

Golang Session

- Harsh Dusane

Topic : Variables

Variables

- Golang is statically typed language, this means that when go lang variables are declared, they either explicitly or implicitly assigned a type even before your program runs.
- You can assign new value to an existing variable, but the value need to be of same type.
- A variable declared within brace brackets {} may be accessed anywhere within the block. The opening curly brace { introduces a new scope that ends with a closing brace }. Inner blocks can access variables within outer blocks. Outer blocks cannot access variables within inner blocks.

Declaring Golang Variables

```
package main

import "fmt"

func main() {
    var i int
    var s string

    i = 10
    s = "Canada"

    fmt.Println(i)
    fmt.Println(s)
}
```

```
package main

import "fmt"

func main() {
    var i int = 10
    var s string = "Canada"

    fmt.Println(i)
    fmt.Println(s)
}
```

```
package main

import (
    "fmt"
    "reflect"
)

func main() {
    var i = 10
    var s = "Canada"

    fmt.Println(reflect.TypeOf(i))
    fmt.Println(reflect.TypeOf(s))
}
```

Short Hand Variable Declaration

```
package main
```

```
import (  
    "fmt"  
    "reflect"  
)
```

```
func main() {  
    name := "John Doe"  
    fmt.Println(reflect.TypeOf(name))  
}
```

```
package main
```

```
import (  
    "fmt"  
)
```

```
func main() {  
    var fname, lname string = "John", "Doe"  
    m, n, o := 1, 2, 3  
    item, price := "Mobile", 2000  
  
    fmt.Println(fname + lname)  
    fmt.Println(m + n + o)  
    fmt.Println(item, "-", price)  
}
```

Scope of Golang Variables Defined by Brace Brackets

```
package main
import "fmt"
var s = "Japan"
func main() {
    fmt.Println(s)
    x := true
    if x {
        y := 1
        if x != false {
            fmt.Println(x)
            fmt.Println(y)
        }
    }
    fmt.Println(x)
}
```

Zero Values

```
package main
```

```
import "fmt"
```

```
func main() {  
    var quantity float32  
    fmt.Println(quantity)
```

```
  
    var price int16  
    fmt.Println(price)
```

```
  
    var product string  
    fmt.Println(product)
```

```
  
    var inStock bool  
    fmt.Println(inStock)
```

```
}
```

Golang Variable Declaration Block

```
package main
```

```
import "fmt"
```

```
var (  
    product = "Mobile"  
    quantity = 50  
    price    = 50.50  
    inStock  = true  
)
```

```
func main() {  
    fmt.Println(quantity)  
    fmt.Println(price)  
    fmt.Println(product)  
    fmt.Println(inStock)  
}
```