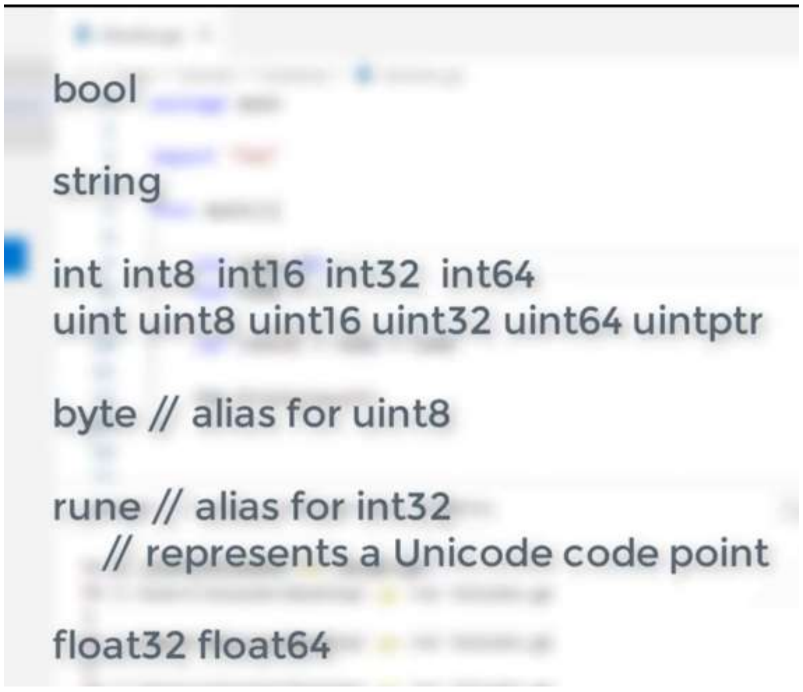
Golang Language Notes

Hello World Program

```
package main  // package that we have created

import "fmt"

func main(){
    fmt.Println("Hello World!")
//Println is a buitin function that we use from fmt package
 }

Output

Hello World!

// in "Go" if we declare any pacakage or dataType then we have to use it other
 it's give error
```

Data Types



```
bool

string

int  int8  int16  int32  int64
uint uint8 uint16 uint32 uint64 uintptr

byte // alias for uint8

rune // alias for int32
     // represents a Unicode code point

float32 float64
```

- ➢ Uint -> support +tive numbers
- ➢ Int -> support data range

Variables

```
package main

import "fmt"

func main(){
    var num=2   // it’s also write as num :=9

    var num3 int
    var num4 int = 4
    var num1,num2 int
    const num9 =9  // value remains constant and not change
    num1,num2=1,2
    fmt.Print(2+3)
    fmt.Print(num)
    fmt.Print(num1+num2)
    fmt.Print(num3)
    fmt.Print(num4)
    fmt.Print(num9)
    num7 :=7
    fmt.Print(num7)
}

Output

5230497
```

Variable Scope

```
package main

import "fmt"

var n=5

func local(){
    fmt.Println(n)
    var n=6
    fmt.Println(n)
}

func main(){
    local()
    fmt.Println(n)
}

Output

6
5
6
```

Loop

- I Go we have only 1 loop i.e. "**For Loop**"

```go
package main

import "fmt"

//in Go we Have only One Loop i.e. FOR LOOP

func main(){
    i :=1
    for i<5 {
        fmt.Println("Loop",i)
        i++
    }

    for j :=5;j<10;j++ {
        fmt.Println("Loops",j)
    }
}
```

Exported built-in function

- Outside main we can use only those Bulit-in functions which start from Capital letter (ex. Use "Println()" but not "toLarge()")

```go
package main

import "fmt"

func main(){
    fmt.Println("hii")
    fmt.toLarge(1000000)
    Demo()
}

func Demo(){
    fmt.Println("user")
}
```

Function

```go
package main

import "fmt"

// add(a int,b int) == add(a,b int)
// we have to specify what type of data we have return
// in Go we have retuen more than 1 values
func add(a int,b int) int{
    out := a+b
    return out
}



func calc(a, b int) (int, int) {
    out1 :=a+b
    out2 :=a-b
    return out1,out2
}

func main(){
    num1 :=1
    num2 :=2
    result := add(num1,num2)
    result1, result2 := calc(num1, num2)
    fmt.Println(result)
    fmt.Println(result1, result2)
}
```

Math Package

- Round – Round of the result
- Ceil – gives greater number
- Floor – gives lower number
- Printf – to print floor value

```
package main

import (
    "fmt"
    "math"
)

func main(){
    var num float64=12
    result :=math.Sqrt(num)
    fmt.Println(result)
    fmt.Printf("%.2f",result) // allow 2 digit agter decimal
    fmt.Printf("%.2g",result)  // it give more precision output
    fmt.Println()
    var intResult1,intResult2,intResult3 =math.Round(result),math.Ceil(result)
,math.Floor(result)
    fmt.Println(intResult1,intResult2,intResult3)
}

Output

3.4641016151377544
3.463.5
3 4 3
```

If , Else , Switch

```
package main

import "fmt"

func main(){
    var num=2
    if(num==2){
        fmt.Println("Equal to",num)
    } else if(num>2){
        fmt.Println("Greater than",2)
    } else{
        fmt.Println("Lesser than",2)
    }
    switch num {
    case 1:
        fmt.Println("One")
    case 2:
        fmt.Println("Two")
    default:
        fmt.Println("None")
    }
}
```

Defer Function

- If we use "defer" than that statement executes first
- If there is more than one "defer" than statement call first which declare at last

```
package main

import "fmt"

func main(){
    a()
}

func a(){
    defer d()
    fmt.Println("a begins")
    defer b()
    fmt.Println("a ends")
}

func b(){
    fmt.Println("in b")
}

func d(){
    fmt.Println("in d")
}

Output
```

```
a begins
a ends
in b
in d
```

➢ For loop using defer

```
package main

import "fmt"

func main(){
    a()
    fmt.Println()
    c()
    for i :=1;i<5;i++ {
        defer fmt.Println(i)
    }
    fmt.Println("Numbers")
}

Output

Numbers
4
3
2
1
```

Struct

```go
package main

import "fmt"

type Student struct {
    rollno int
    name string
}

func main(){
    var s1 = Student{170,"Vaibhav"}
    fmt.Println(s1,s1.rollno,s1.name)
    var s2 = Student{rollno: 171,name: "Vaibhav"}
    fmt.Println(s2,s2.rollno,s2.name)
}
```