

Grep

- **Definierte Bereiche:**

- `[:alnum:]` abc und numerische Ziffern
- `[:alpha:]` abc..
- `[:blank:]` Blank + Tab
- `[:cntrl:]` Kontrollzeichen, z. B. Strg+A
- `[:digit:]` Ziffern
- `[:graph:]` Druckbare Zeichen
- `[:lower:]` abc
- `[:print:]` graph und Blank
- `[:punct:]` . : ;
- `[:space:]` tab, space, CR,LF
- `[:upper:]` ABC
- `[:xdigit:]` hexadezimale Zahlen

.	beliebiges Zeichen (Punkt)
[ABCacZ]	Mengengruppe
[A-Z]	Mengengruppe
[^ABC]	negierte Gruppe
^	Zeilenanfang, an der Spitze
\$	Zeilenende, Dollar ist am Ende
\<	steht für Wortanfang, links oder rechts steht ein space, tab. CR, Anfang
\>	steht für Wortende
\b	steht Wortanfang oder Wortende
()	Gruppe mit Alternative Oder-Bedingung

Wiederholungs-Operatoren: (nur mit Parameter -E) oder egrep

*	0,1,n
?	0,1
+	1,n
{n}	das vorangegangene Zeichen tritt n-mal auf
{n,}	das vorangegangene Zeichen tritt n-mal oder öfter auf
{n,m}	das vorangegangene Zeichen tritt mindestens n-mal und maximal m-mal auf

grep 'ab' *	alle Dateien die ab in einer Zeile haben
'^ab'	"ab" am Anfang
'^abc'	"abc" am Anfang
'abcd\$'	"abcd" am Ende
'cd\$'	"cd" am Ende
'1[23]'	1 und 2 oder 3
'1[23]f'	1 und 2 oder 3 danach f
'1[23][bf]'	1 und 2 oder 3 danach b oder f
'^1[23]'	1 am Anfang danach 2 oder 3
'^[ae]'	a oder e am Anfang
-i '^[Ae]'	a oder e am Anfang oder Unterscheidung ignore case
'^[^ae]'	am Anfang nicht ae
'b.e'	irgendwo b.e
'\.'	'.' liefert alles

grep -E 'abc+'
 -E Extended Regular expression oder egrep

grep -E 'abc?'	
-E '^(ab ed)'	Am Anfang ab oder ed
egrep '[:alnum:]'	
egrep '[:alphanum:]'	
egrep '[:alpha:]'	
egrep '[:digit:]'	
grep '([^()*)a'	nicht () dann beliebig dann a
egrep '([()*)a'	nicht () dann beliebig dann a

egrep '([[:digit:]]{1,3}\.){3}[[:digit:]]{1,3}'

egrep '\$x'	\$ bedeutet Zeilenende
egrep '\\$x'	\\$ bzw. \^ korrekt, aber nun mit Space
egrep '\\$x '	als Wort, tab
egrep '<\$x>'	als Wort, tab, space, auch am Anfang und am Ende

Datum suchen:

```
egrep '\<[0-9]{1,}\.[0-9]{1,}\.[0-9]{1,4}\>' *
a8:12.11.10
a8:1.1.1
a8:01.01.2010
egrep '\<[0-9]{1,}\.[0-9]{1,}\.[0-9]{2,4}\>' *
a8:12.11.10
a8:01.01.2010
```

grep '^abc.abc' a*	abc3abcx abc3abc suchen
egrep '((abc.){2,})' a*	


```
$ grep -c "a" a?
```

```
a1:3
```

```
a2:0
```

```
a3:3
```

```
a4:1
```

```
a5:1
```

```
a6:3
```

```
a7:0
```

```
a8:1
```

```
a9:1
```

```
grep -c '12' a? | sort -r -t : -k 2,2
```

find

Das Kommando „find“ sucht alle Dateinamen, die bestimmten Bedingungen genügen.

Aufruf: find [directory] bedingung aktion

Parameter:

directory Beginn der Suche im Teilbaum

bedingung Es können folgende Bedingungen eingestellt werden:

- name muster Ist erfüllt, wenn der Name dem Muster entspricht
- atime name Ist erfüllt, wenn auf die Datei vor zahl Tagen zugegriffen wurde
- mtime zahl Ist erfüllt, wenn die Datei vor zahl Tagen verändert wurde.
- newer datei Ist erfüllt, wenn die untersuchte Datei nach der letzten Änderung von datei geändert wurde.
- size zahl Ist erfüllt, wenn die Datei zahl viele Blöcke hat.

Beispiele

Beispiele:

```
find /home/paul      type -f *.html
```

Sucht alle Dateien mit der Endung html

```
find /home/otto      ! [-user otto ] -exec ls -l {} \;
```

Bewirkt die Anzeige aller Dateien, die nicht zu Otto gehören.

```
find /home/otto      -atime +10 -print
```

Bewirkt die Anzeige aller Dateien , auf die in den letzten 10 Tagen nicht zugegriffen wurde.

```
find . [ -size +10 -o mtime +7 ] -ls -l {} \;
```

Bewirkt die Anzeige aller Dateien, die eine Größe von mehr als 10 Blöcken haben oder in den letzten 7 Tagen nicht verändert wurden.

Beispiele:

```
find -name a.out
```

- Sucht alle Dateien mit dem Namen a.out, Programmieren mit g++
- find -name a.out -o -name "*.o"
- Sucht alle Dateien und Verzeichnisse mit dem Namen a.out oder mit *.o
- find ./ -type f -name "a?"
- Sucht alle Dateien mit der Maske a?
- find -maxdepth 2 -type f -name "a? "

Sucht alle Dateien mit der Maske a?, Maximal aber ein Unterverzeichnis

Beispiele:

```
find /home/paul      type -f *.html
```

Sucht alle Dateien mit der Endung html

```
find /home/otto      ! [-user otto ] -exec ls -l {} \;
```

Bewirkt die Anzeige aller Dateien, die nicht zu Otto gehören.

```
find /home/otto      -atime +10 -print
```

Bewirkt die Anzeige aller Dateien , auf die in den letzten 10 Tagen nicht zugegriffen wurde.

```
find . [ -size +10 -o mtime +7 ] -ls -l {} \;
```

Bewirkt die Anzeige aller Dateien, die eine Größe von mehr als 10 Blöcken haben oder in den letzten 7 Tagen nicht verändert wurden.

Beispiele:

```
find -name a.out
```

- Sucht alle Dateien mit dem Namen a.out, Programmieren mit g++
- ```
find -name a.out -o -name "*.o"
```
- Sucht alle Dateien und Verzeichnisse mit dem Namen a.out oder mit \*.o
- ```
find ./ -type f -name "a?"
```
- Sucht alle Dateien mit der Maske a?
- ```
find -maxdepth 2 -type f -name "a? "
```

Sucht alle Dateien mit der Maske a?, Maximal aber ein Unterverzeichnis

```
find -type f -name "a?" -exec head {} \;
```

```
find -type f -name "a?" -exec head {} \; | sort
```

```
find -type f -name "a?" -exec head {} \; | sort | uniq
```

<http://publib.boulder.ibm.com/infocenter/aix/v6r1/index.jsp?topic=/com.ibm.aix.cmds/doc/aixcmds2/egrep.htm>