

snippet - rename multiplo di file - UPDATED

Scritto da {ga=blackout}

Stamattina ho imparato a usare il bash script per rinominare tanti file con un solo comando. Prima facevo in maniera molto più dispendiosa e tediosa.

I miei file erano del tipo:

resized_xxxxxxx_large.jpg e li dovevo rinominare in **xxxxxxx_normal.jpg** con xxxxxxxx = 31 caratteri

Allora il prode xanio mi consiglia di usare **rename** per fare il task.

```
{codecitation class="brush: bash;"}  
  
rename -v 's/resized_(w{31})_large.jpg/$1_normal.jpg/' *.jpg
```

```
{/codecitation}
```

oppure con un semplice ciclo for se si tratta solo dell'estensione

```
{codecitation class="brush: bash;"}  
  
for file in $(ls *.jpg); do  
    mv "$file" "${file%_large.jpg}_normal.jpg"done
```

snippet - rename multiplo di file - UPDATED

Scritto da {ga=blackout}

```
for f in *.tgz; do mv $f `basename $f .tgz`.tar.gz; done;
```

{/codecitation}

ancora un altro esempio (by Allanon)

```
{codecitation class="brush: bash;"}  
find $1 -name *.cbz -exec rename -v 's/(w+)(d{1}).cbz/${1}00${2}.cbz/' {} ;
```

{/codecitation}

Ecco uno scorcio di man di **rename**

Syntax

```
rename [ -v ] [ -n ] [ -f ] perlexpr [ files ]
```

-vVerbose: print names of files successfully renamed.-nNo Action: show what files would have been renamed.-fForce: overwrite existing files.perlexprPerl Expression

Regular Expressions

^ matches the beginning of the line \$ matches the end of the line .Matches any single character(character)*match arbitrarily many occurrences of (character)(character)?Match 0 or 1 instance of (character)[abcdef]Match any character enclosed in [] (in this instance, a b c d e or f) ranges of characters such as [a-z] are permitted. The behaviour of this deserves more description. See the page on [grep](#)

for more details about the syntax of lists.

[^abcdef] Match any character *NOT* enclosed in [] (in this instance, any character other than a b c d e or f)
(character){m,n} Match m-n repetitions of (character)
(character){m,} Match m or more repetitions of (character)
(character){,n} Match n or less (possibly 0) repetitions of (character)
(character){n} Match exactly n repetitions of (character)
(expression) Group operator.n Backreference - matches nth group
expression1|expression2 Matches expression1 or expression 2. Works with GNU sed, but this feature might not work with other forms of sed.
w matches any single character classified as a “word” character (alphanumeric or “_”)
W matches any non-“word” characters
 matches any whitespace character (space, tab, newline)
S matches any non-whitespace character
d matches any digit character, equiv. to [0-9]
D matches any non-digit character