

```

// =====
// 関数かけっこ
// fnA と fnB のどちらが先に終了するか
// =====

package main

import (
    "fmt"
    "math/rand"
    "time"
)

func main() {
    fmt.Println("関数かけっこ")

    // チャンネルの作成
    chA := make(chan string)
    chB := make(chan string)

    // ゴルーチン実行
    go fnA(chA)
    go fnB(chB)

    // どちらが先にチャンネルを受け取るか
    select {
    case msg := <-chA:
        // 受け取ったメッセージの表示
        fmt.Printf("%v\n", msg)
        fmt.Println("A の勝ち!")
    case msg := <-chB:
        // 受け取ったメッセージの表示
        fmt.Printf("%v\n", msg)
        fmt.Println("B の勝ち!")
    }
}

// 処理 A
func fnA(ch chan string) {
    r := rand.Intn(5000) // 1~5000 までの乱数を生成
    time.Sleep(time.Millisecond * time.Duration(r)) // 処理を r ミリ秒停止させる
    ch <- "function A is done" // 再開したらチャンネルを送信
}

// 処理 B
func fnB(ch chan string) {
    r := rand.Intn(5000) // 1~5000 までの乱数を生成
    time.Sleep(time.Millisecond * time.Duration(r)) // 処理を r ミリ秒停止させる
    ch <- "function B is done" // 再開したらチャンネルを送信
}

```