NCTU
Golang 2021

Lab 1

2021 / 09 / 23

Variable

How to declare a variable?

Variable

- var variable type
- var v1, v2, v3 type

Variables with initializers

- var variable type = value
- var v1, v2, v3 type = x1, x2, x3
- var v1, v2, v3 = v1, v2, v3

Short variable declarations

- v1 := x1

Variables – Zero values

- 0 for numeric types
- False for the Boolean type
- “” (the empty string) for strings

Variables – Zero values

```
package main

import "fmt"

func main() {
	var i int
	var f float64
	var b bool
	var s string
	fmt.Printf("%v %v %v %q\n", i, f, b, s)
}
```

Output:

Variables – Zero values

```
package main

import "fmt"

func main() {
	var i int
	var f float64
	var b bool
	var s string
	fmt.Printf("%v %v %v %q\n", i, f, b, s)
}
```

Output: 0 0 false “”

fmt.Printf
https://golang.org/pkg/fmt

Basic types

- bool : true false
- string : “Hello”
- int int8 int16 int32 int64 : 123 -123
- uint uint8 uint16 uint32 uint64 123
- byte [231 164 190 229 140 186]
- rune Unicode [31038 21306]
- float32 float64 1.5 -2.5
- complex64 complex128 3+2i

Basic types

```
package main

import (
        "fmt"
        "math/cmplx"
)

var (
        ToBe   bool       = false
        MaxInt uint64     = 1<<64 - 1
        z      complex128 = cmplx.Sqrt(-5 + 12i)
)

func main() {
        fmt.Printf("Type: %T Value: %v\n", ToBe, ToBe)
        fmt.Printf("Type: %T Value: %v\n", MaxInt, MaxInt)
        fmt.Printf("Type: %T Value: %v\n", z, z)
}
```

```
output :
Type: bool Value: false
Type: uint64 Value:
18446744073709551615
Type: complex128 Value: (2+3i)
```

Type casting

```
package main

import "fmt"

func main() {
	i := 42
	f := float64(i)
	u := uint(f)
	fmt.Println(i/4)
	fmt.Println(f/4)
	fmt.Println(u/4)
}
```

Output:

Type casting

```
package main

import "fmt"

func main() {
	i := 42
	f := float64(i)
	u := uint(f)
	fmt.Println(i/4)
	fmt.Println(f/4)
	fmt.Println(u/4)
}
```

Output:
10
10.5
10

Function

How to declare function?

Function – example

```
package main

import "fmt"

func add(x int, y int) int {
        return x + y
}

func main() {
        fmt.Println(add(42, 13))
}
```

Function – abbreviation of input

x int, y int
can be abbreviated as
x, y int

Function – example2

```
package main

import "fmt"

func add(x, y int) int {
        return x + y
}

func main() {
        fmt.Println(add(42, 13))
}
```

Function – multiple return values

```
package main

import "fmt"

func swap(x, y string) (string, string) {
        return y, x
}

func main() {
        a, b := swap("hello", "world")
        fmt.Println(a, b)
        c, _ := swap(“No”, “Yes”)
        fmt.Println(c)
}
```

Function – name return values

```
package main

import "fmt"

func split(sum int) (x int, y int){
        x = sum * 4 / 9
        y = sum - x
        return
}

func main() {
        fmt.Println(split(17))
}
```

Lab 1-1 Broken Calculator

1. 完成lab1.go並把YourGithubUsername資料夾改成自己的github名稱
2. 除法的部份答案直接取整數
3. 把程式的四種功能output擷圖,命名為 學號_姓名_Lab1-1.jpg
4. 把圖檔跟lab1.go放在同一個資料夾
5. 像lab0一樣繳交,branch名稱中有lab0的部份須改為lab1

Gist: https://github.com/benchen216/NCTU-GoProgramming-2021

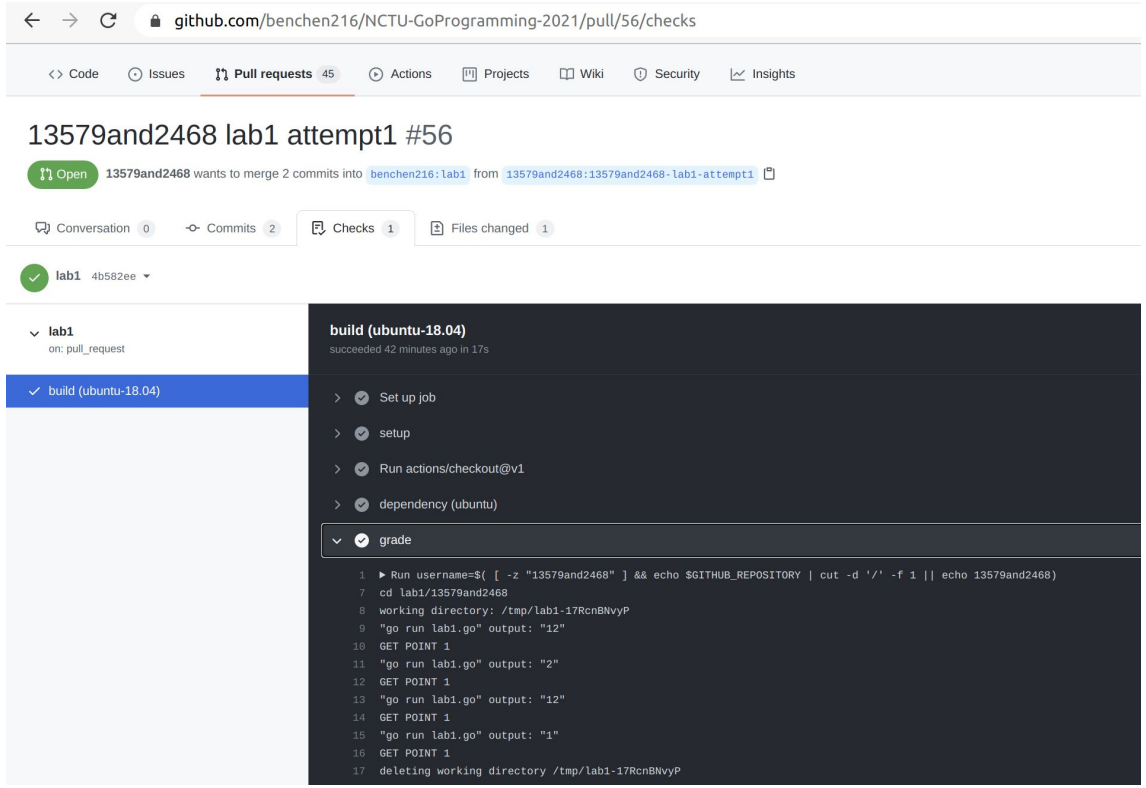
Check your program pass the testcase

綠色勾代表程式編譯有成果，點進去可以看測資是否有通過

可以在這裡看測試結果 ：https://github.com/benchen216/NCTU-GoProgramming-2021/pulls

Check your program pass the testcase

可以在grade看每筆測資的測試情況，或從validate.sh直接看測試資料



上次的作業成績

https://docs.google.com/spreadsheets/d/15HUiFbS6OUx0iyywR2mLib7igOOeHwtfpo9NOhXArK4