**Go Array**

Go supports data structor called array

Which store fixed sequential bytes of same type of element

Declaration of array

var var\_name [size] type.

var var\_name [size] type{value1, value2, value3}

var name[3] string

var balance = [5]float32{1.1, 2.3, 5.4, 17.5, 5.2}

var balance = []float32{1.1, 2.3, 5.4, 17.5, 5.2}

var balance[4] = 17.5

package main

import "fmt"

func main() {

    var n [11]int

    var i, j int

    for i = 0; i < 10; i++ {

        n[i] = i + 100

    }

    for j = 0; j < 10; j++ {

        fmt.Println(j, n[j])

    }

}

0 100

1 101

2 102

3 103

4 104

5 105

6 106

7 107

8 108

9 109

package main

import (

    "fmt"

)

func main() {

    array := []string{"my", "name", "is", "yagnik"}

    /\*

       array = []string

       array[0] = "my"

       array[1] = "name"

       array[2] = "is"

       array[3] = "yagnik"

       fmt.Println("Elements of Array:")

       fmt.Println("Element 1: ", array[2])

    \*/

    // printing simple array

    for i := 0; i < 4; i++ {

        fmt.Printf(array[i])

    }

}

mynameisyagnik

package main

import "fmt"

func main() {

    // 5 row 2 column

    a := [5][2]int{{0, 0}, {1, 5}, {9, 5}, {6, 2}, {7, 2}}

    for i := 0; i < 5; i++ {

        for j := 0; j < 2; j++ {

            fmt.Printf("a[%d][%d] = %d \n", i, j, a[i][j])

        }

    }

}

a[0][0] = 0

a[0][1] = 0

a[1][0] = 1

a[1][1] = 5

a[2][0] = 9

a[2][1] = 5

a[3][0] = 6

a[3][1] = 2

a[4][0] = 7

a[4][1] = 2

Go will allows multi dimensional array

var var\_name[size1] [size2] [size3] …. [sizen] variable\_type

 2D array

var arrayName [ x ][ y ] variable\_type

initialization of 2D array

a = [3] [4] int{

{0,1,2,3},

{4,5,3,6},

{8,4,3,7}

}