Go Go toolchain commands

CONCEPTS

A module (go.mod) is a tree of packages; a package is one directory of .go files. Commands take package patterns: . is the current package, ./... is it plus everything below.

go build only checks and discards non-main packages; for a main package it writes a binary. go install puts the binary in $GOBIN (default $GOPATH/bin).

BUILD AND RUN

go run .
Compile and run the package here

go build ./...
Compile all packages, discard results

go build -o *<path>* **.**
Write the binary to a chosen path

go install *<pkg>*@latest
Install a tool binary into $GOBIN

go build -race
Build with the race detector

go build -ldflags="-X main.version=*<v>***"**
Set a string variable at link time

go build -tags *<tag,...>*
Enable build tags

CROSS-COMPILING

GOOS=*<os>* **GOARCH=***<arch>* **go build**
Build for another platform

CGO_ENABLED=0 go build
Static build without C toolchain

TESTING

go test
Test the package in the current dir

go test ./...
Test every package below here

TESTING (cont.)

go test -v
Print each test and its output

go test -run *<regexp>*
Run only matching tests

go test -count=1
Bypass the test result cache

go test -race
Run with the race detector

go test -cover
Show statement coverage

go test -coverprofile=*<file>*
Write a coverage profile

go tool cover -html=*<file>*
Browse coverage in a browser

go test -bench *<regexp>*
Run benchmarks (. for all)

FORMAT AND CHECK

go fmt ./...
Format files with gofmt -l -w

go vet ./...
Find likely mistakes

DEPENDENCIES

go get *<pkg>*
Add a dependency at the latest version

go get *<pkg>*@*<version>*
Pin a version, branch or commit

go get -u ./...
Upgrade deps of this module

MODULES

go mod init *<module>*
Create go.mod here

go mod tidy
Add missing, remove unused modules

ENVIRONMENT

go env
Print all Go environment settings