68.06) [current loop, but if we couple it with a label,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=72.09) [we can target pretty much anywhere for break to land.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=75) [And then, the continue key word lets us drop the current flow of](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=79.72) [a loop, return to the top where we run any post statements, and](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=82.82) [maybe run through the loop again.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=86.47) [Okay, well, look,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=89.24) [a really quick peek into the Kubernetes code base again, and we'll just go](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=90.55) [for the same deployment controller code that we went with last time. Only](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=94.03) [this time we'll look for for i. Yeah, anyway,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=97.71) [look, it's just easier than looking for for on its own.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=101.83) [Anyway, this one's a classic example.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=106.44) [So we can see it's got a pre‑statement initializing i, told you it](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=110.44) [was idiomatic, then it's evaluating if i is less than the value](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=115.53) [stored in workers, and then, the post statement here increments it](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=119.86) [for each run of the loop.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=123.34) [Then, of course, there's code here inside the loop to run, which this](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=125.94) [time is a call to a function, but it's using the Go key word to start](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=130) [it as a Go routine, and I know that might be a bit complicated, but if](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=133.68) [you've been following along,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=137.18) [you'll know that the wait package here must be being imported somewhere at](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=138.18) [the top, and that this function here is part of that weight package and it's](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=141.78) [exposed because it starts with a capital letter.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=146.09) [Actually, in fact, if we click this, yeah,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=150.14) [look,](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=153.19) [this is the package it's declared in and we can](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=153.69) [even click this, and there it is.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=155.88) [But you know what?](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=159.64) [That's not the point?](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=160.17) [The point is, you will see for loops everywhere in Go, massively useful. Okay, well we're rattling through stuff. Next on the card, arrays and slices.](https://app.pluralsight.com/course-player?clipId=94a0e66d-939c-448b-aa4e-a9f67d1fe015&startTime=161.1)

## [Arrays and Slices](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239)

[Okay,](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=1.54) [arrays and slices. And as usual, the idea is that when we're done, you'll have](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=2.44) [a decent grasp of what they both are, as well as obviously, how to work with](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=7.14) [them in Go. Well, we'll set about things like this.](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=10.77) [First up, we'll nail any theory we're going to need. You know,](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=14.82) [basic stuff like what are arrays and slices, and how are they different? But](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=17.63) [also, pretty much as we're about to see, how slices are better and more powerful](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=22.8) [than arrays. Then, we'll look at the syntax of how to declare them, and we'll](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=28.49) [see how to work with them. Along the way,](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=33.47) [we'll look at some examples, so see how to actually implement them in code.](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=36.27) [And I reckon also, we'll look at how we can manipulate them,](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=40.84) [like grow and shrink them, and change their values and stuff. And yeah,](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=43.77) [that's the plan.](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=48.93) [So first up then, let's go and see if we can explain the theory and the fundamentals.](https://app.pluralsight.com/course-player?clipId=44a0f15a-a6a9-44f5-a48d-3b8e06381239&startTime=50.03)

### [Theory](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e)

[Now then, right at the top here, I want to throw out this warning.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=1.64) [If you've worked with other languages and you've got](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=6.04) [experience with arrays and slices, well,](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=8.82) [the way Go handles them might be different. So, don't just assume,](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=11.52) [oh, yeah, I've done this kind of stuff before.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=15.63) [Let's just crack on.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=17.43) [Don't. Just give me a few minutes of your time to make](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=18.58) [sure we're all on the same page.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=21.9) [Anyway,](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=24.74) [jargon aside, we're basically talking about numbered lists, where](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=25.21) [everything in the list has to be of the same type, so I guess all](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=29.02) [strings or all integers or floats or whatever.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=32.88) [Well, let's break that down.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=35.94) [Like we said, we're talking about lists.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=37.94) [So a list, of course, is here.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=40.83) [And then each item in the list is numbered, or, as we tend to say,](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=42.88) [indexed. As well, though, every item's of the same type. Here,](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=46.36) [we're obviously strings, but it could be floats or ints or something else.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=51.81) [The point is, within the same list, they've all got to be the same type.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=56.14) [And I guess backing that, and I know I've said it a ton of](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=60.54) [times, Go really is picky about types.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=62.68) [Anyway, same again for a list of ints,](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=67.44) [all the same type, oh, and always zero indexed.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=69.83) [Oh, and then this one here, that's illegal.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=75.34) [It's trying to mix types, and the compiler won't have it.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=77.86) [And you know what, I reckon that'll do for the uber‑basics.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=82.64) [Let's switch tack a bit, and we'll see how arrays and slices,](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=86.42) [while being similar, are actually different,](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=90.07) [but they work together.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=93.02) [Talk about confusing.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=95.24) [It's not actually, though. Join me in the next clip, and we'll see how easy it actually is.](https://app.pluralsight.com/course-player?clipId=57f9d85b-8fdf-4b30-82f5-9003cd09cd7e&startTime=96.46)

### [Arrays vs. Slices](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764)

[Right, then. Two quick things before we dive in. Slices are absolutely where the](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=1.64) [action is, like you will rarely see arrays used directly in Go. As well though,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=7.19) [we're going to go a little heavier than normal on the theory here because I](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=14.04) [think it's really important to understand how slices are put together behind the](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=17.92) [scenes, certainly if you plan on being serious with Go. Well, arrays are like](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=21.78) [the lists we just saw.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=27.17) [But importantly, arrays have a fixed length, meaning](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=28.63) [if we create one with six elements,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=32.75) [then we're stuck with six, like there's no adding a seven or an eight.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=35.54) [I mean, technically you can do it, but it is a proper bunch of work.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=40.33) [Like, you'll have to create a new array,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=44.76) [copy the contents of the old one over, do a bunch of renaming, and](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=46.93) [probably a bunch of other stuff I can't think of right now.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=50.13) [So, if only there were a better way.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=53.64) [Captain obvious, say hello to slices.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=57.94) [So at the very highest level, slices look and feel like arrays,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=64.24) [but they're resizable, meaning need to add some elements? No](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=68.36) [sweat. Append a value to the end.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=74) [Need to make the slice shorter? Ha, walk in the park. Just](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=76.1) [create a slice of it, so a slice of a slice. Now then, the way](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=80.18) [it all happens is basically abstraction.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=86.2) [You see, slices are actually built on top of arrays. And I'm](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=89.14) [going to back right up and repeat that again because it is dead](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=93.47) [important to know, so here it is, slices are built on top of arrays. In fact,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=96.55) [it's probably where the name comes from because slices, well yeah,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=102.84) [slices of an array. And maybe actually a picture helps.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=108.27) [So this is an array with 10 elements, then on top of it we](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=113.14) [create a slice of just 3 of them. And it doesn't have to be the](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=117.01) [first three. It could actually be any,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=121.85) [as long as they are contiguous. In fact, drawing it](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=123.63) [like this might be a bit better,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=128.19) [a bit more obvious to see why we call it a slice, yeah. Back here though,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=129.87) [no actual data gets stored in the slice construct.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=135.48) [It's basically pointers pointing to the actual data in the array below,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=139.38) [meaning the slice itself is basically a name,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=144.44) [I guess a type, an offset in the underlying array, and a length,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=148.05) [no actual data.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=151.9) [And just to blow your mind,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=154.54) [we can have multiple slices of underlying arrays. And if we](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=156.99) [do that, as they're basically pointers, they are really cheap. However,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=160.65) [in that model,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=166.37) [obviously changing the value of any element in one of the](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=167.42) [slices changes it in the array below,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=171.81) [meaning every other slice that references it sees the change. And look,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=174.52) [I know it's not rocket science, but it is super important to know,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=180.38) [especially when passing slices as arguments to functions.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=184.2) [So, because slices are references by nature referencing the array, well,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=188.3) [when we pass them to functions, they get passed by reference,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=196.49) [not value.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=199.91) [Well, actually, the slice header,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=201.34) [which is basically jargon for the location of where the](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=203.04) [data actually is in the array, that does get passed as a](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=205.84) [value, so copied to the function.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=208.86) [But as it is actually a reference to the real data in the array,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=212.24) [the overall effect is that of passing by reference.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=216.38) [Oh, now, I know that might sound confusing,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=220.64) [especially if you're new to all of this,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=224.08) [but it should be pretty simple if you watch the functions module where we](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=226.64) [covered passing by reference and passing by value. Basically look,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=231.36) [the slice itself gets passed to the function by value,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=235.84) [so a copy of the slice is passed to the function.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=238.92) [But seeing as how the slice is actually just pointing us to the array,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=241.94) [the actual array values are not passed to the function, just the slice,](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=245.64) [and the slice is basically a reference.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=250.84) [I'm starting to confuse myself.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=254.04) [No, just kidding. But actually, I reckon that is enough on the theory. Let's go and get our hands on.](https://app.pluralsight.com/course-player?clipId=48d30a51-3fe7-4674-bb9a-66bf6cb0a764&startTime=255.98)

### [Working with Slices](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7)

[All alright, to create a slice,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=1.64) [and we'll be working exclusively with slices by the way.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=4.44) [Remember, slices are where the action is.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=7.38) [But, to create one, we can use the built‑in make function.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=10.24) [Now, this accepts three values, type, length, and capacity.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=15.34) [The type, well I guess that's pretty obvious.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=20.23) [Are we making a slice of strings or integers or whatever?](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=22.33) [I reckon the length should be pretty obvious as well.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=26.04) [How many elements do we want in the slice?](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=28.8) [Like how many entries.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=31.54) [But capacity.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=33.54) [What's that about?](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=35.05) [And how is it different to length?](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=36.26) [Well, capacity specifies the maximum size of the slice,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=39.04) [or maybe actually, the expected size.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=42.09) [Basically, under the hood,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=45.44) [this is going to be how big to size the array that's backing it.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=47) [Which is why we did what we did, explaining how slices are backed by arrays.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=51.34) [It's kind of fundamental to know.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=56) [Well, anytime we create a slice, Go creates an array to hold the actual data,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=58.64) [remember?](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=63.42) [And I know, I am beating this point to death,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=64.17) [but a lot of us do learn by repetition, so slices don't actually store data,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=67.07) [it's always stored in the array.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=72.16) [Anyway, look,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=73.58) [if we wanted to define a slice of strings with an](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=74.54) [initial length of 5 and a capacity of 10,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=77.39) [we'd do it like this.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=79.9) [Okay, let's add this here.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=82.34) [It's basically a Printf from the fmt package that displays](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=83.93) [the length and the capacity of the slice.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=87.61) [This here just says, insert the output of len(courses) here,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=89.86) [and the output of cap(courses) here, but a standard base 10 integers.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=93.6) [And then, okay,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=99.34) [len and cap are built‑in functions that return the](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=101.11) [length and capacity of slices.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=103.47) [Anyway, look, we'll get length of slices 5, and capacity is 10.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=106.24) [Well, let's give it a save and a try.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=111.34) [Marvelous.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=115.84) [Now then, all we've actually done so far is declare the slice,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=117.24) [we've not initialized it with any values,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=120.61) [so it's basically an empty slice.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=122.85) [Well, to actually put some data values into it, we can do this.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=126.74) [And look, I could put this anywhere,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=130.74) [but I'm putting it up here below the definition of the slice,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=132.56) [basically, because I think it looks tidier.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=135.32) [But,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=138.54) [what we're saying is stick with this in here as the first entry](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=138.99) [in the slice, remember? Slice is a 0 index,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=143.39) [so the first element is always item 0.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=146.61) [But then, stick this in at element 1, and then this at element 2.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=150.64) [Well, let's add this to print the contents of the slice to the screen,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=156.04) [and it's going to be ugly, but it'll do the job, for now,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=159.32) [we'll throw a loop in later to tidy it up.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=162.64) [I really want to focus at the moment to be on slices and not anything else.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=165.05) [Well, let's give it a whirl.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=170.44) [All right, so the same length and capacity, but now it's showing the entries.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=174.94) [But the thing is, we can make it look a lot tidier.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=181.34) [So instead of declaring the slice here and then initializing it later,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=184.94) [seeing as how we know the values now, let's just do it all in one go](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=188.54) [like this. Now interesting. What do you suppose the length and the](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=191.96) [capacity functions here are going to return now?](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=197.71) [I mean, we've just declared and initialized it with 3 values, but](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=200.54) [we've not specified a length and a capacity.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=204.78) [Well, my guess is we'll get 3s. But before we check, this was a bit ugly,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=207.94) [wasn't it?](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=213.25) [I'd rather see each element on its own line, so we'll do this. Recognize](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=213.74) [that? You should, it's our old friend for range. And we're basically](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=219.32) [saying, range over our courses slice, and for each iteration of the loop,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=224.77) [assign the data value to i, and print i.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=229.62) [And then because Println automatically adds a new line each](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=233.74) [time it runs, we'll get three entries printed over three lines,](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=237.17) [it'll look a lot tidier, trust me.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=241.23) [Well, let's see.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=243.74) [Epic.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=247.54) [Oh and look, length is 3 and so is capacity. Brilliant. Well, let's go and dig a little bit deeper.](https://app.pluralsight.com/course-player?clipId=0c518831-2e5a-4bfc-a405-32b4f558d9d7&startTime=248.16)

### [Getting Under the Hood](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289)

[We've already seen a bit of this in the last clip,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=1.54) [but we can manipulate slice elements by whacking the](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=4.44) [element's index value inside of square brackets.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=8.25) [In fact, remember this.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=12.34) [So, these are the element's index values,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=14.44) [and we populated them with these data values.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=17.28) [Okay, well,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=21.44) [I'm switching to the elements.go file in the slices folder of the GitHub repo.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=22.17) [Alright,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=26.64) [we've got a slice of 10 integers and a Println to put them on the screen.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=27.07) [I reckon this will look tidy enough without putting a loop in.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=31.54) [However, if we put an element number in here like this,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=35.34) [it will only print the value at index position 4,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=39.47) [so let's test it.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=42.62) [Okay, 5, but we wanted element 4.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=49.14) [Now, I'm only messing with you really,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=54.4) [and I'm sure some of you know what's going on.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=57.29) [But just for clarity, like all good computer systems,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=59.36) [Go starts counting at 0, not 1, meaning our index positions are like this.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=63) [So index 4 is data value 5.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=70.09) [And again, I know it's not rocket science, but it is easy to forget.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=74.6) [And when you do, and most of us do from time to time,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=79.13) [but when we do, it can be properly frustrating.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=82.5) [But, just to keep you on your toes, and this makes perfect sense actually,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=87.66) [but functions like len and cap, they obviously start counting at 1.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=92.59) [I mean, we can't have a slice with one value, but then](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=97.43) [have len return its length as 0.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=100.66) [I mean, that would be proper madness. Anyway, as well as referencing elements](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=103.22) [for printing, we can obviously manipulate them as well.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=109.15) [So, this here is going to change the value of element 1 to be a 0.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=112.48) [Now remember,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=118.11) [element 1 is actually the second element, so it's currently holding a 2.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=119.29) [Well,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=125.04) [we'll stick a Println here to verify the change, and let's take that for a spin.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=125.23) [Okay, element 1, now holding a 0.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=131.74) [Well, we can also slice a slice.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=136.94) [In fact, that's how we make a slice smaller.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=140.33) [So to do that,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=143.34) [we create a new slice, and we tell it to point to a](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=144.22) [subsection of the existing slice. Here,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=147.5) [we've got a new slice called sliceOfSlice, and it's basing itself](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=151.3) [off of values 2 through 5 from mySlice. Though, and I'm starting to](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=155.3) [think slices have quite a few quirks,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=161.45) [but the way Go works is it will use values 2 through 4,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=163.74) [so not 2 through 5.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=167.84) [I know. Basically, the first number that we specify is](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=170.94) [inclusive, so we'll be including element 2,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=175.55) [but the last number isn't included, so it'll stop at element 4.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=178.29) [And by my reckoning, we'll get 3, 4, and 5,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=183.98) [which will be elements 2, 3, and 4. Well,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=189.21) [let's whack in this Println, and we'll see if we're right. Bingo!](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=193.11) [Anyway, a couple of points on the syntax of slicing. When creating](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=199.7) [slices or even actually referencing elements, if we don't put](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=206.11) [anything before the colon operator here, index 0 gets implied at](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=209.92) [the start of the slice.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=214.44) [So this here will be 0 to 5,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=216.84) [and it's the same for emitting the value after the colon.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=220.44) [So in this instance, the end of the slice is implied.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=223.59) [Also, just one last quick reminder that passing slices to functions,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=227.34) [although it actually passes by value,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=231.47) [because the slice itself is a reference to an array,](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=234.43) [it's effectively passing by reference. And I reckon that'll do. Next up, we'll see how to expand a slice.](https://app.pluralsight.com/course-player?clipId=ff701c81-f618-41cb-8668-d96db3dab289&startTime=237.25)

### [Appending to Slices](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421)

[I think I mentioned it, probably at the top of the module,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=1.54) [that a major benefit that slices have over arrays is they can grow and shrink.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=5.07) [Well, to grow an array, we can use the built‑in append() function,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=10.16) [and it works like this.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=15.24) [Go take the current capacity of the underlying array,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=16.57) [And for each append(), it adds data to the next slot in the array.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=20.34) [So if we've got a slice here with the length of 1,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=24.94) [but a capacity of 4 and then we add a value,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=27.76) [it'll go to the second element of the array below,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=30.97) [if we add a third, it'll go here, and a fourth here.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=33.35) [But as soon as we add a fifth, Go doubles the size of the array below,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=37.14) [so we'll go from an array with four slots to one with eight.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=42.14) [Only, I don't know, Nigel, I thought you said arrays can't be expanded?](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=45.61) [True.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=50.04) [So Go is actually doing a bit of magic in the background,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=51.24) [creating a new backing array of double the size,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=54.58) [copying over any values,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=58.31) [and doing anything else that needs to make the whole process seamless,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=60.1) [and we don't even notice.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=63.87) [And that's the crack,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=67.24) [fill up the current backing array until it's full. Once we outgrow it,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=69.11) [Go performs a little bit of background magic and doubles its size.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=73.32) [Well, I think it's time to put that to the test with a bit of code.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=78.24) [All right, let's step through that.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=83.24) [Obviously, we're making a slice here with the length of 1 and a capacity of 4,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=85.44) [and then this line here is printing the info to the terminal,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=89.23) [basically just proving our starting point.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=92.12) [In fact, we'll put up here what we're expecting to see.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=94.51) [Anyway, we create a loop here that'll iterate 17 times.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=98.06) [And as we can see, for each iteration,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=104.36) [we're appending a new element to the end of the array,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=107.56) [but we're also printing this line here saying the](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=111.02) [current length is X and the capacity is Y.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=114.16) [So, for the first run through, we'll see array length is 2, but capacity is 4.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=119.24) [Next time, it'll be 3 and 4 and then 4 and 4.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=124.29) [But as soon as we append the fifth value goes,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=127.54) [Go is going to work its magic to double the underlying array size,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=131.46) [which is basically the capacity of the slice.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=135.58) [So, on the screen, we'll get length is 5, but capacity is 8.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=138.62) [Well, we'll rinse and repeat until we get to 9,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=144.74) [at which point, Go should up the array to 16,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=146.86) [more rinsing and repeating until we reached 17, and it should be doubled to 32.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=149.71) [Well, everything looks good on the slides.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=156.14) [Let's see if it actually works like that.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=158.34) [(Laughing) As if there was ever any doubt.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=162.34) [Well, you know what,](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=167.94) [take a second to look at that. But it's essentially what we sketched out on](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=168.92) [the slides, appending to the existing array until we reach its capacity, then](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=172.78) [doubling its size, and rinsing and repeating.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=177) [Magic! Well, there's a couple of other things I want to point out before we do a recap.](https://app.pluralsight.com/course-player?clipId=3f25ffe0-dc40-4ada-bf5e-03210f08f421&startTime=180.44)

### [Miscellaneous](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809)

[All right, let's tidy up a few loose ends.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=1.54) [Simply referencing an array or slice like we do with a variable,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=4.04) [so here the slice is called mySlice, well,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=7.89) [that'll reference the entire thing, like every entry,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=11.3) [which isn't necessarily the same as other languages.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=14.37) [I know, like, for example, doing this in C only returns the first element.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=18.44) [But here in Go, it returns the entire thing.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=22.76) [Also, and I know I showed this earlier,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=26.54) [but for‑range loops are pretty much perfect for working with slices,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=29.35) [and you'll see them together, like, all the time.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=34.45) [But recapping, for‑range returns two values for every iteration of the loop,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=38.74) [so it returns the index value and then the data itself.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=44.83) [Now, actually, this might not be the best example here,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=49.64) [because it is a slice of integers and just might make it look confusing.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=52.85) [But the first run through is going to return index position 0 and data value 1,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=56.82) [second run, index 1 and data value 2,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=63.24) [eventually finishing with index position 4 and data value 5.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=66.27) [However, if you're only interested in the data value,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=72.24) [which is pretty common,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=75.77) [then you can disregard the index value by passing it to](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=77.32) [the underscore like we're doing here.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=80.1) [Well, next up, seeing as we just looked at append,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=82.84) [it is totally possible to append slices to slices.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=86.07) [Well, it kind of is.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=90.84) [You see,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=92.44) [the process doesn't actually append one slice to the other; it appends the](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=93.28) [values from one slice to the end of the existing one.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=97.11) [I mean, look, the result's the same,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=101.24) [but how it happens under the hood maybe isn't what the operation looks like.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=103.07) [But look, honestly, the result is the same.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=107.74) [Now, small print does apply, but it's all pretty obvious,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=111.24) [like you can't append slices of different types.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=114.85) [So, if you've got a slice of strings, you can only append other strings to it.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=117.83) [Anyway, we've got a slice called mySlice with a bunch of ints,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=123.94) [so let's create a new one,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=127.45) [and I know I've gone a bit OTT with the crazy names here.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=130.22) [But the point is it's all integers, and then we append to it like this.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=134.64) [And I get it; i totally looks like‑‑‑well,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=139.4) [actually, first, the formula for the syntax is the slice to append to,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=143.12) [followed by the slice to append from, with ellipses bolted to the end.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=148.06) [So, this one will append newSlice to the end of mySlice.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=153.84) [Only, like I said,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=158.54) [what it's really doing is it is appending the data values](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=160.11) [from newSlice to the end of mySlice.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=163.46) [But again, like I said, the mechanics don't massively matter.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=166.04) [We'll end up with a slice of, what is that, eight elements.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=169.34) [Well, let's throw another Println here to make sure.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=175.14) [But actually, before trying it out, let's quickly step through.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=179.14) [We are declaring and initializing mySlice here as a slice of 5](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=182.64) [integers with a length and a capacity of 5.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=187.09) [And then we're proving that with this line here. Then we're](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=191.24) [showcasing a for‑range loop that'll put the data value from the](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=195.12) [slice in the i variable and print it to screen.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=199.28) [So, we should get lines showing 1 through 5, and we'll get a new line each](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=203.74) [time because Println prints a new line each time it runs. Anyway, then we're](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=209.59) [creating a new slice called newSlice with these three values and appending it](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=215.31) [to the end of mySlice and printing this.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=221.11) [So, we should get mySlice NOW contains [1 2 3 4 5 10 20 30] and has](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=224.72) [a length of whatever and a capacity of whatever.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=232.78) [And I reckon the length will be 8 because of the 8 values, but](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=236.14) [the capacity will be 10, because when we went over the initial](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=240.21) [capacity of 5, Go will double it to 10.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=243.69) [Well, shall we see?](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=247.64) [Epic. I reckon, yeah,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=251.24) [I reckon that's exactly what we expected. Now, maybe you might want to](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=253.98) [pause the video here and just double‑check that,](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=258) [especially if you're following along in your own code.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=260.58) [But yeah, that looks right to me. Anyway, look, next up, quick recap.](https://app.pluralsight.com/course-player?clipId=c331c1be-93ec-4c2f-8c26-f45e9c40d809&startTime=263.24)

### [Recap](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f)

[All right.](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=1.64) [We started out by saying that when it comes to arrays and slices all the action,](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=2.51) [and I do mean, all the action is with slices.](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=7.44) [They are both lists of a single data type, but arrays have got a fixed capacity,](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=12.34) [whereas slices don't. The long and short,](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=17.31) [we can append two slices to make them bigger, and we can](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=20.21) [slice slices to make them smaller.](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=22.91) [The moral of the story, 99% of the time, use slices, not arrays.](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=26.03) [That being said,](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=32.54) [we did dig into the theory a bit and slices are just a bunch of](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=33.59) [pointers that point to the real data stored in arrays.](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=37.01) [But of course, it's all done cleverly, so we basically never need to know.](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=40.34) [We just act on a slice like it's a flexible array. Though, I do think that](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=44.63) [knowing the inner workings helps you understand things like why if](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=49.84) [multiple slices point to the same array elements,](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=55.36) [changing the value in one slice impacts all the](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=57.68) [others pointing to the same elements.](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=61.1) [We also said that the for range loop is the perfect companion for](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=64.24) [slices. Like it iterates over a slice with the loop exiting once the](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=68.34) [end of the slice is reached. And honestly, you'll see the two of](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=73.39) [them used together all the time. But I reckon that's it, and I'm keen to get on to the next topic, maps.](https://app.pluralsight.com/course-player?clipId=475d137d-5928-48e2-be17-6d6599deb46f&startTime=75.95)

## [Working with Maps](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927)

[All right,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=1.74) [maps in Go. And if you've been following along, they're going to look a bit](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=2.37) [familiar because they're quite a bit like arrays and slices,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=6.3) [but there are a couple of differences. I think](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=9.21) [first and foremost, maps are, well,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=12.5) [maps are key sorted lists, or at least, sometimes they are.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=16.78) [Uh, what's that, Nigel? Sometimes? Oh dear, go on, do tell.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=22.14) [And you're going to see this as we go, but, retrieving a map](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=28.14) [without a loop, Go's going to return it in key‑sorted order. But](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=31.78) [when iterating it with a range loop, ha, you get them back in](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=36.15) [random order, or at least, there's no guarantee the order you'll](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=40.44) [get them in. And I get it right, if that sounds a bit confusing,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=44.47) [I don't blame you,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=47.85) [but it will be crystal clear by the end of the module. Anyway, another](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=49.14) [difference with slices is that maps are key‑value pairs. And if you've](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=54.09) [programmed in other languages, you might think hello,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=58.79) [you look like,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=61.77) [I don't know, a dictionary or maybe an associative array](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=62.89) [or a hash table. In fact, yeah. Maps are basically Go's](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=65.82) [implementation of a hash table.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=69.18) [So, as a quick example, and I know, I'm getting ahead of myself a bit,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=71.94) [but I do like seeding ideas before diving in,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=75.9) [but this is a list of some of the bigger football teams in England and](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=79.74) [how many times each one has won the top league. Now,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=83) [it's not a full list and it's not in order or](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=86.79) [anything, it's really just a sampling.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=88.74) [Well, the column on the left is the key, and the one on the right's the](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=90.74) [values. And then, we can dynamically update stuff like adding, and](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=94.5) [removing, and even updating existing entries.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=99.83) [So, yeah go on, let's get rid of Newcastle here and we'll throw in Man City.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=102.03) [And then, what about predicting the future?](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=106.83) [Okay, that's unlikely to happen if you know anything about English football,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=111.04) [but as far as maps go, heck, yeah, that's all doable.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=115.07) [We'll look back on track, here's the plan.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=120.14) [Well look at the basic syntax and getting our hands on with some basics.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=122.64) [We'll look at iterating over them with range loops,](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=126.37) [we'll see how to insert and delete elements, we'll fill in a](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=130.24) [few of the gaps, and finish on a recap. Well, let's crack on.](https://app.pluralsight.com/course-player?clipId=9207978e-99cd-4727-9a92-2f1302347927&startTime=133.38)

### [Getting Started with Maps](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d)

[First up, let's see how to define maps, and like I've said already,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=1.34) [they're a bit similar to slices.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=5.8) [Well, then so are the ways that we declare and initialize them.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=7.98) [Anyway, the basic syntax is map, and then because Go's strict about](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=12.04) [types, we'll declare the key type in square brackets like this and](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=16.16) [then the value type right after it.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=19.45) [Now, a couple of things. The key type has to be a comparable type.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=22.94) [So, a type that can be compared with either the equals or not equals operators,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=27.14) [basically, that's bools, strings,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=33.14) [numeric data types, and pretty much any of the other types like arrays](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=35.35) [and structs, so long as they are composed of bools,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=39.37) [strings, and ints; however, it does not include non‑comparable types,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=42.7) [so, not slices and not functions. As well though, and this one's](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=48.38) [pretty obvious, but key entries have to be unique.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=53.72) [You cannot have two entries in the same map, with the same key.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=58.09) [All right look, enough with the theory. I'm in the titles.go](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=63.34) [file in the maps folder of the courses GitHub repo, and we can](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=67.22) [use the make function like this.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=70.26) [So, this is creating a map called leagueTitles with the key](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=73.74) [type as a string and the data type int.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=78.23) [Now, it's only declaring the map, so it's not initializing any](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=81.34) [values. To do that, we can go with something like this.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=84.96) [So we reference the map, populate the key, a string here,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=89.74) [then the value is an int.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=93.41) [So, Sunderland have won 6 English league titles.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=95.29) [That's actually 6 in the all‑time list by the way. Anyway,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=99.01) [look, same again for Newcastle only, they've only got 4 titles,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=103.05) [which again, if you know much about English football, that is to be expected.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=106.94) [They're not as old as Sunderland, and they don't have as rich a history. Anyway,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=110.98) [as with slices, we can declare and initialize them in a](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=115.84) [single go with the composite literal form.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=119.16) [So, this time we're calling the map recentHead2HeadWins, any Newcastle](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=122.94) [fans are not going to like me. The map keyword again,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=128.2) [we'll have strings and ints again, and then a couple of entries.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=131.53) [Now this time, we're listing the number of wins that each team](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=134.77) [has from the last seven clashes.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=138.12) [Obviously, there's only one winner there, Sunderland](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=140.16) [with 6, Newcastle, oh dear, none.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=142.94) [The other one was a draw.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=145.74) [The thing is though, look,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=147.74) [we'll put a Printf here to display then, and let's give that a save and a try.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=149.24) [All right, the formatting is a bit nasty, well,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=157.34) [so are the stats,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=160.23) [if you're a Newcastle fan. For now though, we only really care that we](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=161.16) [have defined and populated two maps and we can see them.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=165.51) [Oh, but look, see how it's returned Newcastle first both times,](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=170.24) [even though I think, yeah, we've actually added Newcastle second both times.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=174.66) [So that is Go returning them in key order and for Newcastle coming before S for](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=179.89) [Sunderland in the alphabet. But that's the basics. Up next, we'll see how to use a range loop to iterate over them.](https://app.pluralsight.com/course-player?clipId=c0c2745b-72e3-485a-ad70-06b2d690536d&startTime=185.64)

### [Iterating Maps](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3)

[So here we go.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=1.74) [We just saw that Go returns maps in a key‑ordered fashion](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=2.67) [when retrieving them outside of a loop.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=7.65) [Fair play.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=9.81) [But instead, it gets all random if we use a range loop.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=11.14) [Well, to show that,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=15.54) [I am in the maploop.go file from the maps folder in the GitHub repo,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=16.87) [and we're looking at a simple strings to int,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=22.22) [so mapping capital letters to their position in the alphabet.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=25.56) [Well, let's throw in a quick range loop,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=30.34) [and like with slices, the loop executes for each key in the map.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=33.09) [And then for each iteration, it returns two values, the Key and the Value.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=38.5) [So here,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=43.64) [we're assigning the key to a variable called mapKey](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=44.23) [and the value to one called mapVal,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=46.82) [and then foreach loop will print the values to the screen.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=49.18) [Now,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=52.74) [this %v format verb here is saying, print these two variables, but](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=53.34) [instead of printing them as base‑10 integers,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=58.87) [like we've seen with %d already, well, this time, just go](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=60.94) [with whatever the default type is. Anyway,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=64.5) [we should get nine lines saying Key is whatever and](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=67.45) [then Value is whatever, and as well,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=70.65) [if we run it a few times, we should see different ordering.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=73.98) [Remember, Go doesn't guarantee iteration order with range loops. Fair enough.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=76.89) [Let's go and see if it's actually true. Yeah,](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=83.42) [starting at a different offset pretty much every time.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=94.01) [Anyway, up next, we'll have a crack at inserting, updating, and deleting entries from the map.](https://app.pluralsight.com/course-player?clipId=5aa7ecda-11a2-471f-816b-2debd97beee3&startTime=97.82)

### [Updating Maps](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082)

[Okay.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=1.64) [And again, this is the same as with slices, but we reference](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=2.49) [individual map entries with square brackets.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=6.78) [So sticking with the same code file as before, if we lose the](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=10.34) [loop here, and then put this Println in.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=13.53) [Well, like this,](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=16.94) [it will print the entire map, but if we do this, it'll just print C. Well,](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=18.24) [actually, it'll print the value associated with C, so 3. We're basically](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=24.26) [saying, print the value associated with the C key.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=29.02) [Let's give it a try.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=33.84) [Okay, magic.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=37.74) [Well, let's say we want to update a value, maybe change A to 100.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=39.34) [Well, here we can just say testMap "A" equals 100. And don't](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=46.04) [forget the quotes, yeah? The compiler needs to be sure of the](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=52.98) [type, but that will change A to 100.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=56.76) [Well, I'll tell you what,](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=60.44) [let's change this back here to return the entire map, and let's go again.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=61.18) [Magic.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=67.74) [A: 100. And go on, what the heck. Yeah, running it a few](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=68.36) [times, see how it's always ordered by key.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=74.19) [Anyway, if we want to add a new item,](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=79.04) [it's the same syntax. Just call the name of the map, put the key](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=81.31) [in square brackets, and assign it a value.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=84.86) [Now,](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=87.59) [if the key already exists, like it did actually when we just](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=88.4) [updated A, it'll get this here as its new value.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=91.08) [But, if it doesn't exist, the whole thing gets created.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=93.99) [Key and value.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=97.13) [Well, let's take that for a spin.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=100.24) [All right, J equals 1973. And again, Go maintaining its ordering by key.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=104.34) [Well, to delete an element, we use the built‑in delete](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=112.44) [function, and inside the operands, we just name the map](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=115.14) [and tell it which entry to lose.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=118.28) [Okay, so we're updating A, adding J, and then printing this to the screen.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=121.14) [Then we're deleting J, so let's add another Println here to prove it's gone.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=127.04) [So we'll get one line with a J and then one line without.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=130.97) [Let's have a look.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=134.78) [All right, nice one.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=139.44) [There it is in the first line, but it's gone in the second. Right, I've got a few more things I want to show you before we do a recap.](https://app.pluralsight.com/course-player?clipId=69496522-43f8-46c9-8e8c-4e03608e4082&startTime=140.72)

### [Miscellaneous](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e)

[Okay, just a couple of quick things before a recap. Like we just saw with slices,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=1.54) [maps are reference types.](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=6.79) [This means when we pass them to functions,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=9.01) [they get passed by reference. Which in turn means, any](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=11.27) [changes a function makes to the map, well,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=15.15) [they're obviously visible to the caller, but also any other functions.](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=18.2) [In layperson terms, if a function modifies a map,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=22.94) [it really does modify it,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=26.57) [like it's not acting on a copy or anything. No,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=28.06) [functions modify actual map data.](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=31.19) [Well,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=35.14) [a knock‑on effect of this behavior is that maps are cheap like you can](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=35.67) [have a huge old map, but passing it around, cheap as chips. Because](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=39.26) [remember, you're not really passing the map,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=43.64) [are you?](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=45.41) [You're only passing pointers. And on the topic of performance actually, it is](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=45.91) [often considered a good practice to specify the size of a map,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=51.28) [especially if it's a big one.](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=55.09) [I mean, you know have to, and Go can resize them on the go,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=56.73) [but there is a cost to re‑sizing large maps. As well right, and](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=60.26) [it's probably out of scope for this course,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=68.81) [but maps are not thread safe,](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=71.09) [so they are not safe for concurrency. Basically, it's not defined](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=73.88) [what happens to them if written to simultaneously. But you know what, that is enough of that, time for a quick recap.](https://app.pluralsight.com/course-player?clipId=a389dbf4-1e14-440b-a857-68b472068d4e&startTime=78.97)

### [Recap](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a)

[Then, what did we learn?](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=1.54) [Well, first up, we learned how to declare and initialize maps, so we can use](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=3.84) [the make keyword pretty much the same as we do with slices.](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=9.34) [And as Go is strongly typed, we've got to tell it the type](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=13.24) [of the key and the type of the value.](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=17.22) [Anyway, then we confused things over ordering.](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=21.24) [So, as of Go 1.12, which is kind of a while ago, but maps are basically key](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=23.93) [ordered unless we're retrieving them through a range loop.](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=30.13) [We also learned how we can insert,](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=34.84) [update, and even retrieve individual elements using the name of the map](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=37.53) [followed by the elements key in a set of square brackets, and we can use](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=42.18) [the built‑in delete function to delete elements.](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=47.24) [Well, I reckon that's probably it. Oh, no, actually, look, let's not](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=51.64) [forget we just learned that maps are reference types. So when they get](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=54.72) [passed functions, they get passed by reference and not value. And now,](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=58.79) [we definitely are finished. Though, next up, structs, proper cool stuff.](https://app.pluralsight.com/course-player?clipId=363486be-64fb-4211-adeb-408e34281d3a&startTime=63.95)

## [Working with Structs](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1)

[Okay, we've seen a few of the different types of Go supports.](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=1.74) [Obviously, we've worked a bit with strings and ints. But arrays and](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=4.9) [slices, and we've just done maps, these are all types as well. So, strings](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=9.63) [and ints are like the basic types, then pointers,](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=15.31) [slices, maps, even functions, actually, are reference types. And I know, right,](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=18.28) [we've mentioned that as we've been going. But I don't](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=23.54) [actually think we have for functions.](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=26.19) [I mean, I know we've done functions, but we've really](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=28.06) [only scratched the surface. Anyway, despite all of these types,](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=31.29) [there's still going to be times when you need something a bit more](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=35.96) [specialized, and this is where structs come into play.](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=38.57) [Anyway,](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=41.81) [we'll go about the module like this. We'll wrap our heads](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=42.31) [around the high‑level stuff. Actually,](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=45.46) [then we'll give a mention to object‑oriented programming from a Go perspective.](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=47.33) [Then we'll get our hands on defining structs and seeing them in action, and then we'll finish with a recap. Come on.](https://app.pluralsight.com/course-player?clipId=3379db67-fad9-4e3f-a269-e90a6727f9f1&startTime=51.73)

### [What Is a Struct?](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36)

[Right then, at a high level, structs are how we define custom data types,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=1.54) [and like I said,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=6.15) [they are the bizzo when the standard data types don't quite do what you need.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=7.11) [Anyway, the classic example for needing a struct is a geometric shape,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=12.93) [like, I don't know, a circle.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=17.14) [Well, clearly, none of the standard types are fit for a circle.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=19.54) [Like, to accurately describe one, we'd probably want properties like radius,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=23.84) [diameter, and circumference.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=28.13) [Well, halle‑freakin‑lujah,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=30.64) [structs let us define a custom type with all of those fields,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=33.03) [and we can even call it a circle if we want. Then, when that's](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=36.7) [created, we can reference it or create variables like we do any](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=40.86) [other type. Now, yeah, I know we don't need all of those values for a circle.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=45.39) [We can infer some of them. But, you get my point. What we're looking](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=51.57) [at right now is a custom type, perfect for defining and working with](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=55.04) [circles. Now then, just for clarity, although they actually are in this example,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=58.91) [the fields making up a struct do not need to be all of the same type.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=65.34) [Okay, anyway, the point is,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=71.04) [well, two actually. Firstly, this is only defining the type.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=72.61) [We're not instantiating any variables here.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=77.01) [So, we're basically saying, hey,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=79.41) [Go, define me a new type called circle with all of these](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=81.41) [attributes, but don't actually create me one yet.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=84.44) [Well, then, the fields that comprise it, yeah,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=88.44) [they're named and they have set types, but we can pretty](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=91.55) [much have as many of them as we need.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=94.65) [So, like, I don't know, if we created a struct to define a](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=97.84) [new type called person, we could have, well,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=102.38) [honestly, loads of fields, but things like name and age, preferred pronouns,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=105.24) [eye color, hair color, whatever.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=109.8) [We've pretty much got free reign.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=111.98) [And I reckon that's the basics. Now,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=115.34) [we are going to get hands on with them in a second,](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=117.81) [but I do just want to make a quick side step to mention object‑oriented programming.](https://app.pluralsight.com/course-player?clipId=1252eaba-126e-4ea7-b8ca-d7d4e3163d36&startTime=119.95)

### [Object-oriented Programming and Go](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4)

[Okay, so a very quick sidestep here, and,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=1.44) [look, you know what, if you're not already into object‑oriented programming,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=3.81) [honestly,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=7.84) [just hit that forward button down there, save yourself a](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=8.58) [couple of minutes. Still here? Okay, well, look,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=11.13) [just in case you're not an object‑oriented programmer and you ignored](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=16.34) [my advice and didn't hit the forward button, well,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=20.28) [object‑oriented programming is pretty much an application of](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=22.96) [the old UNIX philosophy of creating smaller,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=26.68) [modular, reusable components, objects, yeah,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=29.83) [that specialize in doing one thing,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=32.84) [but then bringing lots of those specialist objects together to form a useful](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=35.44) [application. Anyway, if you've got experience with OO,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=39.74) [then I'm sorry to say, Go does not have an object type, and,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=44.32) [actually, it doesn't have a class keyword either.](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=49.36) [So, while structs might look like we're heading towards OO with Go,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=52.84) [I'm really sorry, we're not. Because they're just not objects in the full sense.](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=57.92) [I mean, yeah,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=63.09) [they're a bunch of named fields and you can associate methods with them,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=64) [but they don't support inheritance.](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=67.02) [So, no true objects,](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=70.34) [no classes, and no inheritance. Basically, no OO with Go, and that's by](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=72.38) [design. Now, yeah, like all things, you can be clever and make Go do things](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=77.43) [it was never designed to do, but I wouldn't recommend it. If you're going to](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=83.19) [be using Go, honestly, leave whatever OO you can behind you, and work with](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=87.1) [Go the way it was intended.](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=92.25) [But look, I'm waffling. Structs are not OO in Go, but they are quality, so let's get our hands on with them.](https://app.pluralsight.com/course-player?clipId=4c031618-5d3d-4807-911f-885c7c6283e4&startTime=95.04)

### [Defining Structs](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6)

[All right,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=1.64) [no more slides. And actually, I am in the struct.go file in the structs](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=2.14) [folder of the GitHub repo. Now to define a struct in go,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=6.64) [we can use the type keyword. Then, we give it a name.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=10.73) [Now look,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=15.14) [geometric shapes have been done to death with structs, so I'm](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=15.79) [going with Pluralsight courses. Though to be fair,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=19.58) [I've probably overdone them a bit in this course.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=22.45) [Anyway, look,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=24.71) [we're going to call our new type courseMeta. Then we go struct because](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=25.68) [it's going to be a struct. Now, actually, we can define new types as](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=31.29) [simple ints and strings and the likes if we want to.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=36.79) [So yeah, even though there's already an int type,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=40.04) [we can create a new int type that's a simple int as well. And the reason we do](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=43.36) [this sometimes is we can assign methods to custom types,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=48.24) [basically attaching a function to the type. Anyhow, look,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=51.88) [we slapped the fields inside a set of curlys, and of](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=56.65) [course, we make sure they have unique names.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=59.07) [Well, as we're going with metadata for Pluralsight](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=62.14) [courses, we're having author as a string, level as a](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=64.63) [string, and then rating as a float64.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=67.59) [Now we could have more, but that'll do for now. And now then, I know](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=71.54) [I'm over‑stressing this, but in case it's new to you,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=78.2) [all this is doing is defining a new type.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=81.26) [So it's called courseMeta, and it's got these properties, we're not](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=84.23) [actually creating a variable of this type yet.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=88.12) [Well, to actually do that, we can go either this or this.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=91.54) [Now,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=97.34) [both of these declare a new variable called](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=97.56) [gettingStartedWithK8s, but importantly,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=99.44) [instead of its type being an int or something,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=102.99) [it's of our new courseMeta type.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=105.62) [But again, these only declare the variable, so they](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=108.84) [initialize all the fields with default 0 values.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=112.19) [But there is a difference between the two, using the](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=115.29) [new keyword here yields a pointer.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=117.7) [Now then, we're not going to use either of those, but I am going](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=121.94) [to leave them on there just for reference.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=125.35) [Okay, the composite literal form looks like this.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=128.94) [So we're declaring a new variable again called](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=133.11) [gettingStartedWithK8s, but this time we're initializing it as well.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=135.66) [So why not, go on. From the top real quick.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=140.84) [We're defining a new type here called courseMeta with three named fields.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=144.05) [These are a couple of ways to create the variable of this new type,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=148.54) [but they only declare it and they don't initialize it.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=152.46) [Well, down here we are declaring and initializing in one shot.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=155.59) [Well, I'll tell you what,](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=161.64) [let's have a Println here to show its values and we'll give this a try.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=162.93) [Okay, not bad. Well, next up, we'll see how to work a little bit more intricately with structs.](https://app.pluralsight.com/course-player?clipId=a923ab23-8c46-4c27-b138-9ccb5303e7a6&startTime=169.04)

### [Working with Structs](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872)

[Right then. We referenced individual fields in a](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=1.54) [struct with the period operator.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=4.69) [So, I'll tell you what, let's change this Println here to be this.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=7.54) [So, we've got some texts saying Author of Getting Started](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=11.54) [with Kubernetes is and then we're referencing the author](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=14.62) [field from the new variable.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=18.17) [So, it is variable name,](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=21.34) [period, and then the field we're referencing. And remember, this is the](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=23.28) [gettingStartedWithK8s variable of type courseMeta,](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=27.87) [which is a custom type created as a struct. Boom.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=32.02) [Well, let's see if it works.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=37.95) [Ah, magic.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=43.54) [Obviously, we can use the period to more than just](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=45.34) [retrieve fields, we can change them as well.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=47.5) [So, probably not a good idea,](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=50.84) [but let's say the impossible happened and everybody fell out](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=53.1) [of love with the course. I don't know,](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=56.09) [maybe the rating plummets from 5 stars to 1.2, or something horrific like that.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=58.13) [Well, it's just like working with the other types we've been doing all course.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=64.04) [So if you've already declared an instance,](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=67.74) [you can use the equals operator to assign new values.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=70.07) [We'll put this Println here to show the new rating, and let's give it a run.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=74.44) [Okay, all good, or not so much actually,](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=81.04) [if you are me, the author of that course.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=83.73) [But look, that's it.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=86.11) [That is the basics of struct.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=87.33) [I mean, obviously, there is a ton more,](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=89.12) [we could spend all day and fry our brains on the stuff,](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=91.27) [but for a beginner‑level course, that is done. Time for a quick recap before moving on.](https://app.pluralsight.com/course-player?clipId=15916c4d-d4af-4e0f-b773-b9e9206dc872&startTime=94.95)

### [Recap](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a)

[Then we've seen a bunch of Go's built‑in types,](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=1.44) [ints, floats, strings, even arrays, slices,](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=4.43) [and maps,](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=7.01) [and that's only a few of them. B ut there's still loads of times](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=7.98) [when we need something a bit more specialized,](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=12.44) [and that's where structs to come into play.](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=15.09) [They are literally Go's way of defining custom types that we](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=17.13) [can use just as if they're regular variables.](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=21.11) [So, under the hood, a struct is a new type with an arbitrary number of fields,](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=25.74) [each field is named and has its own specific type, and each field can be of a](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=31.21) [different type; they don't all have to be the same.](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=36.59) [I think in our example, we use the type command to create a new type](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=39.94) [called courseMeta of type struct, and we used it to store details of a](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=43.67) [top quality Pluralsight course called Getting Started with Kubernetes.](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=48.33) [Though, actually, just to be clear,](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=52.14) [the type was called courseMeta, it had three fields for](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=55.42) [storing the author of the course,](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=59.32) [its intended audience level, and a customer rating.](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=61.34) [Then we created a variable called gettingStartedWithK8s, and the](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=64.34) [type of that variable, instead of it being a slice, or an integer,](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=69.2) [or whatever, it was a courseMeta type.](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=72.62) [Whew! Marvelous! Well, we also saw how to manipulate individual](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=77.04) [fields with the period operator, and I reckon that was about it, and](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=82.24) [honestly, that is the basics of structs.](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=87.67) [Next up, and oh my goodness,](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=89.81) [our last module, but it is a proper good one, concurrency. See you there.](https://app.pluralsight.com/course-player?clipId=ca474ae2-f58d-4780-ae24-eee24004fc3a&startTime=91.72)

## [Concurrency in Go](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b)

### [Module Intro](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b)

[Right then, our final, final topic of the course,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=1.64) [and it's a really good one and a real puller actually.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=5.51) [Like,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=9.02) [it's one of the major features that brings both developers and projects to Go.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=9.75) [But, you know what, it's also a topic that's easily misunderstood.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=14.84) [Anyway, as the title suggests, we're exploring concurrency in Go,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=20.44) [and we'll go about things like this.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=24.73) [First off, we'll figure out what concurrency is,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=27.74) [we'll explain how Go implements it,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=30.63) [and at some point we're going to compare and contrast it with parallelism,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=32.83) [because, yeah, for sure, they are related topics,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=36.81) [but they're absolutely not the same thing.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=40.14) [Anyway, we'll also get our hands on coding some simple examples,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=43.64) [and we'll cover off channels as we go.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=47.35) [Now then, looking at that, there's probably going to be a bunch of theory,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=51.64) [especially up top.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=55.53) [And I just ask that you stick with me on this,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=57.44) [because it really is important,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=59.55) [but you can also see we will be getting our hands dirty.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=61.42) [Okay, well, one last thing before we get going.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=65.44) [The aim of the game is to give you a decent picture of what](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=68.03) [concurrency is and how to start implementing it in Go.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=71.47) [But remember, we're only a beginner course,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=76.04) [so you're not going to walk away ready to implement](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=78.13) [concurrency in a major project at work, but,](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=81.15) [you definitely will be bootstrapped and ready to take your next steps. So, look, as per this plan, let's crack on with the theory.](https://app.pluralsight.com/course-player?clipId=56ac8bc3-91b2-4a12-9e17-ddefb19e227b&startTime=84.5)

### [Explaining Concurrency](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b)

[Okay.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=1.54) [So what even is concurrency?](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=2.17) [Well,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=5.54) [at the highest level it's about creating multiple](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=6) [processes, and I use the term processes lightly.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=8.89) [But, it is about creating multiple processes that execute](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=11.64) [independently. Now, while I do use the term process lightly,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=16.43) [I am very serious about the term independently.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=21.26) [Like, I do not mean simultaneously and I don't mean in parallel.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=24.35) [Yeah,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=30.54) [go on, maybe an analogy will help. Though, it is just an](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=30.87) [analogy, so don't take it too far. Anyway look,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=34.56) [most mornings when I start work,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=38.23) [I've got a bunch of things I need to do that day.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=40.58) [Like obviously, I've got project work to do, like editing](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=42.99) [videos and planning and testing labs and maybe finishing](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=46.01) [book chapters and the likes.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=49.47) [But, I've also got a bunch of admin stuff as well,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=51.74) [like emails and messages that demand answering.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=54.24) [So let's say I've got a couple of project‑related](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=58.04) [emails to start the day with. Well,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=59.9) [instead of firing off the first one and then putting my feet up](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=62.23) [and sitting around waiting for a response.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=65.17) [You know what?](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=67.54) [I'll just crack straight on with the next one and then when](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=68.16) [that's sent and I'm waiting for responses to them.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=71.12) [I don't know, yeah,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=73.49) [go on. Let's say I finished recording and editing a new module](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=74.41) [for a video course, so I set it off converting to an MP4 and](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=77.08) [uploading to Dropbox or whatever.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=80.58) [Well, while all that's happening,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=83.44) [I'll go and check through a chapter of one of my books that's been translated,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=86.05) [you know,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=88.95) [making sure all the images are still in the right places and the](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=89.43) [fonts all line up and all that kind of jazz.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=91.91) [Well, okay, I've got three things going here.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=94.94) [I'm waiting on responses to a couple of emails, I'm waiting on a video](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=97.79) [production and upload, and I'm reviewing a translated book chapter.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=100.95) [Well,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=106.24) [here's the difference between concurrency and parallelism. At no point, am](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=106.55) [I working on any of those three tasks at the same time. I'm not, like, I](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=112.04) [wrote one email at a time, and then only after I'd sent them, did I switch](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=117.65) [to the video editing and uploading.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=122.59) [And again,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=124.94) [only when I kicked off the upload did I switch to the book](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=125.6) [translation. And that people, is concurrency.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=128.62) [So instead of firing off the emails and sitting around waiting for responses,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=133.04) [which might not actually come until later in the week, by the way,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=136.47) [I cracked on with something else, the video work. And then, when](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=139.31) [that started producing and uploading, instead of hanging around](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=142.91) [again and waiting for that to happen,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=145.64) [which could be several hours with large videos, I cracked](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=147.59) [on with reviewing the book translation.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=150.73) [But remember, I'm just one person, so I only did one](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=154.44) [of those at any point in time.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=158.74) [If let's say I want to work on the book translation at the same time](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=162.04) [as writing emails and doing the video stuff, well,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=165.71) [either I need a major brain upgrade or I need to hire some help.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=168.74) [Now one last thing actually,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=175.34) [if I'm doing the video work or the book stuff and I get a](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=176.75) [notification that I've got an email response,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=179.55) [then heck yeah, I can break from what I'm doing and I can check that email.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=182.12) [But the point is, I do have to break from whatever it is I'm doing at that time.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=187.44) [Like, I can't do both at once. So, concurrency is](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=192.04) [dealing with lots of things at once,](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=197.49) [whereas parallelism is doing lots of things at once.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=199.65) [Well, let's have a quick look at how we sometimes achieve this in computer programs.](https://app.pluralsight.com/course-player?clipId=91b632ba-9c19-46be-a5a8-7057745d8b4b&startTime=205.44)

### [Concurrency in Computer Programs](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f)

[In order to understand Go's implementation of concurrency,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=1.54) [I think we need to be clear on a very quick bit of computer science.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=4.16) [So, computers have processors, and for this simple example,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=8.9) [actually, let's assume it's got a single core.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=12.73) [Old school, I know, but stick with me.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=15.75) [Well, then we've got an application, and when we run it,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=19.34) [the OS creates a process, which is basically the runtime instance of the program.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=22.46) [So it's the process that tracks memory, file handles, I/O, all of that jazz.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=28.24) [And also, the process starts out with a single thread,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=33.22) [threads being what actually run on processor cores and execute the program code.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=36.18) [Well, the program starts with a single thread,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=42.84) [though it can launch more,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=44.87) [but it is the job of the OS to schedule threads on cores.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=46.28) [Okay, magic, why are you telling us this, Nigel?](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=51.84) [Well, it is a common way of achieving concurrency,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=54.08) [and you know what, even parallelism, if you've got multiple cores.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=56.98) [So, for parallelism,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=60.69) [all the OS does is it runs different threads in parallel on different cores.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=62.3) [You know what, though, back to concurrency.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=68.84) [The orange thread here could be the first email I sent in the morning,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=70.88) [then this blue one could be the second, the green one,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=74.6) [the video processing, and then whatever this is,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=77.22) [plum?](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=79.62) [I don't know, but it could be the book translation.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=80.52) [So, four threads or four things on the go,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=84.34) [but only a single core in my head or on the processor,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=87.96) [so basically only one of these actively working at any one point in time.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=91.84) [And that's threads, and, yeah, they're pretty low level,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=98.55) [they're kind of complicated, they're actually a bit resource hungry,](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=102.71) [and if you're not careful, they can drag down performance.](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=106.37) [Well, it's just as well that Go doesn't use threads for concurrency. Wait, what?](https://app.pluralsight.com/course-player?clipId=964852ad-1652-46da-bf21-fb1fde27e31f&startTime=110.54)

### [Go's Concurrency Model](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb)

[Okay, hang on, Nigel.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=1.64) [What is the point of talking about threads if Go doesn't even use them?](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=3.21) [Well, explaining concurrency the way we just did with threads,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=8.14) [it's actually useful for understanding the concepts of concurrency.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=12.03) [But, Go does actually use threads; it's just,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=15.83) [it abstracts them through a lightweight construct layered on top.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=20.4) [Well, this lightweight abstraction is called a goroutine,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=25.74) [and if you know your computer science, they're a bit like green threads,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=29.17) [or they are,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=33.02) [at least from the fact that they're not scheduled by the OS;](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=33.63) [it's the Go runtime that schedules them.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=36.28) [Fair enough.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=39.64) [But why use this model instead of traditional OS threads?](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=40.84) [Well, a bunch of reasons actually.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=45.24) [So, goroutines are way lighter than OS threads.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=47.06) [I think the initial stack size, it's tiny,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=51.06) [like 2K, so way smaller than around a MB for OS threads.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=53.67) [Plus, they can grow and shrink as needed.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=57.91) [As well though, Go does all of the heavy lifting with goroutines,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=60.57) [and that is a mahoosive bonus.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=65.02) [Because it means you and I as developers,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=68.43) [we can just crack on with our app code; we don't need to](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=71.12) [learn about OS‑specific thread stuff.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=73.87) [As well though, switches are expensive with OS threads.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=77.17) [Well,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=80.91) [they're expensive with goroutines as well; it's just](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=81.49) [they happen way less with goroutines.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=84.03) [So Go layers goroutines on top of threads, yeah.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=87.84) [And then when a goroutine blocks,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=90.72) [like maybe it's waiting on network I/O or something,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=92.86) [no sweat.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=95.78) [Go just swaps it out for another goroutine, but running on the same thread.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=96.61) [And, I mean, yeah, of course there's overhead with doing that,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=102.8) [but it is massively less than scheduling a full new thread.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=106.6) [Whew, now, a footnote here.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=111.84) [Even with goroutines,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=114.1) [there's going to be block conditions needing a brand‑new thread.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=115.92) [It's just that happens way less often.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=119.3) [And as well, goroutines have faster startup times and,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=122.84) [thanks to channels,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=126) [multiple goroutines can easily and safely communicate and share data.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=127.06) [Whew, look, the long and short,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=133.24) [goroutines let us switch a lot of work on and off a single thread.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=136.17) [And I think I've said a few times,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=140.57) [they work threads to within an inch of their lives.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=142.3) [And you know what, due to the cost of threads,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=146.24) [if you can do something on a single thread,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=148.69) [I'm telling you, you absolutely should.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=151.16) [I mean,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=153.95) [way too many cycles get lost into the black hole of queuing and](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=154.41) [swapping threads on and off processor cores. Look,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=158.87) [when it comes to threads, less is definitely more.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=162.33) [Now then, as we wrap the theory, just in case you're interested,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=167.14) [the concurrency model used by Go is the actor model, or, if you like](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=171.17) [to sound technical and you want to impress people,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=175.72) [it's a modern implementation of the communicating sequential](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=177.88) [processes model, CSP, if you like your acronyms. Anyway,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=181.14) [though, in this model,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=185.55) [actors safely pass messages between each other via channels. Well,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=186.95) [in Go, actors are goroutines, and channels, well,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=191.48) [they're called channels.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=194.79) [So, I suppose we need to explain channels.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=196.84) [I'll tell you what, trying to keep this brief is harder than I thought.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=199.76) [But look, channels are just like pipes.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=202.7) [One goroutine puts data onto the channel, and another goroutine](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=205.16) [grabs it off, simple. And you know what, for now,](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=209.16) [I'm just going to leave channels there. Let's go and see how some of this stuff actually works.](https://app.pluralsight.com/course-player?clipId=073eaf6a-ee2b-433c-826a-1a0e1b42cedb&startTime=212.55)

### [Writing a Concurrent Program](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176)

[Right then.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=1.64) [We've got a simple program here,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=2.15) [package stuff, imports, a main function, and a couple of anonymous functions.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=3.95) [So as things stand,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=8.64) [execution comes in at main, then our first anonymous function gets called.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=10.02) [Though actually, I don't think we've actually mentioned anonymous functions.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=15.33) [Yeah, no, I don't think we have.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=20.04) [Well look, Go let's you create functions without names,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=21.54) [so nothing in the operands here. And then sticking an empty set of](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=25.39) [operands at the end makes it self executing. Anyway, program flow](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=29.01) [will come in, and we'll hit the sleep here. That will put the entire](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=34.36) [program to sleep for 5 seconds.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=37.85) [Once they're up, we'll print Hello to the terminal and drop back into main.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=40.24) [This next anonymous function will be called, that'll print Pluralsight to the](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=44.94) [screen, and then as there's nothing else left to execute, the program will](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=48.32) [exit. Net will get Hello Pluralsight. Now look,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=52.03) [I know it's far from special,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=57.68) [but it's program flow that we're interested in, so](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=59.18) [execution is top to bottom, one line at a time,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=62.77) [one function at a time.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=65.59) [So, when we hit the sleep here,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=67.02) [the entire program blocks and we do nothing for 5 seconds,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=69.64) [literally nothing.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=73.72) [Well, when the 5 seconds are up,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=75.84) [we're back in business stepping through one line at a time.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=77.8) [Well, I'll tell you what, before we do anything else, we should probably see it.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=81.34) [Okay, so this is waiting for the 5 seconds. All right.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=87.94) [Okay.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=92.98) [Tell you what, let's go and add some concurrency. And you know what?](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=93.42) [It could not be easier, just slap the go keyword here in front](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=97.51) [of each function, and it makes them into goroutines. And as](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=101.59) [we'll see, it totally changes flow.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=105.48) [So going from the top again,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=108.74) [we step through a line at a time until we hit our first anonymous function.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=110.74) [But, that's a goroutine now, so this time when we hit the sleep, only this](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=115.1) [goroutine blocks. So, flow picks up here with the next anonymous function,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=120.5) [which is also a goroutine, and that gets switched onto the thread and starts](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=125.08) [executing. And of course, it's fast, yeah,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=129.78) [like, it is not going to take 5 seconds,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=133.25) [that's for sure.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=135.94) [So, we'll get Pluralsight printed to the screen and that](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=136.71) [goroutine lets it. When the 5 seconds is up,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=140.01) [this one will resume execution so get back onto the thread and it will](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=143.5) [print Hello. Net result will get Pluralsight Hello.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=147.27) [Now there's one more thing we need to do before we can actually run this,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=152.14) [func main here is also a goroutine, it's the main one,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=155.55) [so as soon as it exits the whole program does, and for us, that means](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=159.74) [our main function is going to exit before any of this completes. Well,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=164.92) [definitely this one here with the 5‑second sleep. So, we need to use](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=169.7) [waitGrp from sync package so that our goroutines have a way to tell main](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=174.46) [when they're finished. So we'll import sync up here and then we'll add](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=178.81) [these lines here.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=183.33) [Okay, then we'll add 2 to the waitGrp, so one for each](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=187.04) [goroutine. This way main's going to wait until both report](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=190.6) [back before it continues and exits.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=193.51) [Anyway, look, that should make those good to go. So, remember to save](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=195.62) [your changes, and we'll give it a try. Now remember, we're expecting to](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=199.61) [see Pluralsight, then a bit of a pause, and then Hello. Okay, sleeping](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=203.98) [for a few seconds, and there we go.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=209.65) [Quality. So our code executed top to bottom, one line at a time](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=212.51) [until our first goroutine fired. At that point,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=217.56) [execution continued and a second goroutine was created for this](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=221.57) [function here, but it didn't get any time on the thread until the](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=224.68) [first one blocked on the 5‑second sleep. Well,](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=228.53) [as soon as that happened, the sleeper got bumped off the thread, and the](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=232.6) [second one got its moment. It punched out Pluralsight and it reported back to](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=235.58) [waitGrp that it was done. After 5 seconds was up, the first one here came back](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=240.76) [to life, printed Hello to the terminal, and also reported back that it was](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=245.51) [done. Then at that point, main continues. Well, there's no more code to](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=249.78) [execute and the program exited.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=254.67) [Brilliant. Though I know, not rocket science, but not bad at explaining the](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=258.24) [high‑level concepts of goroutines. And it shows a potential pitfall, our text](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=265.28) [came out back to front because of the delay in scheduling. So, you need to keep](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=270.69) [that in mind when working with concurrency. Anyway, let's take a look at channels.](https://app.pluralsight.com/course-player?clipId=5ea8f0d8-53f3-4fec-97c6-b7e148edc176&startTime=276.5)

### [Channels](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f)

[So we mentioned before that Goroutines use channels to safely share data.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=1.54) [Well, there's two types of channel, buffered and unbuffered.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=5.94) [And unbuffered channels can't hold data.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=10.64) [So any Goroutine putting data onto one blocks until there's a receiver on the](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=13.21) [other end. I guess kind of forcing synchronous behavior. Well, on the other](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=17.74) [hand, buffered channels, they can hold data, so a Goroutine can drop data onto](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=22.85) [it and crack straight on with whatever it was doing without having to care if](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=27.91) [there's a receiver on the other end.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=31.53) [Okay, marvelous.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=34.74) [Well, to create an unbuffered channel, we can use make with the chan](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=35.78) [keyword. And then, obviously, Go wants to know the type of data the channel](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=39.74) [is going to hold. Remember, Go is obsessed with types. But, you know what,](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=44.02) [for an unbuffered channel, that's it.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=49.34) [But to make it buffered, all we do is add buffers. So](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=52.24) [this one here will have 5, and because it's for int, it](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=56.32) [means it can hold up to 5 ints.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=59.87) [Well, like we said,](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=63.74) [the effect of a buffered channel is that Goroutines](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=65.04) [using it don't need to block.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=67.89) [In fact, actually, as long as the channel is not full, other Goroutines](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=69.84) [can come along after and can also put data on it.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=74.05) [So proper asynchronous behavior. Now, obviously, if a buffered channel is full,](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=77.18) [then, for sure, any Go routines wanting to use it are going to block until it](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=83.47) [frees up. And then, likewise, if a receiver is trying to grab data off of a](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=87.38) [channel and the data's not there yet, well that blocks until it is. And you know](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=93.14) [what, that's enough for a taster. Let's go and wrap the module.](https://app.pluralsight.com/course-player?clipId=2b9e7a13-f277-4193-bce6-4e47d4b9e60f&startTime=98)

### [Recap](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06)

[Well, that's the module.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=1.54) [In fact, if you've been following along, that's the course pretty much.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=3.02) [Anyway, concurrency. We learned that it's about spinning lots of plates,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=8.44) [so dealing with lots of things,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=12.63) [but it's not about doing lots of things at the same time.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=14.96) [That's parallelism.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=19) [So I think I said I can fire off a bunch of emails,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=21.14) [and while I wait on responses,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=24.65) [I can crack on with other stuff, maybe edit a video. Then,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=26.79) [if I reply to an email that comes in,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=31.26) [of course, I can park the video stuff and I can reply to the email. Then](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=33.44) [when I'm done with that, I pick up the video work again.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=37.23) [So yeah, like we said, we're only doing one thing at a time,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=40.64) [but we're actually dealing with lots of things.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=45.32) [Anyway, Go has native support for concurrency via goroutines, which are just](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=48.64) [regular old functions prefixed with the go keyword. Though,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=53.72) [you've got to be careful of the scheduling and timing impact that can](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=58.52) [have. Remember, we saw Hello Pluralsight printed as Pluralsight Hello](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=61.33) [because of some blocking and scheduling.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=67.01) [Well, under the hood,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=70.34) [goroutines are built on threads, but they're more](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=71.56) [lightweight and they are way easier to code.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=74.63) [They also use a construct called a channel to share data, and we just said that](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=78.24) [channels can be buffered or unbuffered, with the buffered kind allowing](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=82.27) [asynchronous behavior where senders can drop data onto the channel and crack](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=86.72) [straight on with whatever else they're doing.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=91.01) [Well,](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=94.94) [that's us just about done. Join me for one last super quick module to discuss where to go and what to do next.](https://app.pluralsight.com/course-player?clipId=0d6e5db1-2a0b-4422-9fa2-5829e3e7fb06&startTime=95.59)

## [What Next?](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72)

### [What Next?](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72)

[Right then, we're done.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=1.64) [Amazing, yeah? And you know what? Massive congrats if you've made it all the](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=3.4) [way through. You should be rightly proud of yourself. And you know what else?](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=8.59) [If you've been with me from the start, honestly,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=13.88) [thank you.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=16.7) [I genuinely appreciate your time.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=17.65) [But as well, looking at the stuff we've covered,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=20.64) [you should have a decent grasp of Go and some of this stuff that it can do.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=23.57) [But we've only scratched the surface, and I'm telling you,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=29.44) [there isn't much that Go can't do.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=33.47) [Now look, if you like what you've seen,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=37.24) [then honestly, check out all the other stuff we have got here](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=39.17) [on Pluralsight. We've honestly got loads, and we're adding more](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=42.74) [all the time. And you know what?](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=46.06) [It's not just video courses, I always recommend you get your hands](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=48.33) [on as much as possible, and then any tests, and self‑assessments,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=52.16) [and things that we provide, seriously,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=56.34) [you should do them and don't stress about them.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=58.41) [Just enjoy them.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=61.14) [They're always good fun.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=62.25) [Anyway,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=65.14) [away from Pluralsight, and I suppose aside from the obvious of](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=65.61) [get your hands on and hack a few of your own projects, but get](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=69.08) [involved with the community.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=72.51) [So things like conferences,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=75.14) [I love them, they're always a great way to learn](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=76.94) [new stuff and meet great people.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=79.18) [So maybe search for gophercon and golang conferences, and you know, what even](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=81.36) [golang meetups. There's probably a meet up close to you.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=86.36) [So yeah, check out our videos, and our hands‑on, and our tests, and the likes,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=91.54) [get yourself a project to take your skills to the next](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=95.45) [level, and get involved with the community.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=97.88) [And I reckon that's it.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=101.04) [So just a final thanks from me.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=102.31) [Honestly, I've invested a ton of effort into this course, so I hope you liked it.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=105.74) [If you're into the infrastructure side of things,](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=110.94) [then I highly recommend my containers and Kubernetes courses.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=113.2) [Just search my name on Pluralsight, I reckon you'll find me.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=117.44) [But yeah, honestly, it has been an absolute pleasure spending time with you. I'm Nigel, and have fun coding.](https://app.pluralsight.com/course-player?clipId=bcbeaf93-9f5d-4033-8f2c-276aea23cb72&startTime=120.44)