

Chapter 1 : The Go Programming Language | Pluralsight

Go is an open source programming language that makes it easy to build simple, reliable, and efficient software. Download Go Binary distributions available for Linux, macOS, Windows, and more.

Go provides two features that replace class inheritance. Go interfaces were designed after protocols from the Smalltalk programming language. Since conformance to a Go interface is checked statically by the Go compiler except when performing a type assertion, the Go authors prefer the term structural typing. Any object of type T for which functions exist matching all the required methods of interface type I is an object of type I as well. The definition of type T need not and cannot identify type I. For example, if Shape, Square and Circle are defined as: Interfaces can embed other interfaces with the effect of creating a combined interface that is satisfied by exactly the types that implement the embedded interface and any methods that the newly defined interface adds. The language constructs to do so are the type assertion, [63] which checks against a single potential type, and the type switch, [64] which checks against multiple types. It is similar to the Object class in Java or C and is satisfied by any type, including built-in types like int. Reader is public but bzip2. Concurrency refers not only to CPU parallelism, but also to asynchrony: A function call prefixed with the go keyword starts a function in a new goroutine. The built-in switch-like select statement can be used to implement non-blocking communication on multiple channels; see below for an example. Go has a memory model describing how goroutines must use channels or other operations to safely share data. The actor style can be simulated in Go by maintaining a one-to-one correspondence between goroutines and channels, but the language allows multiple goroutines to share a channel or a single goroutine to send and receive on multiple channels. Unlike previous concurrent programming languages such as Occam or Limbo a language on which Go co-designer Rob Pike worked, [80] Go does not provide any built-in notion of safe or verifiable concurrency. Various studies have been done into the effectiveness of this approach. The study found that the non-expert tended to write divide-and-conquer algorithms with one go statement per recursion, while the expert wrote distribute-work-synchronize programs using one goroutine per processor. Specifically, unless a program explicitly synchronizes via channels or other means, writes from one goroutine might be partly, entirely, or not at all visible to another, often with no guarantees about ordering of writes. Composition and delegation are in fact largely automated by struct embedding; according to researchers Schmager et al.

Chapter 2 : Go Programming Language, The | InformIT

The Go Programming Language Go is an open source programming language that makes it easy to build simple, reliable, and efficient software. Gopher image by Renee French, licensed under Creative Commons Attributions license.

In October, hundreds of companies helped us understand how enterprises are using Go by taking the Go company questionnaire. These surveys and questionnaires have played an enormous role in driving changes to our language and community, from our new code of conduct , to our latest release Go 1. Today we are conducting the Go user survey. Please help us shape the future of Go by participating in the minute Go user survey by November 30th: [Go User Survey](#) This survey is confidential and anonymous. We need as many Gophers as possible to participate in this survey to help us better understand our global user base. In the subsequent years, the Go community has grown significantly and has settled on idiomatic patterns for cloud-based applications. Today, Google Cloud is announcing a new Go 1. Starting with Go 1. Context values, or HTTP clients. Write your Go application however you prefer, add an app. [Specifying Dependencies](#) describes how the new runtime supports vendoring and modules experimental for dependency management. [Main](#) , which is now entirely optional. Furthermore, the application code is completely portable—there are no ties to the infrastructure that your application is deployed on. If you need to use external dependencies, you can add those dependencies to a vendor directory or to a go. With the application code complete, create an app. Create an account with <https://cloud.google.com/sdk/> Install the Cloud SDK on your system. With all the setup complete, you can deploy using one command: `gcloud app deploy` There is a free tier. Check out the [getting started guide](#) or the [migration guide](#) and deploy an app to the new runtime today! This post goes into more detail about Wire, a dependency injection tool provided with Go Cloud. What problem does Wire solve? Dependency injection is a standard technique for producing flexible and loosely coupled code, by explicitly providing components with all of the dependencies they need to work. In Go, this often takes the form of passing dependencies to constructors: Replacing one implementation of a service with another can be painful because it involves modifying the dependency graph by adding a whole new set of dependencies and their dependencies In practice, making changes to initialization code in applications with large dependency graphs is tedious and slow. Dependency injection tools like Wire aim to simplify the management of initialization code. You describe your services and their dependencies, either as code or as configuration, then Wire processes the resulting graph to figure out ordering and how to pass each service what it needs. Why is this part of Go Cloud? Bucket can swap implementations without changing their application logic. However, the initialization code is inherently provider-specific, and each provider has a different set of dependencies. For example, constructing a GCS blob. Bucket requires a `gcp.HTTPClient`, which eventually requires `google.Credentials`, while constructing one for S3 requires an `aws.Config`, which eventually requires AWS credentials. Thus, updating an application to use a different blob. Bucket implementation involves exactly the kind of tedious update to the dependency graph that we described above. There are a number of dependency injection frameworks out there. We think this approach has several advantages: Runtime dependency injection can be hard to follow and debug when the dependency graph gets complex. Nothing is obfuscated by an intervening framework doing "magic". In particular, problems like forgetting a dependency become compile-time errors, not run-time errors. Wire uses Go types to connect components with their dependencies. How does it work? Wire has two basic concepts: Providers are ordinary Go functions that "provide" values given their dependencies, which are described simply as parameters to the function. DB based on some connection info. Build with the list of providers or provider sets that are needed to construct the end result: Wire helpfully tells us the line number and types involved. We can either add a provider for it to wire. Build, or add it as an argument: `Wire.Build(&wire.Builder{Providers: []Provider{...}})` There is no dependency on Wire at runtime: As you can see, the output is very close to what a developer would write themselves. How can I get involved and learn more? We appreciate any input you have about your experience with Wire! We need your help to create the best programming language for developing simple, reliable, and scalable software. To do this, we need to better understand how companies are using Go. Please help by participating in a 7-minute company questionnaire. We only need one response per company or per department for larger

companies. Please take this 7-minute questionnaire by October 30th: [Go Company Questionnaire](#) The questionnaire is confidential, but not anonymous. The Go project leadership will use your responses to better understand how companies use Go and in what ways we can improve their experience. We would like as many companies as possible to participate to help us better understand our global user base. Please help us spread the word by sharing this post on your social network feeds, at meetups, and in other communities. By [Ran Tao, Steve Francia](#) [Go 2 Draft Designs](#) 28 August Yesterday, at our annual Go contributor summit, attendees got a sneak peek at preliminary drafts of possible designs for changes to error handling and generics. The development of Go 2 was announced last year and we are excited to share updates with you today. For a quick overview, watch this short message we just played at Gophercon We invite everyone in the Go community to learn more about the designs and help us improve them. See the [index](#) for more articles. Except as noted , the content of this page is licensed under the [Creative Commons Attribution 3](#).

Chapter 3 : Go by Example: Hello World

Go (often referred to as Golang) is a programming language designed by Google engineers Robert Griesemer, Rob Pike, and Ken Thompson. Go is a statically typed, compiled language in the tradition of C, with the added benefits of memory safety, garbage collection, structural typing, and CSP-style concurrency.

This language led to the creation of softwares in a very easy and efficient manner. Want to know how? Well, there are quite a few reasons behind this. The best part is we will be discussing each of them here. It is a statically-typed language having the syntax similar to that of C. This language is also capable of providing garbage collection, type safety, dynamic-typing capability, many advanced built-in types such as variable length arrays, key-value maps and much more. Go programming language was created with the aim of combining ease of programming by interpreted and dynamically typed language, and with the full efficiency of statistically typed language as well. And to our surprise, they did so. The standard libraries prove to be helpful while coding. With the help of standard libraries, we can make the use of relevant functions in our programs. For better understanding, we can say its portable as well. It offers unit testing features such as a simple mechanism to write the unit test, parallel with the code because of this we can understand our code coverage by our own tests. As simple as that. Static Typing- Go has a static type functionality for its programmers. Applications of Go programming language Though Go is a new programming language but it is widely used in some of the popular areas of IT Information and technology. Netflix “ If you are into movies and shows then you must have heard about this website. The chances of your visit on this website are also very high at least once in life. And you will be surprised to know that the Go programming language is used in two of its server architectures. DropBox “ One of the popular file hosting services in the world dropbox is also powered by this language. Dropbox offers services like cloud storage, file synchronization, personal cloud, and client software. Red hat is a public cloud application development and hosting service which runs on AWS. And here too everything is powered by Go programming language. Resources and Links to Learn Go programming language Similar to any other language there are tons and tons of resources available to learn Go as well. You can learn it offline by taking numerous classes in your city or you can just opt for an online course. The main advantage of online courses is that they offer free tutorials. This facilitates you to learn without paying anything. So here we are listing some of the best available online sources.

Chapter 4 : Go (programming language) - Wikipedia

Go language is a programming language initially developed at Google in the year by Robert Griesemer, Rob Pike, and Ken Thompson. It is a statically-typed language having syntax similar to that of C.

Chapter 5 : Go Programming Language

The Go Programming Language is the authoritative resource for any programmer who wants to learn Go. It shows how to write clear and idiomatic Go to solve real-world problems. It shows how to write clear and idiomatic Go to solve real-world problems.

Chapter 6 : Documentation - The Go Programming Language

Go! is an agent-based programming language in the tradition of logic-based programming languages like Prolog. It was introduced in a paper by Francis McCabe and Keith Clark.

Chapter 7 : Go Programming language - For beginners

Go by Example Go is an open source programming language designed for building simple, fast, and reliable software.

Check out the first example or browse the full list below.

Chapter 8 : Go! (programming language) - Wikipedia

The Go Programming Language is a new programming language from Google. It is very different than many other modern programming languages, and is a great substitute for C or C++. In this course we will cover most of the Go programming language and look at some of the neat things we can do with Go.

Chapter 9 : Go (programming language) | Google Wiki | FANDOM powered by Wikia

*The Go Programming Language Alan A. A. Donovan Google Inc. Brian W. Kernighan Princeton University New York
â€¢ Boston â€¢ Indianapolis â€¢ San Francisco.*