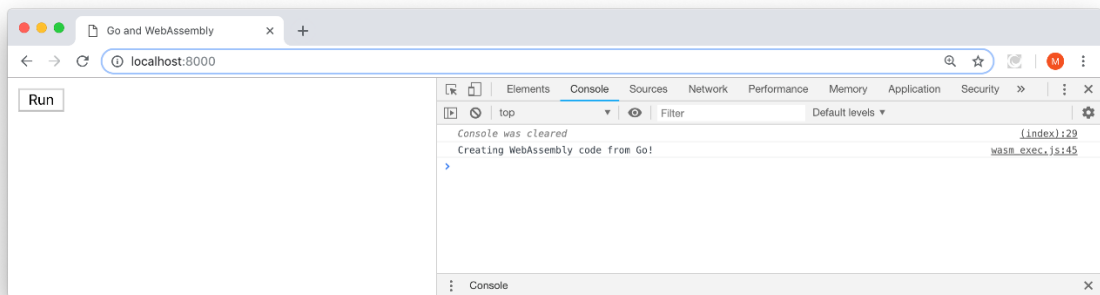
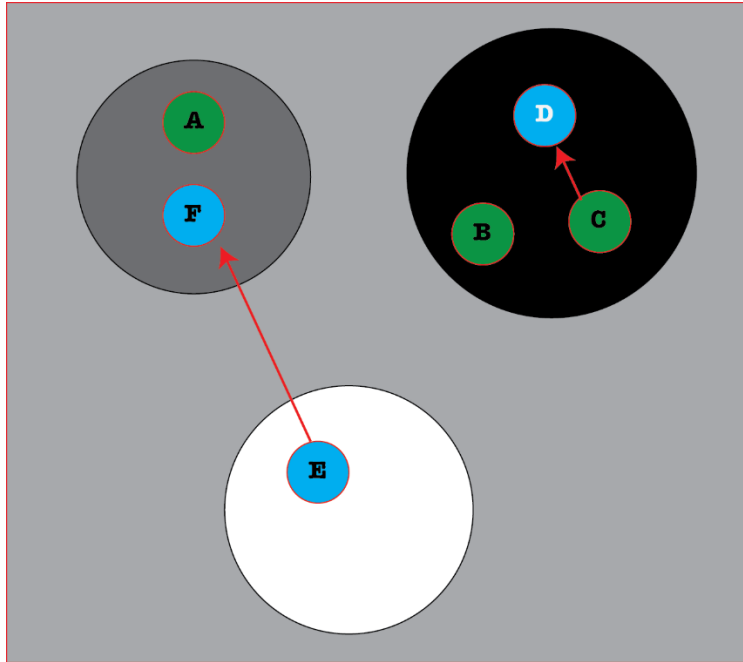
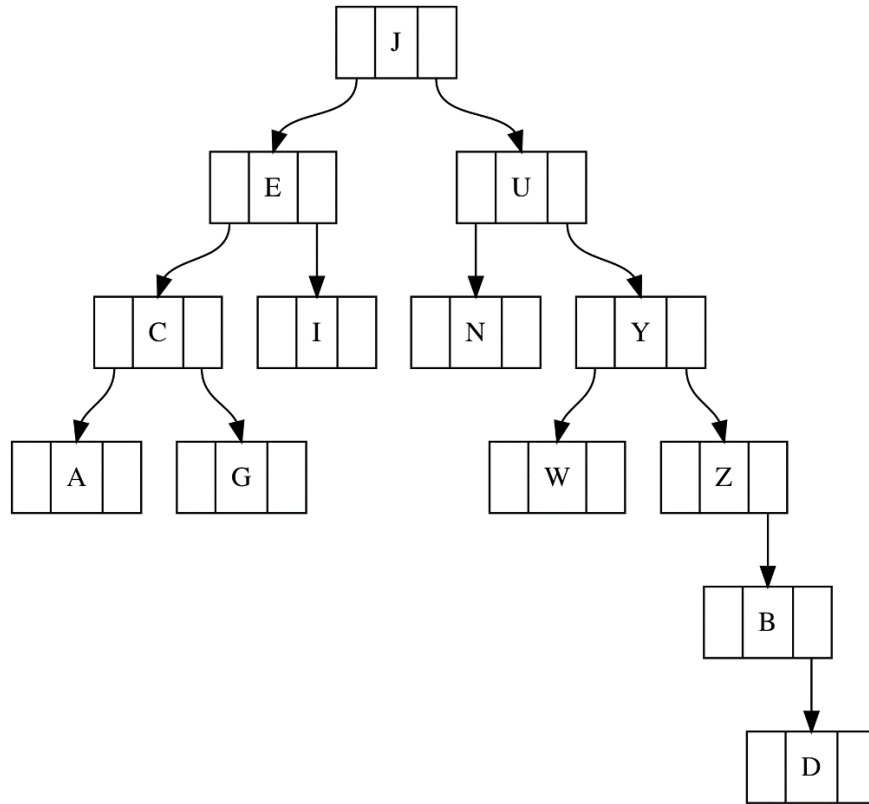
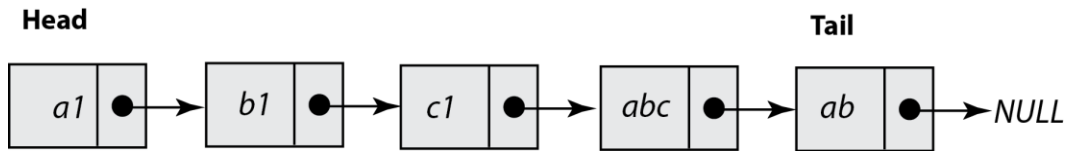
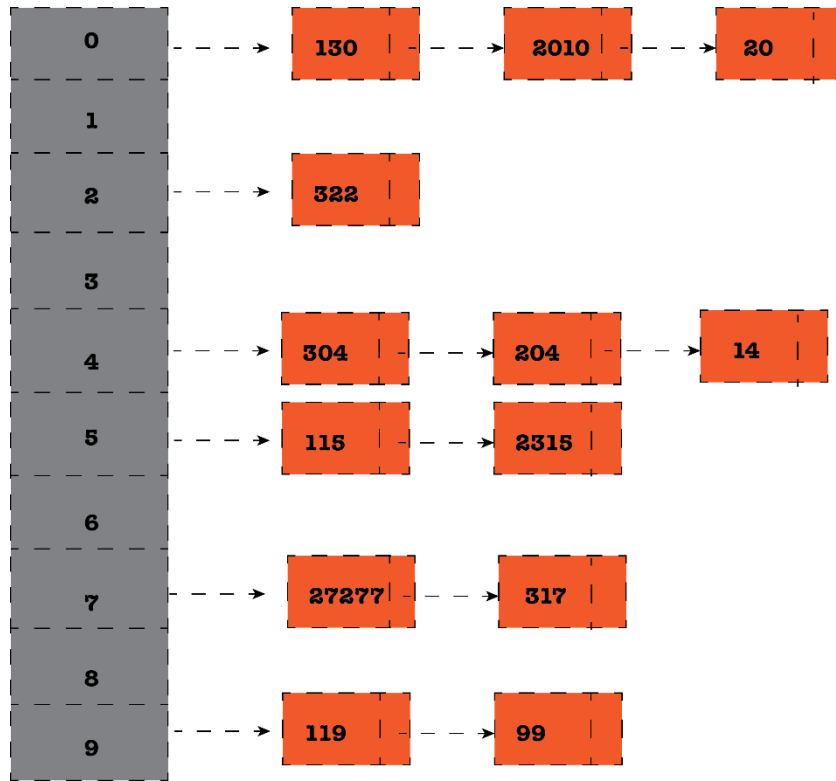


Chapter 2: Understanding Go Internals

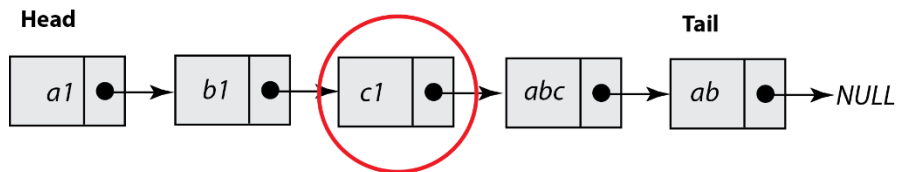


Chapter 5: How to Enhance Go Code with Data Structures

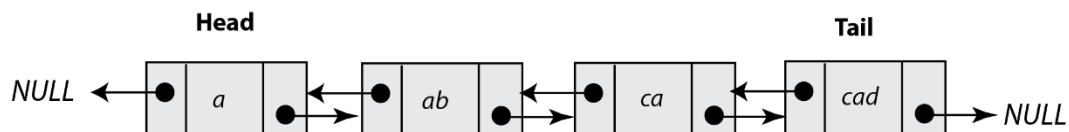
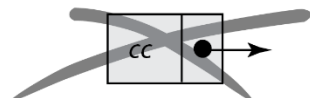
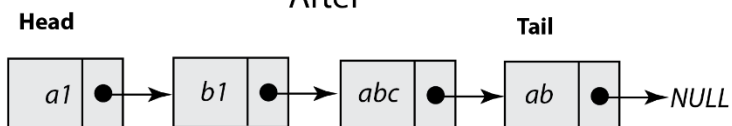




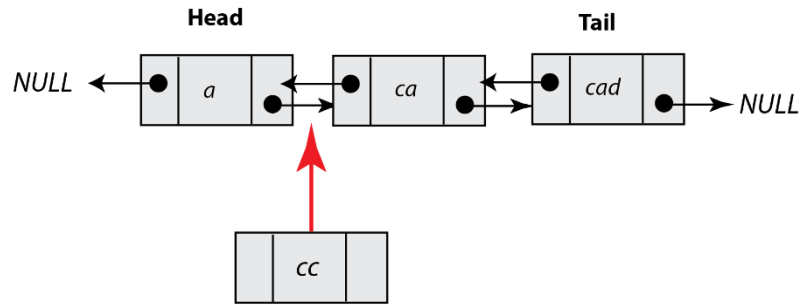
Before



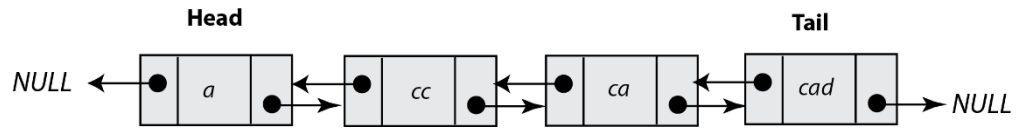
After



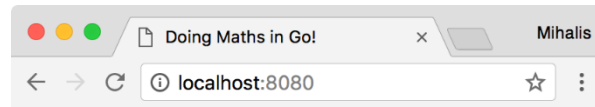
Before



After

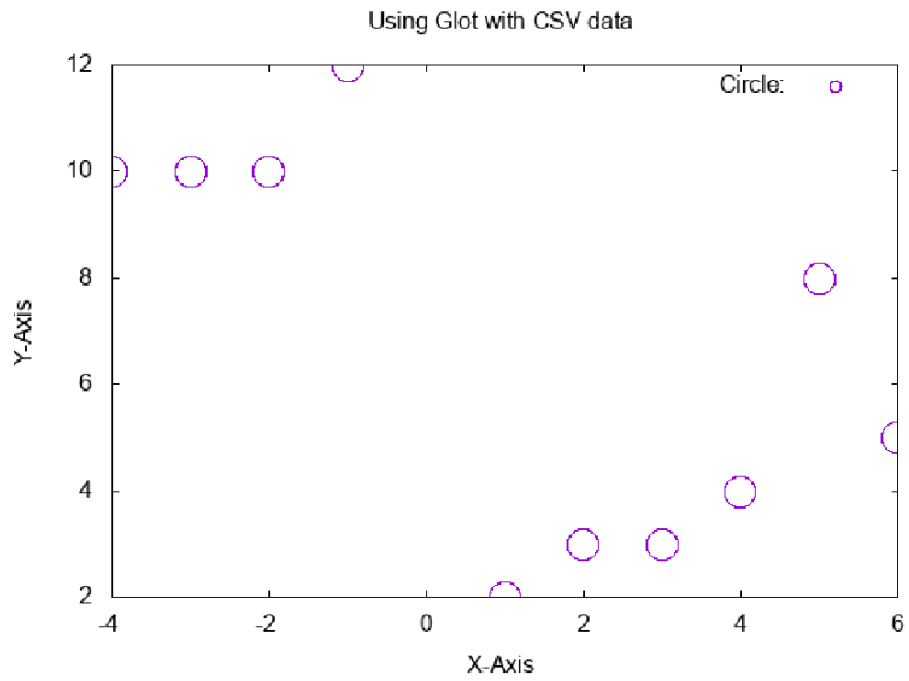


Chapter 6: What You Might Not Know About Go Packages and Functions



Number	Double	Square
20	40	400
21	42	441
22	44	484
23	46	529
24	48	576
25	50	625
26	52	676
27	54	729
28	56	784
29	58	841
30	60	900
31	62	961
32	64	1024
33	66	1089
34	68	1156
35	70	1225
36	72	1296
37	74	1369
38	76	1444
39	78	1521
40	80	1600
41	82	1681
42	84	1764
43	86	1849
44	88	1936
45	90	2025
46	92	2116
47	94	2209
48	96	2304
49	98	2401

Chapter 8: Telling a UNIX System What to Do



Chapter 10: Concurrency in Go – Advanced Topics

Global queue



Local queue



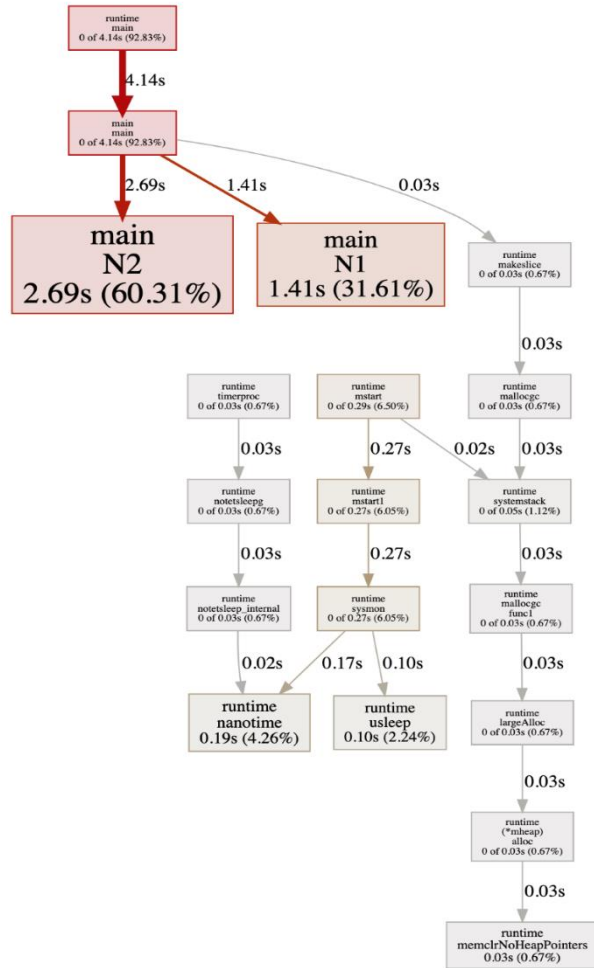
P1

Local queue



P2

Chapter 11: Code Testing, Optimization, and Profiling



unknown cpu x +
localhost:8080/ui/source

pprof VIEW SAMPLE REFINE Search regexp unknown.cpu

main.N2

/Users/mtsouk/Desktop/mGo2nd/Mastering-Go-Second-Edition/ch11/profileMe.go

Total:	2.69s	2.69s (flat, cum)	60.31%
41	-	-)
42	-	-	return true
43	-	-)
44	-	-)
45	-	-	func N2(n int) bool {
46	90ms	90ms	for i := 2; i < n; i++ {
47	2.60s	2.60s	if (n % i) == 0 {
48	-	-	return false
49	-	-	}
50	-	-)
51	-	-	return true
52	-	-	}

main.N1

/Users/mtsouk/Desktop/mGo2nd/Mastering-Go-Second-Edition/ch11/profileMe.go

Total:	1.41s	1.41s (flat, cum)	31.61%
32	-	-	return fo[n]
33	-	-)
34	-	-)
35	-	-	func N1(n int) bool {
36	-	-	k := math.Floor(float64(n/2 + 1))
37	60ms	60ms	for i := 2; i < int(k); i++ {
38	1.35s	1.35s	if (n % i) == 0 {
39	-	-	return false
40	-	-	}
41	-	-)
42	-	-	return true
43	-	-	}

runtime.nanotime

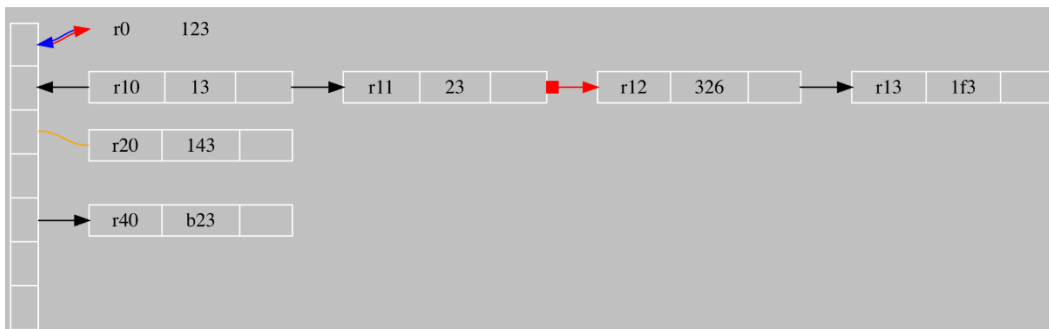
/usr/local/Cellar/go/1.12.4/libexec/src/runtime/sys_darwin.go

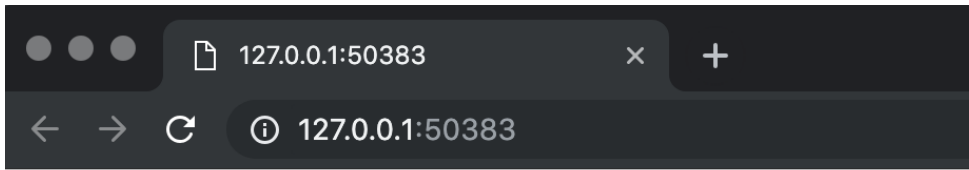
Total:	190ms	190ms (flat, cum)	4.26%
222	-	-	func open_trampoline()
223	-	-)
224	-	-	//go:nosplit
225	-	-	//go:cgo_unsafe_args
226	-	-	func nanotime() int64 {
227	190ms	190ms	var x struct {
228	-	-	t int64 // raw timer
229	-	-	numer, denom uint32 // conversion factors. nanoseconds = t * numer / denom.
230	-	-	}
231	-	-	libcCall(unsafe.Pointer(funcPC(nanotime_trampoline)), unsafe.Pointer(&x))
232	-	-	// Note: Apple seems unconcerned about overflow here. See

runtime.usleep

/usr/local/Cellar/go/1.12.4/libexec/src/runtime/sys_darwin.go

Total:	100ms	100ms (flat, cum)	2.24%
202	-	-)
203	-	-	//go:nosplit
204	-	-	//go:cgo_unsafe_args
205	-	-	func usleep(usec uint32) {





[View trace](#)

[Goroutine analysis](#)

[Network blocking profile](#) (↓)

[Synchronization blocking profile](#) (↓)

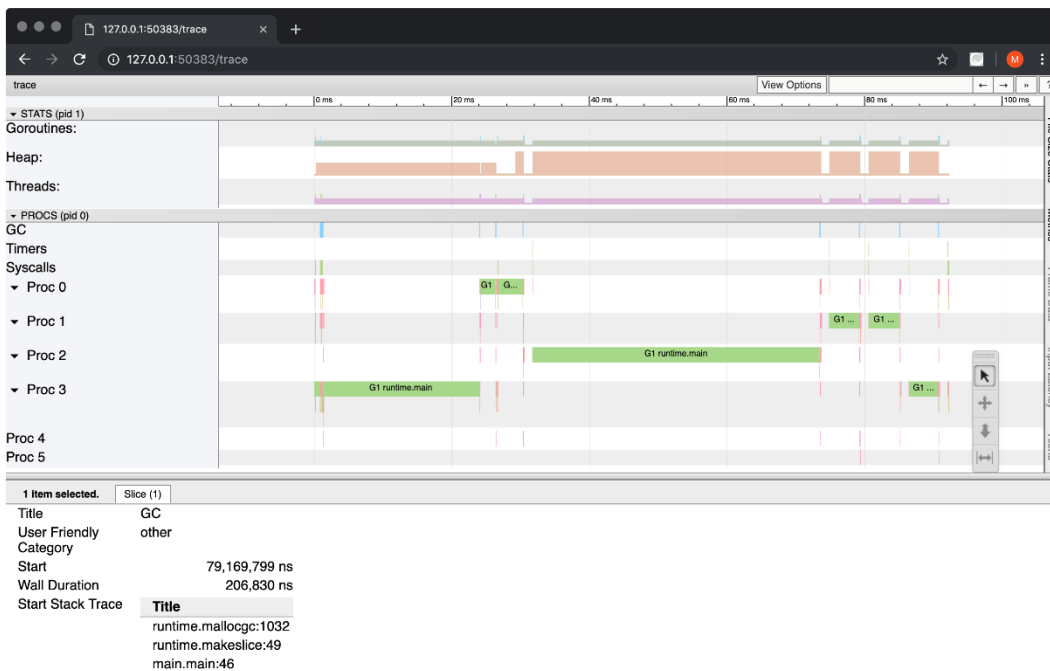
[Syscall blocking profile](#) (↓)

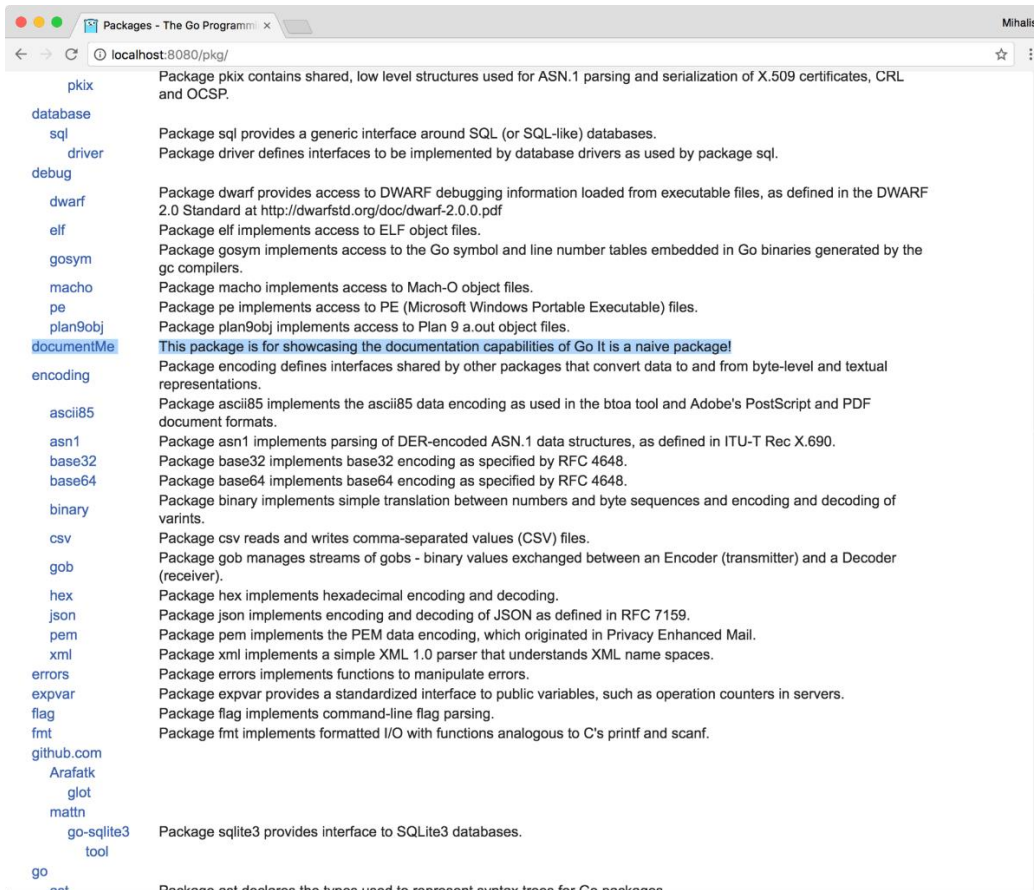
[Scheduler latency profile](#) (↓)

[User-defined tasks](#)

[User-defined regions](#)

[Minimum mutator utilization](#)





documentMe - The Go Program x Mihalis
localhost:8080/pkg/documentMe/

The Go Programming Language Documents Packages The Project Help Blog Search

Package documentMe

```
import "documentMe"
```

[Overview](#)
[Index](#)
[Examples](#)

Overview

This package is for showcasing the documentation capabilities of Go It is a naive package!

Index

Constants
func F1(n int) int
func S1(s string) int

Examples

F1
S1

Package files

documentMe.go

Constants

Pie is a global variable This is a silly comment!

```
const Pie = 3.1415912
```

func F1

```
func F1(n int) int
```

documentMe - The Go Progr... x Mihalis

localhost:8080/pkg/documentMe/

```
func F1(n int) int
```

The F1() function returns the double value of its input integer A better function name would have been Double()!

▼ Example

Code:

```
fmt.Println(F1(10))
fmt.Println(F1(2))
```

Output:

```
1
55
```

func S1

```
func S1(s string) int
```

The S1() function finds the length of a string It iterates over the string using range

▼ Example

Code:

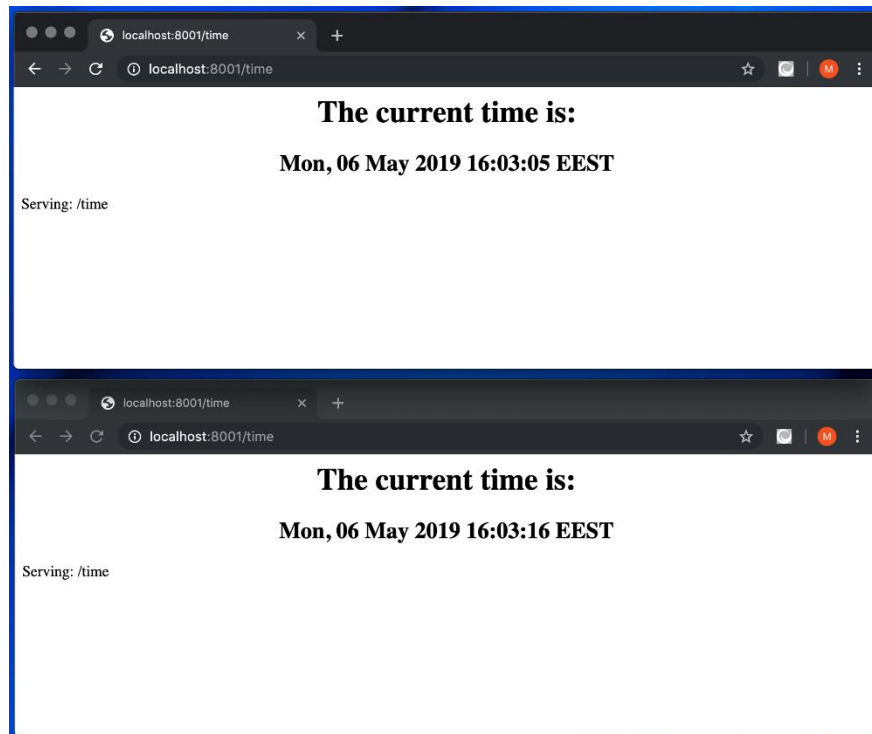
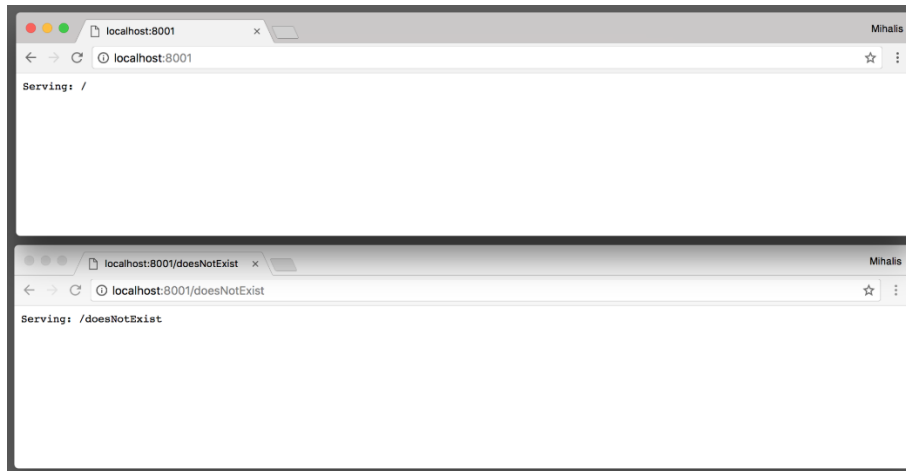
```
fmt.Println(S1("123456789"))
fmt.Println(S1(""))
```

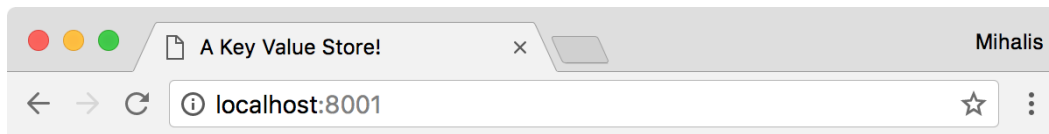
Output:

```
9
0
```

Build version go1.10.
Except as noted, the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a BSD license.
[Terms of Service](#) | [Privacy Policy](#)

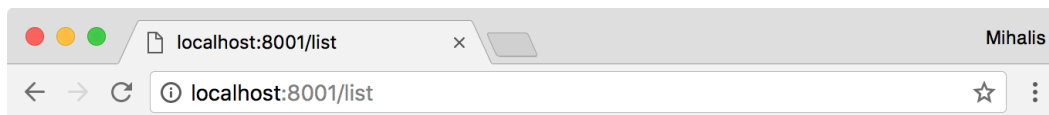
Chapter 12: The Foundations of Network Programming in Go





[Home sweet home!](#) [List all elements!](#) [Change an element!](#) [Insert new element!](#)

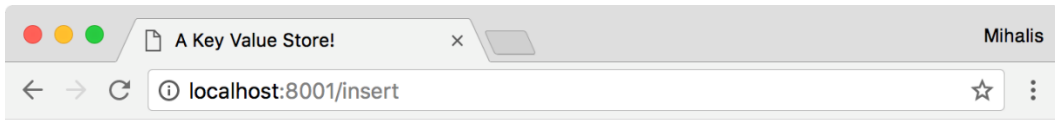
Welcome to the Go KV store!



[Home sweet home!](#) [List all elements!](#) [Change an element!](#) [Insert new element!](#)

The contents of the KV store are:

- **12** with value: {Mihalis Tsoukalos 123}
- **21** with value: {Dimitris Tsoukalos 123}
- **123** with value: {Vassilis Tsoukalos 123456}



[Home sweet home!](#) [List all elements!](#) [Change an element!](#) [Insert new element!](#)

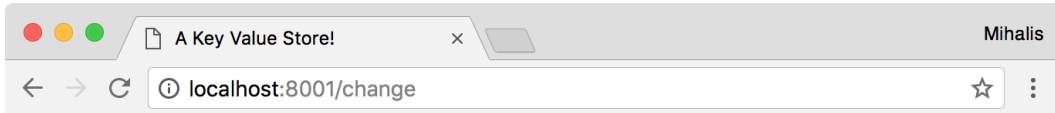
Please fill in the fields:

Key:

Name:

Surname:

Id:



[Home sweet home!](#) [List all elements!](#) [Change an element!](#) [Insert new element!](#)

Please fill in the fields:

Key:

Name:

Surname:

Id:

Chapter 14: Machine Learning in Go

