## **Exercise on Errors**

Here is an exercise you can solve to test your understanding!

WE'LL COVER THE FOLLOWING ^

Assignment

## Assignment #

Copy your Sqrt function from the earlier exercises and modify it to return an error value. (This online assignment may be useful)

Sqrt should return a non-nil error value when given a negative number, as it doesn't support complex numbers.

Create a new type

type ErrNegativeSqrt float64

and make it an error by giving it a

func (e ErrNegativeSqrt) Error() string

method such that ErrNegativeSqrt(-2).Error() returns

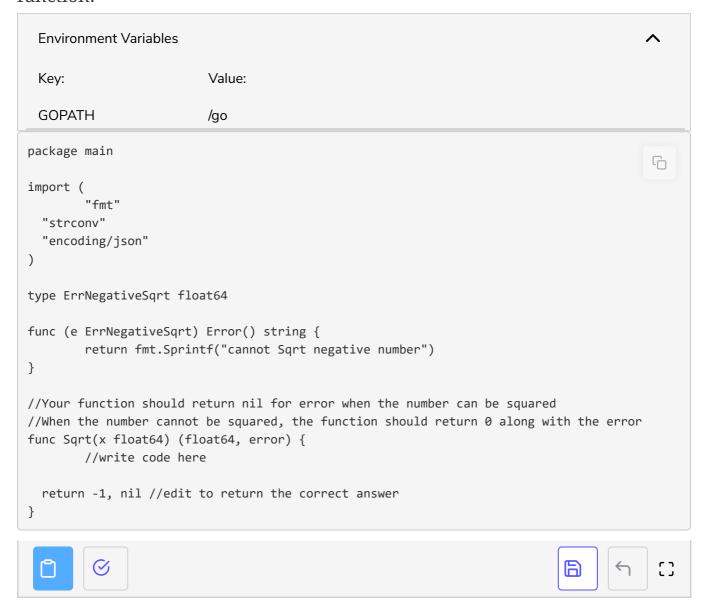
cannot Sqrt negative number: -2.

Note: a call to fmt.Print(e) inside the Error method will send the program into an infinite loop. You can avoid this by converting e first:

fmt.Print(float64(e)). Why?

Change your Sqrt function to return an ErrNegativeSqrt value when given a negative number.

Given below is some starter code, add your code to implement the required function:



**Tip**: When doing an inferred declaration of a float, you can omit the decimal value and do the following:

```
z := 1.

// same as

// z := 1.0
```