
Go
Everything You Wanted to Know About Go
and
Didn't Know to Ask

Raymond E. Marcil
<marcilr@gmail.com>

Revision 52 (February 4, 2013)

Abstract

Working at GCI Network Services, OSS I needed a location for Go documentation.
Hence this document.

Contents

Contents	2
List of Figures	3
List of Tables	3
List of Definitions and Abbreviations	4
Introduction	5
Comments	5
Packages	6
Functions	7
Variables	8
Flow Control	9
Types	10
Examples	11
Verbatim	12
Appendix	13
Links	13

List of Figures

1	File and Directory Structure	12
---	--	----

List of Tables

List of Definitions and Abbreviations

- **MOA** - Municipality of Anchorage

Introduction

Introduction to the L^AT_EX Template with Examples.

Comments

COMMENTS Comment — *Sean Weems, Spring 2003*

We should get the COMMENTS column searchable via the landrecords application before we do much anything else – shouldn't be too hard.

Errata: Plats spanning multiple sections

A few anomalies can be observed in the AKPLATS table. Specifically plats exist that span multiple sections. Since the table only has a single column, SCODE, that accepts a single section code, SGU (Status Graphics Unit) has handled this problem by entering multiple rows in the table, each with a different section that point to the same plat or file. Multiple section plats are indicated by setting the TCODE column to the value 37, and making an appropriate notation like *Section 24-25-26-27* in the REMARKS column.

[FIXME: Perhaps the SCODE column should accept an array of sections?]

Packages

Functions

Variables

Flow Control

Flow control of code using conditionals like `for`, `if`, `else`, `switch`, and `defer`.¹

¹“Flow control statements: `for`, `if`, `else`, `switch` and `defer`. Learn how to control the flow of your code with conditionals, loops, switches and defers.”
<https://tour.golang.org/list>

Types

Type such as arrays, structs, slices, and maps.²

²“More types: structs, slices, and maps. Learn how to define types based on existing ones: this lesson covers structs, arrays, slices, and maps.”
<https://tour.golang.org/list>

Examples

Verbatim

“The verbatim environment is a paragraph-making environment that gets L^AT_EX to print exactly what you type in. It turns L^AT_EX into a typewriter with carriage returns and blanks having the same effect that they would on a typewriter.”³

```
\begin{verbatim}
  text
\end{verbatim}
```

Figure formatting with verbatim

The following figure leverages verbatim for proper formatting:

```
gis/raster/
  dnr/
    map_library/
      plats/
        SP/YYYYMMDD/*.pdf          # indexed
        HI/YYYYMMDD/*.pdf          # Indexed
        ASLS/YYYYMMDD/*.pdf        # Indexed
      recorded-plats/
        YYYYMMDD/*.pdf
  blm/
    easements_17b/YYYYMMDD/*.pdf   # indexed
    mtp/YYYYMMDD/*.pdf              # non-indexed
    usrs/YYYYMMDD/*.pdf             # indexed
    usrs-notes/YYYYMMDD/*.pdf       # indexed
    uss/YYYYMMDD/*.pdf              # indexed
    uss-notes/YYYYMMDD/*.pdf        # indexed
    usms/YYYYMMDD/*.pdf             # indexed
    usms-notes/YYYYMMDD/*.pdf       # indexed
  usgs/
    drg/
      collared/
        250K/
        63K/
        25K/
        24/
      decollared/
      tools/
      missing\_data/
  dem/
  doq/
  topo/
```

Figure 1: File and Directory Structure

Appendix

Links

A Tour of Go

The tour covers the most important features of the language, mainly: Basics (Packages, variables, functions, flow control statements: for, if, else, switch, defer, more types: structs, slices, and maps), Methods and interfaces, and Concurrency.

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

List of integrated circuit packaging types

From Wikipedia, the free encyclopedia

https://en.wikipedia.org/wiki/List_of_integrated_circuit_packaging_types

Package Testing

`"import testing"`

The Go Programming Language

<https://golang.org/pkg/testing/>

Package types

`import "go/types"`

Package types declares the data types and implements the algorithms for type-checking of Go packages. Use `Config.Check` to invoke the type checker for a package. Alternatively, create a new type checked with `NewChecker` and invoke it incrementally by calling `Checker.Files`.

<https://golang.org/pkg/go/types/>

The Go Programming Language

<https://golang.org/pkg/go/types/>

The Go Programming Language Specification

Version of January 5, 2016

The Go Programming Language

<https://golang.org/ref/spec#Selectors>