

# Go – Kurzreferenz

K.Topolewski – <http://www.weltchecker.de/>

## Kommentare

---

// ... einzeilig  
/\* ... \*/ mehrzeilig

## Pakete

---

```
package main
```

```
import "fmt"  
import ("fmt"; "math")
```

```
import (  
    "fmt"  
    "math"  
)
```

Verwendung

```
import "fmt" -> fmt.Printf  
import Format "fmt" -> Format.Printf  
import . "fmt" -> Printf
```

## Konstanten

---

vordefiniert: *true*, *false*, *iota*

```
const pi = 3.14  
const (  
    pi float32 = 3.14 // typisiert  
    r = 2 // nicht typisiert  
    text = "Hallo"  
)
```

## Variablen

---

normale Deklaration

```
var Bezeichner string  
var Bezeichner string = "Text"  
var Bezeichner = "Text"
```

```
var x, y, z int  
var x, y, z int = 1, 2, 3
```

```
var a, b, c = 1, true, "text"
```

```
var (  
    pi float32 = 3.14  
    r = 2  
)
```

kurze Deklaration

```
Bezeichner := "text"  
x := 1
```

## Typen

---

```
type Eingabe string  
type Koordinaten struct {  
    x int8  
    y int8  
}  
type (  
    eingabe string  
    Ergebnis []int8  
)
```

## Datentypen

---

bool: true, false

Integer – nicht vorzeichenbehaftet

```
uint8 -> 0 ... 255 8 Bit  
uint16 -> 0 ... 65535 16 Bit  
uint32 -> 0 ... 4294967295 32 Bit  
uint64 -> 0 ... 18446744073709551615 64 Bit
```

Integer – vorzeichenbehaftet

```
int8 -> -128 ... 127 8 Bit  
int16 -> -32768 ... 32767 16 Bit  
int32 -> -2147483648 ... 2147483647 32 Bit  
int64 -> -9223372036854775808 ... 9223372036854775807 64 Bit
```

int, uint – 32 bit oder 64 Bit, ja nach Betriebssystem

byte

Alias für uint8

reelle Zahlen – gemäß IEEE-754

float32, float64

```
0.  
12.34  
1.e+0  
12.34e-5  
.12345
```

komplexe Zahlen – reelle Zahlen mit imaginärer Einheit

complex64, complex128

```
0i  
12.34i  
1.e+0i  
12.34e-5i  
.12345i
```

String – UTF-8

```
"abc", "unicodebeispiel"
```

Array; Feld fester Länge

[Länge]WertTyp

```
[10]string  
[32]int8
```

Slice – Feld variabler Länge

[]WertTyp

Struct

struct {Variable; Variable; ...Variable}

```
var Person struct {x, y int8; A string}
```

Map

[SchlüsselTyp]WertTyp

```
map[string]string  
map[string]int32
```