Getting started with Go

The best place to begin is the short tutorial, "A Tour of Go". They have it set up using a "sandbox" in which you can edit and run Go code without installing the toolchain or setting up a workspace.

https://tour.golang.org/welcome/1

The main other document you'll need is "Effective Go", which gives details and answers most of the questions that come to mind as you go through the first tutorial.

https://golang.org/doc/effective_go.html

Another nice resource, if you want some indigth into the rationale for Go, is the interview with Andrew Gerrand (one of the Go team at Google) on Software Engineering Radio,

http://www.se-radio.net/2014/03/episode-202-andrew-gerrand/

Installing the tools is easy. Go to https://golang.org/ and follow the download link to get an appropriate binary. The documents "Getting Started" and "How to Write Go Code" might be helpful, see https://golang.org/doc/

Building

I'll mention one thing that wasn't obvious until I read the documentation several times. If you have a single file, say hello.go, you can run the command

go build hello.go

and it will create an executable called hello that you can run. As soon as you want to build a package, however, you have to use the unintuitively-named command go install instead of go build. And in order to do that, you need to have an environment variable GOPATH to point to a root directory containing three subdirectories named src, lib, and bin. (These are just defaults.)

As a concrete example, say I start with the directory structure

```
workspace/src/foo/hello.go
workspace/src/bar/printer.go
workspace/pkg/
workspace/bin/
```

where hello.go is in a main package:
package main
import "bar"

func main() {
    bar.DoPrint("Steve")
}

and printer.go is in a non-main package

package bar
import "fmt"

func DoPrint(s string) {
    fmt.Println("Hello " + s)
}

Then I can run the following commands from the workspace directory:

    smkautz$ export GOPATH=/Documents/isu/cs430/go/workspace/
    smkautz$ cd src
    smkautz$ go install foo bar

And end up with the structure,

    workspace/src/foo/hello.go
    workspace/src/bar/printer.go
    workspace/pkg/darwin_amd64/bar.a
    workspace/bin/foo

That is, the non-main package is compiled into a library in the pkg directory, and the main package is compiled into an executable in the bin directory.

    smkautz$ ./bin/foo
    Hello Steve

**Eclipse**

I've also tried installing the Eclipse plugin and found it works well. You get a Go perspective you can switch to, you'll get the three standard directories src, pkg, and bin, and the "New" wizard will offer to create Go files. However, you have to manually create the package directories you want within the src folder.

You get red squiggles for errors. They don't go away when you correct them; you have to explicitly run the build target again.

The console will show you the go install command being executed when you select the "build" target.

The standard key combination Ctrl-Shift-F runs the gofmt tool, which is nice.