

# Week Report 6

---

## Summary on Wildcards

### A. Summary of the Presentation.

- Wildcards:
  - Definition: Wildcards represent letter and characters use to specify a file name for searches.
  - Wildcards are also know as metacharacter wildcards.
    - You can use a wildcard to get a long list of all files in the current directory starting with new.
    - Use wildcards to manage to directories faster
    - Move or delete a group of files
    - Locate files on a portion of their filenames
    - create files and directories quicker
  - Wildcard is a star (\*) character.
    - Example: `ls *.txt`
      - This will match all files edn in .txt
- Benefits of using Wildcard:
  - when you want to list all files with a particular file extension
  - when you remember a portion of the name
  - when you want to copy, move or move all files that match a particular naming convention
  - Example:
    - For example, `ls *.txt` will match all files that end in .txt regardless of the size of the file name.

```

Terminal
File Edit View Terminal Tabs Help

[16:27:55](adrian@G752VL2 dir)
>ls *.txt 1
1233 file.txt 'another file.txt' _file.txt
[16:28:01](adrian@G752VL2 dir)
>ls *.txt *.pdf 2
1233 file.txt 'another file.txt' f2.pdf f3.pdf _file.txt
[16:28:12](adrian@G752VL2 dir)
>ls file.* 3
ls: cannot access 'file.*': No such file or directory
[16:28:23](adrian@G752VL2 dir)
>ls *file.* 4
1233 file.txt 'another file.txt' _file.txt
[16:28:34](adrian@G752VL2 dir)
>

```

!

- Formula: List only the CSS files

When working with wildcards, keep in mind two things:

- **What do the files have in common?**
  - This is the part of the file name which is the same in all the files you want to match. For example, the file extension.
- **What part of the file name is irrelevant?**
  - This is the part of the file name that it is irrelevant in the matching. In our example, this is the file name

Let's assume that you want to list only the **css** files inside this directory in a single column.

The command for achieving this would be **ls -l \*.css**

The **\*** wildcard replaces the name of the files because we do not care about the file name in this instances. In the command **we keep the extension because that is what the files must have in common.**

```

1/1 + [ ] [ ]
1: Terminal
~/website via [ ]
→ ls -X1
assets
docs
scripts
main.css
media.css
reset.css
style.css
index.html
script.js
background.png
~/website via [ ]
→ ls -X1 *.css
main.css
media.css
reset.css
style.css
~/website via [ ]
→

```

!

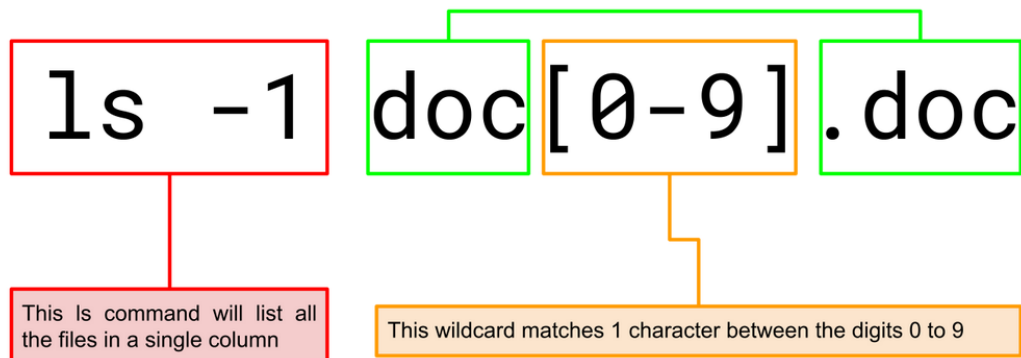
- List files inside a single column.
  - First: ls -X1
  - Second: ls -X1 \*.css
- The Wildcard (\*) replaces the name of the files because we do not care about the file name in this instances.
- The Wildcard:
  - The brackets wildcard match a single character in a range.
  - The brackets wildcard use the exclamation mark to reverse the match.
    - Example: match everything except vowels [!aeiou]

- Example: any character except numbers [!0-9]
- Examples:
  - 1. To match all files that you have a vowel after letter f:
    - Example: `ls f[aeiou]*`
  - 2. To match all files that do not have a vowel after letter f:
    - Example: `ls f![aeiou]*`
  - 3. To match all the files whose name does not have a number in their file name:
    - Example: `ls [!0-9]*`
  - 4. to match all files whose name begins with a letter from a-p or start with letter s or c:
    - `ls[a-psc]*`

- Whilecard Rule:

## Let's breakdown an example

This is the part of the file name which we care about. The files we are trying to list must start with the word **doc** and must be **.doc file**. However, we do not care about the number of the file so we use a wildcard to match all files from **doc0.doc** to **doc9.doc**



## Using Wildcards / File Globbing (quick reference)

Wildcard	Description
*	Matches zero or more characters in a filename
?	Matches any one character in a filename
[acf]	Matches one of multiple characters in a filename; in this example, a, c, or f
[a-f]	Matches one of a range of characters in a filename; in this example, any character from a through f
[!a-f]	Matches filenames that don't contain a specified range of characters; in this example, filenames that don't contain a through f

- Using Brace Expansion

- Brace expansion {} is not a wildcard
  - To create a whole directory structure in a single command:
    - `mkdir -p music/{jazz,rock}/ {mpfiles, videos, offiles}/new{1..3}`
    -
- To create a N number of files use:
  - `touch website {1..5}.html`
  - `touch file{A..Z}.html`
- to remove multiple files in a single directory
  - `rm -r{dir1,dir2,dir3,file.txt,file.py}`