

HOWTO zur Shell-Quotierung

(C) 2006-2017 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>  
 OSTC Open Source Training and Consulting GmbH  
<http://www.ostc.de>

\$Id: shell-quoting-HOWTO.txt,v 1.17 2018/07/31 06:13:41 tsbirn Exp \$

Dieses Dokument beschreibt die verschiedenen Verfahren zur Quotierung von Shell-Sonderzeichen.

## INHALTSVERZEICHNIS

- 1) Shell-Sonderzeichen
- 2) Quotierung
- 3) Beispiel

### 1) Shell-Sonderzeichen

Die Shell kennt eine Reihe von Sonderzeichen, die nicht für sich selbst stehen, sondern von ihr speziell interpretiert werden und dann bestimmte Funktionen auslösen (26 Stück!):

```
+-----+
| * ? [ ] = $ < > | & \ |
| " ' ` ; ( ) { } ~ ^ ! # |
|           <Space> <Tabulator> <Return> |
+-----+
```

### 2) Quotierung

Sollen diese Zeichen nicht ihre Sonderbedeutung haben, sondern als ganz normaler Text interpretiert werden, so sind sie vor der Shell zu "schützen", der Fachausdruck dafür ist "quotieren" (zitieren). Die Shell kennt 3 verschiedene Möglichkeiten des Schützens von Zeichen, die für unterschiedliche Anwendungsfälle benötigt werden:

Typ	Bedeutung
'...'	ALLE Sonderzeichen in ... abschalten [tick, single quote]
"..."	ALLE Sz. BIS AUF \$, \$(...), `...`, \, ! abschalten [double quote]
\Z	GENAU EIN (folgendes) Sonderzeichen Z abschalten [backslash]

Die Quotierung per "..." wird auch als "schwach" (weak) bezeichnet, da sie nicht alle Sonderzeichen abschaltet. Die Quotierung per '...' und \ wird als "stark" (strong) bezeichnet, weil sie ALLE Sonderzeichen abschaltet.

Hier eine Übersicht über die von den Quotierungszeichen geschützten (\*) bzw. NICHT geschützten (-) Sonderzeichen (CR=Zeilenende, Hst=Kmdo-History, e=Ende, w=wörtlich übernommen, i=Ignoriert):

Typ	\$VAR	Kmdo-Substit '...'\$(...)	Hst !	\	Glob *?[]	"	'	Whitespace (TAB/NL/SP)	C R	Rest
"..."	-	-	-	-	*	e	*	*	w	*
'...'	*	*	*	*	*	*	e	*	w	*
\Z	*	*	*	*	*	*	*	*	i	*

Zweck der beiden Quotierungs-Varianten "..." und '...' ist es, zu unterscheiden, ob man den WERT einer Variablen (oder die AUSGABE eines Kommandos) nach dem Einsetzen in eine Kommandozeile NOCHMALS von der Shell interpretieren lassen möchte oder nicht. Bei "..." wird nochmal interpretiert, bei '...' nicht.

Beispiel:

```
VAR="*"      # Variable VAR den Text "*" (STERN) zuweisen
echo $VAR    # => Alle Dateinamen des akt. Verz. (2x ausgewertet)
echo "$VAR"  # => * (1x ausgewertet)
echo '$VAR'  # => $VAR (0x ausgewertet)
echo \$VAR   # => $VAR (0x ausgewertet)
```

HINWEIS: Die Zeichen `...' (backtick, left quote) führen das dazwischen stehende Kommando aus und setzen das Ergebnis (die Standard-Ausgabe des Kommandos) an dieser Stelle ein:

