1 Golang CheatSheet

LANGUAGES

- PDF Link: cheatsheet-golang-A4.pdf, Category: languages
- Blog URL: https://cheatsheet.dennyzhang.com/cheatsheet-golang-A4
- \bullet Related posts: Ruby CheatSheet, Python CheatSheet, #denny-cheatsheets

File me Issues or star this repo.

1.1 Golang Conversion

Name	Comment
Convert string to int	i, _ := strconv.ParseInt("12345", 10, 64)
Convert string to int	i, err := strconv.Atoi("-42")
Convert string to list	L := strings.Split("hi,golang", "")
Convert string to []byte	[]byte("abcXX")
Convert string to byte	<pre>byte(str1[])</pre>
Convert byte to string	<pre>string(byte('a'))=</pre>
Convert string to float32	$f, _ := strconv.ParseFloat("3.1415", 32)$
Convert int to float32	0.5*float $32(age)+7>=$ float $32(age2)$
Convert int to string	s := strconv.Itoa(-42). Notice: not string(-42)
Convert rune to string	string(rune1)
Convert string list to string	strings.Join(list, ", ")
Convert int list to string	<pre>fmt.Sprintf("%s", 1)</pre>
Convert list to byte	byteI := byte(65)
Convert byte to int	<pre>int(byte('a'))</pre>
Convert byte to string	<pre>string(byte('a'))</pre>
Convert bytes to string	<pre>string([]byte("abcXX"))</pre>
Convert int32 to int32 Pointer	<pre>func int32Ptr(i int32) *int32 { return &i }</pre>
Convert string[] to string	strings.Join([]string{"a", "b"}, ",")
Format string	<pre>fmt.Sprintf("%.3f", float64(v)/1000)</pre>
Format string	<pre>fmt.Sprintf("At %v, %s", e.When, e.What)</pre>
Format string	<pre>fmt.Printf("int: %d, float: %f, bool: %t\n", 123, 78.9, true)</pre>

1.2 Golang Quality Improvement Tools

Name	Comment
gosec	Golang security checker, gosec ./
golangci-lint	lint check, golint
errcheck	errcheck checks that you checked errors, errcheck ./
delve	Delve is a debugger for the Go programming language.
ginkgo	BDD Testing Framework for Go
mock	GoMock is a mocking framework for Golang
envtest	provides libraries for integration testing by starting a local control plane
go-junit-report	Convert go test output to junit xml
gocover-cobertura	go tool cover to XML (Cobertura) export tool
gocovmerge	Merge coverprofile results from multiple go cover runs

1.3 Golang Dependencies

Name	Comment
goimports	updates your Go import lines, adding missing ones and removing unreferenced ones
dep	Go deps, now recommend go modules
Install go dep	go get -u github.com/golang/dep/cmd/dep
Go modules	export GO111MODULE=on, go mod download
Initialize new module in current directory	go mod init XXX
Download modules to local cache	go mod download
Add missing and remove unused modules	go mod tidy
Make vendored copy of dependencies	go mod vendor
Reference	Link: Using Go Modules

1.4 Deep Dive Into Golang

• Go's original target was networked system infrastructure, what we now call cloud software

Name	Comment
Garbage Colection	
Golang goroutine	
Golang return a tuple	func dfs(root *TreeNode, max *float64) (sum int, cnt int), LeetCode:
Use strings.Builder, instead of string	LeetCode: Unique Email Addresses
Variable Conversion	float64(x_int/y_int) != float64(x_int)/float64(y_int), LeetCode: Max
For a list of objects, pass by value or reference	f(1 []*TreeNode) vs f(1 *[]*TreeNode), LeetCode: Lowest Common Ancest

1.5 Golang Errors

Name	Comment
does not support indexing	*variable[0] -> (*variable)[0]

1.6 Golang Common

Name	Comment
Upgrade golang to 1.12 in mac	brew upgrade go, go version
Reference	Link: The Go Programming Language Specification

1.7 Golang Code Structure & Common Algorithms

Name	Comment
Online Go Playgroud	https://play.golang.org/
One line if statement	$if a >= 1 \{ fmt.Print("yes") \}$
Declare variables with initializers	var ischecked, v , $str = false$, 2, "yes!"
goroutine	Define functions to run as distince subprocesses
switch	$\operatorname{code}/\operatorname{example}$ -switch.go
queue	LeetCode: Number of Recent Calls
bfs	m code/tree-bfs.go
trie tree	code/tree-trie.go

1.8 Syntax Sugar: From Python To Golang

Name	Python	Golang
sum slice	sum([1, 2, 3])	$\operatorname{sum} := 0; \operatorname{for} i := \operatorname{range} \operatorname{nums} \{ \operatorname{sum} + = \operatorname{nums}[i] \}$
Get last item	nums[-1]	nums[len(nums)-1]
For	for i in range(10):	${ m for} \ { m i} := 0; { m i} < 10; { m i} + +$
Loop list	for num in [1, 2]	$for num := range[[int{1, 2} { fmt.Print(num) }]$
Loop string	for ch in str:	$for _, ch := range str \{ fmt.Print(ch) \}$
Iterator	for num in nums:	for $_$, num := range nums {fmt.Print(num)}
While	while isOK:	for isOK
Check ch range	<pre>ord(ch) in range(ord('a'), ord('z')+1)</pre>	ch >='a' && $ch <=$ 'z'
Get min	min(2, 6, 5)	
Check is nil	root is None	root == nil
Reverse list	nums[::-1]	Need to create your own function. Weird!

1.9 Surprises In Golang

Name	Comment
Modulus returns negative numbers	In golang, $-3\% 2 == -1$

Updated: February 2, 2020

$1.10 \quad Golang \; Array/List/Slice$

Name	Comment
Make a array	var a [2]string; a[0]="hello"; a[1]="world"
Create array with given values	$l := [6] int \{2, 3, 7, 5, 11, 13\}$
Create array with given values	$l := [string{"a", "c", "b", "d"}]$
Create dynamically-sized arrays	$\mathrm{a} := \mathrm{make}([]\mathrm{int}, 5)$
Create dynamically-sized arrays	$\mathrm{a} := \mathrm{make}([]\mathrm{int},1,5) \;//\; 5 \; \mathrm{is\; capacity}$
Sort string array	<pre>sort.Strings(1); fmt.Print(1)</pre>
Sort int array	sort.Ints(1) //in-place change
Golang sort one array by another array	LeetCode: Largest Values From Labels
Sort in descending order	<pre>sort.Sort(sort.Reverse(sort.IntSlice(keys)))</pre>
Append item	l = append(l, "e")
Append items	$l = \operatorname{append}(l, "e", "b", "c")$
Append item to head/prepend	<pre>1 = append([]string{"a"}, 1)</pre>
Remove last item	1 = 1[:len(1)-1]
Remove item by index	1 = append(1[0:1], 1[2:])
Slices of a array	var 12 = 1[1:3] // Notice: it's a reference
Copy a list	b := make([]int, len(a)); copy(b, a)
Join two lists	11 = append(11, 12)
Use pointer of array list	$\operatorname{code/pointer-array.go}$

1.11 Golang String

Name	Comment
Format string	fmt.Sprintf("At %v, %s", e.When, e.What)
Format string	<pre>fmt.Printf("int: %d, float: %f, bool: %t\n", 123, 78.9, true)</pre>
Padding zero	<pre>fmt.Printf("%02d:%02d", 2, 10)</pre>
Split string	<pre>var L = strings.Split("hi,golang", ",")</pre>
Replace string	<pre>var str2 = strings.Replace("hi,all", ",", ";", -1)</pre>
Replace string	strings.Replace("aaaa", "a", "b", 2) $//{ m bbaa}$
Split string by separator	<pre>strings.Split(path, " ")</pre>
Count characters	strings.Count("test", "t")
Substring	strings.Index("test", "e")
Join string	strings.Join([]string{"a","b"}, "-")
Repeat string	strings.Repeat("a", 2) // aa
Lower string	strings.ToLower("TEST")
Trim whitespace in two sides	<pre>strings.TrimSpace("\t Hello world!\n ")</pre>
Trim trailing whitespace	strings.TrimRight("\t Hello world!\n ", "\n ")
Concact string	<pre>fmt.Sprintf("%s%s", str1, str2)</pre>
Reference	Link: package strings

${\bf 1.12}\quad {\bf Golang\ Integer/Float}$

Name	Comment
Int max	MaxInt32 = 1
Int min	$MinInt32 = -1 \ll 31 \text{ golang math}$
Pass int as reference	sample code

1.13 Golang Env

Name	Comment
GOPATH	It is called as the workspace directory for Go programs
GOROOT	The location of your Go installation. No need to set any more
go env	Show a full list of environment variables
Reference	Link: GOPATH, GOROOT, GOBIN

1.14 Golang Package management

Name	Comment
go mod	Link: go modules
go get fix	GO111MODULE=off go get -fix ./
go mod replace url	go mod edit -replace

1.15 Golang Ascii

Name	Comment
get character ascii	byte('0')
ascii offset	<pre>fmt.Println(string('B' + byte('a')-byte('A')))</pre>

$1.16 \quad Golang\ Dict/Hashmap/Map$

Name	Comment
Create dict	map[string]int{"a": 1, "b": 2}
Create dict	<pre>make(map[string]int)</pre>
Check existence	$_, \mathrm{ok} := \mathrm{m[k]}$
Delete key	delete(m, "k1")
Create a map of lists	m := make(map[string][]string)
Get keys of a map	Loop the dictionary and append the key to a list
Use (x, y) as hashmap key	$m[[2]int\{2, 2\}] = true, code/example-hashmap-arraykey.go$

1.17 Golang Networking

Name	Comment
golang http	code/example-http.go

1.18 Golang Goroutines

Name	Comment
Basic goroutine	code/example-goroutine.go

1.19 Golang Inteface

Name	Comment
Hash map with both key and value dynamic	<pre>map[interface{}]interface{}</pre>
Define and use interface	$\operatorname{code}/\operatorname{example}$ -interface.go
Convert map[interface {}]interface {} to map[string]string	${\rm code/interface\hbox{-}conversion.go}$

1.20 Golang Files & Folders

Name	Comment
Read files	code/example-read-file.go
Write files	code/example-write-file.go

1.21 Golang Math

Name	Comment
pow(2, 3)	int(math.Pow(2, 3)) // Default is float64
sqrt	math.Sqrt(100)
Get rand	rand.Intn(100).rand.Float64()

1.22 Golang Bit Operator & Math

Name	Comment
Shift left	fmt.Print(1 \ll 10) $//$ 1024
Shift right	fmt.Print(1024 » 3) // 128

Updated: February 2, 2020

1.23 Golang BBD Testing

Name	Summary
ginkgo	BDD Testing Framework for Go http://onsi.github.io/ginkgo/
Ubuntu install ginkgo	apt-get install golang-ginkgo-dev
gomega	Ginkgo's Preferred Matcher Library
Add tags to tests	// +build availability, go test -vtags=availability ./test/e2e/

1.24 Golang Misc

Name	Comment
Golang sleep	time.Sleep(4* time.Second)
Golang logging	<pre>import "log", log.Info, log.Print, log.Error(err)</pre>
Golang print function name	runtime.Callers

1.25 More Resources

https://play.golang.org/

 $\verb|https://tour.golang.org/list|$

https://golang.org/doc/

https://github.com/a8m/go-lang-cheat-sheet License: Code is licensed under MIT License.

Updated: February 2, 2020