

Overview

- Batch rename files or folders within Alfred using [Regular Expression](#)
- Add additional steps such as find and replace, remove space, remove diacriticals, capitalize, add current date, file modified date, custom number sequence etc.
- Quick look to preview all the new names
- Recent Expressions
- Presets, including default ones ready to be used

Triggering

- keyword **regex** - main workflow keyword
- keyword **regex?** - learn about the workflow
- Hotkey - show Alfred with the keyword **regex**
- Hotkey - use the last expression with no interaction
- File Action - rename files or folders selected in Alfred file browser

Note that the main keyword and hot keys require you to select item(s) in the Finder first.

How to Use

Rename Only

You can use the workflow to rename a single file without using a regular expression, like in Finder:

1. Select a single file in the Finder (or in Alfred file browser)
2. Type RegEx
3. Type the new filename e.g. **new name.txt**

You can optionally use modifiers and filename additions (see below).

Rename with Regular Expression

Here is a command line with all features, including the optional ones, as explained below:

```
regular expression@new filename $d-$m-$y ##[1]{find@replace}{-modifiers}
```

Step by step:

- (1) Type the regular expression to group elements of the previous filename:

```
^([ ^ ]+) (.*)\.(.*)
```

The workflow will group using **\$1**, **\$2**, **\$3** and so on. Later you can use them to build a new filename.

- (2) When the regular expression is finished type then an **at sign** at the end:

```
^([ ^ ]+) (.*)\.(.*)@
```

- (3) After the at sign type the new filename (here you can use the groups formed in the first step):

```
^([ ^ ]+) (.*)\.(.*)@ $1.$3
```

- (4) **Optionally** add current date or any other filename constants (read below about them):

```
^([ ^ ]+) (.*)\.(.*)@ $1 $d-$m-$y.$3
```

- (5) **Optionally** add a custom sequence of numbers e.g. with two leading zeros starting at one (read how to build a custom sequence below):

```
^([ ^ ]+) (.*)\.(.*)@ $1 $d-$m-$y ##[1].$3
```

- (6) **Optionally** find and replace characters in the original filename utilizing the syntax **{find@replace}**:

```
^([ ^ ]+) (.*)\.(.*)@ $1 $d-$m-$y ##[1].$3{__@-}
```

- (7) **Optionally** utilize one of the modifiers (read about them below) to e.g. set all letters to lowercase:

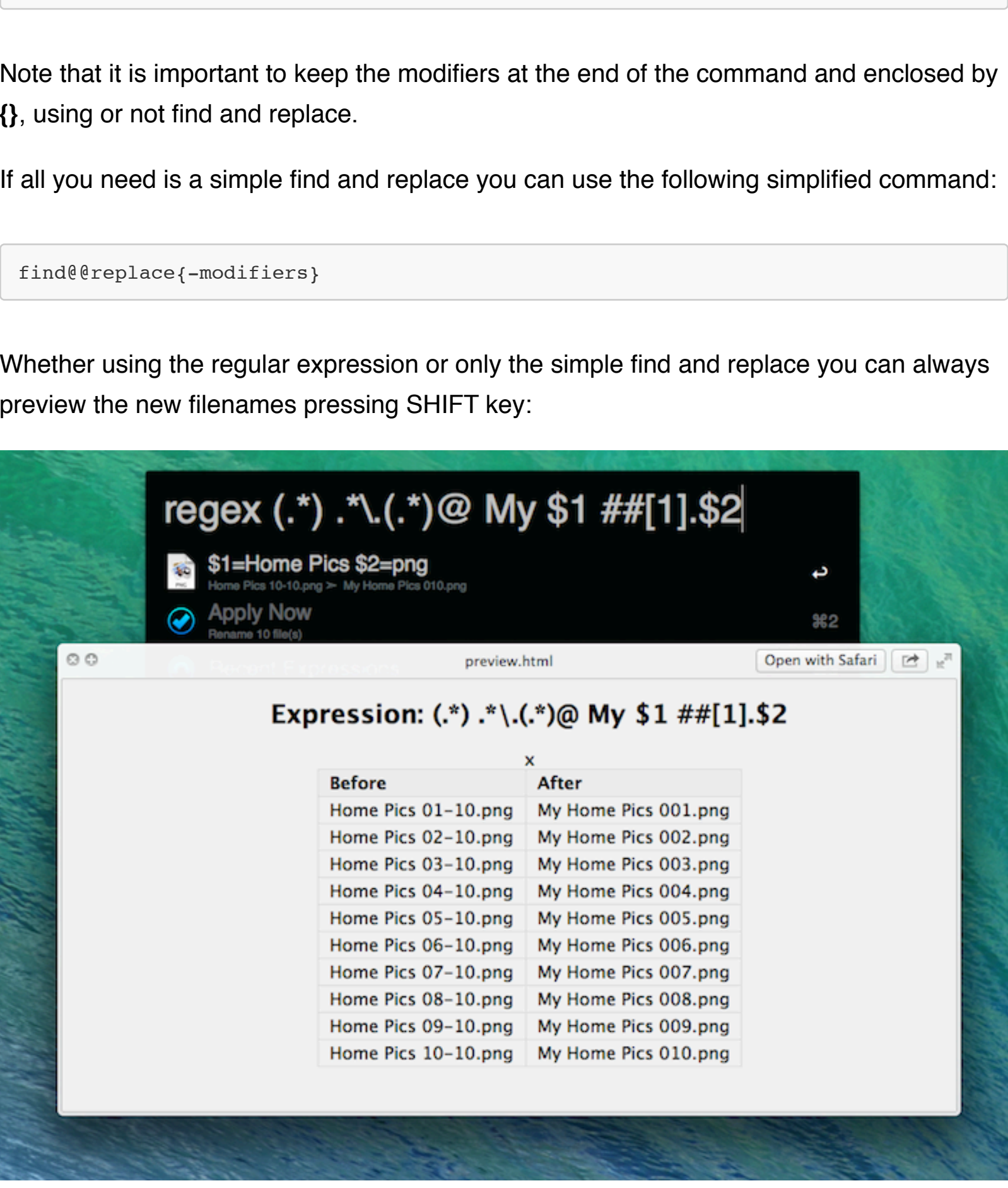
```
^([ ^ ]+) (.*)\.(.*)@ $1 $d-$m-$y ##[1].$3{__@-}{-l}
```

Note that it is important to keep the modifiers at the end of the command and enclosed by {}, using or not find and replace.

If all you need is a simple find and replace you can use the following simplified command:

```
find@replace{-modifiers}
```

Whether using the regular expression or only the simple find and replace you can always preview the new filenames pressing SHIFT key:



Modifiers

- -d to remove diacriticals
- -s to remove space
- -c to Capitalize Name
- -t to Title name
- -u to UPPERCASE
- -l to lowercase
- -_ to replace underscore to space
- -b to replace space to underscore

Use one or more at the end of the command enclosed by {}, for example:

```
(.*)\.(.*)@ $1.$2{-s-u}  
-@_{-u}
```

New Filename Constants

You can add the following constants to the new filename:

Current Date

- \$d = day e.g. 02
- \$m = month e.g. 04
- \$mm = month e.g. April
- \$y = year e.g. 2013
- \$h = hour e.g. 02
- \$n = minutes e.g. 54
- \$s = seconds e.g. 30

Created

- \$cd = day e.g. 02
- \$cm = month e.g. 04
- \$cmm = month e.g. April
- \$cy = year e.g. 2013
- \$ch = hour e.g. 02
- \$cn = minutes e.g. 54
- \$cs = seconds e.g. 30

Modified

- \$mod = day e.g. 02
- \$mom = month e.g. 04
- \$momm = month e.g. April
- \$moy = year e.g. 2013
- \$moh = hour e.g. 02
- \$mon = minutes e.g. 54
- \$mos = seconds e.g. 30

Image Dimensions

- \$iw = width in pixels
- \$ih = height in pixels

Sequence

You can add a number sequence to the new filename by utilizing # (each one represents a leading zero) followed by the start number enclosed in brackets e.g. **###[1]** which means 3 leading zeros starting at 1.

Create New Folder and Move

Files

You can also use the regular expression to create a new folder and move the file(s) there.

Utilize a forward slash ("/") before the new name:

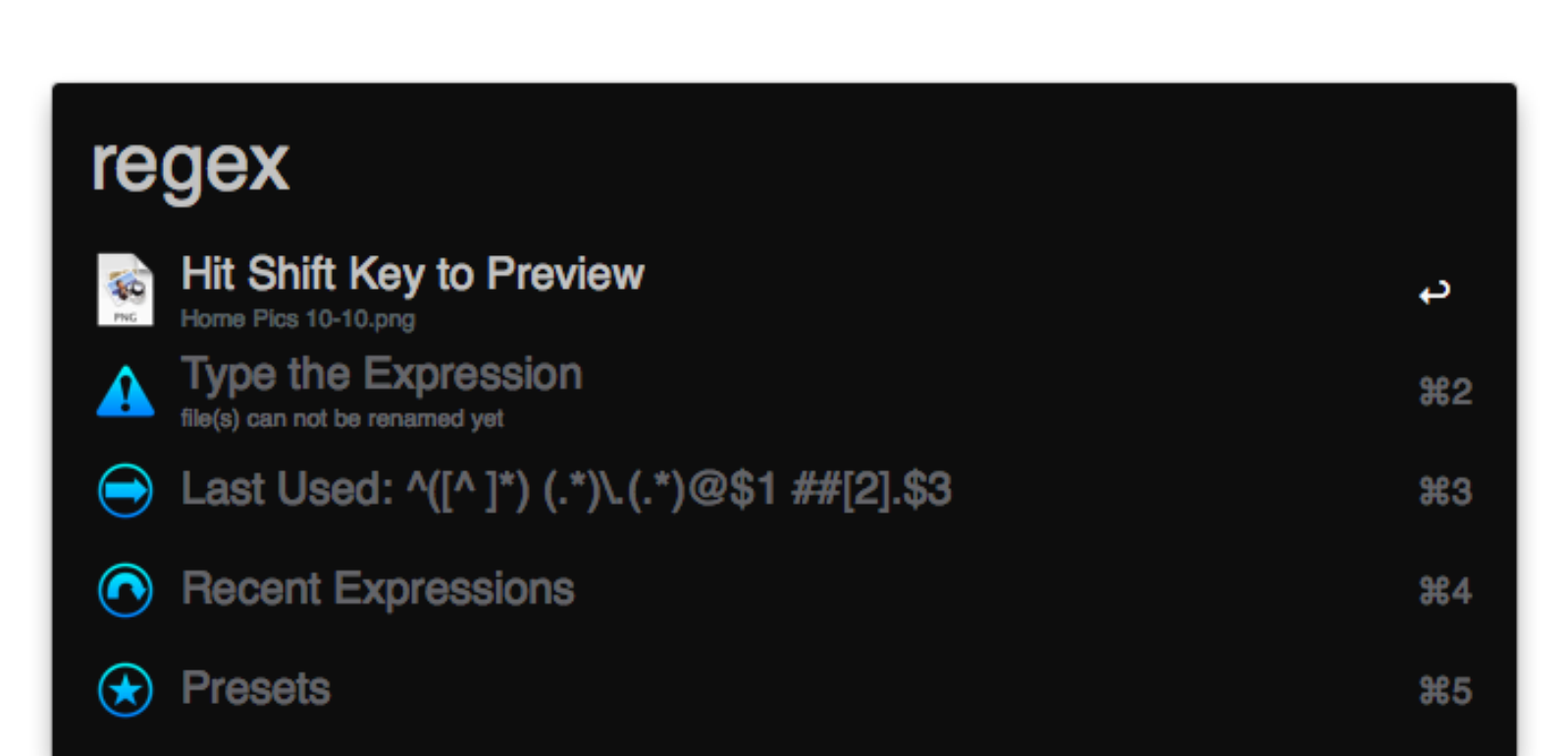
```
(.*)@$m-$y/file.png
```

A new folder will be created based on current month (\$m) and year (\$y) and the file will be renamed and moved to there.

Recent Expressions and Presets

To make things faster the workflow:

- Display the last used expression
- In Recent Expressions there are a list of the last 40 used expressions
- In Presets there are some ready to use expressions a long with the user favorites ones



The workflow saves the last used regular expression in the main workflow menu, a list of recent regular expressions and a list of Presets.

To add a Recent expression to Presets just type a plus symbol followed by the preset name: **+For My Videos**

While in Presets just type a minus symbol "-" to remove from Presets.