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

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

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.

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 *n* times.
{n,}	Previous item appears *n* times or more.
{,m}	Previous item appears *n* times maximum.
{n,m}	Previous item appears between *n* and *m* 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)