Returning Errors

This lesson talks about Displaying error messages in Go and use of fmt package for printing

WE'LL COVER THE FOLLOWING

Returning an Error Message

In the previous lessons, we learned how to implement interfaces in Go. We will now look at how we can display error messages within them. An **error** is anything that can describe itself as an error string. The idea is captured by the predefined, built-in interface type, error, with its single method, `Error`, returning a string:

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

The `fmt` Package

The `fmt` package's various print routines automatically know to call the method when asked to print an error.

```
package main

import (
    "fmt"
    "time"
)

type MyError struct {
    When time.Time
    What string
}

func (e *MyError) Error() string {
    return fmt.Sprintf("at %v, %s",
        e.When, e.What)
}


func run() error {
    return &MyError{
        time.Now(),
        "it didn't work",
    }
}

func main() {
    if err := run(); err != nil {
        fmt.Println(err)
    }
}
```

To better enhance your understanding of this concept, an exercise on errors is provided at the end of the chapter.