

Golang Session

- Harsh Dusane

Topic : Golang For Loops

Golang - traditional for Statement

```
package main

import "fmt"

func main() {

    k := 1
    for ; k <= 10; k++ {
        fmt.Println(k)
    }

    k = 1
    for k <= 10 {
        fmt.Println(k)
        k++
    }

    for k := 1; ; k++ {
        fmt.Println(k)
        if k == 10 {
            break
        }
    }
}
```

Golang - for range Statement

```
package main

import "fmt"

func main() {

    // Example 1
    strDict := map[string]string{"Japan": "Tokyo", "China": "Beijing", "Canada": "Ottawa"}
    for index, element := range strDict {
        fmt.Println("Index :", index, " Element :", element)
    }

    // Example 2
    for key := range strDict {
        fmt.Println(key)
    }

    // Example 3
    for _, value := range strDict {
        fmt.Println(value)
    }
}
```

Golang - range loop over string

```
package main

import "fmt"

func main() {
    for range "Hello" {
        fmt.Println("Hello")
    }
}
```

Golang - Infinite loop

```
package main

import "fmt"

func main() {
    i := 5
    for {
        fmt.Println("Hello")
        if i == 10 {
            break
        }
        i++
    }
}
```