

# **grep** - Mustersuche



# Inhaltsverzeichnis

- [Beschreibung](#)
- [Syntax](#)
- [Endestatus \(Rückgabewert\)](#)
- [Erste Beispiele](#)
- [Farbliche Hervorhebung der Treffer](#)
- [Die wichtigsten Optionen](#)

- Rekursive Suche
- Kontextsteuerung
- Reguläre Ausdrücke

# Beschreibung

`grep` ist ein Kommando zur Mustersuche in Text-Dateien und Text-Datenströmen. Es ist ein mächtiges und häufig genutztes Kommando. Gerne wird in einer Pipeline genutzt, um die Ausgabe eines anderen Kommandos zu filtern.

Diese Lektion enthält eine Einführung in die Nutzung von `grep`. Weiter gehende Informationen finden Sie mit `grep --help` oder in der Manpage `man grep`.

# Syntax

```
grep [options] pattern [file ...]
```

`grep` schreibt die Trefferzeilen in die Standardausgabe (STDOUT).

- `[options]`: `grep` bietet viele Optionen. Die wichtigsten sind Gegenstand dieser Lektion.
- `pattern` ist das Muster, nach dem gesucht werden soll. Das Muster kann eine einfache Zeichenfolge oder ein regulärer Ausdruck (regex) sein. (Mit regulären Ausdrücken können sehr komplexe Suchmuster definiert werden. Sie sind Gegenstand einer anderen

- `file ...`: Eine oder mehrere Dateien, in denen das Muster gesucht werden soll. Wird keine Datei angegeben, wird von `grep` der Standard-Eingabestrom (STDIN) gelesen. Werden mehrere Dateiargumente angegeben, dann wird vor den Trefferzeilen der Dateiname, in dem die Trefferzeile gefunden wurde, ausgegeben.

Weitere Optionen und Infos zu `grep` finden Sie mit `grep --help` oder in der Manpage mit `man grep`.

# Erste Beispiele

**Beispiel 1:** das Muster `root` in der Datei `/etc/passwd` suchen.

Bei der Angabe nur eines Dateiarguments wird der Dateiname vor der Trefferzeile nicht ausgegeben.

```
hermann@debian:~$ grep root /etc/passwd  
root:x:0:0:root:/root:/bin/bash
```

**Beispiel 2:** das Muster `root` in den Dateien `/etc/passwd` und `/etc/group` suchen

Bei der Angabe mehrerer Dateiargumente wird der Dateiname (mit einem Doppelpunkt getrennt) vor der Trefferzeile ausgegeben.

```
hermann@debian:~$ grep root /etc/passwd /etc/group
/etc/passwd:root:x:0:0:root:/root:/bin/bash
/etc/group:root:x:0:
```



### Beispiel 3: das Muster `root` im Eingabestrom suchen

Wird kein Dateiargument angegeben, wird der Eingabestrom (STDIN) gelesen und nach dem Muster durchsucht.

```
hermann@debian:~$ # grep reads from STDIN (= from the pipe)
hermann@debian:~$ cat /etc/passwd | grep root
root:x:0:0:root:/root:/bin/bash
```

```
hermann@debian:~$ # grep reads from STDIN (= from the redirected open file)
hermann@debian:~$ grep root < /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

# Endestatus (Rückgabewert)

Nach der Ausführung des Kommandos kann der Endestatus mit `echo $?` abgefragt werden.

- **0** - wenn mindesens ein Treffer gefunden wurde
- **1** - wenn kein Treffer für das Muster gefunden wurde
- **2** - beim Auftreten eines Fehlers

# Ausgabe und Fehlerausgabe unterdrücken

Ist man nur am Endestatus, jedoch nicht an der Ausgabe und Fehlerausgabe von `grep` nicht interessiert, so kann man diese nach `/dev/null` umleiten.

Einfacher ist es, stattdessen die Optionen `-q` und `-s` zu nutzen:

- `-q` oder `--quiet` oder `--silent`: unterdrückt die Standardausgabe
- `-s` oder `--no-messages`: unterdrückt die Fehlerausgabe

# Beispiele ohne Ausgabeunterdrückung

```
hermann@debian:~$ grep root /etc/passwd # pattern 'root' matches  
root:x:0:0:root:/root:/bin/bash  
hermann@debian:~$ echo $?  
0
```

```
hermann@debian:~$ grep foobar /etc/passwd # pattern 'foobar' does not match  
hermann@debian:~$ echo $?  
1
```

```
hermann@debian:~$ grep root /etc/shadow # error: no permission  
grep: /etc/shadow: Keine Berechtigung  
hermann@debian:~$ echo $?  
2
```

# Beispiele mit Ausgabeunterdrückung

```
hermann@debian:~$ grep -q -s root /etc/passwd # pattern 'root' matches  
hermann@debian:~$ echo $?  
0
```

```
hermann@debian:~$ grep -q -s foobar /etc/passwd # pattern 'foobar' does not match  
hermann@debian:~$ echo $?  
1
```

```
hermann@debian:~$ grep -q -s root /etc/shadow # error: no permission  
hermann@debian:~$ echo $?  
2
```

# Farbliche Hervorhebung der Treffer

Die Option `--color=auto` hebt die Treffer farblich (rot) hervor.

Durch die Definition des passenden Alias kann die Option `--color=auto` dauerhaft aktiviert werden.

```
hermann@debian:~$ alias grep='grep --color=auto'
```

Nach der Definition des Alias wird jedes Vorkommen des Musters `root` farblich hervorgehoben.

```
hermann@debian:~$ grep root /etc/passwd # pattern match displayed in red  
root:x:0:0:root:/root:/bin/bash
```

In diesem Foliensatz kann die farbliche Hervorhebung der Treffer allerdings nicht dargestellt werden.

Der Alias ist jedoch für die weiteren Kommandos in einer interaktiven Shell-Sitzung sinnvoll.

# Die wichtigsten Optionen

- `-i` oder `--ignore-case`: Groß-/Kleinschreibung ignorieren

```
hermann@debian:~$ grep -i bat ~/repo/README.md  
- *win-bat*: Windows Batch-Dateien
```



- `-v` oder `--invert-match`: Zeilen ausgeben, die das Muster nicht enthalten

```
hermann@debian:~$ ls -ld D*
drwxr-xr-x 2 hermann hermann 4096 14. Nov 23:17 Desktop
drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Dokumente
drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Downloads
hermann@debian:~$ ls -ld D* | grep -v Desk
drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Dokumente
drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Downloads
```

- `-n` oder `--line-number`: Zeilennummern der Trefferzeilen auch ausgeben

```
hermann@debian:~$ ls -ld D*
drwxr-xr-x 2 hermann hermann 4096 14. Nov 23:17 Desktop
drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Dokumente
drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Downloads
hermann@debian:~$ ls -ld D* | grep -n Desk
1:drwxr-xr-x 2 hermann hermann 4096 14. Nov 23:17 Desktop
hermann@debian:~$ ls -ld D* | grep -v -n Desk
2:drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Dokumente
3:drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Downloads
```

- `-c` oder `--count`: Anzahl der Treffer ausgeben

```
hermann@debian:~$ ls -ld D*
drwxr-xr-x 2 hermann hermann 4096 14. Nov 23:17 Desktop
drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Dokumente
drwxr-xr-x 2 hermann hermann 4096 29. Okt 17:43 Downloads
hermann@debian:~$ ls -ld D* | grep -c Desk
1
hermann@debian:~$ ls -ld D* | grep -c -v Desk
2
```

- `-l` oder `--files-with-matches`: Dateinamen ausgeben, die das Muster enthalten
- `-L` oder `--files-without-match`: Dateinamen ausgeben, die das Muster nicht enthalten

```
hermann@debian:~$ grep root /etc/passwd /etc/group /etc/fstab
/etc/passwd:root:x:0:0:root:/root:/bin/bash
/etc/group:root:x:0:
hermann@debian:~$ grep -l root /etc/passwd /etc/group /etc/fstab
/etc/passwd
/etc/group
hermann@debian:~$ grep -L root /etc/passwd /etc/group /etc/fstab
/etc/fstab
```

# Rekursive Suche

Bei der rekursiven Mustersuche (`-r`) können auch Verzeichnisse als Argumente angegeben werden. Standardmäßig werden alle Dateien in den durch die Verzeichnisse definierten Dateihierarchien durchsucht. Dabei werden symbolische Links ignoriert. Mit der Option `-R` werden auch symbolische Links verfolgt. Dabei mit `--include` und `--exclude` Dateimuster angegeben werden, um nur bestimmte Dateien zu durchsuchen oder bestimmte Dateien auszuschließen.

- `-r` oder `--recursive`: Rekursiv in Unterverzeichnissen suchen
- `-R` oder `--dereference-recursive`: Rekursiv in Unterverzeichnissen suchen und symbolische Links folgen
- `--include=file-pattern`: Nur Dateien mit dem angegebenen Dateimuster durchsuchen
- `--exclude=file-pattern`: Dateien mit dem angegebenen Dateimuster ausschließen

```
hermann@debian:~$ grep -r -l -s --include '*host*' debian /etc
/etc/hosts
/etc/hostname
/etc/ssh/ssh_host_rsa_key.pub
/etc/ssh/ssh_host_ecdsa_key.pub
/etc/ssh/ssh_host_ed25519_key.pub
```

`grep -r debian /etc` durchsucht rekursiv das Verzeichnis `/etc` nach dem Muster `debian`. Die Option `-l` gibt nur die Dateinamen aus, die das Muster enthalten. `-s` unterdrückt die Fehlerausgaben (z.B. wegen fehlender Berechtigungen). Mit `--include '*host*'` werden nur Dateien durchsucht, die das Muster `host` im Dateinamen enthalten.

# Nach mehreren Mustern suchen

Durch die wiederholte Angabe der Option `-e pattern` können mehrere Suchmuster angegeben werden. Es werden alle Zeilen ausgegeben, die mindestens eines der Muster enthalten.

```
hermann@debian:~$ grep -e root -e hermann /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
33:hermann:x:1000:1000:Hermann Hueck,,,:/home/hermann:/bin/bash
```



# Kontextsteuerung

Normalerweise gibt `grep` nur die Trefferzeilen aus. Mit den Optionen `-B`, `-A`, und `-C` kann der Kontext der Trefferzeilen gesteuert werden. Der Kontext ist die Anzahl der Zeilen, die vor und/oder nach der Trefferzeile ausgegeben werden.

- `-B n` oder `--before-context=n`: n Zeilen vor dem Treffer ausgeben
- `-A n` oder `--after-context=n`: n Zeilen nach dem Treffer ausgeben
- `-C n` oder `-n` oder `--context=n`: n Zeilen vor und nach dem Treffer ausgeben

```
hermann@debian:~$ grep -n -C 2 -e root -e hermann /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
2-daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3-bin:x:2:2:bin:/bin:/usr/sbin/nologin
--
31-rtkit:x:109:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
32-colord:x:110:120:colord ... :/var/lib/colord:/usr/sbin/nologin
33:hermann:x:1000:1000:Hermann Hueck,,,:/home/hermann:/bin/bash
34-harry:x:1001:1001:Harry Potter:/home/harry:/bin/bash
35-hermine:x:1002:1002:Hermine Granger:/home/hermine:/bin/bash
```

Statt `-C 2` kann auch `-2` oder `--context=2` verwendet werden.

# Reguläre Ausdrücke

Reguläre Ausdrücke werden in einer eigenen Lektion behandelt. Hier nur ein Beispiel: In `/etc/fstab` sollen alle Zeilen für ext4-Dateisysteme ausgegeben werden. Dabei sollen Kommentarzeilen, die das Muster `ext4` enthalten, unterdrückt werden.

```
hermann@debian:~$ grep ext4 /etc/fstab # pattern 'ext4' matches
# root file system is of type ext4
UUID=1cbc22fc-4dad-4f2f-9e7d-d58329c86015 /    ext4      errors=remount-ro 0      1
hermann@debian:~$ grep ext4 /etc/fstab | grep -v '^ *#' suppresses comment lines
UUID=1cbc22fc-4dad-4f2f-9e7d-d58329c86015 /    ext4      errors=remount-ro 0      1
```

Das erste `grep`-Kommando gibt alle Zeilen aus, die das Muster `ext4` enthalten. Das zweite `grep`-Kommando filtert mit dem Muster `'^ *#'` (Beliebig viele Leerzeichen gefolgt von einem `#` am Zeilenanfang) die Kommentarzeilen heraus.

- `-E` oder `--extended-regexp`: Interpretiere das Muster als erweiterten regulären Ausdruck. `grep -E` ist äquivalent zum Kommando `egrep`.