# 1. What is glob

|  |
| --- |
| "Glob" is the common name for a set of Bash features that match or expand specific types of patterns.  Some synonyms for globbing (depending on the context in which it appears) are [pattern matching](http://tiswww.case.edu/php/chet/bash/bashref.html#SEC36), pattern expansion, filename expansion, and so on.   A glob may look like \*.txt and, when used to match filenames, is sometimes called a "wildcard".  Traditional shell globs use a very simple syntax, which is less expressive than a [RegularExpression](http://mywiki.wooledge.org/RegularExpression). |

|  |  |  |
| --- | --- | --- |
| Even if a file contains internal whitespace, the expansion of a glob that matches that file will still preserve each filename as a single word. For example, comparision:  **List to test**:  [root@TESTBED-VOD-CMS dath]# ls | grep '\.txt$'  0987543432-count.txt  103.237.144.19-count.txt  abnormalGetRequest.txt  abnormalRequest.txt  get-request-count.txt  result .txt => Name with whitespace  suspectedUserId.txt  test.txt  userIdcount.txt  **Test**  Save case:  [1](http://mywiki.wooledge.org/glob#CA-aaf1d6a7185af10f101f4ac82c9de2edee960d7c_1) # This is safe even if a filename contains whitespace:  [2](http://mywiki.wooledge.org/glob#CA-aaf1d6a7185af10f101f4ac82c9de2edee960d7c_2) for f in \*.tar; do  [3](http://mywiki.wooledge.org/glob#CA-aaf1d6a7185af10f101f4ac82c9de2edee960d7c_3) tar tvf "$f"  [4](http://mywiki.wooledge.org/glob#CA-aaf1d6a7185af10f101f4ac82c9de2edee960d7c_4) done  Fail case:  [7](http://mywiki.wooledge.org/glob#CA-aaf1d6a7185af10f101f4ac82c9de2edee960d7c_7) for f in $(ls | grep '\.tar$'); do  [8](http://mywiki.wooledge.org/glob#CA-aaf1d6a7185af10f101f4ac82c9de2edee960d7c_8) tar tvf "$f"  [9](http://mywiki.wooledge.org/glob#CA-aaf1d6a7185af10f101f4ac82c9de2edee960d7c_9) done  **Result**:  Save case:   |  | | --- | | + for f in '\*.txt'  + tar tvf 'result .txt'  tar: This does not look like a tar archive  tar: Skipping to next header  tar: Exiting with failure status due to previous errors |   Fail case:   |  | | --- | | + for f in '$(ls | grep '\''\.txt$'\'')'  + tar tvf result  tar: result: Cannot open: No such file or directory  tar: Error is not recoverable: exiting now  + for f in '$(ls | grep '\''\.txt$'\'')'  + tar tvf .txt  tar: .txt: Cannot open: No such file or directory  tar: Error is not recoverable: exiting now |   The first example has no such problem, because the filenames produced by the glob do not undergo any further word-splitting.  In the second example above, the output of ls is filtered, and then the result of the whole pipeline is [divided into words](http://mywiki.wooledge.org/WordSplitting): result and .txt  For more such examples, see [BashPitfalls](http://mywiki.wooledge.org/BashPitfalls). |