Die Sprache Go

In fünf Minuten

by Hauke Stieler

15. Mai 2018

- Robert Griesemer *(JavaScript engine V8)*
- Rob Pike *(Plan 9, Inferno, Limbo, UTF-8)*
- Ken Thompson *(UNIX, shell)*

- wird kompiliert
- imperativ
- objektorientiert
- stark & statisch typisiert
- garbage Collection
- gute Unterstützung für Nebenläufigkeit

- `go build [file]`
- `go run file`
- `go test [file]`
- `go get [url]`

```go
package main

import "fmt"

func main() {
    fmt.Println("hello world")
}
```

Keywords:

`break`	`default`	`func`	`interface`	`select`
`case`	`defer`	`go`	`map`	`struct`
`chan`	`else`	`goto`	`package`	`switch`
`const`	`fallthrough`	`if`	`range`	`type`
`continue`	`for`	`import`	`return`	`var`

Constants:

`true`	`false`	`nil`	`iota`

Functions:

```
new         len         complex     panic
make        cap         real        recover
close       append      imag
copy
delete
```

Basic types:

```
int     int8     int16     int32     int64
uint    uint8    uint16    uint32    uint64    uintptr

float32      float64
complex64    complex128

bool    byte     rune      string    error
```

Operators:

```
*    /    %    &    &^   <<   >>
+    -    ^    |
==   !=   <    <=   >    >=
&&
||
```

Variablen:

```
1  var x int = 0
2  var x int
3  var x = 0
4
5  // for local variables only
6  x := 0
```

Slice:

```
1  s := make([]string, 0)
2  fmt.Println(cap(s)) // 0
3  s = append(s, "hello")
4  fmt.Println(cap(s)) // 1
5  s = append(s, "hello")
6  fmt.Println(cap(s)) // 2
7  s = append(s, "hello")
8  fmt.Println(cap(s)) // 4
9  s = append(s, "hello")
10 s = append(s, "hello")
11 fmt.Println(cap(s)) // 8
12 //...
```

Schleifen:

```go
for i := 0; i < 10; i++ {
    fmt.Println(i)
}

// or:

i := 0
condition := true
for condition {
    fmt.Println(i)
    i++
    if i == 10 {
        condition = false
    }
}
```

Synchronisation von Threads:

```
1  done := make(chan bool)
2
3  go func(from string) {
4      for i := 0; i < 3; i++ {
5          fmt.Println(from, ":", i)
6          time.Sleep(time.Second)
7      }
8
9      done <- true
10 }("value")
11
12 <-done
13
14 fmt.Println("done")
```