

2019 TECH(K)NOW Day Taipei

Let's **=GO** For Web!

Benjamin Wang @ Golang Taiwan

2019.4.20



Hi, I'm Benjamin Wang

Golang Taiwan Core member

ASUS Engineer
- Golang, Android, Node.js, ...

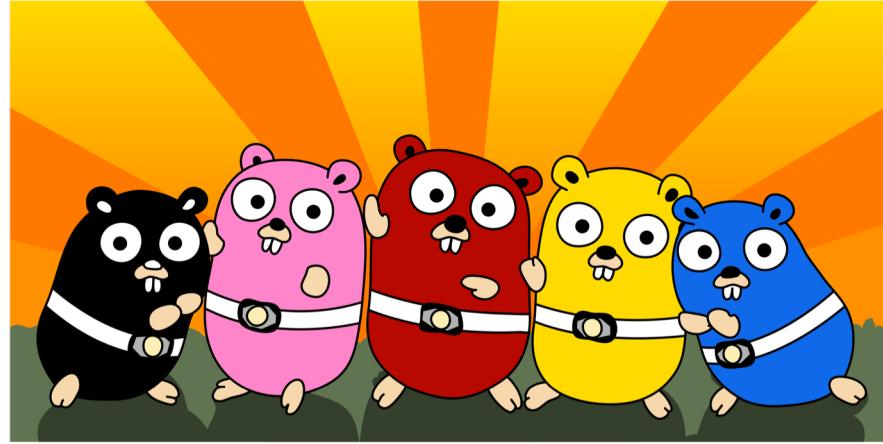
V-Hub.org Co-founder

Tiger Accelerator DevOps

 <https://t.me/syxuan>



Golang Taiwan



meetup

<https://www.meetup.com/golang-taipei-meetup/>

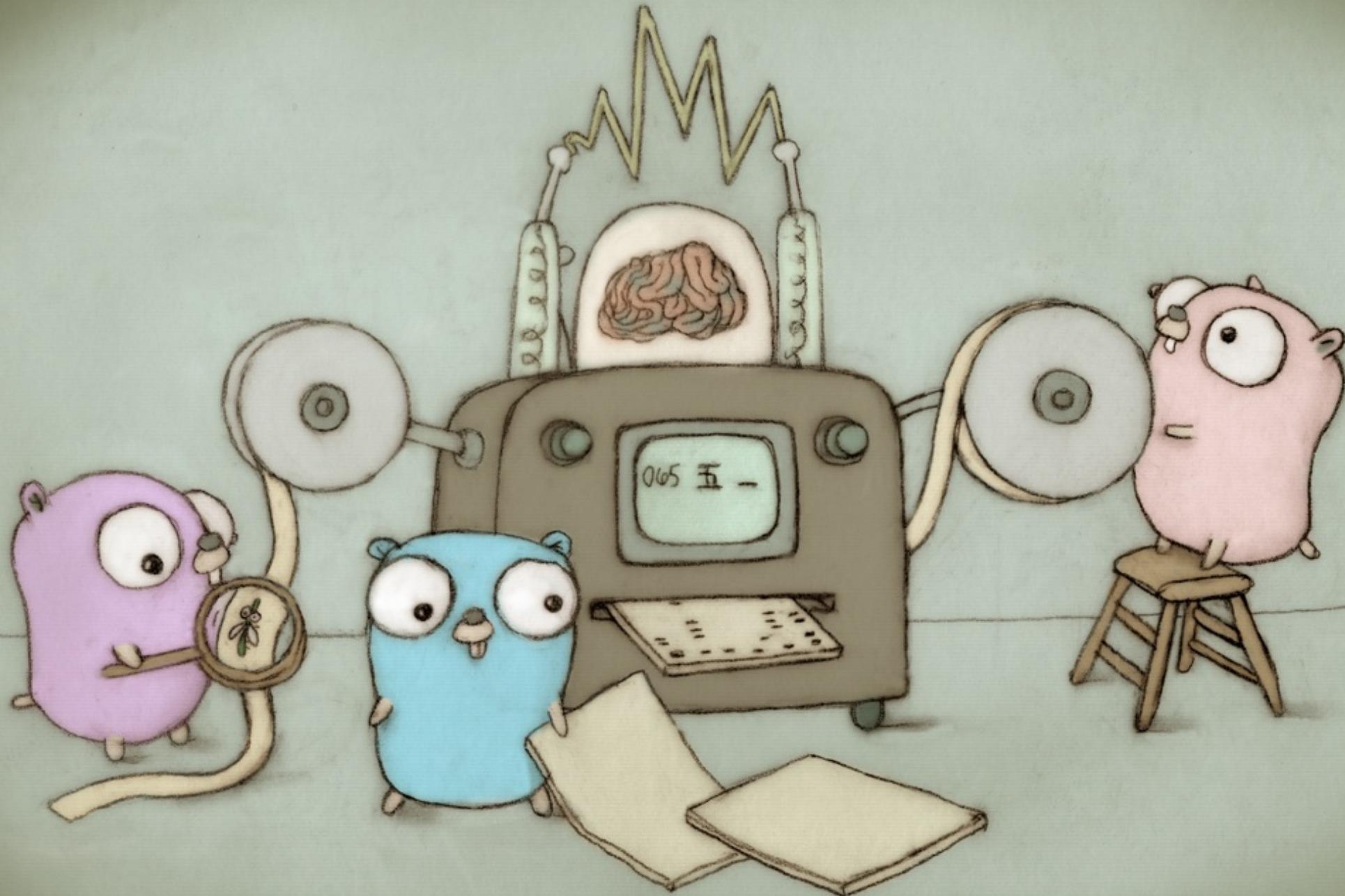


<https://t.me/golangtw>



<https://www.facebook.com/groups/269001993248363/>



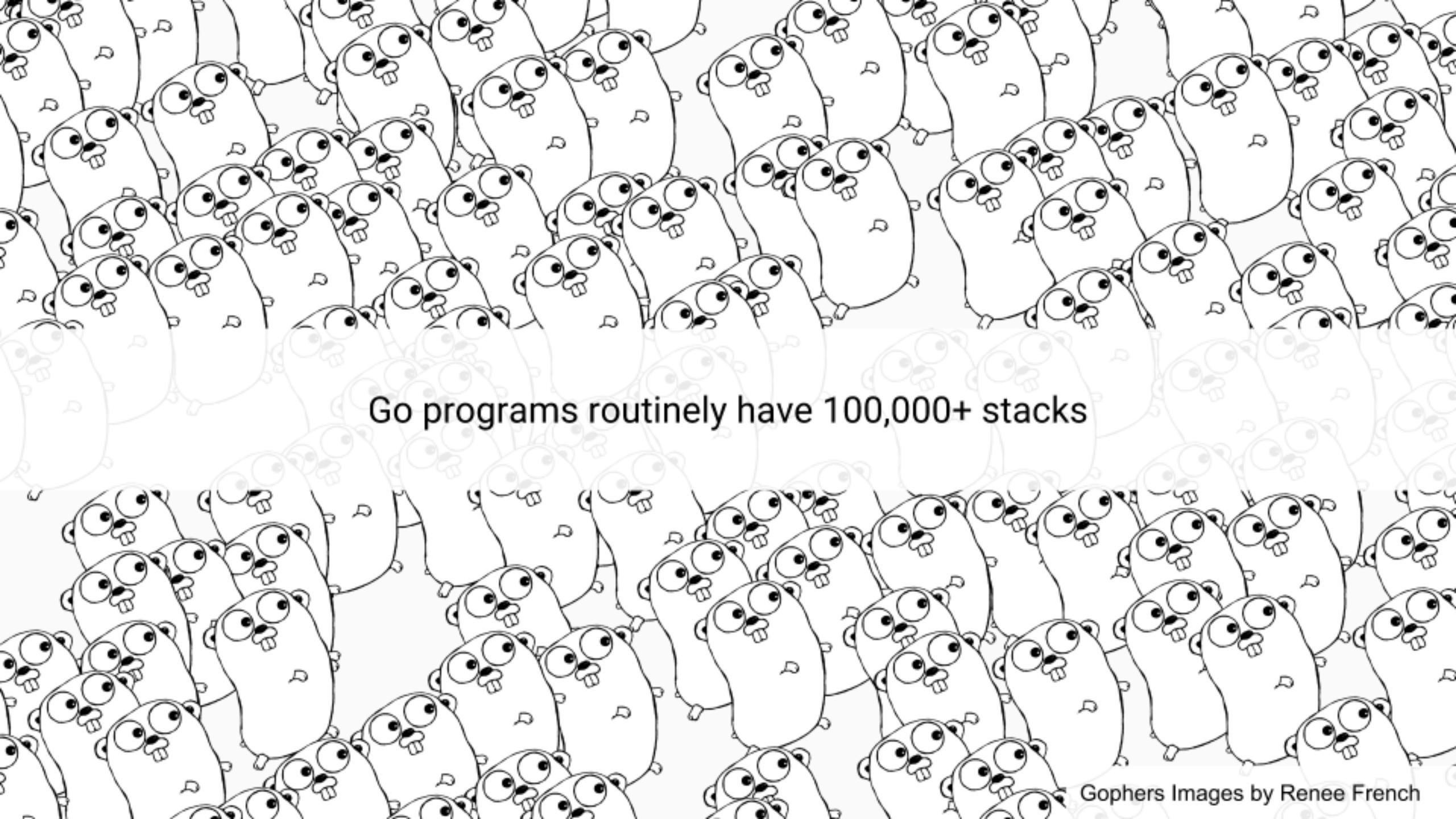


Robert Griesemer
- HotSpot, V8

Rob Pike
- UTF-8,
- Plan 9 from Bell Labs

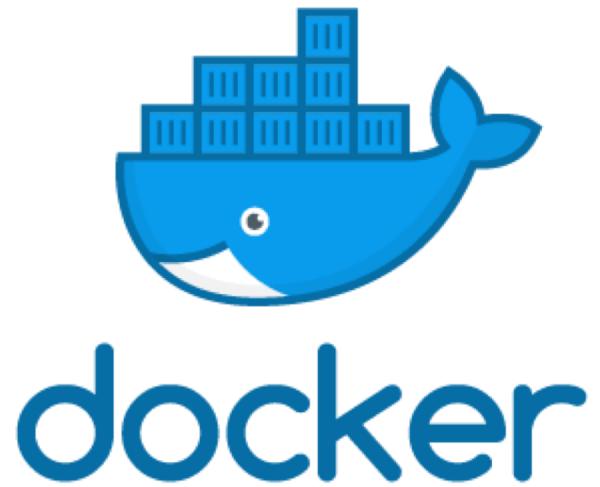
Ken Thompson
- UNIX, B, C++, Plan 9



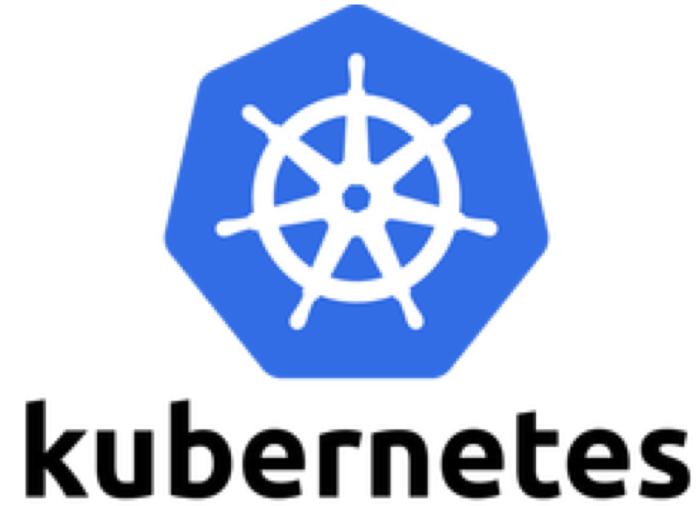


Go programs routinely have 100,000+ stacks

2013.03



2014.06





A photograph of a lecture hall from the perspective of the back rows. A large audience of mostly young adults is seated in rows of dark wooden chairs, facing a stage on the left. On the stage, a man in a grey shirt stands behind a white table, gesturing with his hands as if speaking. Another person is seated at a podium to his right. The room has brick walls and a large window on the left side.

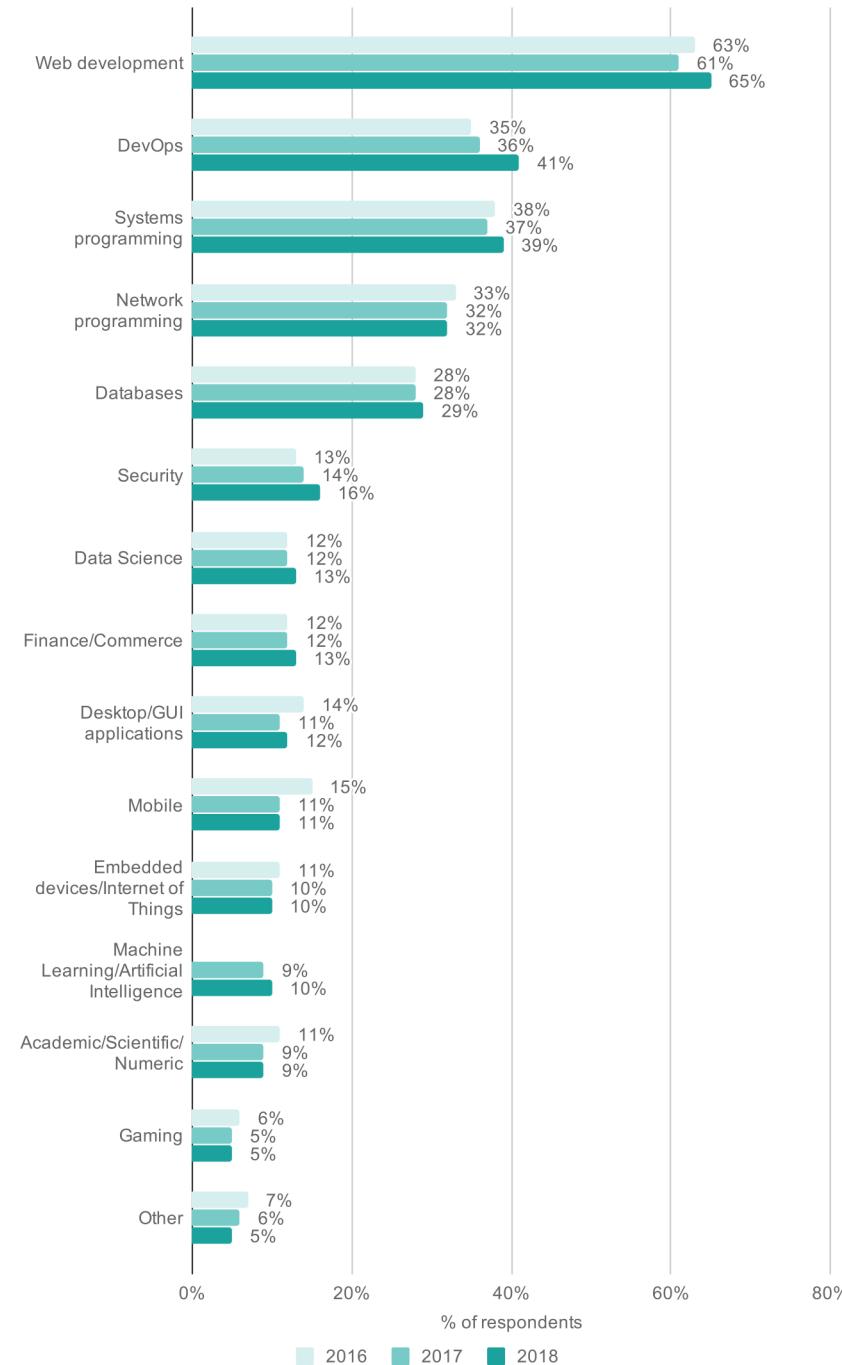
Communication

A close-up, black and white photograph of a young child's face. The child has dark hair and is looking upwards and to the right with a thoughtful expression. Their hands are resting on a dark wooden piano keyboard. The background is blurred, showing the interior of a room with a piano and some furniture.

What can you do?

<https://blog.golang.org/survey2018-results>

I work in the following areas: (select all that apply)



Getting Started



<https://github.com/jmhobbs/party-gopher>

CONTENTS

1. Intrudoction
2. Install
3. Basic Knowledge
4. RESTful API
5. HTML Template
6. Static Assets
7. Post Form
8. User Login
9. Cookie
10. Socket

Keywords

```
break    default    func    interface select
case    defer      go      map      struct
chan    else       goto    package  switch
const   fallthrough if      range    type
continue for        import  return  var
```

<https://golang.org/doc/install>

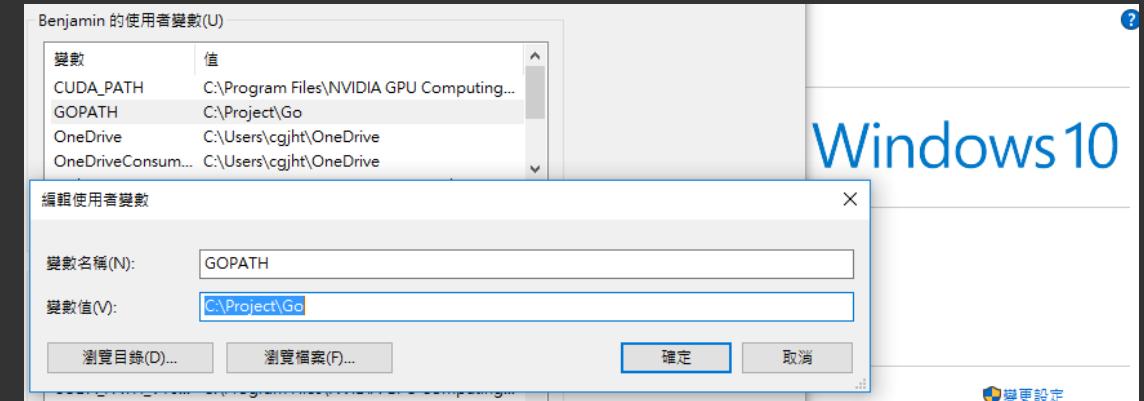
Linux, macOS

Edit \$HOME/.profile or .bash_profile or .bashrc

```
export GOROOT=/usr/local/go
export GOPATH=[PathToProject]
export PATH=$PATH:$GOROOT/bin
```

```
[~] ~ cat .bashrc
alias l="ls"
alias ll="ls -al"
export GOROOT=/usr/local/go
export GOPATH=$HOME/go
export PATH=$PATH:$GOROOT/bin
[~] ~
```

Windows



\$GOPATH/src/hello/hello.go

```
package main  
  
import "fmt"  
  
func main() {  
    fmt.Println("hello, go")  
}
```

```
→ hello go run hello.go  
hello, go  
→ hello go build  
→ hello ./hello  
hello, go  
→ hello █
```

In \$GOPATH, you must have three folders as follows

src: for source files whose suffix is .go, .c, .g, .s

pkg: for compiled files whose suffix is .a

bin: for executable files

Package

```
import (
    "fmt"
)

func main() {
    fmt.Println("hello go")
}
```

Define variables

```
var variableName type
var vname1, vname2, vname3 type
var (
    vname1 type
    vname2 type
    vname3 type
)
vname1, vname2, vname3 := v1, v2, v3
```

Error types

```
err := errors.New("emit macho dwarf: elf header  
corrupted")  
if err != nil {  
    fmt.Println(err)  
}
```

if

```
if x > 10 {  
    fmt.Println("x is greater than 10")  
} else {  
    fmt.Println("x is less than 10")  
}
```

```
if x := computedValue(); x > 10 {  
    fmt.Println("x is greater than 10")  
} else {  
    fmt.Println("x is less than 10")  
}
```

for

```
sum := 0;
for index:=0; index < 10 ; index++ {
    sum += index
}

for ; sum < 1000; {
    sum += sum
}

for sum < 1000 {
    sum += sum
}
```

for (cont.)

```
for k,v:=range map {  
    fmt.Println("map's key:",k)  
    fmt.Println("map's val:",v)  
}
```

```
for _, v := range map{  
    fmt.Println("map's val:", v)  
}
```

Functions

```
func max(a, b int) int {
    if a > b {
        return a
    }
    return b
}

func main() {
    x := 3
    y := 4

    max_xy := max(x, y) // call function max(x, y)

    fmt.Printf("max(%d, %d) = %d\n", x, y, max_xy)
}
```

Functions (cont.)

```
func SumAndProduct(A, B int) (int, int) {
    return A+B, A*B
}

func main() {
    x := 3
    y := 4

    xPLUSy, xTIMESy := SumAndProduct(x, y)

    fmt.Printf("%d + %d = %d\n", x, y, xPLUSy)
    fmt.Printf("%d * %d = %d\n", x, y, xTIMESy)
}
```

Functions (cont.)

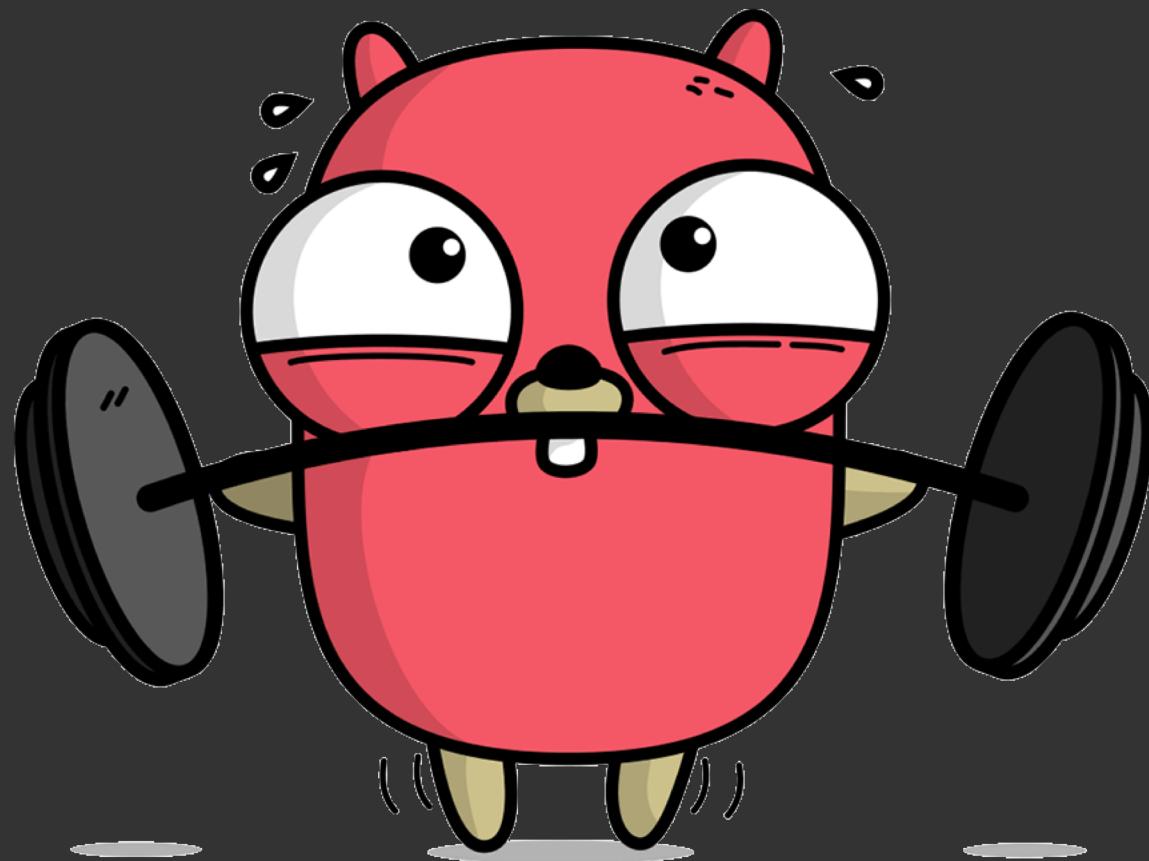
```
func SumAndProduct(A, B int) (add int, Multiplied int) {  
    add = A+B  
    Multiplied = A*B  
    return  
}
```

Struct

```
type person struct {  
    name string  
    age int  
}
```

```
var P person
```

```
P.name = "Astaxie"  
P.age = 25  
fmt.Printf("The person's name is %s", P.name)
```



<https://www.calhoun.io/guide-to-go/>