Die Sprache Go

In fünf Minuten

by Hauke Stieler

4. Dezember 2022

- 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]`

```
1  package main
2
3  import "fmt"
4
5  func main() {
6      fmt.Println("hello world")
7  }
```

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:

```
1  for i := 0; i < 10; i++ {
2      fmt.Println(i)
3  }
4
5  // or:
6
7  i := 0
8  condition := true
9  for condition {
10     fmt.Println(i)
11     i++
12     if i == 10 {
13         condition = false
14     }
15 }
```

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")
```