

Programmieren mit R für Einsteiger

1. Grundlagen / 1.1 Syntax



Berry Boessenkool



frei verwenden, zitieren

2022-03-16 23:15

```
4*9
```

```
## [1] 36
```

Code in diesen Folien ist mit grauen Kästen hinterlegt und farbig markiert (syntax-highlighting). R Ausgaben stehen hinter `##`.

`[1]` wird im Abschnitt 2.3 (Vektoren) erklärt.

Kommentare (nach einem Hashtag) werden von R ignoriert.
Sie machen Code für Menschen verständlich = leichter lesbar:

```
2 + 4.8 # Punkt als Dezimaltrennzeichen
```

```
## [1] 6.8
```

```
2+4.8 # Leerzeichen egal für R, hilfreich für Lesbarkeit
```

```
3^2
```

```
## [1] 9
```

```
sqrt(81) # Wurzel (square root)  
## [1] 9
```

```
abs(-12) # Betrag (absolute value)  
## [1] 12
```

```
log(100) # natürlicher Logarithmus (ln) mit Basis e (2.72)  
## [1] 4.60517
```

```
log10(100) # Logarithmus mit Basis 10  
## [1] 2
```

```
exp(3) # Exponentialfunktion  $e^3$   
## [1] 20.08554
```

Zuweisung (assignment): ein Objekt mit Daten im Workspace anlegen

```
alter <- 15.4
```

Rstudio: Tastaturtasten **ALT** + **-** drücken für **<-**

```
alter # ist jetzt im Workspace (quasi der R Speicher)  
## [1] 15.4
```

```
alter + 5  
## [1] 20.4
```

alter ist unverändert 15.4. Zum Ändern überschreiben:

```
alter <- 37.1  
alter # immer die aktuelle Version, keine Geschichte dabei  
## [1] 37.1
```

Groß-/Kleinschreibung (case) beachten:

```
Alter # ist kein existierendes Object  
## Fehler: Objekt 'Alter' nicht gefunden
```

```
ls() # Eigene Objekte im Workspace auflisten
```

```
## [1] "alter"
```

```
rm(alter) # ein Objekt löschen
```

```
pi # Eingebaute Konstante
```

```
## [1] 3.141593
```

```
pi <- 3 # das kann überschrieben werden
```

```
sin(pi/2) # aber dann kommt nicht mehr ganz 1 raus ...
```

```
## [1] 0.997495
```

Empfehlung: bestehende Namen wie `pi`, `sin` nicht verwenden. Wenn ein eigenes Objekt `pi` existiert, wird nicht das eingebaute `pi` genutzt.

Gute Objektnamen sind **kurz aber aussagekräftig**, zB `tempMaxBerlin`

Übliche Konventionen sind `lowerCamelStandard` und `unter_strich`.

Den alten `punkt.standard` bitte nicht mehr verwenden. Der hat in anderen Programmiersprachen eine besondere Bedeutung.

Funktionen werden mit runden Klammern aufgerufen (ausgeführt):

```
log(7.4) # Funktionsaufruf  
## [1] 2.00148
```

```
log(x=7.4) # explizite Benennung des Arguments  
## [1] 2.00148
```

Argumente haben Namen. Diese können weggelassen werden, sofern sie in der richtigen Reihenfolge stehen.

`log` hat ein weiteres Argument `base`. Wenn das (wie bisher) nicht angegeben wird, wird 2.718 verwendet. Das ist der Standardwert für `base` (default). Für benutzerdefinierte Basis:

```
log(x=200, base=12)  
## [1] 2.1322
```

Argumentnamen können abgekürzt werden, sofern sie einmalig sind:

```
log(200, b=12)  
## [1] 2.1322
```

Syntax, Objekte, Operatoren, Funktionen:

- ▶ `+`, `-`, `*`, `/`, `^`
- ▶ Leerzeichen und Kommentare (`#`) machen Code leichter lesbar
- ▶ `pi`, `sqrt`, `abs`, `log`, `log10`, `exp`
- ▶ Objekte: `ALT` + `-` für Zuweisungspfeil (`<-`)
- ▶ Objektnamen Case sensitive
- ▶ `ob_jekt` oder `objekt`, nicht: `pi`, `daten`,
- ▶ `ls`, `rm`
- ▶ `funktion(1, argument=2, arg=3)`

Operatoren 2.0 ; Exponentialdarstellung

```
sin(15 * pi/180) # Grad zu dezimal  
## [1] 0.247404
```

```
factorial(5) # Fakultät:  $n! = 1*2*3*4*...*n$   
## [1] 120
```

```
exp(1) # eulersche Zahl e  
## [1] 2.718282
```

```
e^3 # geht so nicht  
## Fehler: Objekt 'e' nicht gefunden
```

```
3.91 * 10^-3 # wissenschaftliche Schreibweise: 3,91 E-3  
## [1] 0.00391
```

```
3.91e-3 # scientific notation: keine Leerzeichen erlaubt  
## [1] 0.00391
```

```
1e6 # schnell eine Million schreiben (ohne 6 Nullen)  
## [1] 1e+06
```

```
options(scipen=9) # bis 1e9 wird ab jetzt ausgeschrieben  
1e6 ; 1e14  
## [1] 1000000  
## [1] 1e+14
```


Technisch kann `=` statt `<-` für Zuweisungen verwendet werden. Gemäß **style guide** sollte `=` nur für `Funktion(arg="wert")` verwendet werden.

`median(x <- 1:10)` erstellt auch `x` im `globalenv()` workspace, `median(x=1:10)` nicht.

blog.revolutionanalytics.com, csgillespie.wordpress.com