Variables,Values and Types

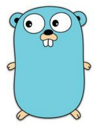
Saber Mesgari

Variable

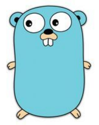
- A storage unit of a particular data type.

- Golang Is Static Type

- Golang Types:
  https://gist.github.com/thatisuday/c17e05de591c2e2021ab402e4c2d4bdc



bool	Boolean data type. It can store value `true` or `false`.	false
string	String data type. It can store UTF-8 string. All strings in go are UTF-8 by default. Unlike JavaScript, string is only encapsulated in double quotes.	empty string
int	Integer data type. It can store 32-bit or 64-bit signed integer. A 32-bit system will allocate 32 bits of memory and 64-bit system will allocate 64 bits of memory. Hence 32-bit system can store -2147483648 to 2147483647 and 64-bit system can store -9223372036854775808 to 9223372036854775807	0
uint	Integer data type. Same as `int`, `uint` can store 32 bits or 64 bits **unsigned** integer.	0
int8	Integer data type. System will allocate 8 bits of memory to store an integer. Hence it can store values between -128 to 127.	0
uint8	Integer data type. Same as `int8`, `uint8` can store 8-bit **unsigned** integer. Hence it can store values between 0 to 255.	0
int16	Integer data type. System will allocate 16 bits of memory to store an integer. Hence it can store values between -32768 to 32767.	0
uint16	Integer data type. Same as `int8`, `uint8` can store 16-bit **unsigned** integer. Hence it can store values between 0 to 65535.	0
int32	Integer data type. System will allocate 32 bits of memory to store an integer. Hence it can store values between -2147483648 to 2147483647.	0
uint32	Integer data type. Same as `int8`, `uint8` can store 32-bit **unsigned** integer. Hence it can store values between 0 to 4294967295.	0
int64	Integer data type. System will allocate 64 bits of memory to store an integer. Hence it can store values between -9223372036854775808 to 922337203685477580.	0
uint64	Integer data type. Same as `int8`, `uint8` can store 64-bit **unsigned** integer. Hence it can store values between 0 to 18446744073709551615.	0
float32	Float data type. System will allocate 32 bits of memory to store a float value. Hence it can store values between -3.4E+38 to +3.4E+38.	0
float64	Float data type. System will allocate 64 bits of memory to store a float value. Hence it can store values between -1.7E+308 to +1.7E+308.	0
complex64	Go supports complex numbers out of this box. `complex64` has `float32` real part and	0+0i



Define Variables

```go
var variableName dataType = initialValue

var integer1 int = 15
var integer2 int8 = -25
var integer3 int32 // default 0
var float1 float32 = 63.2
var string1 string = "Hello World!"
var boolean1 bool // default false
var boolean2 bool = true
```

Type Inference

```
var variableName = initialValue
```

```
var integer1 = 52 // int
var string1 = "Hello World" // string
var boolean1 = false // bool
```

Short Hand Notation

```
variableName := initialValue
```

```
integer1 := 52 //int
string1 := "Hello World" //string
boolean1 := false //bool
```

Multiple Variable Declaration

```
var var1, var2, var3 int



var var1, var2, var3 int = 1, 2, 3
var var1, var2, var3 = 1, 2.2, false
var1, var2, var3 := 1, 2.2, false


var (
    var1        = 50
    var2        = 25.22
    var3 string = "Telefonía"
    var4 bool
)
```

Type Conversion

```go
var1 := 10 // int
var2 := 10.5 // float64
// illegal
// var3 := var1 + var2
// legal
var3 := var1 + int(var2) // var3 => 20



var1 := "Hello"
var2 := []int32(va1)
```