

# Notes 4 Wildcards

---

## Wildcards

- Represent letters and characters used to specify a file name for searches

### The \* Wildcard

- A star(\*) alone matches anything and nothing and matches any number of characters.
- Examples
  - `ls *.txt`
    - will match all files that end in .txt regardless of the size of the file name
  - `ls *.txt *.odf`
    - lists all .txt and .pdf files
  - `ls file.*` list all files with the string "file" regardless of extension
- Mostly used when you need to list files with a particular file extension, when you do not remember the complete name of a file but you remember a portion of the name, when you want to copy, move, or remove all files that match a particular naming convention.

### The ? Wildcard

- Matches precisely one character.
- useful when working with hidden files(dot files)
  - to list all hidden files use "`ls .??*`"
- Examples
  - `ls ./.*`
    - list all hidden files in current directory
  - `ls b??k*`
    - lists all files that have two letters between b and k

### The [] Wildcard

- Matches a single character in a range
- You can use the ! to reverse the match
- examples
  - `ls f[aeiou]*`
    - lists all files that have a vowel after the letter f
  - `ls f![aeiou]*`
    - lists all files that do not have a vowel after the letter f
  - `ls *[0-9]*`
    - lists files that have at least one number in the file name

### Brace Expansion

- Not a wildcard but another feature of bash that allows you to generate arbitrary string to use with commands
- Example

- `mkdir -p music/{jazz,rock}/{mp3files,videos,oggfiles}/new{1..3}`
  - create a whole directory structure in one command
- `touch website{1..5}.html`
  - creates n number of files
- `rm -r {dir1,dir2,file.txt,file.py}`
  - removes multiple files in a single directory