Go Quick Reference version α

Basic Types

Slices

Maps

Channels

Type Declarations

```go
//struct type
type myType struct {
    embeddedType
    protectedMember int
    PublicMember    string
}
//type
type intList []int
```

Functions

```go
func returnNumberOne() int {
    return 1
}
// multiple return values
func returnTwoInts() (int,int) {
    return 2,3
}
// methods have receievers
func (t *myType) getProt() int {
    return t.protectedMember
}
```

Interfaces

```go
type protGetter interface {
    // function signatures only
    getProt() int
}
```

Operators

Unary	+	-	!	^	*	&	<-
Multiplication	*	/	%	«	»	&	&^
Addition	+	-	|	^			
Comparison	==	!=	<	<=	>	>=	
Logical	&&						
	||						

precedence

^x = bitwise compliment of x

Declaration

```go
var num int      // declare an integer
var num int = 1 // declare & initialize
var num = 1      // type inferred
num := 1         // short declaration
var num, foo int = 1, 2 // multiple
// Unicode works
var 名 = "アラスカ"
// constant
const name = "golang"
// group-declaration
var (
    foo int
    bar int = 1
    baz = 2
)
const (
    fooConst = iota // 0
    barConst        // 1
    bazConst = "Baz"
)
// camelCase when multi-word
myFavoriteRodent = "gopher"
// identifiers with uppercase first
// letters are exported
const PackageName = "my package"
var BadIdea = "mutate from anywhere!"
```

Short Declaration

```go
// Can only use in function body
x := aFunction() // type inferred
// can "re-declare" with at least one
// new variable
var y int
x, z := 1, 2 // ok
x, y := 3, 4 // ERROR
// creates block-local variables
for i:= 0 ; i < 10; i++ {
    fmt.Println(i)
}
fmt.Println(i) // ERROR
// beware of shadowing
```

Iota

Structure - Exectutable

```go
package main
import (
    "pkg"
    "pkg2"
)
import "fmt" //or one per-line
const (
    numberOne = 1
)
const hello = "Hello, 世界"
//main.main required for executable
func main() {
    // Code
}
```

Structure - Package

```go
package mylib
func CallMeFromOutside
```

Format verbs

Simpler than C's
MOAR TABLE

```go
package anothermain
import (
"fmt"
)
func main() {
    fmt.Println("Gopher, save me")
}
```