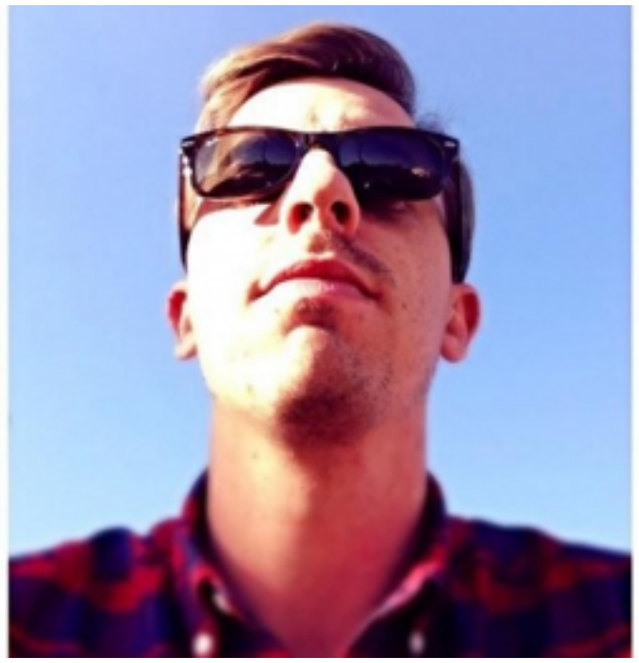

@scottmotte


SendGrid®

Go is hawt

Installation

Step 1

http://code.google.com/p/go/downloads/list?q=OpSys-OSX+Type-Installer

Step 2

```
// hello.go
package main
import "fmt"

func main() {
    fmt.Printf("hello, world\n")
}
```

Step 3

go run hello.go

That's it!

Environment

Step 1

```
mkdir -p ~/go/bin
mkdir -p ~/go/pkg
mkdir -p ~/go/src
```

Step 2

export GOPATH=$HOME/go

add this to your .bashrc, .profile, or .vimrc

Step 3

mkdir -p $GOPATH/src/github.com/your_username

Step 4

```
cd $GOPATH/src/github.com/
your_username
mkdir hello-go
cd hello-go
vim hello-go.go
```

Step 5

```
// hello.go
package main
import "fmt"

func main() {
    fmt.Printf("hello, world\n")
}
```

Step 6

go install github.com/your_username/hello-go

That installs the program as a binary to $GOPATH/bin/hello-go.

Step 7

$GOPATH/bin/hello-go

Optionally, add $GOPATH/bin to your PATH. Then you could just run: hello-go

Enjoy. Start writing Go.

Deploying

https://github.com/scottmotte/heroku-go-hello

```
git clone https://github.com/scottmotte/heroku-go-hello.git
cd heroku-go-hello
heroku create -b https://github.com/kr/heroku-buildpack-go.git
git push heroku master
```

Iterate and build

```
package main
import "fmt"
func main() {
  var a string = "initial"
  fmt.Println(a)

  var b, c int = 1, 2
  fmt.Println(b, c)

  var e int
  fmt.Println(e)

  f := "short"
  fmt.Println(f)
}
```

```
package main
import "fmt"
func main() {
  fmt.Println("go" + "lang")
  fmt.Println("1+1=", 1+1)
  fmt.Println(true && false)
  fmt.Println(true || false)
}
```

```
package main
import "fmt"
func main() {
  var array [5]int
  fmt.Println(array)

  array[4] = 44
  fmt.Println(array)
  fmt.Println(array[4])

  b := [5]int{1,2,3,4,5}
  fmt.Println(b)

  twoD := [2][3]int{{1,2,3}, {4,5,6}}
  fmt.Println(twoD)
}
```

```
package main
import "fmt"
func main() {
  slice := []string{"dude", "ted", "bill"}
  fmt.Println(slice)

  slice[0] = "newdude"
  fmt.Println(slice)
}
```

```
package main
import "fmt"
func main() {
  a := []string{"1", "2"}
  fmt.Println(a)

  // make(map[key_type]value_type)
  m := map[int]string{1: "one", 2: "two"}
  fmt.Println(m)
}
```

```
package main
import "fmt"
func main() {
  i := 1
  for i<=3 {
    fmt.Println(i)
    i = i + 1
  }

  for j := 5; j <= 12; j++ {
    fmt.Println(j)
  }

  ...
```

```
  ...
    for {
      fmt.Println("dude")
      if i >= 100 {
        break
      }
      i = i + 1
    }
  }
```

```
package main
import "fmt"
//funcs
func plus(a int, b int) int {
  return a + b
}
//multiple return values
func vals() (int, int) {
  return 88, 77
}
...
```

```
...
func main() {
  result := plus(9,3)
  fmt.Println(result)

  a, b := vals()

  fmt.Println(a)
  fmt.Println(b)
}
```

Resources

- http://golang.org
- https://github.com/scottmotte/heroku-go-hello
- http://github.com/scottmotte/go-by-example

Follow me on Twitter at

@scottmotte