

Go talks

Check out <http://talks.golang.org> for presentations for some of the talks. For a comprehensive, curated and searchable index, try [GopherVids](#) from Damian Gryski.

Official

Introductory Talks

An introduction to Go.

Russ Cox's Tour of Go

[\[video and discussion\]](#)

Three things that make Go fast, fun, and productive: interfaces, reflection, and concurrency. Builds a toy web crawler to demonstrate these.

Go: a simple programming environment

[\[video\]](#) [\[another video\]](#) [\[slides\]](#)

Go is a general-purpose language that bridges the gap between efficient statically typed languages and productive dynamic language. But it's not just the language that makes Go special – Go has broad and consistent standard libraries and powerful but simple tools.

This talk gives an introduction to Go, followed by a tour of some real programs that demonstrate the power, scope, and simplicity of the Go programming environment.

Get Started with Go

[\[video\]](#)

Get a feel for the language and its standard libraries and tools in this session, where we go through installing Go and writing some simple but useful programs.

Go Programming

[\[video\]](#) [\[code\]](#)

A presentation delivered by Rob Pike and Russ Cox at Google I/O 2010. It illustrates how programming in Go differs from other languages through a set of examples demonstrating features particular to Go. These include concurrency, embedded types, methods on any type, and program construction using interfaces.

The Go Tech Talk

[\[video\]](#) [\[slides\]](#)

An hour-long talk delivered by Rob Pike at Google in October 2009. The language's first public introduction. The language has changed since it was made, but it's still a good introduction.

Development in Go

Writing Web Apps in Go

[\[video\]](#) [\[slides\]](#)

A talk by Rob Pike and Andrew Gerrand presented at Google I/O 2011. It walks through the construction and deployment of a simple web application and unveils the [Go runtime for App Engine](#).

Real World Go

[\[video\]](#) [\[slides\]](#)

A talk by Andrew Gerrand presented at Google I/O Bootcamp 2011. It gives a broad overview of Go's type system and concurrency model and provides four examples of Go programs that solve real problems.

Building Integrated Apps on Google's Cloud Platform

[\[video\]](#)

A talk by Andrew Gerrand presented at Google Developer Day Japan 2011. It discusses the development of a web application that runs on Google App Engine and renders raytraced that it stores on Google Cloud Storage.

High Performance Apps with Go on App Engine

Google I/O, May 2013

[\[video\]](#) [\[slides\]](#)

Practical Go Programming

[\[video\]](#) [\[slides\]](#) [\[code\]](#)

This talk presents the development of a complete web application in Go. It looks at design, storage, concurrency, and scaling issues in detail, using the simple example of an URL shortening service.

Lexical Scanning in Go

[\[video\]](#)

This GTUG talk by Rob Pike discusses the detailed design of a lexical scanner that uses Go's features in expressive combinations. (The discussion near the end about avoiding goroutines at initialization is obsolete: Go 1 allows goroutines in init functions so the extra complexity is unnecessary.)

Go in Production

Google I/O, June 2012

[\[video\]](#)

Since Go's release in 2009 many companies (besides Google, of course) have used the language to build cool stuff. In this session programmers from several companies will share their first-hand experience using Go in production environments.

Go: code that grows with grace

[\[video\]](#) [\[slides\]](#)

One of the Go Programming Language's key design goals is code adaptability; that it should be easy to take a simple design and build upon it in a clean and natural way. In this talk I describe a simple "chat roulette" server that matches pairs of incoming TCP connections, and then use Go's concurrency mechanisms, interfaces, and standard library to extend it with a web interface and other features. Although the function of the program changes dramatically, the inherent flexibility of Go allows the original design to remain intact as it grows.

Implementing a bignum calculator

[\[video\]](#) [\[slides\]](#)

Rob Pike describes his interpreter for an APL-like calculator language.

Go in Go

[\[video\]](#) [\[slides\]](#)

Rob Pike speaks on moving the Go toolchain from C to Go

Concurrency in Go

Go concurrency patterns

Google I/O, June 2012

[\[video\]](#)

Advanced Concurrency Patterns

[\[video\]](#) [\[slides\]](#)

Google I/O, May 2013

Concurrency is the key to designing high performance network services. This talk expands on last year's popular Go Concurrency Patterns talk to dive deeper into Go's concurrency primitives, and see how tricky concurrency problems can be solved gracefully with simple Go code.

Design of Go

The Expressiveness Of Go

[\[slides\]](#)

A discussion of the qualities that make Go an expressive and comprehensible language. The talk was presented by Rob Pike at JAOO 2010. The recording of the event was lost due to a hardware error.

Another Go at Language Design

[\[video\]](#) from Sydney University) [\[slides\]](#)

A tour, with some background, of the major features of Go, intended for an audience new to the language. The talk was presented at OSCON 2010. This talk was also delivered at Sydney University in September 2010.

Go Emerging Languages Conference Talk

[\[video\]](#) [\[slides\]](#)

Rob Pike's Emerging Languages Conference presentation delivered in July 2010. Talk abstract:

Go's approach to concurrency differs from that of many languages, even those (such as Erlang) that make concurrency central, yet it has deep roots. The path from Hoare's 1978 paper to Go provides insight into how and why Go works as it does.

The State of Go

June 2014

[\[video\]](#) [\[slides\]](#)

February 2015

[\[video\]](#) [\[slides\]](#)

May 2015

[\[video\]](#) [\[slides\]](#)

Miscellaneous

The Go frontend for GCC

[\[paper\]](#)

A description of the Go language frontend for gcc. Ian Lance Taylor's paper delivered at the GCC Summit 2010.

The Go Promo Video

[\[video\]](#)

A short promotional video featuring Russ Cox demonstrating Go's fast compiler.

Meet the Go team

Google I/O, June 2012

[\[video\]](#)

A panel discussion with David Symonds, Robert Griesemer, Rob Pike, Ken Thompson, Andrew Gerrand, and Brad Fitzpatrick.

Fireside Chat with Go team

Google I/O, May 2013

[\[video\]](#)

A fireside chat with Andrew Gerrand, Brad Fitzpatrick, David Symonds, Ian Lance Taylor, Nigel Tao, Rob Pike, Robert Griesemer, Sameer Ajmani.

The State of the Gopher

[\[video\]](#) [\[slides\]](#)

Unofficial

Talks by members of the community.

Let's Go, or introduction to Go

[\[video \(starting at 14:35\)\]](#) [\[slides\]](#) [\[source\]](#)

This talk gives an introduction to Go in Russian.

What are Go modules and how do I use them?

[Paul Jolly](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#)

What else is in Go 1.11

[Daniel Marti](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#)

Sneak peak at the Go 1.11 release

Get Going with WebAssembly

[Johan Brandhorst](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#) [\[code wasm\]](#) [\[code grpc\]](#)

In this talk, Johan introduces you to the WebAssembly port in Go 1.11 and how it can help when dealing with JavaScript madness :)

Go and Mongo - and how it's changing

[DJ Walker-Morgan](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#)

Building a simple concurrency teaching language with Go

[Nicholas Ng](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#)

In this talk Nicholas presents the design and implementation of a simple language designed for teaching concurrency theory (process calculi), implemented in Go. He covers some of Go's static analysis tools used in the implementation and show how you can use them too!

Introducing Remoto

[Mat Ryer](#) at [LondonGophers](#)

[\[video\]](#)

Mat shares the first glimpse of a new project that aims to make building RPC services easy. gRPC isn't good for clients (especially web), and RESTful designs sometimes lead to confusing APIs. Remoto lets you define your service with a Go interface, and generate everything you need to build and consume the service.

Go Swagger

[Simone Trubian](#) at [LondonGophers](#)

[\[video\]](#)

Simone gives an overview of the Go Swagger command line tool and briefly explain how he used it to improve productivity in designing REST API's.

ORMs in Go

[Renato Serra](#) at [LondonGophers](#)

[\[video\]](#)

Renato explains where ORMs can help, what the options were and what it's been like to use one.

Unused parameters in Go code

[Daniel Marti](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#)

Daniel talks about how to use SSA and callgraphs to write powerful code analysis tools. In particular, he demonstrates how to detect unused parameters in functions.

Lies, Damn Lies, and Benchmarks

[Amnon](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#)

Amnon discusses why microbenchmarks can be misleading for optimising real world systems, why data layout is often more significant than code structure, and how Go can help us in the quest for performance.

A debugger from scratch

[Liz Rice](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#) [\[code\]](#)

Liz explains how a debugger works by building one in a few lines of Go. This includes mapping between Go source code and the machine code instructions it compiles to, and using the ptrace system call to set break points and examine and modify the running process.

Fast Fractal Fun With SDL

[Sue Spence](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#) [\[code\]](#)

Go programs which create images such as the Mandelbrot & Julia sets often output an image file. I will show how to use Go bindings for the Simple Directmedia Layer library to output them on a display device instead.

Concurrency: a Journey from Ruby to Go

[Mathilda Thompson](#) at [LondonGophers](#)

[\[video\]](#)

Go in a Polyglot Environment

[Kevin McKelvin](#) at [LondonGophers](#)

[\[video\]](#)

In this talk Kevin goes through his experience of adopting Go, moving to a polyglot environment, successes and challenges, and how Go fits into his company's overall architecture and strategy.

Delivering Go Services

[Zak Knill](#) at [LondonGophers](#)

[\[video\]](#)

Delivering Go Services: After introducing Go to your company, and deploying your first go service. What are the next steps? This talk focuses on some of the things that come next, touching on the fabled "New service to prod in X (10, 20, 30) mins", as well as some gotchas along the way.

Go-ing Lambda

[David Blooman](#) at [LondonGophers](#)

[\[video\]](#)

Go-ing Lambda - A year in production: How we(FundApps) used Go in lambda functions to build a service for importing/scraping/parsing data for financial services to build API's on top of. Tips and tricks of lambda functions in Go, limitations, performance and using the Apex framework.

The RED method

[Tom Wilkie](#) at [LondonGophers](#)

[\[video\]](#)

We'll also have a section dedicated to those of you who are hiring or looking to get hired (if we'll miss it like last time, please don't be afraid to remind us).

Abusing Go's net package for fun and profit

[Michał Witkowski](#) at [LondonGophers](#)

[\[video\]](#)

This talks into the details of how Go's composition-based philosophy, as applied to the net package, can be creatively leveraged to beautiful and useful hacks that significantly augment the functionality of the stack. We'll explore the net.Conn, and how one can (ab)use them in creative ways. We'll take a peek into net/http, and explore how the http.Handler and http.Roundtripper interfaces can be creatively appropriated to build useful middleware. We'll then dig even deeper into the net/http internals and how they related tls.Conn and x/net/http2, to understand how they work, and armed with that knowledge we'll demonstrate some of our most beautiful hacks.

2018's stringer

[Daniel Marti](#) at [LondonGophers](#)

[\[video\]](#) [\[slides\]](#)

2018's stringer - a demonstration of new features you likely haven't heard of.