Textverarbeitung mit Perl
Dozentin: Miriam Butt
Universität Konstanz, Sommer 2009
Dokument erstellt von Veronika Walther

Perl – Übersicht Befehle 1

Numerische Operatoren

2 + 3	2 plus 3, oder 5
5.1 - 2.4	5,1 minus 2,4, oder 2,7
3 * 12	3 mal 12 = 36
14 / 2	14 geteilt durch 2, oder 7
10.2 / 0.3	10,2 geteilt durch 0,3, oder 34
10 / 3	grundsätzlich Fließkomma-Division, also etwa 3,3333333...

Strings in einfachen Anführungszeichen

'Fred'	die vier Zeichen: F, r, e und d
'Barney'	diese sechs Zeichen
' '	der Nullstring (kein Zeichen)
'Ein \' kann Ihren String früher beenden, als Sie denken.'	
'Das letzte Zeichen dieses Strings ist ein Backslash: \\'	
'Hallo\n'	Hallo, gefolgt von einem Backslash, gefolgt von einem n
'Hallo Welt!'	Hallo, Zeilenumbruch, Welt! (insgesamt 11 Zeichen)
'\'\\'	ein einfaches Anführungszeichen, gefolgt von einem Backslash

Strings in doppelten Anführungszeichen

"Barney"	das Gleiche wie 'Barney'
"Hallo Welt\n"	Hallo Welt und ein Newline-Zeichen
"Das letzte Zeichen ist ein doppeltes Anführungszeichen: \" "	
"Kaffee\tTee"	ein Kaffee, ein Tabulator und ein Tee

Escapes für Strings in doppelten Anführungszeichen

Ausdruck	Bedeutung
\n	newline, Zeilenumbruch
\r	return, Wagenrücklauf
\t	tab, Tabulator
\f	form-feed, Seitenvorschub
\b	backspace, Rückschritt
\a	bell, Tonsignal
\e	Escape (ASCII Escape-Zeichen)
\007	jeder oktale ASCII-Wert (hier: 007=Tonsignal)
\x7f	jeder hexadezimale ASCII-Wert (hier: 7f=DEL, Löschzeichen)
\cC	jedes Control-Zeichen (hier: Conntrol-C)

\\	backslash
\“	double quote, doppeltes Anführungszeichen
\l	den folgenden Buchstaben klein schreiben
\L	alle folgenden Buchstaben klein schreiben bis zum nächsten \E
\u	den folgenden Buchstaben groß schreiben
\U	alle folgenden Buchstaben groß schreiben bis zum nächsten \E
\Q	alle nicht-alphanumerischen Zeichen mit Backslash schützen bis zum nächsten \E
\E	hebt \L, \U, oder \Q auf

String Operatoren

“Hallo” . “Welt”	das Gleiche wie “HalloWelt”
“Hallo” . ‘ ’ . “Welt”	das Gleiche wie ‘Hallo Welt’
‘Hallo Welt’ . “\n”	das Gleiche wie “Hallo Welt\n”
“Fred” x 3	ergibt “FredFredFred”
“Barney” x (4+1)	ergibt “Barney“ x 5, oder “BarneyBarneyBarneyBarneyBarney”
5 x 4	ist eigentlich “5” x 4, also “5555”

Skalare Zuweisung

$fred = 17;	$fred den Wert 17 zuweisen
$barney = ‘Hallo’;	$barney den String ‘Hallo’ zuweisen
$barney = $fred + 3;	$barney den gegenwärtigen Wert von $fred plus 3 zuweisen (20)
$barney = $barney * 2;	$barney ist nun $barney mal 2 (40)

Binäre Zuweisung

```
$fred    = $fred + 5;      ohne den Operator für binäre Zuweisung
$fred    += 5;             mit dem Operator für binäre Zuweisung
$barney  = $barney * 5;
$barney  *= 5;
```

Ausgabe mit print

```
print “Hallo Welt\n”;      Ausgabe von Hallo Welt, gefolgt von einem Newline-Zeichen
print “Die Antwort ist ”;
print 6 * 7;
print “.\n”;
```

Auch möglich ist folgender Befehl (hierbei handelt es sich um eine Liste):

```
print “Die Antwort ist “, 6 * 7, ” .\n”;
```

(aus Learning Perl/Einführung in Perl, O‘Reilly)