

### Usage

Grep standard output (i.e. a stream of text)

```
grep [-options] 'string'
```

Grep the content of a file

```
grep [-options] 'string' filename
```

Wildcards are accepted in *filename*.

### General Regular Expression Processor

| Operation                                       | Option | Example                                                                    |
|-------------------------------------------------|--------|----------------------------------------------------------------------------|
| Find a string in 1 or more files                |        | grep 'string' filename1 filename2 ... filename <i>n</i>                    |
| Case insensitive search                         | i      | grep -i 'string' filename                                                  |
| Use regular expressions (regex)                 |        | grep 'regex' filename                                                      |
| Look for words                                  | w      | grep -w 'word' filename                                                    |
| Display <i>n</i> lines after matching string    | A      | grep -A <i>n</i> 'string' filename                                         |
| Display <i>n</i> lines before matching string   | B      | grep -B <i>n</i> 'string' filename                                         |
| Display <i>n</i> lines around matching string   | C      | grep -C <i>n</i> 'string' filename                                         |
| Recursive grep                                  | r      | grep -r 'hackers-club.cn' /var/log/apache2/archives/                       |
| Return all lines which don't match the pattern  | v      | grep -v 'warning' /var/log/syslog                                          |
| Use regex                                       | e      | grep -e 'string1' -e 'string2' filename<br>grep --regexp 'string' filename |
| Return lines starting with 'al'                 |        | grep -e '^al' filename                                                     |
| Use extended regex                              | E      | grep -E 'apache wheel root' filename                                       |
| Get lines containing 1+ w                       |        | grep -E 'w+' filename                                                      |
| Get lines with 3 w in a row (www)               |        | grep -E 'w{3}' filename                                                    |
| Get lines containing between 3 and 6 m in a row |        | grep -E 'm{3,6}' filename                                                  |
| Get lines containing jason or jackson           |        | grep -E 'ja(s)cks)on' filename                                             |
| Count results                                   | c      | grep -c 'error' /var/log/syslog                                            |
| Display filename                                | l      | grep -l 'string' /var/log/*                                                |
| Only show the matching part of the string       | o      | grep -o 'string' filename                                                  |
| Show line number                                | n      | grep -n 'string' filename                                                  |

**About grep -E:** In basic regular expressions the meta-characters '?', '+', '{', '|', '(', and ')' lose their special meaning; instead use the backslashed versions '\?', '\+', '\{', '\|', '\(', and '\)'.  
GNU grep -E emulates classic meta-characters. The command 'grep -E '{1}' searches for the 2-character string '{1}' instead of reporting an error. POSIX allows this behavior as an extension, but portable scripts should avoid it.



By **TME520** (TME520)  
[cheatography.com/tme520/](https://cheatography.com/tme520/)  
[tme520.com](https://tme520.com)

Published 4th May, 2015.  
Last updated 25th February, 2020.  
Page 1 of 2.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

### Regular expressions : wildcards

|             |                                                            |
|-------------|------------------------------------------------------------|
| .           | Any character.                                             |
| ?           | Optional and can only occur once.                          |
| *           | Optional and can occur more than once.                     |
| +           | Required and can occur more than once.                     |
| {n}         | Previous item appears exactly <i>n</i> times.              |
| {n,}        | Previous item appears <i>n</i> times or more.              |
| {,m}        | Previous item appears <i>n</i> times maximum.              |
| {n,m}       | Previous item appears between <i>n</i> and <i>m</i> times. |
| [[:alpha:]] | Any lower and upper case letter.                           |
| [[:digit:]] | Any number.                                                |
| [[:alnum:]] | Any lower and upper case letter or digit.                  |
| [[:space:]] | Any whitespace.                                            |
| [A-Za-z]    | Any lower and upper case letter.                           |
| [0-9]       | Any number.                                                |
| [0-9A-Za-z] | Any lower and upper case letter or digit.                  |

### Regular expressions : anchors and positions

|     |                    |                                                     |
|-----|--------------------|-----------------------------------------------------|
| ^   | Beginning of line. | grep '^Once upon a time' /home/livres/lovestory.txt |
| \$  | End of line.       | grep 'divorced.\$' /home/livres/lovestory.txt       |
| ^\$ | Empty line.        |                                                     |
| \<  | Start of word.     | grep '\<love\>' /home/manga/seinen.txt              |
| \>  | End of word.       |                                                     |

### Characters to escape

|    |                        |
|----|------------------------|
| \  | .                      |
| [  | ^                      |
| \$ | '                      |
| *  | - (start of line only) |



By **TME520** (TME520)  
[cheatography.com/tme520/](https://cheatography.com/tme520/)  
[tme520.com](https://tme520.com)

Published 4th May, 2015.  
 Last updated 25th February, 2020.  
 Page 2 of 2.

Sponsored by **Readable.com**  
 Measure your website readability!  
<https://readable.com>