**Formal parameters**

Formal parameters says always stick to value in main variable if we used the same value in any other function.

Let us say  variable a is declared in global and local both

And same called by the function

Then function will take priority from local only. See the below program

Formal parameters are treated as local variables with-in that function and they take preference over the global variables. For example −

package main

import "fmt"

/\* global variable declaration \*/

var a int = 20;

func main() {

   /\* local variable declaration in main function \*/

   var a int = 10    **// This value is always a preference**

   var b int = 20

   var c int = 0

   fmt.Printf("value of a in main() = %d\n",  a);

   c = sum( a, b);

   fmt.Printf("value of c in main() = %d\n",  c);

}

/\* function to add two integers \*/

func sum(a, b int) int {

   fmt.Printf("value of a in sum() = %d\n",  a);

   fmt.Printf("value of b in sum() = %d\n",  b);

   return a + b;

}

value of a in main() = 10

value of a in sum() = 10

value of b in sum() = 20

value of c in main() = 30

For loop as while loop

package main

import "fmt"

func main() {

    var i int32

    i = 0

    for i < 5 {

        fmt.Println("This loop runs five time")

        i++

    }

}

This loop runs five time

This loop runs five time

This loop runs five time

This loop runs five time

This loop runs five time

For loop as a do while loop

//There is no do while loop in the go

// There are few ways with the help of for loop we can define do loop

package main

import "fmt"

func main() {

    var i int = 0

    for {

        fmt.Println("This loop will run 5 times", i)

        i++

        if i >= 5 {

            break

        }

    }

}

PS C:\Go\_WorkSpace\forasdowhile> go run forasdowhile.go

This loop will run 5 times 0

This loop will run 5 times 1

This loop will run 5 times 2

This loop will run 5 times 3

This loop will run 5 times 4

Break statement in go

package main

import "fmt"

func main() {

    var i int = 10

    for {

        fmt.Println(i)

        i++

        if i > 15 {

            break

        }

    }

}

PS C:\Go\_WorkSpace\breakloop> go run break.go

10

11

12

13

14

15

/\*package main

import "fmt"

func main() {

    var i int = 10

    for i < 20 {

        i++

        fmt.Println(i)

        //continue

    }

}\*/

package main

import "fmt"

func main() {

    /\* local variable definition \*/

    var a int = 10

    /\* do loop execution \*/

    for a < 20 {

        if a == 15 {

            /\* skip the iteration \*/

            a = a + 1

            continue

        }

        fmt.Printf("value of a: %d\n", a)

        a++

    }

}

value of a: 10

value of a: 11

value of a: 12

value of a: 13

value of a: 14

value of a: 16

value of a: 17

value of a: 18

value of a: 19

Go to statement

package main

import "fmt"

func main() {

    learnGoTo()

}

func learnGoTo() {

    fmt.Println("a")

    goto FINISH

    fmt.Println("b")

FINISH:

    fmt.Println("c")

}

PS C:\Go\_WorkSpace\goto> go run goto.go

a

c