# 流程控制

作者：少林之巅

# 目录

1.if else语句块

2. for语句

3. switch语句

# if else语句

1. 基本语法

```
if condition {
    //do something
}
```

```
if condition {
   //do something
} else if condition {
   //do something
} else {
    //do something
}
```

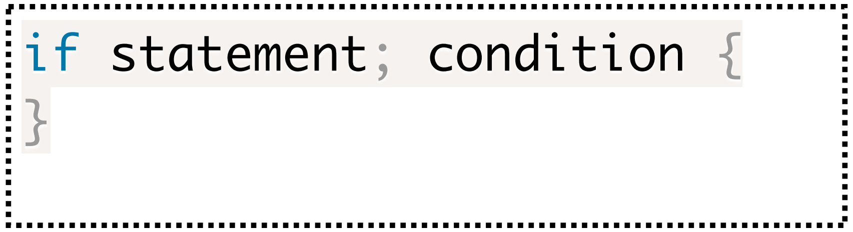
# if else语句

2. 练习一

```go
package main

import (
    "fmt"
)

func main() {
    num := 10
    if num % 2 == 0 { //checks if number is even
        fmt.Println("the number is even")
    }  else {
        fmt.Println("the number is odd")
    }
}
```

# if else语句

3. 基本语法



```
if statement; condition {
}
```

# if else语句

4. 练习二

```go
package main

import (
    "fmt"
)

func main() {
    if num := 10; num % 2 == 0 { //checks if number is even
        fmt.Println(num,"is even")
    }  else {
        fmt.Println(num,"is odd")
    }
}
```

# if else语句

5. 练习三

```go
package main

import (
    "fmt"
)

func main() {
    num := 99
    if num <= 50 {
        fmt.Println("number is less than or equal to 50")
    } else if num >= 51 && num <= 100 {
        fmt.Println("number is between 51 and 100")
    } else {
        fmt.Println("number is greater than 100")
    }

}
```

# 循环

1. Go语言中只有一种循环 for

```
for initialisation; condition; post {
}
```

# 循环

2. 练习一

```go
package main

import (
    "fmt"
)

func main() {
    for i := 1; i <= 10; i++ {
        fmt.Printf(" %d",i)
    }
}
```

# 循环

3. break，终止循环

```go
package main

import (
    "fmt"
)

func main() {
    for i := 1; i <= 10; i++ {
        if i > 5 {
            break //loop is terminated if i > 5
        }
        fmt.Printf("%d ", i)
    }
    fmt.Printf("\nline after for loop")
}
```

# 循环

4. continue，终止本次循环

```go
package main

import (
    "fmt"
)

func main() {
    for i := 1; i <= 10; i++ {
        if i%2 == 0 {
            continue
        }
        fmt.Printf("%d ", i)
    }
}
```

# 循环

5. 练习一

```go
package main

import (
    "fmt"
)

func main() {
    i := 0
    for ;i <= 10; { // initialisation and post are omitted
        fmt.Printf("%d ", i)
        i += 2
    }
}
```

# 循环

6. 练习二

```go
package main

import (
    "fmt"
)

func main() {
    i := 0
    for i <= 10 { // initialisation and post are omitted
        fmt.Printf("%d ", i)
        i += 2
    }
}
```

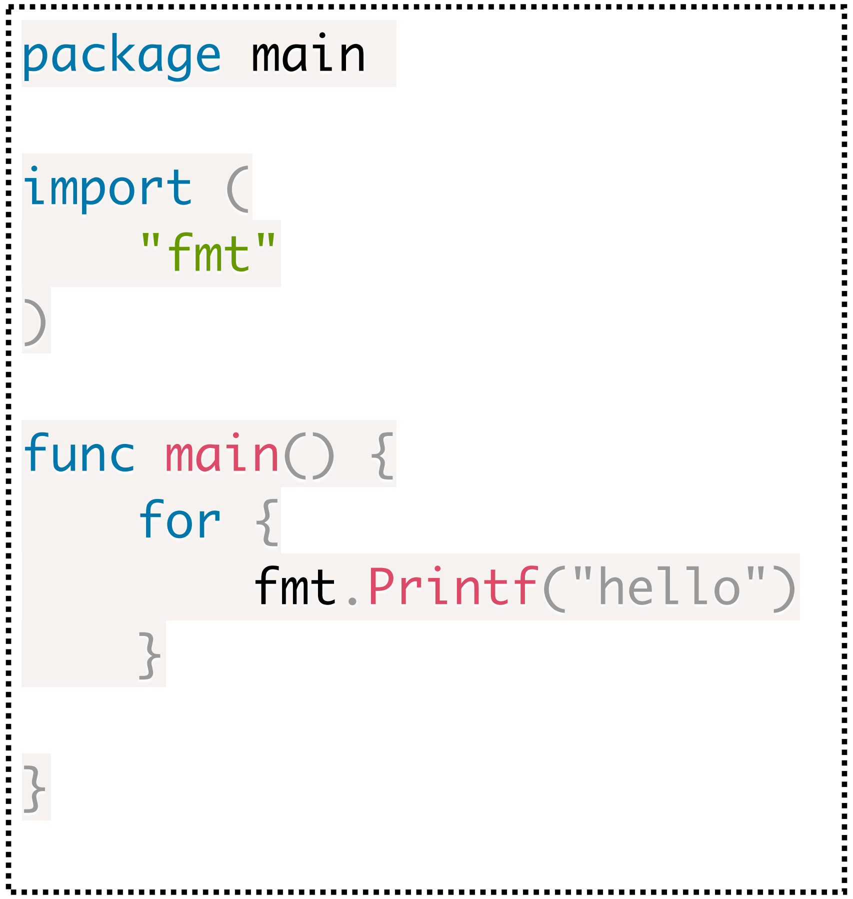
# 循环

## 7. 练习三

```go
package main

import (
    "fmt"
)

func main() {
    for no, i := 10, 1; i <= 10 && no <= 19; i, no = i+1, no+1 {
        fmt.Printf("%d * %d = %d\n", no, i, no*i)
    }

}
```

# 循环

8. 无限循环



```go
package main

import (
    "fmt"
)

func main() {
    for {
        fmt.Printf("hello")
    }

}
```

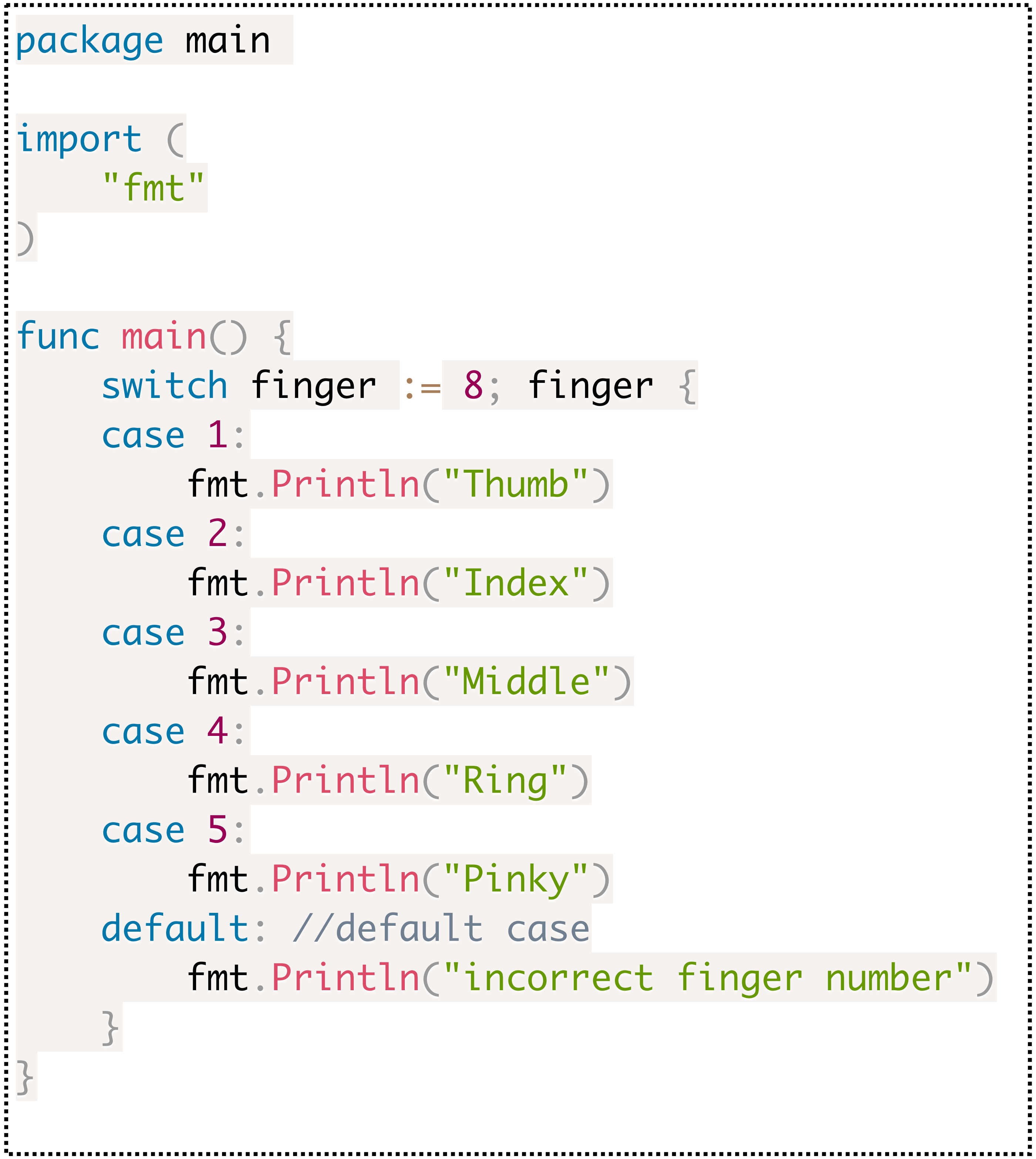
# switch语句

1. switch

```go
package main

import (
    "fmt"
)

func main() {
    finger := 4
    switch finger {
    case 1:
        fmt.Println("Thumb")
    case 2:
        fmt.Println("Index")
    case 3:
        fmt.Println("Middle")
    case 4:
        fmt.Println("Ring")
    case 5:
        fmt.Println("Pinky")

    }
}
```

# switch语句

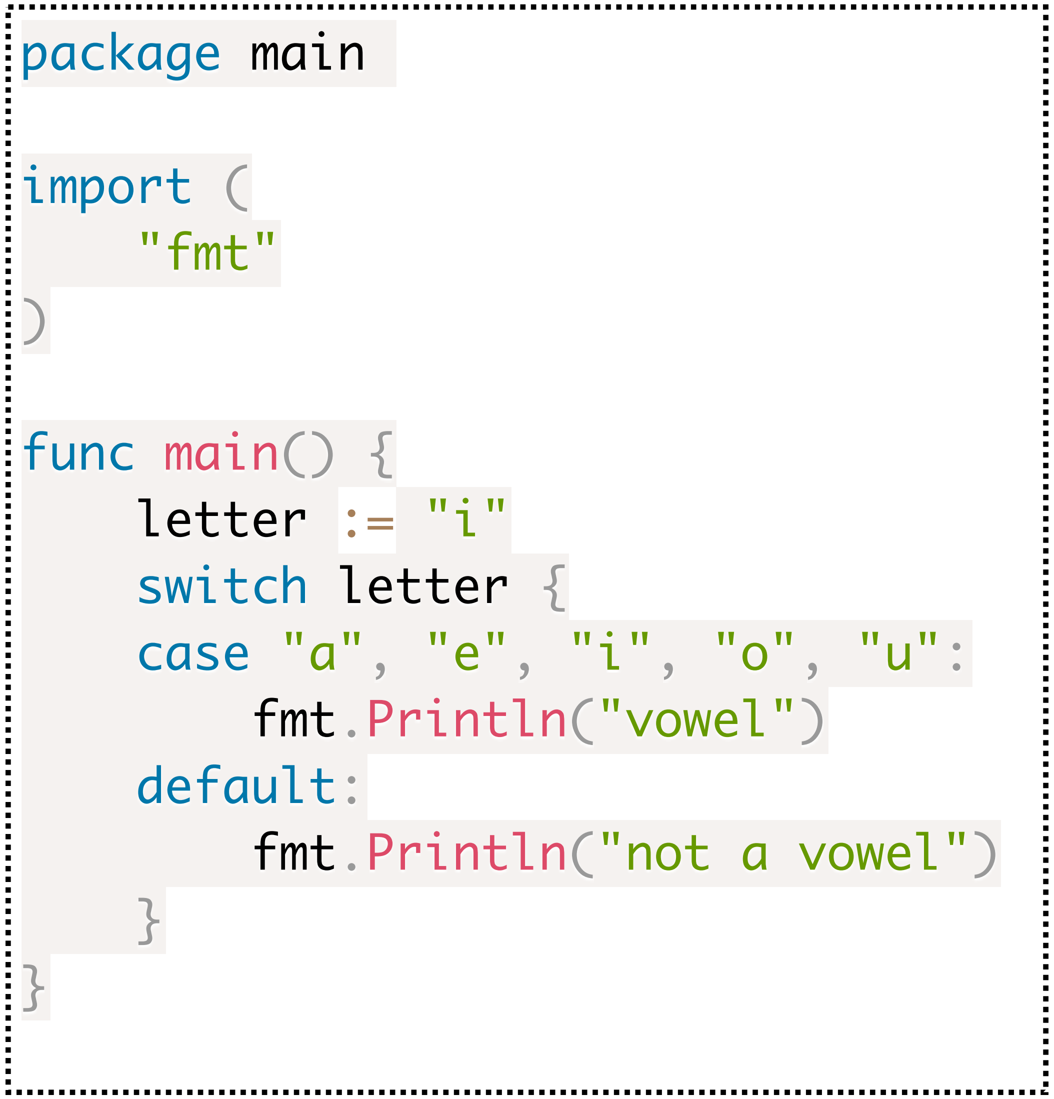
2. Switch default



```go
package main

import (
    "fmt"
)

func main() {
    switch finger := 8; finger {
    case 1:
        fmt.Println("Thumb")
    case 2:
        fmt.Println("Index")
    case 3:
        fmt.Println("Middle")
    case 4:
        fmt.Println("Ring")
    case 5:
        fmt.Println("Pinky")
    default: //default case
        fmt.Println("incorrect finger number")
    }
}
```

# switch语句

## 3. Switch



```go
package main

import (
    "fmt"
)

func main() {
    letter := "i"
    switch letter {
    case "a", "e", "i", "o", "u":
        fmt.Println("vowel")
    default:
        fmt.Println("not a vowel")
    }
}
```

# switch语句

4. Switch case 条件判断

```go
package main

import (
    "fmt"
)

func main() {
    num := 75
    switch { // expression is omitted
    case num >= 0 && num <= 50:
        fmt.Println("num is greater than 0 and less than 50")
    case num >= 51 && num <= 100:
        fmt.Println("num is greater than 51 and less than 100")
    case num >= 101:
        fmt.Println("num is greater than 100")
    }

}
```

# switch语句

5. Switch fallthrough

```go
package main

import (
    "fmt"
)

func number() int {
        num := 15 * 5
        return num
}

func main() {

    switch num := number(); { //num is not a constant
    case num < 50:
        fmt.Printf("%d is lesser than 50\n", num)
        fallthrough
    case num < 100:
        fmt.Printf("%d is lesser than 100\n", num)
        fallthrough
    case num < 200:
        fmt.Printf("%d is lesser than 200", num)
    }

}
```