

Go 1.21 is released! - The Go Programming Language

Eli Bendersky, on behalf of the Go team 8 August 2023

The Go Blog

Today the Go team is thrilled to release Go 1.21, which you can get by visiting the [download page](#).

Go 1.21 is packed with new features and improvements. Here are some of the notable changes; for the full list, refer to the [release notes](#).

The Profile Guided Optimization (PGO) feature we [announced for preview in 1.20](#) is now generally available! If a file named `default.pgo` is present in the main package's directory, the `go` command will use it to enable a PGO build. See the [PGO documentation](#) for more details. We've measured the impact of PGO on a wide set of Go programs and see performance improvements of 2-7%.

The [go tool](#) now supports [backward](#) and [forward](#) language compatibility.

Language changes

New built-in functions: [min](#), [max](#) and [clear](#).

Several improvements to type inference for generic functions. The description of [type inference in the spec](#) has been expanded and clarified.

In a future version of Go we're planning to address one of the most common gotchas of Go programming: [loop variable capture](#). Go 1.21 comes with a preview of this feature that you can enable in your code using an environment variable. See [the LoopvarExperiment wiki page](#) for more details.

Standard library additions

New [log/slog](#) package for structured logging.

New [slices](#) package for common operations on slices of any element type. This includes sorting functions that are generally faster and more ergonomic than the [sort](#) package.

New [maps](#) package for common operations on maps of any key or element type.

New [cmp](#) package with new utilities for comparing ordered values.

Improved performance

In addition to the performance improvements when enabling PGO:

The Go compiler itself has been rebuilt with PGO enabled for 1.21, and as a result it builds Go programs 2-4% faster, depending on the host architecture.

Due to tuning of the garbage collector, some applications may see up to a 40% reduction in tail latency.

Collecting traces with [runtime/trace](#) now incurs a substantially smaller CPU cost on amd64 and arm64.

A new port to WASI

Go 1.21 adds an experimental port for [WebAssembly System Interface \(WASI\)](#), Preview 1 (GOOS=wasip1, GOARCH=wasm).

To facilitate writing more general WebAssembly (Wasm) code, the compiler also supports a new directive for importing functions from the Wasm host: `go:wasmimport`.

Thanks to everyone who contributed to this release by writing code, filing bugs, sharing feedback, and testing the release candidates. Your efforts helped to ensure that Go 1.21 is as stable as possible. As always, if you notice any problems, please [file an issue](#).

Enjoy Go 1.21!