

Weekly Report: Wildcard

| WILDCARDS/FILE GLOBBING CHEAT SHEET | | |
|---|---|--|
| The * Wildcard | The ? Wildcard | |
| List all txt and python files | Copy all the files that have 2 characters between 2 letters. | |
| ls -A *.txt *.py | cp Downloads/b??k.pdf Documents/ | |
| List all the files that have 'demo' in the name | List all the files with a 2 letter file extension | |
| ls -A *demo* | ls -A Scripts/*.?? Programs/program.?? Downloads/setup*.?? | |
| Move all the files inside a directory | Remove all the hidden files in a given directory | |
| mv Pictures/* ~/Backup/ | rm Documents/.??*.doc | |
| Delete all files that start with a given word | List all the hidden files that have a 4 letter file extension | |
| rm Downloads/copy* Documents/new*.docx | ls -A .??*.???? | |

| The [] wildcard | | |
|---|--|--|
| List all the text files that start with an uppercase letter and all the python files that start with a number | | |
| ls -A [A-Z]*.txt [0-9]*.py | | |
| List all the ruby files that do not start with a number. | | |
| ls -A [!0-9]*.rb | | |
| List all the files that have one of the characters in a set before the extension | | |
| List *[xyz].* | | |
| List all files whose name begins with any 3 combination of numbers and the current user's username: | | |
| ls -A [0-9][0-9][0-9]\$USER* | | |

| Wildcard | Matches | Example |
|----------|--|-----------------------|
| * | 0 or multiple characters | ls *.pdf |
| ? | 1 character | ls program?.py |
| [] | 1 character from a given set of characters | ls document[A-Z].doc |
| [!] | The opposite of the given set | ls new-doc[!0-9].docx |

| POSIX CHARACTER CLASSES | | |
|-------------------------|--|--|
| POSIX class | Equivalent to | Matches |
| [aInum:] | [A-Za-z0-9] | Digits, uppercase and lowercase letters |
| [alpha:] | [A-Za-z] | Upper- and lowercase letters |
| [ascii:] | [\x00-\x7F] | ASCII characters |
| [blank:] | [\t] | Space and TAB characters only |
| [cntrl:] | [\x00-\x1F\x7F] | Control characters |
| [digit:] | [0-9] | Digits |
| [graph:] | [* [:cntrl:]] | Characters which have graphic representation |
| [lower:] | [a-z] | Lowercase letters |
| [print:] | [[[:graph:]]] | Graphic characters and space |
| [punct:] | [~!@#\$%^&*()'+,-;=<>?~][~!@#\$%^&*()'+,-;=<>?~] | Punctuation characters except letters and digits |
| [space:] | [\t\n\r\f\v] | All whitespace characters |
| [upper:] | [A-Z] | Uppercase letters |
| [word:] | [A-Za-z0-9_] | Word characters |
| [xdigit:] | [0-9A-Fa-f] | Hexadecimal digits |

Brace expansion and how to use it

- Brace expansions are useful techniques that help you do every command in short time since with a single command you can create modify and even delete a bunch of files and all kind of things in the terminal and in the work space your are in.
- How to use it: there are several ways to use brace expansions first one is using a script in the terminal stating with `#!/bin/bash` of the specific task you want to use for the other one is using brackets `{}` also the regular way using `/`` in the terminal for the script can use it as this is a complete document specifying what you need to modify or do. `echo {one,two,three,four}``` also with for i in {3..7} do echo $i done and expanding names and directories ls *.txt`