NUMERIC DATA TYPES IN GO

What are Numeric Values

Numeric data types in programming are data types that represent numeric values. We have three numeric data types in Go:

- integers,
- floating point,
- and complex numbers

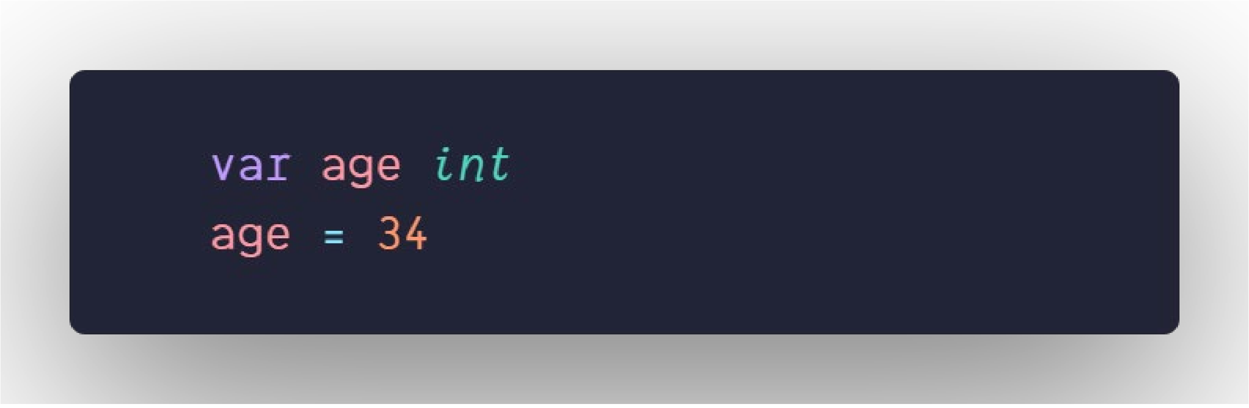
Integers

Integers represent a whole number.

Zero value is 0.

Default type is int.

Integer declaration:



```
var age int
age = 34
```

We have two types of Integers:

- Signed integers hold negative, zero and positive whole numbers
- Unsigned integers hold zero and positive whole numbers

Signed Integer Types

Type Name	Min Value	Max Value
int8	-128	127
int16	-32768	32767
int32	-2147483648	2147483647
int64	-9223372036854775808	9223372036854775807
int	either 32 or 64 bits (int32 or int64)	

Unsigned Integer Types

Type Name	Min Value	Max Value
uint8	0	255
uint16	0	65535
uint32	0	4294967295
uint64	0	18446744073709551615
uint	either 32 or 64 bits (uint32 or uint64)	
uintptr	contain the bit pattern of any pointer	

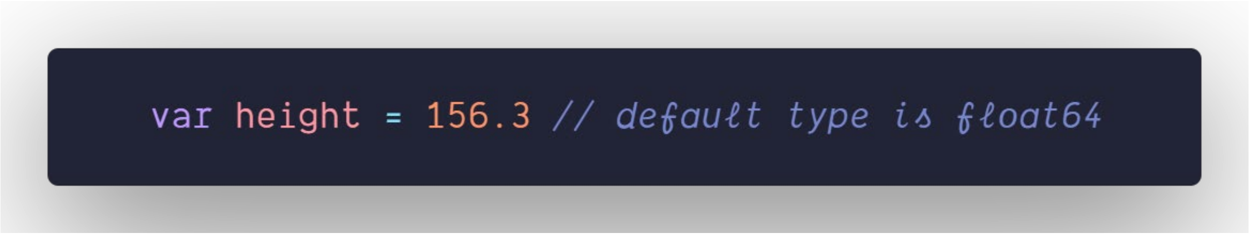
We have two type aliases:

- byte which is an alias for uint8
- rune which is an alias for int32

Floating-point Numbers

Floating point represents a number with a decimal point.

Zero value is 0.



```
var height = 156.3 // default type is float64
```

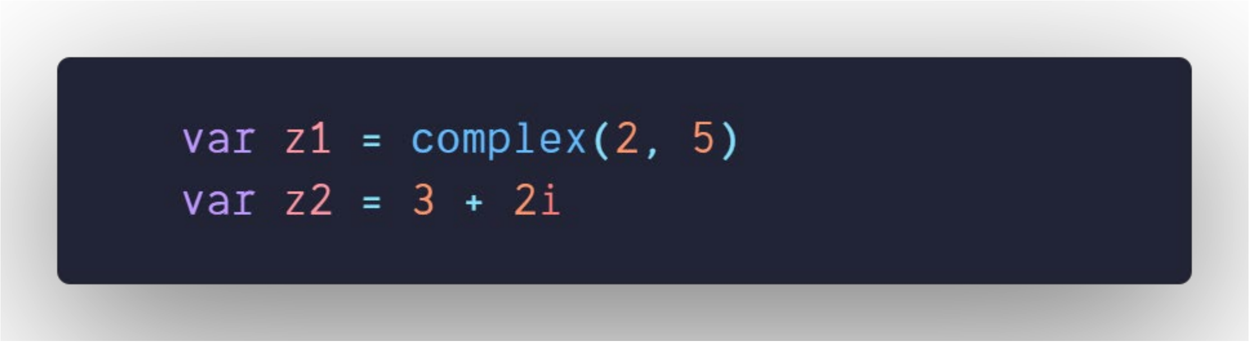
Type Name	Min Value	Max Value
float32	-3.4e+38	3.4e+38
float64	-1.7e+308	1.7e+308

Complex Numbers

Zero value is 0 for both real and imaginary part.

Default type is complex128.

We can declare complex numbers using complex built-in function or initialization syntax:



```
var z1 = complex(2, 5)
var z2 = 3 + 2i
```

Type Name	Description
complex64	32-bit floating point for real and imaginary parts
complex128	64-bit floating point for real and imaginary parts

Reflect Package

The reflect.TypeOf() function can tell us about the type of any value. We can use a Zero function to check what is the zero value for a particular type.

Link to the package documentation: https://pkg.go.dev/reflect