

Go If-else

Go if statement :- the if statement in Go is used to test the condition. If it evaluates to true, the body of the statement is executed. If it evaluates to false, if block is skipped.

Syntax :

```
if(boolean_expresssion){  
    // body  
    // body will execute only if the expression  
    result is true.  
}
```

Example :-

```
package main  
  
import "fmt"  
  
func main() {  
  
    var a int = 10  
    if a%2 == 0 {  
        fmt.Println("a is even number")  
    }  
}
```

Output :-

```
a is even number
```

Go if-else :- The if-else is used to test the condition. If condition is true, if block will executed otherwise else block will execute.

Syntax:-

```
if(boolean_expression){  
    // statement  
    // execute only if boolean expression is true  
}  
else{  
    //statement  
    // execute if boolean expression is false  
}
```

Example :-

```
package main  
  
import "fmt"  
  
func main() {  
    var a int = 10  
    if a%2 == 0 {  
        fmt.Println("a is even")  
    } else {  
        fmt.Println("a is odd")  
    }  
}
```

Output:-

```
a is even
```

Go If else-if :- If else if statements are used to execute one statement from multiple conditions.

We can have N numbers of if-elseif statement.

Example :-

```
package main

import "fmt"

func main() {
    fmt.Println("Enter Marks")
    var input int
    fmt.Scanln(&input)

    if input < 0 || input > 100 {
        fmt.Print("Please enter valid number")
    } else if input >= 0 && input < 50 {
        fmt.Print("Fail")
    } else if input >= 50 && input < 60 {
        fmt.Print("D Grade")
    } else if input >= 60 && input < 70 {
        fmt.Print("C Grade")
    } else if input >= 70 && input < 80 {
        fmt.Print("B Grade")
    } else if input >= 80 && input < 90 {
        fmt.Print("A Grade")
    } else if input >= 90 && input <= 100 {
        fmt.Print("A+ Grade")
    }
}
```

```
Enter Marks
58
D Grade
```

Go Nested if-else :- We can also nest the if-else statement to execute one statement from multiple condition.

Syntax :-

```
if(boolean_expression 1){  
    // statement will execute only if  
expression 1 is true  
    if(boolean_expression 2){  
        // statement will execute only if  
expression 1 and expression 2 both are true  
    }  
}
```

Example :-

```
package main  
  
import "fmt"  
  
func main() {  
  
    var x int = 10  
    var y int = 20  
  
    if x >= 10 {  
        fmt.Print("In outer if condtion statement\n")  
        if y >= 10 {  
            fmt.Print("In nested if statement\n")  
        }  
    }  
  
    fmt.Println("Value of x is ", x)  
    fmt.Println("Value of y is ", y)  
}
```

Output:-

```
In outer if condtion statement  
In nested if statement  
Value of x is  10  
Value of y is  20
```