

# The Go Programming Language



## Getting Started

[Install the Go tools](#)  
[Test your installation](#)  
[Uninstalling Go](#)  
[Getting help](#)

Your download should begin shortly. If it does not, click [this link](#).

### Install the Go tools

If you are upgrading from an older version of Go you must first [remove the existing version](#).

### Linux, Mac OS X, and FreeBSD tarballs

[Download the archive](#) and extract it into `/usr/local`, creating a Go tree in `/usr/local/go`. For example:

```
tar -C /usr/local -xzf go1.10.3.linux-amd64.tar.gz
```

(Typically these commands must be run as root or through sudo.)

Add `/usr/local/go/bin` to the PATH environment variable. You can do this by adding this line to your `/etc/profile` (for a system-wide installation) or `$HOME/.profile`:

```
export PATH=$PATH:/usr/local/go/bin
```

**Note:** changes made to a profile file may not apply until the next time you log into your computer. To apply the changes immediately, just run the shell commands directly or execute them from the profile using a command such as `source $HOME/.profile`.

### Installing to a custom location

The Go binary distributions assume they will be installed in `/usr/local/go` (or `c:\Go` under Windows), but it is possible to install the Go tools to a different location. In this case you must set the `GOROOT` environment variable to point to the directory in which it was installed.

For example, if you installed Go to your home directory you should add commands like the following to `$HOME/.profile`:

```
export GOROOT=$HOME/go1.X  
export PATH=$PATH:$GOROOT/bin
```

**Note:** `GOROOT` must be set only when installing to a custom location.

## Test your installation

Check that Go is installed correctly by setting up a workspace and building a simple program, as follows.

Create your [workspace](#) directory, `$HOME/go`. (If you'd like to use a different directory, you will need to [set the GOPATH environment variable](#).)

Next, make the directory `src/hello` inside your workspace, and in that directory create a file named `hello.go` that looks like:

```
package main

import "fmt"

func main() {
    fmt.Printf("hello, world\n")
}
```

Then build it with the go tool:

```
$ cd $HOME/go/src/hello
$ go build
```

The command above will build an executable named `hello` in the directory alongside your source code. Execute it to see the greeting:

```
$ ./hello
hello, world
```

If you see the "hello, world" message then your Go installation is working.

You can run `go install` to install the binary into your workspace's `bin` directory or `go clean -i` to remove it.

Before rushing off to write Go code please read the [How to Write Go Code](#) document, which describes some essential concepts about using the Go tools.

## Uninstalling Go

To remove an existing Go installation from your system delete the `go` directory. This is usually `/usr/local/go` under Linux, Mac OS X, and FreeBSD or `c:\Go` under Windows.

You should also remove the Go `bin` directory from your `PATH` environment variable. Under Linux and FreeBSD you should edit `/etc/profile` or `$HOME/.profile`. If you installed Go with the [Mac OS X package](#) then you should remove the `/etc/paths.d/go` file. Windows users should read the section about [setting environment variables under Windows](#).

## Getting help

For help, see the [list of Go mailing lists, forums, and places to chat](#).

Report bugs either by running “**go bug**”, or manually at the [Go issue tracker](#).

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)