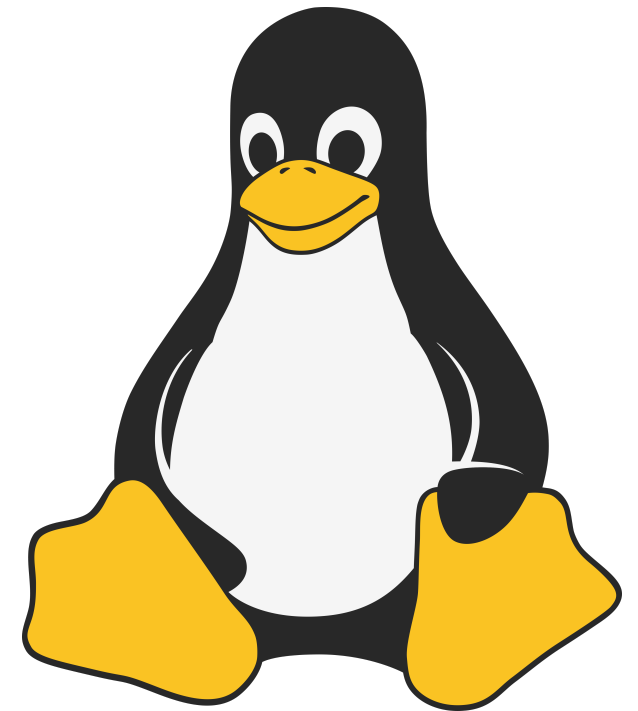


Wildcards für Dateinamen



BASH
THE BOURNE-AGAIN SHELL



Inhaltsverzeichnis

- [Wildcards - was sind sie?](#)
- [Regeln](#)
- [Beispiele mit Erläuterungen](#)
- [Shell-Beispiele](#)

Wildcards - was sind sie?

Wildcards sind Platzhalterzeichen, die in der Shell verwendet werden, um Dateinamen zu spezifizieren. Als Wildcards werden folgende Zeichen verwendet :

- ***** (Sternchen) - steht für eine beliebige Anzahl beliebiger Zeichen (auch keine)
- **?** (Fragezeichen) - steht für ein einzelnes beliebiges Zeichen
- **[...]** (eckige Klammern) - steht für ein einzelnes Zeichen aus einer Menge von Zeichen, die in den Klammern aufgeführt sind. Innerhalb der eckigen Klammern gelten folgende Regeln:

- ○ Die erlaubten Zeichen können als einzelne Zeichen angegeben werden, z.B. `[aAxX]` steht für ein Zeichen, das `a`, `A`, `x` oder `X` sein kann.
- Ein Bereich von Zeichen (die im ASCII-Zeichensatz aufeinander folgen) kann durch zwei Zeichen getrennt werden, die durch einen Bindestrich (`-`) verbunden sind, z.B. `[a-z]` steht für ein beliebiges Kleinbuchstaben-Zeichen.
- Durch ein Ausrufezeichen (`!`) oder ein Caret (`^`) nach der öffnenden eckigen Klammer kann eine Negation angegeben werden, z.B. `[!a-z]` oder `[^a-z]` steht für ein Zeichen, das alles sein kann, außer einem Kleinbuchstaben.

Regeln

- Wildcards werden von der Shell interpretiert, bevor ein Befehl ausgeführt wird. Sie können in jedem Befehl verwendet werden, in dem Dateinamen vorkommen sollen.
- Passt ein Wildcard-Muster auf mindestens einen oder mehrere Dateinamen, wird das Muster durch die Dateinamen ersetzt (expandiert), auf die es passt.
- Passt ein Wildcard-Muster auf keinen Dateinamen, bleibt das Muster unverändert.

- Der führende `.` eines Dateinamens wird nicht von einem Wildcard-Muster erfasst.
- Die Einträge `.` und `..` werden von Wildcards nicht erfasst.
- Wildcards können in jedem Pfadteil eines Dateinamens verwendet werden.
- Wildcard-Zeichen sind auch Shell-Sonderzeichen. Sie müssen daher maskiert werden (mit `\`, `'...'` oder `"..."`), wenn sie nicht expandiert, sondern als normale Zeichen behandelt werden sollen.

Beispiele mit Erläuterungen

- `*.txt` - steht für alle Dateien, die auf `.txt` enden
- `*.jpg` - steht für alle Dateien, die auf `.jpg` enden
- `*.?` - steht für alle Dateien, die einen Punkt an vorletzter Stelle und ein einziges aber beliebiges Zeichen an der letzten Stelle (nach dem Punkt) enthalten
- `*.*` - steht für alle Dateien, die mindestens einen `.` enthalten
- `*` - steht für alle Dateien (außer den Dateien, die mit `.` beginnen)
- `???.txt` - steht für alle Dateien, die genau drei Zeichen vor `.txt` enthalten. Das erste Zeichen darf kein `.` sein.

- `[abc]*` - steht für alle Dateien, die mit `a`, `b` oder `c` beginnen
- `[a-z]*` - steht für alle Dateien, die mit einem Kleinbuchstaben beginnen
- `[A-Z]*` - steht für alle Dateien, die mit einem Großbuchstaben beginnen
- `[0-9]*` - steht für alle Dateien, die mit einer Ziffer beginnen
- `[a-zA-Z]*` - steht für alle Dateien, die mit einem Buchstaben beginnen
- `[!a-z]*` oder `[^a-z]*` - steht für alle Dateien, die nicht mit einem Kleinbuchstaben beginnen

Shell-Beispiele

```
hermann@debian:~$ cd /etc/apt
hermann@debian:/etc/apt$ ls -l
insgesamt 36
drwxr-xr-x 2 hermann hermann 4096 10. Nov 13:07 apt.conf.d
drwxr-xr-x 2 hermann hermann 4096 10. Nov 13:07 auth.conf.d
drwxr-xr-x 2 hermann hermann 4096 10. Nov 13:07 keyrings
-rw-r--r-- 1 hermann hermann  150 10. Nov 13:07 listchanges.conf
drwxr-xr-x 2 hermann hermann 4096 10. Nov 13:07 listchanges.conf.d
drwxr-xr-x 2 hermann hermann 4096 10. Nov 13:07 preferences.d
-rw-r--r-- 1 hermann hermann 1054 10. Nov 13:07 sources.list
-rw-r--r-- 1 hermann hermann    0 10. Nov 13:07 sources.list~
drwxr-xr-x 2 hermann hermann 4096 10. Nov 13:07 sources.list.d
drwxr-xr-x 2 hermann hermann 4096 10. Nov 13:07 trusted.gpg.d
```

```
drwxrhermann@debian:/etc/apt$ echo sources*  
sources.list sources.list~ sources.list.d
```

```
hermann@debian:/etc/apt$ echo *.d  
apt.conf.d auth.conf.d listchanges.conf.d preferences.d sources.list.d trusted.gpg.d
```

```
hermann@debian:/etc/apt$ echo *list*  
listchanges.conf listchanges.conf.d sources.list sources.list~ sources.list.d
```

```
hermann@debian:/etc/apt$ echo *list.*  
sources.list.d
```

```
hermann@debian:/etc/apt$ echo *list*.conf*  
listchanges.conf listchanges.conf.d
```

```
hermann@debian:/etc/apt$ echo *list*.conf  
listchanges.conf
```

```
hermann@debian:/etc/apt$ echo [a-p]*  
apt.conf.d auth.conf.d keyrings listchanges.conf listchanges.conf.d preferences.d
```

```
hermann@debian:/etc/apt$ echo *[ft]  
listchanges.conf sources.list
```

```
hermann@debian:/etc/apt$ echo *[fts]  
keyrings listchanges.conf sources.list
```

```
hermann@debian:/etc/apt$ echo *[d~]  
apt.conf.d auth.conf.d listchanges.conf.d preferences.d sources.list~ sources.list.d trusted.gpg.d
```

```
hermann@debian:/etc/apt$ echo /etc/host*  
/etc/host.conf /etc/hostname /etc/hosts /etc/hosts.allow /etc/hosts.deny
```

```
hermann@debian:/etc/apt$ echo *  
apt.conf.d auth.conf.d keyrings listchanges.conf listchanges.conf.d preferences.d ...
```

```
hermann@debian:/etc/apt$ echo .* # no match here  
.*
```

```
hermann@debian:/etc/apt$ cd  
hermann@debian:~$ echo .* # all hidden files in home directory  
.bash_aliases .bash_history .bash_logout .bash_profile .bash_profile.log .bashrc ...
```