



链滴

# Go 边看边练 - 《Go 学习笔记》系列 (五)

作者: [88250](#)

原文链接: <https://ld246.com/article/1438070631857>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## ToC

- [Go 边看边练 - 《Go 学习笔记》系列 \(一\) - 变量、常量](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(二\) - 类型、字符串](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(三\) - 指针](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(四\) - 控制流1](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(五\) - 控制流2](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(六\) - 函数](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(七\) - 错误处理](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(八\) - 数组、切片](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(九\) - Map、结构体](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(十\) - 方法](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(十一\) - 表达式](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(十二\) - 接口](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(十三\) - Goroutine](#)
  - [Go 边看边练 - 《Go 学习笔记》系列 \(十四\) - Channel](#)
- 

### 2.4.4 Switch

分支表达式可以是任意类型，不限于常量。可省略 `break`，默认自动终止。

```
x := []int{1, 2, 3}
i := 2

switch i {
  case x[1]:
    println("a")
  case 1, 3:
    println("b")
  default:
    println("c")
}
```

输出:

a

如需要继续下一分支，可使用 `fallthrough`，但不再判断条件。

```
x := 10

switch x {
  case 10:
    println("a")
    fallthrough
  case 0:
    println("b")
}
```

输出:

```
a
b
```

省略条件表达式, 可当 if...else if...else 使用。

```
switch {
  case x[1] > 0:
    println("a")
  case x[1] < 0:
    println("b")
  default:
    println("c")
}
```

```
switch i := x[2]; { // 带初始化语句
  case i > 0:
    println("a")
  case i < 0:
    println("b")
  default:
    println("c")
}
```

## 2.4.5 Goto, Break, Continue

支持在函数内 **goto** 跳转。标签名区分大小写, 未使用标签引发错误。

```
func main() {
  var i int
  for {
    fmt.Println(i)
    i++
    if i > 2 {
      goto BREAK
    }
  }
}
```

```
BREAK:
  fmt.Println("break")
```

```
EXIT: // Error: label EXIT defined and not used
}
```

配合标签, `break` 和 `continue` 可在多级嵌套循环中跳出。

```
<iframe style="border:1px solid" src="https://wide.b3log.org/playground/571e693b091b555dec40312b7d12ca9.go?embed=true" width="99%" height="600"></iframe>
```

附: `break` 可用于 `for`、`switch`、`select`, 而 `continue` 仅能用于 `for` 循环。

```
x := 100

switch {
  case x >= 0:
    if x == 0 { break }
    println(x)
}
```

下一篇: <https://hacpai.com/article/1438164538421>

- 
- 本系列是基于雨痕的《Go 学习笔记》(第四版) 整理汇编而成, 非常感谢雨痕的辛勤付出与分享!
  - 转载请注明: 文章转载自: 黑客与画家的社区 [<https://hacpai.com>]
  - 如果你觉得本章节做得不错, 请在下面打赏一下吧~

---

#### 社区小贴士

- 关注标签 [golang] 可以方便查看 Go 相关帖子
- 关注作者后如有新帖将会收到通知