**Goroutines:**

func show\_value(i int){

for i<10{

fmt.Println(i)

i = i+1

}

}

func main() {

show\_value(0)

fmt.Println("We have called the function")

}

package main

import "fmt"

func show\_value(i int){

for i<10{

fmt.Println(i)

i = i+1

}

}

func main() {

go show\_value(0)

fmt.Println("We have called the function")

}

package main

import "fmt"

func show\_value(i int){

for i<10{

fmt.Println(i)

i = i+1

}

}

func main() {

go show\_value(0)

fmt.Println("We have called the function")

fmt.Scanln()

}

package main

import ("fmt"

"time"

)

func show\_value(i int){

for i<10{

fmt.Println(i)

time.Sleep(200 \* time.Millisecond)

i = i+1

}

}

func main() {

go show\_value(0)

fmt.Println("We have called a goroutine")

show\_value(0)

fmt.Println("We have called a function")

}

**Multiple goroutines:**

package main

import ("fmt"

"time"

)

func show\_value(i int){

for i<10{

fmt.Println(i)

time.Sleep(200 \* time.Millisecond)

i = i+1

}

}

func show\_value1(c int){

for c<110{

fmt.Println(c)

time.Sleep(200 \* time.Millisecond)

c = c+1

}

}

func main() {

go show\_value(0)

fmt.Println("We have called a goroutine show\_value")

go show\_value1(100)

fmt.Println("We have called a goroutine show\_value1")

fmt.Scanln()

}

**Wait for Goroutines:**

package main

import ("fmt"

"time"

"sync"

)

func show\_value(i int, wg \*sync.WaitGroup){

for i<10{

fmt.Println(i)

time.Sleep(200 \* time.Millisecond)

i = i+1

}

wg.Done()

}

func show\_value1(c int, wg \*sync.WaitGroup){

for c<110{

fmt.Println(c)

time.Sleep(200 \* time.Millisecond)

c = c+1

}

wg.Done()

}

func main() {

var wg sync.WaitGroup

wg.Add(2)

go show\_value(0, &wg)

fmt.Println("We have called a goroutine show\_value")

go show\_value1(100, &wg)

fmt.Println("We have called a goroutine show\_value1")

wg.Wait()

fmt.Println("Done")

}