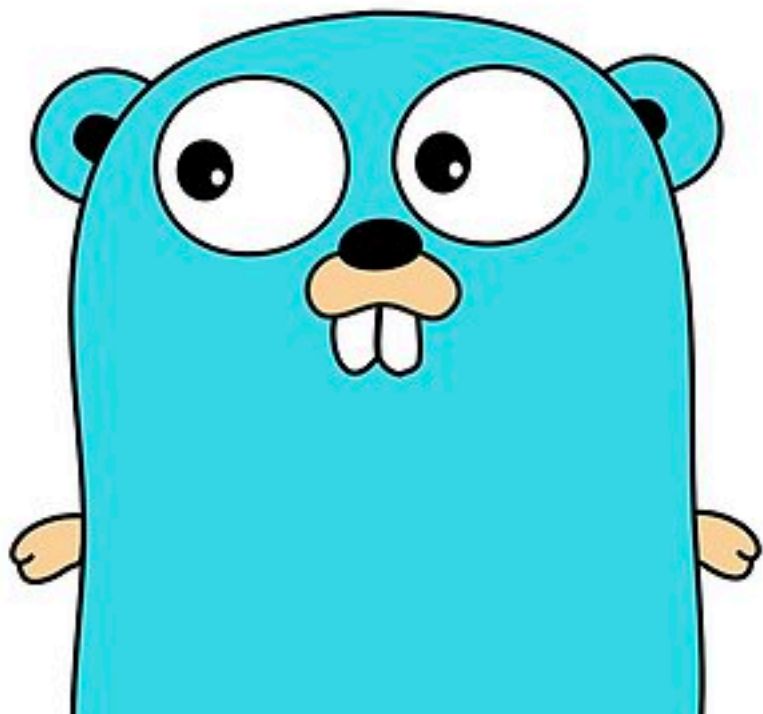


Don't panic()

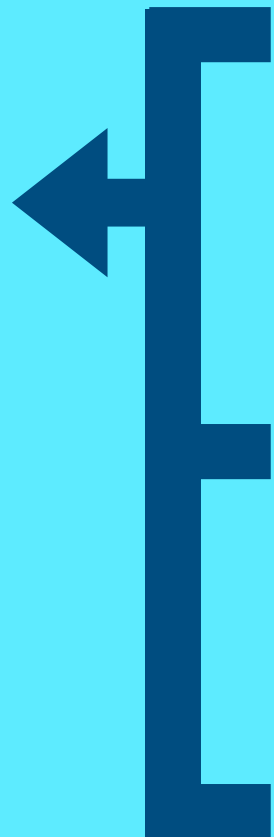
It's just an **error**



Merten van Gerven

# The **error** type

```
type error interface {  
    Error() string  
}
```



```
type MyError string  
  
func (err MyError) Error() string {  
    return err  
}  
  
MyError("houston, we have a problem!")  
  
—  
  
// What we have here is  
errors.New("a failure to communicate")  
  
—  
  
fmt.Errorf("i just can't do it captain");
```

# A typical use case

```
var err error

err = initLogger(options.LogLevel)
if err != nil {
    return fmt.Errorf(
        "failed to initialise logger with level '%s': %v",
        options.LogLevel,
        err,
    )
}

err = profiling.Start(
    "host.server.com:8086",
    "",
    "",
    "user",
    profiling.PrecisionMilliseconds,
)
if err != nil {
    return fmt.Errorf("failed to initialise profiling: %v", err)
}

err = initDB(options.MasterDSN)
if err != nil {
    return fmt.Errorf("failed to connect to master database: %v", err)
}

...
```

# To err is human...

Any action handled by a system that you can't control is a possible point of failure

When you can't control it, you CAN choose how to proceed after a failure.

Consider the severity of an **error** before passing it up the stack

But why can't I just panic()?

**IN THIS WE  
TRUST**



# Handling errors

The **error** should always be the last return value of a function

Errors can be **nil** and should be if there is no **error**

If you can safely recover from the **error**, you should.

Use panic() only when NOT doing so would cause further harm.

Always wrap errors you receive to provide a logical chain of execution - this just helps debug things.

# The state of errors

## Error handling and Go

<https://blog.golang.org/error-handling-and-go>

## Errors by example

<https://gobyexample.com/errors>

## Various packages for handling errors

<https://github.com/golang/go/wiki/Projects#error-handling>

## Marcel van Lohuizen

Go 2 error values today - a proposal for Go 1.13

<https://www.youtube.com/watch?v=SeVxmQl9Wmk>

# import "github.com/whitecypher/work"

```
work := work.Make()

// init logger
work.Add(func() (err error) {
    return initLogger(options.LogLevel)
})

// init profiler
work.Add(func() (err error) {
    return profiling.Start(
        "host.server.com:8086",
        "",
        "",
        "user",
        profiling.PrecisionMilliseconds,
    )
})

// init db
work.Add(func() (err error) {
    return initDB(options.MasterDSN)
})

err := work.Do()
if err != nil {
    fmt.Println(fmt.Errorf("the #@!$ hit the fan: %v", err))
}
```





Thanks for listening

[merten.vg@gmail.com](mailto:merten.vg@gmail.com)