ZSH Globbing (extended)

glob expression

- `ls <path>/*/*` : lista todos los ficheros a 2 niveles del actual
- `ls <path>/**/*` : lista todos los ficheros en cualquier directorio
- `ls <path>/**/*.txt` : lista todos los ficheros .txt en cualquier directorio por debajo
- `ls **/README.*`: lista todos los ficheros README en cualquier directorio por debajo
- `ls **/*(.)`: listar solo ficheros
- `ls **/*(/)`: listar solo directorios
- coindidencia parcial:

```
## Starts with READ
$ ls **/(READ)*.*
## Ends With READ
$ ls **/*(READ).*
## Contains READ Anywhere
$ ls **/*(READ)*.*
```

glob operators

mucho mas:http://zsh.sourceforge.net/Doc/Release/Expansion.html#Filename-Generation

- `ls <path>/**/*<1-10>.txt` : lista cualquier fichero contenido en cualquier subdirectorio que acabe entre 1 y 10 (y .txc)
- `ls <path>/**/[a]*.txt` : ficheros que empiezan por a
- `ls <path>/**/(ab|bc)*.txt` : ficheros que empieza por ab* o bc*
- `ls <path>/**/[^cC]*.txt` : ficheros que no empiecen por **c** o **C**

```
# All files that start with A
$ ls **/[A]*(.)
# All files that start with A or a
$ ls **/[Aa]*(.)
# All Files that contain the number
$ ls **/*[1]*(.)
# Any files that end in a vowel
$ ls **/*[aeiouy](.)
# Files that don't start with A or a
$ ls **/[^Aa](.)
# Files that end in a number
$ ls **/*<1-10>(.)
```

glob qualifiers

- http://zsh.sourceforge.net/Doc/Release/Expansion.html
- se usa una notación entre paréntesis para darle más criterios a las búsquedas
- se pueden agrupar los diferentes *qualifiers* (no permite espacios, lo que hace ciertas sentencias difíciles de leer)
- `/` : solo directorios

- `.` : solo ficheros regulares
- Ln : indica tamaño exacto → **L2m** : 2MB
    - + : mayor que → **L+2k** : mayor de 2k
    - - : menor que
- m : modificación
    - - : modificados en las últimas X unidades (entre)
    - + : modificados más allá (antes de)
    - unidades:
        - M : meses → **mM-1** : último mes
        - w : semanas
        - h : horas
        - m : minutos
        - s : segundos
- o / O : ordena (ascendente / descendente) cualquier otro cualificador (pero va por separado, no inscrustado en el qualificador que queremos ordenar
    - [a,b] : muestra los ficheros entre posiciones **a** y **b**
    - [c] : muestra solo el fichero de posición **c**

ejemplo uso cualificadores

```
ls -l ./**/*(.mh-24) # ficheros modificados las últimas 24h
ls -l ./**/*(.omh-24) # ERROR
ls -l ./**/*(.mh-24 om[1,4]) # últimas 24h, ordenados de más
reciente a menos, solo del 1 al 4
```

- e : estring → evalua una condición y solo pasan los que la cumplen
    - `ls ./**/*(e:'[[ $REPLY COND ]]':)`
        - uso de **:** como separadores
        - **[[COND]]** al uso
        - $REPLY va conteniendo cada elemento que cumpla con **/**/***

Ejemplos

- `print -l <path>/**/*(/)` : solo directorios
- `print -l <path>/**/*(.)` : solo ficheros regulares
- `print -l <path>/**/*(L0)` : ficheros vacíos
- `print -l <path>/**/*(Lk+3)` : ficheros más grandes de 3k
- `print -l <path>/**/*(mh-1)` : ficheros modificados en la última hora
- `ls -l <path>/**/*(om[1,3])` : mostrar ordenadamente ficheros modificados y mostrar los 3 últimos
- `ls -l zsh_demo/**/*(.Lm-2mh-1om[1,3])` ≡ `ls -l zsh_demo/**/*(. Lm-2 mh-1 om [1,3])` : ficheros de menos de 2MB, modificados en la última hora, mostrar los tres primeros (por orden de modificación

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/linux:zsh:globbing

Last update: **23/03/2020 22:57**