

Welcome to the Go wiki, a collection of information about the [Go Programming Language](#). [Awesome Go](#) is another great resource for Go programmers, curated by the Go community.

## Questions about Go

See [Questions](#).

Please do not ask questions by editing or creating a wiki page. Any such changes will be silently reverted.

## Contributing

- This wiki is open to editing by any member of the Go community with a GitHub account.
- If you would like to add a new page, please first open an issue in the [Go issue tracker](#) with the prefix 'wiki' to propose the addition. Clearly state why the content does not fit into any of the existing pages.
- Because renaming of pages in the wiki can break external links, please open an issue before renaming or removing any wiki page.

## Table of Contents

- [Getting started with Go](#)
- [Working with Go](#)
- [Learning more about Go](#)
- [The Go Community](#)
- [Using the go toolchain](#)
- [Additional Go Programming Wikis](#)
- [Online Services that work with Go](#)
- [Troubleshooting Go Programs in Production](#)
- [Contributing to the Go Project](#)
- [Platform Specific Information](#)
- [Release Specific Information](#)
- [Questions](#)

## Getting started with Go

- [The Go Tour](#) is the best place to start.
- [Effective Go](#) will help you learn how to write idiomatic Go code.
- [Go standard library documentation](#) to familiarize yourself with the standard library.
- [Use the Go Playground](#) to test out Go programs in your browser.
- Still not convinced? Check out this list of [Go Users](#) and a few of their [Success stories](#). We've also assembled a long list of reasons [why you should give Go a try](#).
- Read about the companies which have [switched from other languages to Go](#).

## Working with Go

Ready to write some Go code of your own? Here are a few links to help you get started.

- Install and Setup your Environment
  - Start here: [Official Installation Documentation](#)
  - If you prefer to install from source, [read this first](#).
  - [InstallFromSource](#) - Additional tips on source installs.
  - Windows user? [Install and configure Go, Git, and Atom for Windows](#)

- Mac user? [How I start - Go](#) - A step-by-step guide to installing Go and building your first web service.
- Having installation problems? [InstallTroubleShooting](#).
- Make sure you have your [\\$GOPATH environment variable set correctly](#).
  - If you need additional tips on using [\\$GOPATH](#), [go here](#).
- [MultipleGoRoots](#) - More advanced information on working with multiple go installations and the `$GOROOT` variable.
- [Go IDEs and Editors](#) - Information on how to use your favorite editor with Go.
- [Tools for working with Go code](#) - Formatting, linting, vetting, refactoring, navigation, and visualization.
- Finding Go Libraries and Packages
  - Start here: [Go open source projects](#).
  - Search for Go packages: [pkg.go.dev](#)
  - Visualization of the [Go open-source package graph](#)
- [Modules](#) - documentation on the dependency management system built into the Go command, added in 1.11.
- [Managing your dependencies](#) - An overview of the tools you can use to manage third-party packages (vendoring).
- Publishing Go Packages as Open Source
  - Getting ready to publish your package? [Start here](#).
  - [The Go Checklist](#) - A comprehensive guide for publishing a project.
  - [How to layout your GitHub repo](#) to make it easy for other Go programmers to use with the `go get` command.
  - [Go Package, Go](#) - A few recommendations for making Go packages easy to use.

## Learning more about Go

Once you have an overview of the language, here are resources you can use to learn more.

- [Learning Go](#) - A collection of resources for learning Go - beginner to advanced.
  - [Best Practices for a New Go Developer](#) - Insights from Go community members.
  - [Server programming](#) - Building web, mobile, and API servers.
  - [More on concurrency](#)
  - [More on error handling](#)
  - [More on testing](#)
  - [More on mobile - Android and iOS](#)
- [Books](#) - A list of Go books that have been published (ebook, paper)
- [Blogs](#) - Blogs about Go
  - [Podcasts](#) - Podcasts and episodes featuring Go
- Videos, Talks, and Presentations
  - [GopherVids](#) (currently unreachable) is a searchable index of videos about Go.
  - [GoTalks](#) - A collection of talks from Go conferences and meetups.
  - [Livestreams](#) - live interactive Go project streams with members of the community
  - [Screencasts](#)
- [Articles](#) - A collection of articles to help you learn more about Go.
- [Training](#) - Free and commercial, online, and classroom training for Go.
- [University Courses](#) - A list of CS programs and classes using Go.
- [Resources for non-English speakers](#)

## The Go Community

Here are some of the places where you can find Gophers online. To get a sense of what it means to be a member of the Go community, read [Damian Gryski's keynote from the GolangUK 2015 conference](#) or watch [Andrew Gerrand's closing keynote from GopherCon 2015](#).

- Where Gophers hangout online:
  - [The Go Forum](#) - An all-purpose discussion forum for the Go community.
  - [Gophers Slack Channel](#) - For real-time chat ([request membership](#)).
  - [Golang News](#) - For curated links about Go Programming.
  - There is also a [/r/golang](#) sub-reddit.
  - On Twitter, follow the [@golang](#) account and keep tabs on the [#golang](#) hashtag.
  - We've also got a landing page on [Stack Overflow](#) for Go Q&A.
  - Matrix enthusiasts are invited to join [#Go:matrix.org](#).
  - Discord users are welcome at the [Discord Gophers](#) server.
  - Hashnode users talk and write about Go in [Hashnode Go community](#).
- Mailing Lists
  - The mailing list for Go users is [golang-nuts](#) - very high traffic.
    - Before you post, [check to see if it's already been answered](#), then read [these tips on how to ask a good question](#)
  - For discussions about the core Go open source project, join [golang-dev](#).
  - To get just our release announcements, join [golang-announce](#)
- User Groups & Meetups - There are [Go Meetups in many cities](#)
  - [GoBridge](#) - Volunteers helping underrepresented communities to teach technical skills and to foster diversity in Go.
  - [Women Who Go](#)
  - See here for [additional information GoUserGroups](#)
  - [GoDiscourse](#) - Go Discourse is an open-source Go-based forum from `hello world`.
- [Conferences](#) - A list of upcoming and past Go conferences and major events.
- [Companies using Go](#) - A comprehensive list of companies using Go throughout the world.
- Learn more about the [Go Gopher images](#) by Renee French.

## Using the Go Toolchain

- Start with the standard documentation for the `go` command [available here](#)
- Start here to learn about [vendoring](#).
- See also [PackageManagementTools](#) for package management tools.
- [Cross Compilation](#)
- Shared libraries and Go (buildmode)
  - [Go Shared Libraries](#) - Examples for creating and using shared libraries from Go and Python.
  - [Sharing Go Packages with C](#) - by [@ralch](#).
  - [Calling Go libraries from Python](#) - by Filippo Valsorda
  - [Calling Go libraries from Ruby](#) - by Peter Hellberg
  - [Calling Go libraries from Swift](#) - by Jaana Burcu Dogan
  - [Build a Ruby Gem with a Go native extension](#) - by @jondot
  - [gohttplib](#) - An experiment in using Go 1.5 buildmode=c-shared.
- See the wikis below for additional details:
  - [GoGetTools](#)
  - [GoGetProxyConfig](#)

- [cgo](#)
- [CompilerOptimizations](#)
- [GccgoCrossCompilation](#)
- [GcToolchainTricks](#)
- [GoGenerateTools](#)
- [Go Tooling Essentials](#) - by Jaana Burcu Dogan

## Additional Go Programming Wikis

- Concurrency
  - [Timeouts](#) - Abandon async calls that take too long
  - [LockOSThread](#)
  - [MutexOrChannel](#) - When to use one vs the other
  - [RaceDetector](#) - How to detect and fix race conditions
- Working with Databases
  - [database/sql](#) - Online tutorial for working with the database/sql package.
  - [SQLDrivers](#)
  - [SQLInterface](#)
- From other languages
  - [Go for Java Programmers](#)
  - [Go for C++ Programmers](#)
- Strings
  - [GoStrings](#)
  - [String Matching](#)
- [Comments](#)
- [CommonMistakes](#)
- [Errors](#)
- [GcToolchainTricks](#)
- [InterfaceSlice](#)
- [lota](#)
- [MethodSets](#)
- [PanicAndRecover](#)
- [Range](#)
- [RateLimiting](#)
- [SignalHandling](#)
- [SimultaneousAssignment](#)
- [SliceTricks](#)
- [Switch](#)
- [TableDrivenTests](#)

## Online Services that work with Go

If you're looking for services that support Go, here's a list to get you started.

- Cloud Computing - Go is well supported by most cloud service providers.
  - [Amazon Web Services](#)
  - [Azure](#)
  - [DigitalOcean](#)
  - [GE Predix](#)
  - [Google Cloud Platform for Go](#)

- [Heroku](#)
- [IBM Bluemix](#)
- [OpenStack](#)
- [Vscale](#)
- [Aliyun](#)
- [Tencent](#)
- See here for [information on additional providers](#)
- [Continuous Integration and Continuous Deployment](#) - Go is well supported by most CI/CD frameworks
- Monitoring/Logging
  - [OpsDash](#) - Go-based cluster monitoring platform.
- Package and Dependency Management
  - [Gopkg.in](#) is a source for stable Go libraries, provided by Gustavo Niemeyer.

## Troubleshooting Go Programs in Production

- Understand the performance of your Go apps using the [pprof package](#)
- Heap Dumps
  - [heapdump13](#)
  - [heapdump14](#)
  - [heapdump15](#)

## Contributing to the Go Project

- Start by reading the [Go Contribution Guidelines](#)
- If you'd like to propose a change to the Go project, start by reading the [Go Change Proposal Process](#)
  - An archive of [design documents is also available](#)
- Go releases happen at ~6-month intervals. [See here for more information](#)
- Want to know more about how the [Go source X-repositories are structured?](#)
- The Go project requires that all code be reviewed before it is submitted.
  - Read more about our [code review practices](#)
  - If you're commenting on code under review, please read [these guidelines](#)
- Issues
  - Bug reports and feature requests should be filed using the [GitHub issue tracker](#)
  - Want to understand how we [handle issues that are reported?](#)
- Project Dashboards
  - [Go Builds Dashboard info](#)
  - [Performance Dashboard info](#)
  - [Download failed logs and debugging](#)

## Platform Specific Information

- See [MinimumRequirements](#) for minimum platform requirements of current Go ports.
- Considering porting Go to a new platform? [Read our porting policy first](#)
- [Mobile](#)
- [Ubuntu](#)
- [Windows](#)
  - [WindowsBuild](#)
  - [WindowsCrossCompiling](#)
  - [WindowsDLLs](#)

- [GoArm](#)
- [ChromeOS](#)
- [Darwin](#)
- [DragonFly BSD](#)
- [FreeBSD](#)
- [Linux](#)
- [NativeClient](#)
- [NetBSD](#)
- [OpenBSD](#)
- [Plan 9](#)
- [Solaris](#)

Notes:

- Please refrain from changing the title of the wiki pages, as some of them might be linked to from [golang.org](http://golang.org) or other websites.