

Go Switch Statements

The switch statement executes one statement from multiple conditions. It is similar to if-else-if chain statement.

Syntax :-

```
switch var1{
    case val1:
        // statement
    case val2:
        // statement
    case val3:
        //statement
    default
        // default statement
}
```

- ➔ Switch statement in Go is more flexible. In the above syntax, var1 is a variable which can be of any type, and val1, val2, ... are possible values of var1.
- ➔ In switch statement, more than one values can be tested in a case, the values are presented in a comma separated list **like:** case val1, val2, val3:
- ➔ If any case is matched, the corresponding case statement is executed.

Example:-

```
package main

import "fmt"

func main() {
    fmt.Print("Enter a Number")
    var input int
    fmt.Scanln(&input)
    switch input {
    case 10:
        fmt.Print("The value is 10")
    case 20:
        fmt.Print("The value is 20")
    case 30:
        fmt.Print("The value is 30")
    }
```

```

    case 40:
        fmt.Print("The value is 40")
    default:
        fmt.Print("It is not 10,20,30,40 ")
    }
}

```

Output :-

```

Enter a Number10
The value is 10

```

Or

```

Enter a Number30
The value is 30

```

Switch statement with fall-through

- ➔ the break keyword is implicit. So automatic fall-through is not the default behaviour of switch statement in Go language.
- ➔ For fall-through in Go switch statement, use the keyword “fallthrough” at the end of the branch.

Example:-

```

package main

import "fmt"

func main() {
    var k = 30
    switch k {
    case 10:
        fmt.Print("<=10\n")
        fallthrough
    case 20:
        fmt.Print("<=20\n")
        fallthrough
    case 30:
        fmt.Print("<=30\n")
        fallthrough
    }
}

```

```
case 40:  
    fmt.Print("<=40\n")  
    fallthrough  
case 50:  
    fmt.Print("<=50\n")  
    fallthrough  
case 60:  
    fmt.Print("<=60\n")  
    fallthrough  
default:  
    fmt.Print("Default Case")  
}  
}
```

Output:-

```
<=30  
<=40  
<=50  
<=60  
Default Case
```