Golang Session

- Harsh Dusane

Topic : Golang Maps

Map initialization and declaration

```
package main

import "fmt"

var employee = map[string]int{"Mark": 10, "Sandy": 20}

func main() {
        fmt.Println(employee)
}
```

```
package main

import "fmt"

func main() {
        var employee = map[string]int{}
        fmt.Println(employee)        // map[]
        fmt.Printf("%T\n", employee) // map[string]int
}
```

```
package main

import "fmt"

func main() {
        var employee = make(map[string]int)
        employee["Mark"] = 10
        employee["Sandy"] = 20
        fmt.Println(employee)

        employeeList := make(map[string]int)
        employeeList["Mark"] = 10
        employeeList["Sandy"] = 20
        fmt.Println(employeeList)
}
```

Maps Sample Codes – Create, Add, Update

```
package main

import "fmt"

func main() {
        var employee = make(map[string]int)
        employee["Mark"] = 10
        employee["Sandy"] = 20

        // Empty Map
        employeeList := make(map[string]int)

        fmt.Println(len(employee))     // 2
        fmt.Println(len(employeeList)) // 0
}
```

```
package main

import "fmt"

func main() {
        var employee = map[string]int{"Mark": 10, "Sandy": 20}

        fmt.Println(employee["Mark"])
}
```

```
package main

import "fmt"

func main() {
        var employee = map[string]int{"Mark": 10, "Sandy": 20}
        fmt.Println(employee) // Initial Map

        employee["Rocky"] = 30 // Add element
        employee["Josef"] = 40

        fmt.Println(employee)
}
```

```
package main

import "fmt"

func main() {
        var employee = map[string]int{"Mark": 10, "Sandy": 20}
        fmt.Println(employee) // Initial Map

        employee["Mark"] = 50 // Edit item
        fmt.Println(employee)
}
```

Maps – Delete Items

```
package main

import "fmt"

func main() {
        var employee = make(map[string]int)
        employee["Mark"] = 10
        employee["Sandy"] = 20
        employee["Rocky"] = 30
        employee["Josef"] = 40

        fmt.Println(employee)

        delete(employee, "Mark")
        fmt.Println(employee)
}
```

Iterate over a Map

```
package main

import "fmt"

func main() {
    var employee = map[string]int{"Mark": 10, "Sandy": 20,
        "Rocky": 30, "Rajiv": 40, "Kate": 50}
    for key, element := range employee {
        fmt.Println("Key:", key, "=>", "Element:", element)
    }
}
```

Truncate Map

```
package main

func main() {
        var employee = map[string]int{"Mark": 10, "Sandy": 20,
                "Rocky": 30, "Rajiv": 40, "Kate": 50}

        // Method - I
        for k := range employee {
                delete(employee, k)
        }

        // Method - II
        employee = make(map[string]int)
}
```

Sort Map Keys

```
package main

import (
        "fmt"
        "sort"
)

func main() {
        unSortedMap := map[string]int{"India": 20, "Canada": 70, "Germany": 15}

        keys := make([]string, 0, len(unSortedMap))

        for k := range unSortedMap {
                keys = append(keys, k)
        }
        sort.Strings(keys)

        for _, k := range keys {
                fmt.Println(k, unSortedMap[k])
        }
}
```

Sort Map Values

```
package main

import (
        "fmt"
        "sort"
)

func main() {
        unSortedMap := map[string]int{"India": 20, "Canada": 70, "Germany": 15}

 // Int slice to store values of map.
        values := make([]int, 0, len(unSortedMap))

        for _, v := range unSortedMap {
                values = append(values, v)
        }

 // Sort slice values.
        sort.Ints(values)

 // Print values of sorted Slice.
        for _, v := range values {
                fmt.Println(v)
        }
}
```

Merge Maps

```
package main

import "fmt"

func main() {
        first := map[string]int{"a": 1, "b": 2, "c": 3}
        second := map[string]int{"a": 1, "e": 5, "c": 3, "d": 4}

        for k, v := range second {
                first[k] = v
        }

        fmt.Println(first)
}
```