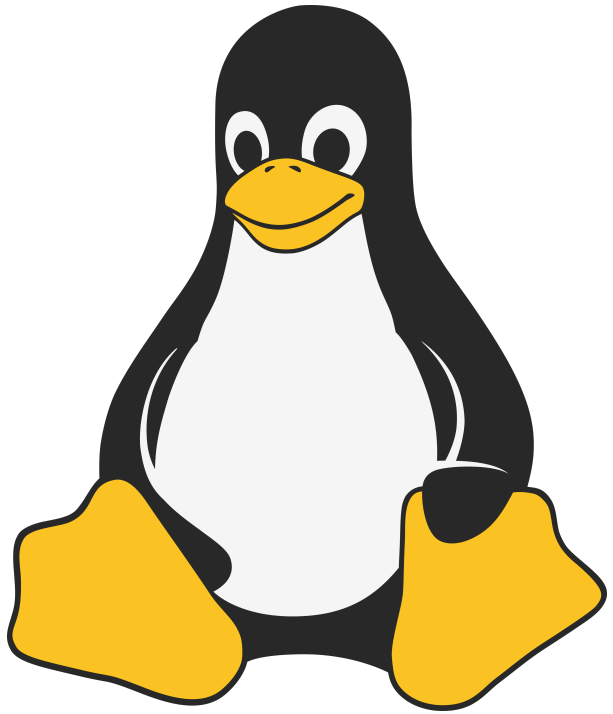
Shell-Sonderzeichen (2) und deren Maskierung



Inhaltsverzeichnis

Sonderzeichen (Meta-Zeichen)

- Die Shell verwendet eine Reihe von Sonderzeichen (auch als Meta-Zeichen bezeichnet), die eine besondere Bedeutung haben.
- Diese Sonderzeichen verlieren (falls sie nicht maskiert sind) ihre normale Zeichenbedeutung und werden von der Shell bei der Verarbeitung der Kommandos (auch in Shell-Skripten) interpretiert.
- Sollen sie ihre Shell-Sonderbedeutung verlieren und ihre Zeichenbedeutung wiedererlangen, müssen sie maskiert werden.

Zum Inhaltsverzeichnis ...

Sonderzeichen	Bedeutung
`#`	Kommentarzeichen
Leerzeichen	Wort-Trenner, Token-Trenner
`>`	Ausgabe-Umlenkung
`>>`	Ausgabe-Umlenkung (Anhängen)
`<`	Eingabe-Umlenkung
`<<`	Eingabe-Umlenkung (Here-Document) (nicht behandelt in Chap04)
`|`	Pipe und Kommando-Trenner

Zum Inhaltsverzeichnis ...

Sonderzeichen	Bedeutung
`\n` (Newline)	Kommando-Trenner
`;`	Kommando-Trenner
`&&`	Logisches UND und Kommado-Trenner
`||`	Logisches ODER und Kommando-Trenner
`$`	Variablen-Substitution
`$(command)`	Kommando-Substitution
`` `command` ``	Kommando-Substitution (alte Syntax)

Zum Inhaltsverzeichnis ...

Sonderzeichen	Bedeutung
`*`	Dateinamens-Wildcard: Platzhalter für eine beliebige Zeichenkette
`?`	Dateinamens-Wildcard: Platzhalter für ein einzelnes Zeichen
`[...]`	Dateinamens-Wildcard: Platzhalter für ein einzelnes Zeichen aus einer Zeichenklasse
`{cmd1; cmd2; ...}`	Kommandogruppe, Ausführung in der aktuellen Shell
`(cmd1; cmd2; ...)`	Kommandogruppe, Ausführung in einer Subshell

Zum Inhaltsverzeichnis ...

Sonderzeichen	Bedeutung
`\`	Backslash, maskiert das direkt nachfolgende Zeichen
`'...'`	Einfache Anführungszeichen, maskieren die eingeschlossenen Zeichen ausser `'`
`"..."`	Doppelte Anführungszeichen, maskieren die eingeschlossenen Zeichen ausser `"`, `\`, `$` und `` ` ``

- Variablen-Substitution und Kommando-Substitution (auch die alte Syntax mit `` `...` ``) sind in doppelten Anführungszeichen für die Shell sichtbar und werden interpretiert.

Zum Inhaltsverzeichnis ...

Maskierung von Sonderzeichen

Manchmal möchte man ein Sonderzeichen seiner besonderen Bedeutung berauben, d.h. das Sonderzeichen soll als normales Zeichen behandelt werden. Z.B. soll das Semikolon (`;`) nicht als Kommando-Trenner, sondern als normales Semikolon behandelt werden. Oder das Dollarzeichen (`$`) soll nicht als Variablen-Substitution, sondern als Dollarzeichen (z.B. für eine Preisangabe) behandelt werden. In diesen Fällen muss das Sonderzeichen maskiert werden.

Zum Inhaltsverzeichnis ...

- Die Shell hat eigene Sonderzeichen für die Maskierung von Sonderzeichen (s.o.).
- Nach der Bearbeitung der Sonderzeichen durch die Shell werden die Maskierungszeichen (`\` , `'...'` , `"..."`) entfernt.

```
hermann@debian:~/$ echo Mein Heimatdorf ist     $HOME.
Mein Heimatdorf ist /home/hermann.
hermann@debian:~/$ echo "Mein Heimatdorf ist     $HOME."
Mein Heimatdorf ist     /home/hermann.
hermann@debian:~/$ echo "Mein Heimatdorf ist     \$HOME."
Mein Heimatdorf ist     $HOME.
hermann@debian:~/$ echo 'Mein Heimatdorf ist     $HOME.'
Mein Heimatdorf ist     $HOME.
```

```
hermann@debian:~$ echo 'Der Preis beträgt: $ 100'
Der Preis beträgt: $ 100
hermann@debian:~$ echo "Der Preis beträgt: \$ 100"
Der Preis beträgt: $ 100
hermann@debian:~$ echo Der Preis beträgt: \$ 100
Der Preis beträgt: $ 100
```


Zum Inhaltsverzeichnis ...

```
hermann@debian:~$ echo 'Zitat: "Irren ist menschlich."'
Zitat: "Irren ist menschlich."
hermann@debian:~$ echo "Zitat: \"Irren ist menschlich.\""
Zitat: "Irren ist menschlich."
hermann@debian:~$ echo Zitat: \"Irren ist menschlich.\"
Zitat: "Irren ist menschlich."
```

```
hermann@debian:~$ echo '--'\''--'
--'--
hermann@debian:~$ echo "--'--"
--'--
hermann@debian:~$ echo '--'"'"'--'
--'--
```