

Perl-Workshop, Teil II

Ingo Blechschmidt,
Michael Hartmann

7. Februar 2007



Inhalt

1 Oft verwendete Funktionen

- Numerische Funktionen
- Stringfunktionen

2 Tastatureingabe

- Beispiel
- Besprechung
- Falle: chomp
- Übung

Numerische Funktionen

abs:	Betrag	abs -5	# 5
int:	„Runden zur 0“	int 7.9	# 7
atan2:	Arcustangens	atan2 (1, 0)	# $\pi/2$
sin:	Sinus	sin 0	# 0
cos:	Kosinus	cos 0	# 1
exp:	e-Funktionswert	exp 0	# 1
log:	Natürl. Logarithmus	log 1	# 0
sqrt:	Wurzel	sqrt 9	# 3

Stringfunktionen

String in Kleinbuchstaben:

```
lc:          lc "BaR"          # bar
```

Erster Buchstabe des Strings mit Kleinbuchstaben:

```
lcfirst:    lcfirst "BaR"     # baR
```

String in Großbuchstaben:

```
uc:          uc "BaR"          # BAR
```

Erster Buchstabe des Strings mit Großbuchstaben:

```
ucfirst:    ucfirst "bAr"     # BAr
```

Stringfunktionen (Fortsetzung)

Stringlänge:

```
length "Länge"           # 5
```

String umkehren:

```
reverse "Lager"         # regaL
```

Zeilenumbruch am Stringende entfernen:

```
my $var = "Kohle\n";  
chomp($var);           # $var: "Kohle"
```

Unicode/UTF-8 vs. ISO-8859-15/Fixed-8-Bit

- Quellcodedatei UTF-8-kodiert (gut)? →
`print length "Länge"; # 6 (!)`
- `use utf8`, um perl mitzuteilen, dass Quellcode UTF-8-kodiert ist:
`use utf8;`
`print length "Länge"; # 5`

Konvertierung nach UTF-8 mit Vim

```
:set fileencoding=utf-8  
:w
```

Tastatureingabe

```
1 #!/usr/bin/perl
2
3 use warnings;
4 use strict;
5
6 print "Gib deinen Namen ein: ";
7 my $name = <STDIN>; # Name einlesen
8 chomp $name;      # "\n" entfernen
9
10 print "Gib dein Alter ein: ";
11 # Name einlesen und zugleich "\n" entfernen
12 chomp(my $alter = <STDIN>);
13 # my $alter = chomp(<STDIN>) ist FALSCH!!!
14
15 print "Hallo $name, in 10 Jahren bist du "
16       . ($alter+10) . ".\n";
```

Tastatureingabe

1 Zeile einlesen:

```
my $zeile = <STDIN>;
```

2 Zeilenumbruch entfernen:

```
chomp $zeile;
```

...oder kürzer (TIMTOWTDI)...

1 Zeile einlesen und Zeilenumbruch entfernen:

```
chomp(my $zeile = <STDIN>);
```

Falle: chomp

- `chomp "hallo\n": Fehler`
- `my $var = chomp "hallo\n": Fehler`
- `my $orig = "hallo\n";`
`my $var = chomp($orig); # →`
`$orig: "hallo"; $var: 1`
- `chomp` verändert die Variable und gibt zurück, ob ein Zeilenumbruch entfernt wurde (0 oder 1)



Übung

Ein Programm soll zwei Zahlen von der Kommandozeile einlesen, und den Betrag der Multiplikation beider Zahlen ausgeben.

Mögliche Lösungen

```
1 #!/usr/bin/perl
2
3 use warnings;
4 use strict;
5
6 chomp(my $faktor1 = <STDIN>);
7 chomp(my $faktor2 = <STDIN>);
8
9 print abs($faktor1*$faktor2) . "\n";
```

```
1 #!/usr/bin/perl
2
3 use warnings;
4 use strict;
5
6 print abs(<STDIN>*<STDIN>) . "\n"; # Kontext
```

Mögliche Lösungen

```
1 #!/usr/bin/perl
2
3 use warnings;
4 use strict;
5
6 chomp(my $faktor1 = <STDIN>);
7 chomp(my $faktor2 = <STDIN>);
8
9 print abs($faktor1*$faktor2) . "\n";
```

```
1 #!/usr/bin/perl
2
3 use warnings;
4 use strict;
5
6 print abs(<STDIN>*<STDIN>) . "\n"; # Kontext
```

Hausaufgaben

- 1** Euro-Dollar Umrechner schreiben:
Das Programm soll einen Eurobetrag als Zahl von der Kommandozeile einlesen und ihn in Dollar umrechnen. (1 EUR = 1,2956 USD)
- 2** gelerntes Wiederholen, v.a.:
 - Zahlen (+, -, *, /, **, %, ++, --, abs, int, sqrt)
 - Strings (., x, lc, lcfirst, ucfirst, length, reverse, chomp)
 - skalare Variablen (Deklaration, Zuweisung)
 - Ein-/Ausgabe (print, <STDIN>)
- 3** folgendes Anschauen:
 - Kontrollstrukturen (if, elsif, else)
 - Schleifen (while)

Bildnachweis

- <http://gnosislivre.org/twiki/pub/PerlMongersSSA/WebHome/camel.gif>
- <http://www.ophase.bwl.uni-muenchen.de/images/achtung.gif>