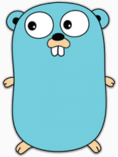
Programming in Go

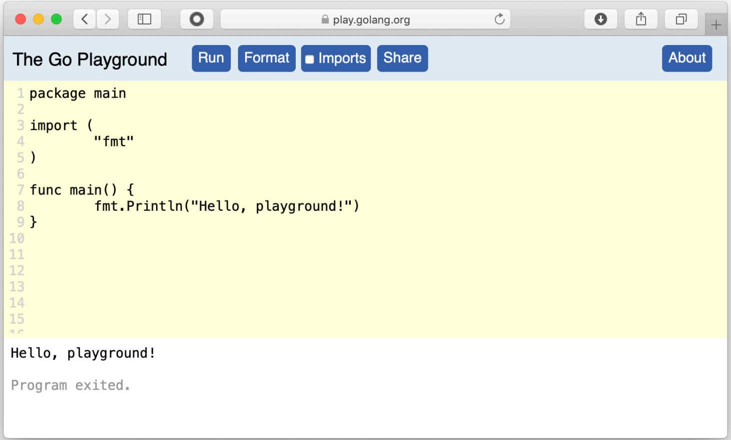
Matt Holiday
Christmas 2020



Hello, world!

Hello, playground!

Simple programs run at https://play.golang.org



```go
package main

import (
	"fmt"
)

func main() {
	fmt.Println("Hello, playground!")
}
```

```
Hello, playground!

Program exited.
```

Playground limitations

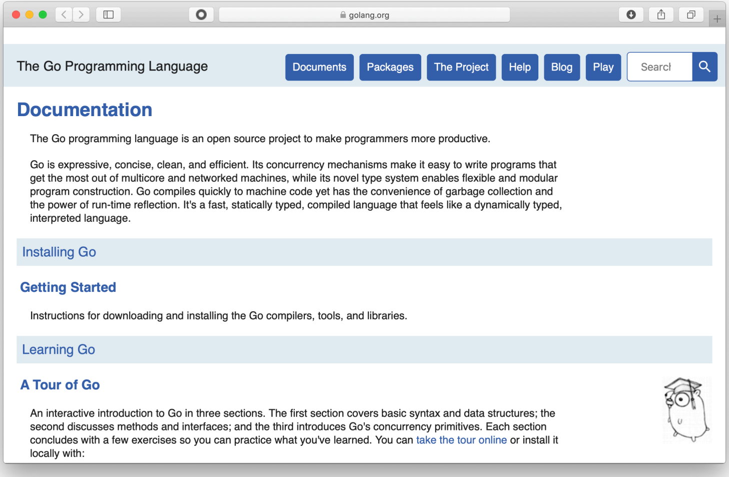
There's no input or output *except* writing to

- `stdout`
- `stderr`

For obvious reasons, you can't write to files or open a network socket or run a web server

More information

Get all the info at https://golang.org/doc/



The Go Programming Language

Documentation

The Go programming language is an open source project to make programmers more productive.

Go is expressive, concise, clean, and efficient. Its concurrency mechanisms make it easy to write programs that get the most out of multicore and networked machines, while its novel type system enables flexible and modular program construction. Go compiles quickly to machine code yet has the convenience of garbage collection and the power of run-time reflection. It's a fast, statically typed, compiled language that feels like a dynamically typed, interpreted language.

Installing Go

Getting Started

Instructions for downloading and installing the Go compilers, tools, and libraries.

Learning Go

A Tour of Go

An interactive introduction to Go in three sections. The first section covers basic syntax and data structures; the second discusses methods and interfaces; and the third introduces Go's concurrency primitives. Each section concludes with a few exercises so you can practice what you've learned. You can take the tour online or install it locally with:

Hello, world!

What the simplest program looks like:

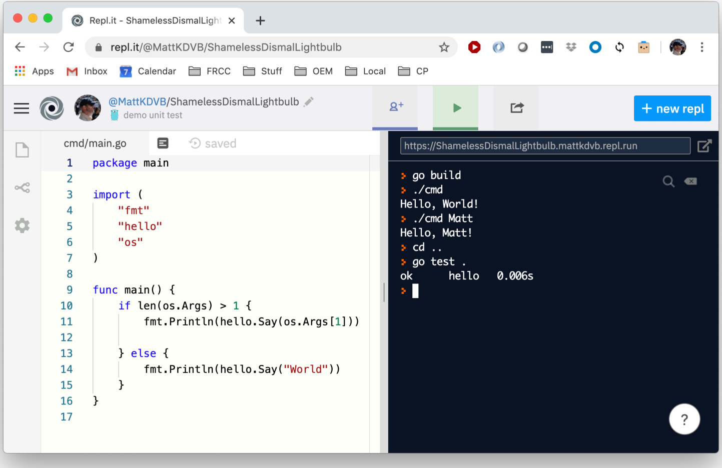
```go
package main

import "fmt"

func main() {
    fmt.Println("Hello, world!")
}
```

Hello, repl.it!

You can also use repl.it


Repl.it - ShamelessDismalLight
repl.it/@MattKDVB/ShamelessDismalLightbulb
Apps
Inbox
Calendar
FRCC
Stuff
OEM
Local
CP
@MattKDVB/ShamelessDismalLightbulb
demo unit test
+ new repl
cmd/main.go
saved
https://ShamelessDismalLightbulb.mattkdvb.repl.run
?

```go
package main

import (
	"fmt"
	"hello"
	"os"
)

func main() {
	if len(os.Args) > 1 {
		fmt.Println(hello.Say(os.Args[1]))

	} else {
		fmt.Println(hello.Say("World"))
	}
}

```

```
> go build
> ./cmd
Hello, World!
> ./cmd Matt
Hello, Matt!
> cd ..
> go test .
ok      hello   0.006s
> 
```

Installation

Start from the Go downloads page: https://golang.org/dl

Mac: use `brew install go` (or use the installer package)

Homebrew installation: https://brew.sh

Windows: open the installer (MSI) file and follow the prompts to install the Go tools

(otherwise you can download a ZIP file, but you have to set some environment stuff)

Linux: download the archive and extract it into `/usr/local`, creating a Go tree in `/usr/local/go`

```
    $ sudo tar -C /usr/local -xzf go1.15.6.linux-amd64.tar.gz
```

and don’t forget to add `/usr/local/go/bin` to `$PATH` (Linux)

Running a program

From the command line:

```
$ go run .
Hello, world!
```

Later we’ll talk about how to build binaries that stick around