

## QUIZ

Define a constant in Go called `bestBook` and assign it the value `"Candide"`.

```
package main

import "fmt"

func main() {
    const bestBook = "Candide"
}
```



You got it!

Which of the following would be considered a literal in Go?

`uint16`

`const literalVal = 13.89`

`var literalVal literal`

`13.89`



Correct! A number or string without a name is called a "literal".

What does the number `5` refer to in the following error message:

```
./main.go:5:7: eyeColor declared and not used
```

5 is the *number of times* this code raised an error.

5 is the *column number* that the error occurs in.

5 is the *line number* that the error occurs on!



Correct! Error messages first display the file name, then the line, then the column, then the message indicating what has gone wrong.

Which of the following types allows for a value range from -128 to 127?

uint8

float32

int8



Correct! `int8` can be any integer value between -128 and 127

complex64

Which of the following keywords is necessary to create a new variable in Go without immediately assigning a value to it?

new

const

let

var



Correct! `var` `variableName` `type` is the way to create such a variable.

Which of the following types is not a Go builtin?

int64

bool

complex64

float8



Go only supports two float types: `float32` and `float64`.

What does the `:=` operator do in Go?

Assigns a new value to a constant.

`:=` is not a legal Go operator.

Creates a new variable with the value appearing afterwards and infers type based on that value.



Correct! `:=` is used to infer type and create a new variable.

Update a string literal with a new message.

What would the following code print out?

```
package main
import "fmt"

func main() {
    var emptyString string
    fmt.Println(emptyString)
}
```

"emptyString"

This error: `./main.go:6:15 emptyString not defined`

Nothing!



Correct! This code would print out the "empty string" which is indistinguishable from it printing nothing.

What is wrong with this block of code?

```
package main

import "fmt"

func main() {
    var coolMessage string
    coolMessage = 'Hello and welcome to Go!'
    fmt.Println(coolMessage)
}
```

Assignment of a string value to `coolMessage` should use double-quotes (") instead of single-quotes (')



Correct! Single-quotes are for a specific character (also called a *rune*), but string values need double-quotes.