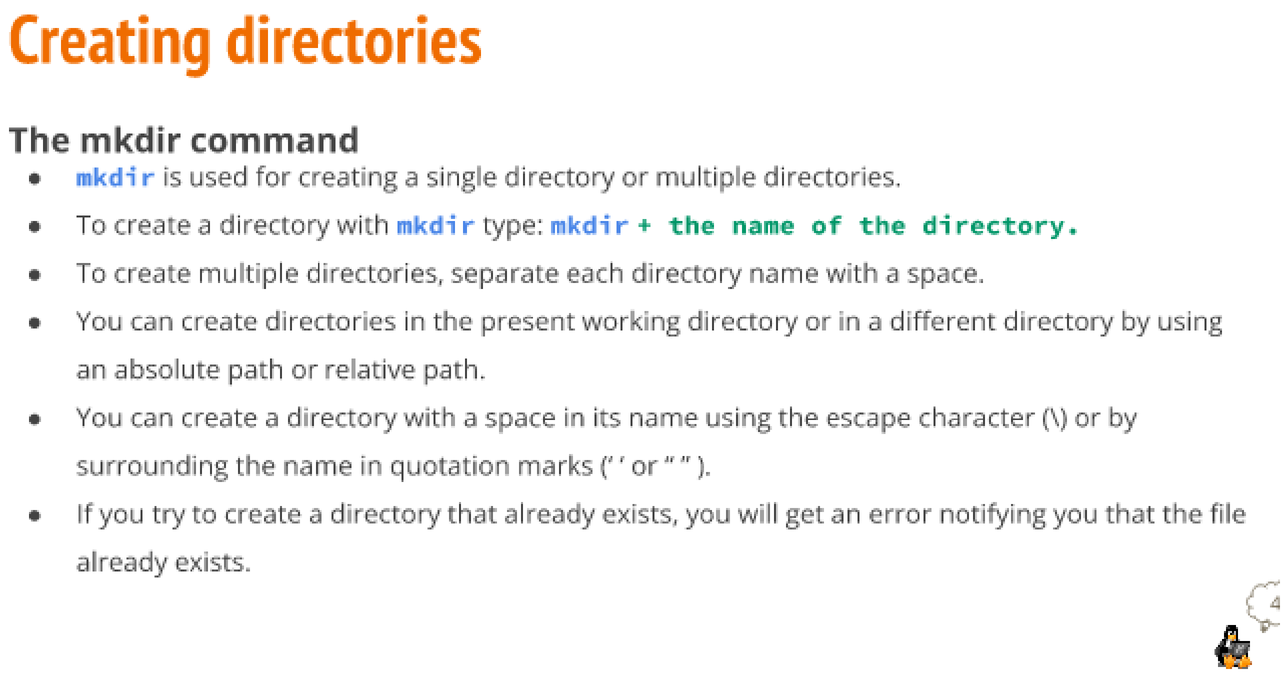
Lecture 4 | Managing Files and Directories

Creating Files and Directories

- MKDIR Command is used for creating a single directory or multiple directories.

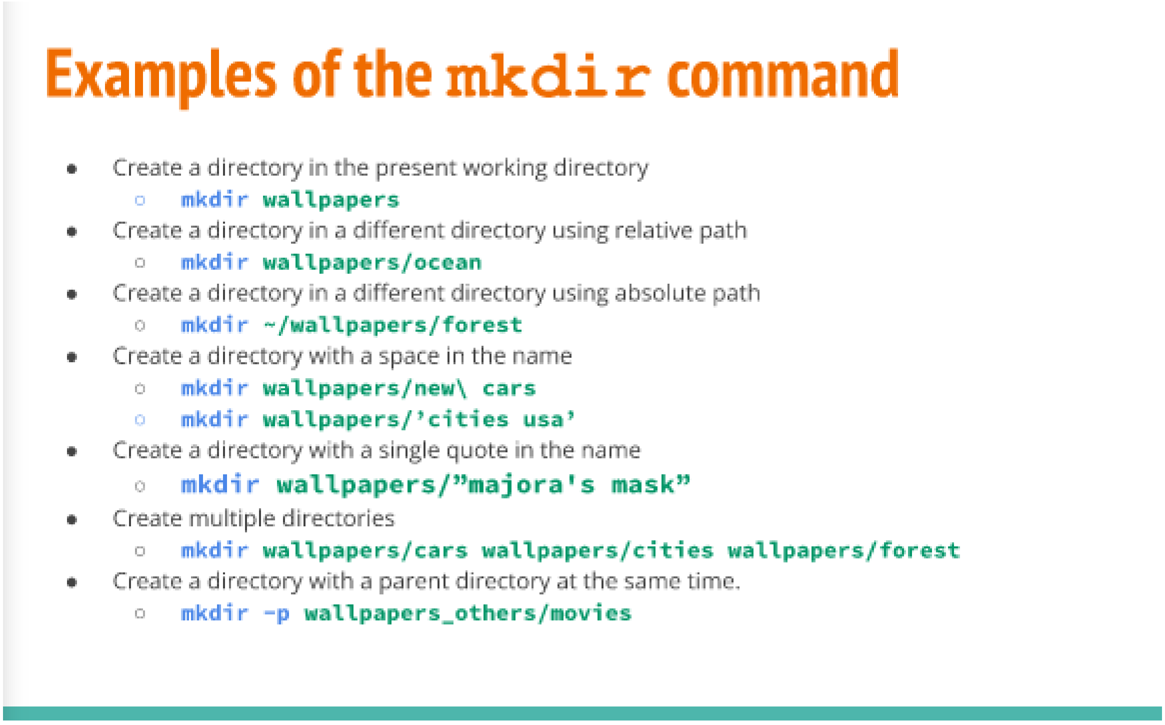


Creating directories

The mkdir command

- `mkdir` is used for creating a single directory or multiple directories.
- To create a directory with `mkdir` type: `mkdir + the name of the directory.`
- To create multiple directories, separate each directory name with a space.
- You can create directories in the present working directory or in a different directory by using an absolute path or relative path.
- You can create a directory with a space in its name using the escape character (\) or by surrounding the name in quotation marks (' ' or " ").
- If you try to create a directory that already exists, you will get an error notifying you that the file already exists.

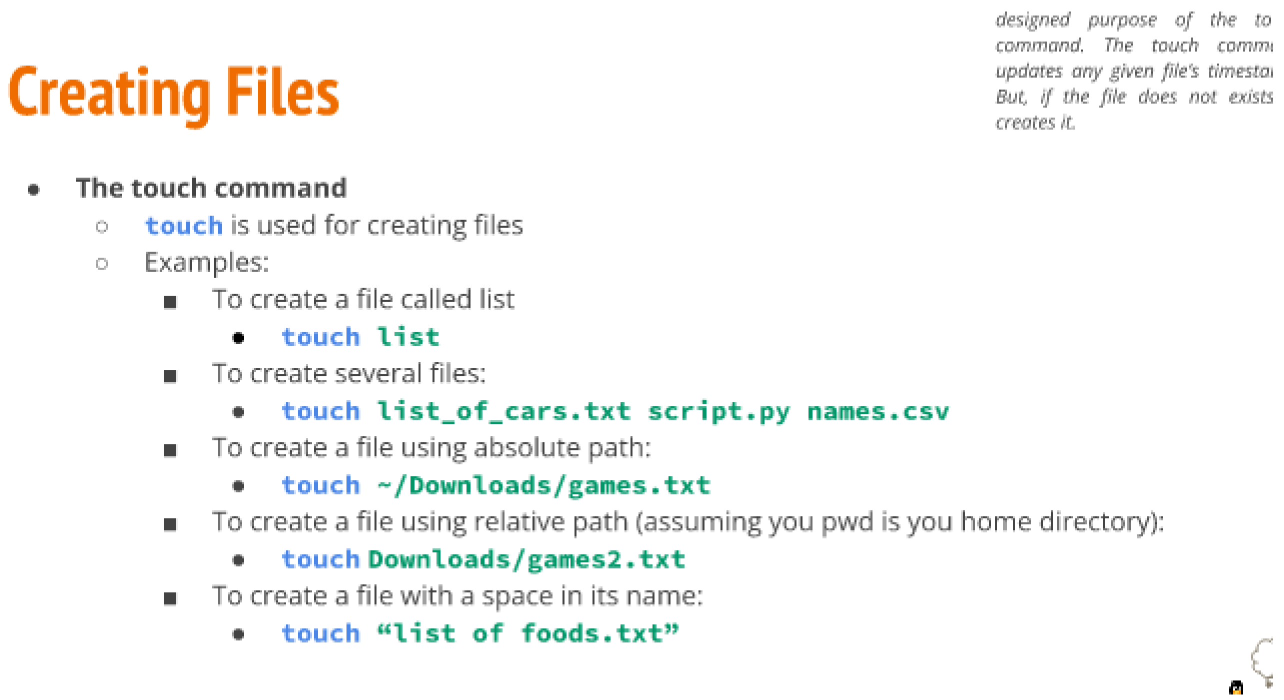
Examples



Examples of the `mkdir` command

- Create a directory in the present working directory
  - `mkdir wallpapers`
- Create a directory in a different directory using relative path
  - `mkdir wallpapers/ocean`
- Create a directory in a different directory using absolute path
  - `mkdir ~/wallpapers/forest`
- Create a directory with a space in the name
  - `mkdir wallpapers/new\ cars`
  - `mkdir wallpapers/’cities usa’`
- Create a directory with a single quote in the name
  - `mkdir wallpapers/”majora's mask”`
- Create multiple directories
  - `mkdir wallpapers/cars wallpapers/cities wallpapers/forest`
- Create a directory with a parent directory at the same time.
  - `mkdir -p wallpapers_others/movies`

Creating files

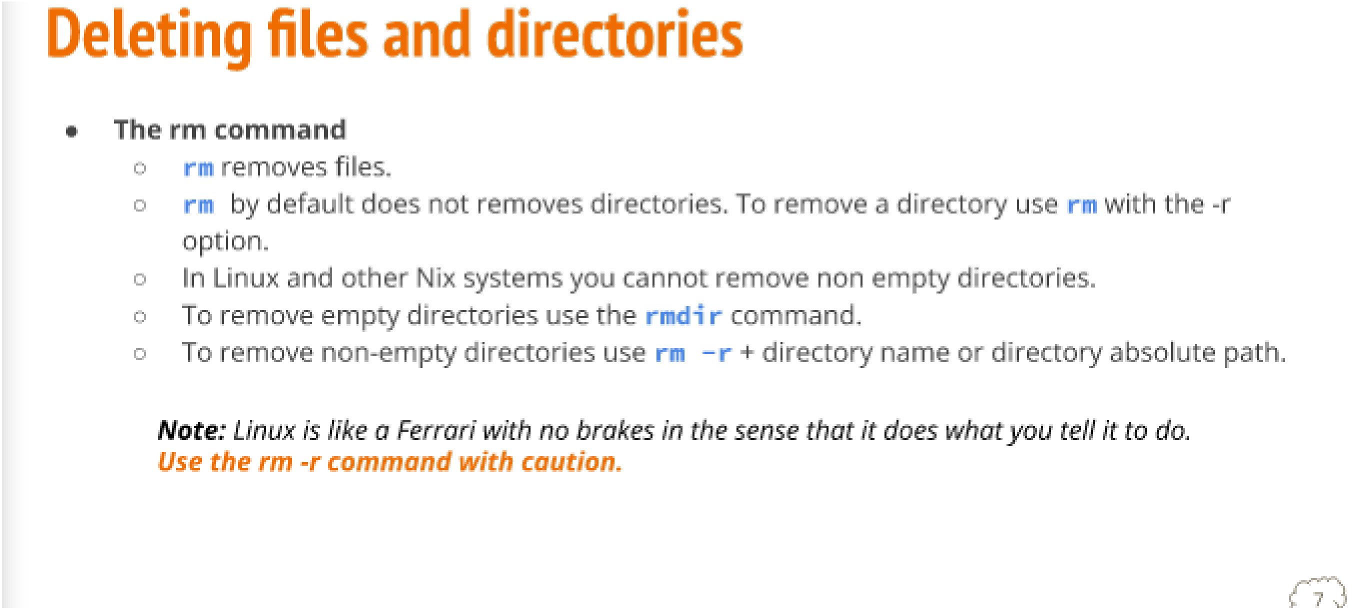


designed purpose of the to command. The touch comm updates any given file's timesta But, if the file does not exists creates it.

Creating Files

- **The touch command**
  - `touch` is used for creating files
  - Examples:
    - To create a file called list
      - `touch list`
    - To create several files:
      - `touch list_of_cars.txt script.py names.csv`
    - To create a file using absolute path:
      - `touch ~/Downloads/games.txt`
    - To create a file using relative path (assuming you pwd is you home directory):
      - `touch Downloads/games2.txt`
    - To create a file with a space in its name:
      - `touch “list of foods.txt”`

Deleting Files and directories

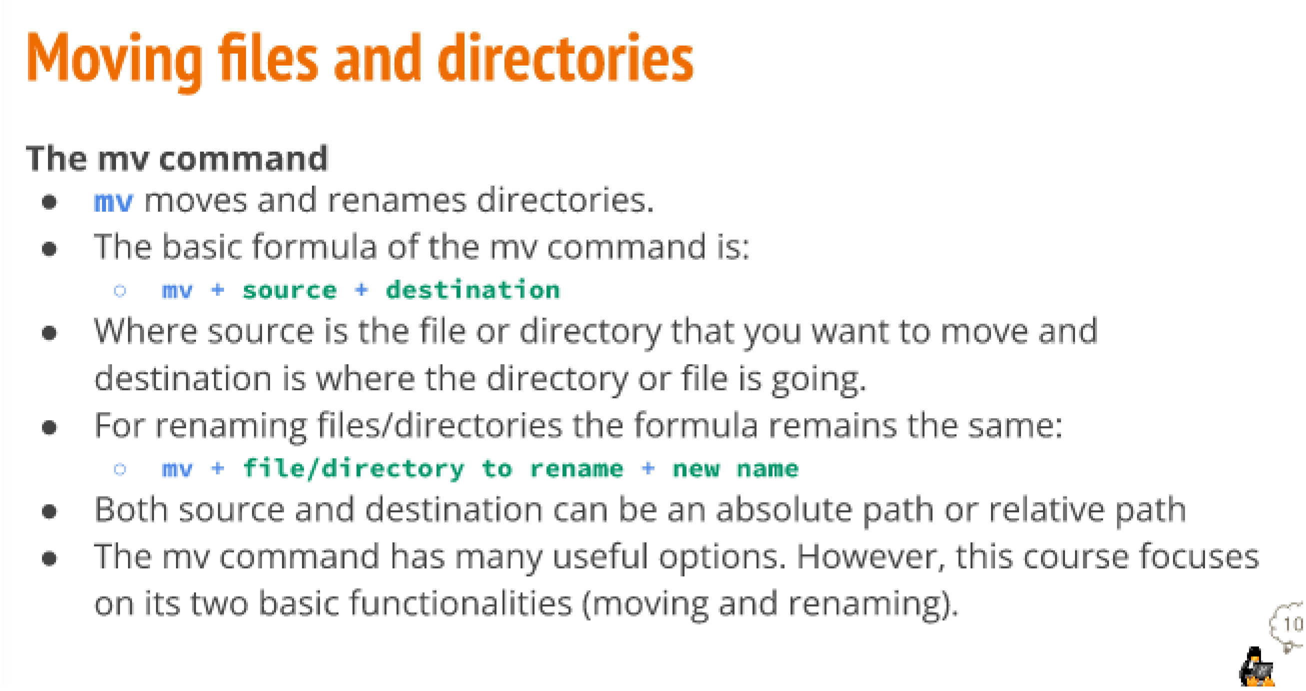


Deleting files and directories

- **The rm command**
  - `rm` removes files.
  - `rm` by default does not removes directories. To remove a directory use `rm` with the -r option.
  - In Linux and other Nix systems you cannot remove non empty directories.
  - To remove empty directories use the `rmdir` command.
  - To remove non-empty directories use `rm -r` + directory name or directory absolute path.

Note: *Linux is like a Ferrari with no brakes in the sense that it does what you tell it to do.*
Use the rm -r command with caution.

Moving Files and directories



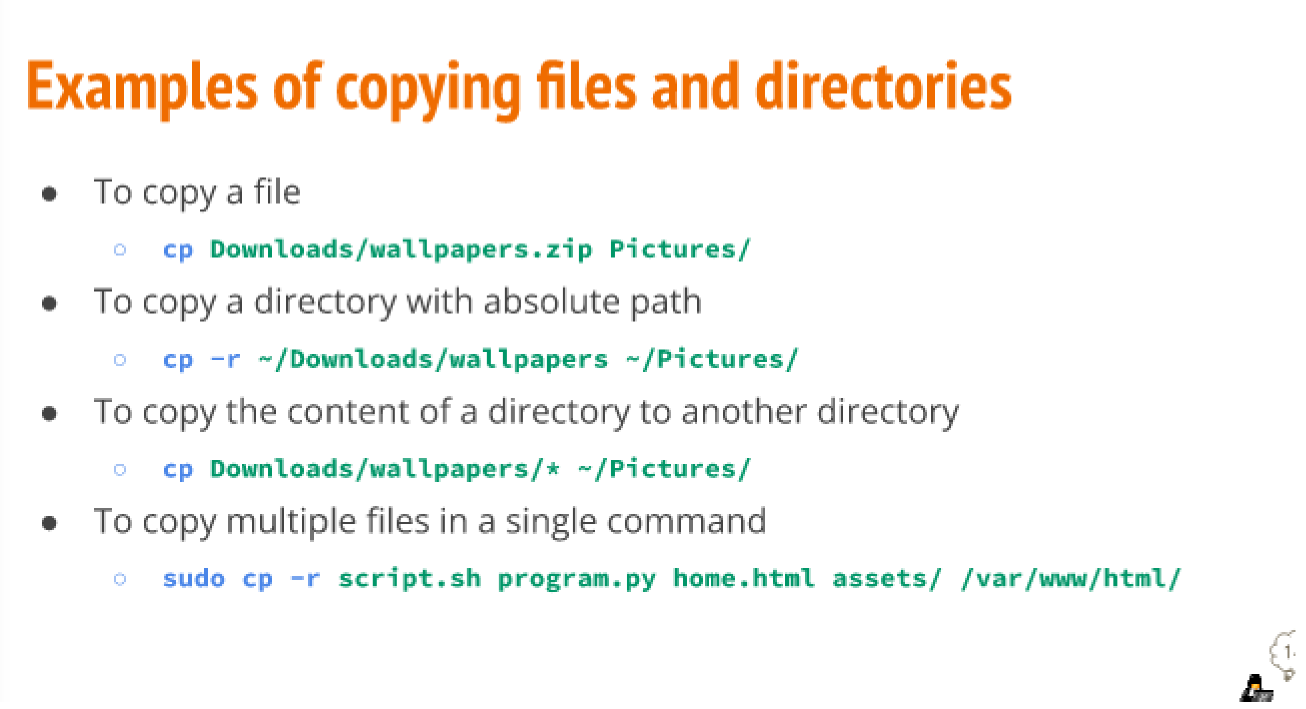
Moving files and directories

The mv command

- `mv` moves and renames directories.
- The basic formula of the mv command is:
  - `mv + source + destination`
- Where source is the file or directory that you want to move and destination is where the directory or file is going.
- For renaming files/directories the formula remains the same:
  - `mv + file/directory to rename + new name`
- Both source and destination can be an absolute path or relative path
- The mv command has many useful options. However, this course focuses on its two basic functionalities (moving and renaming).

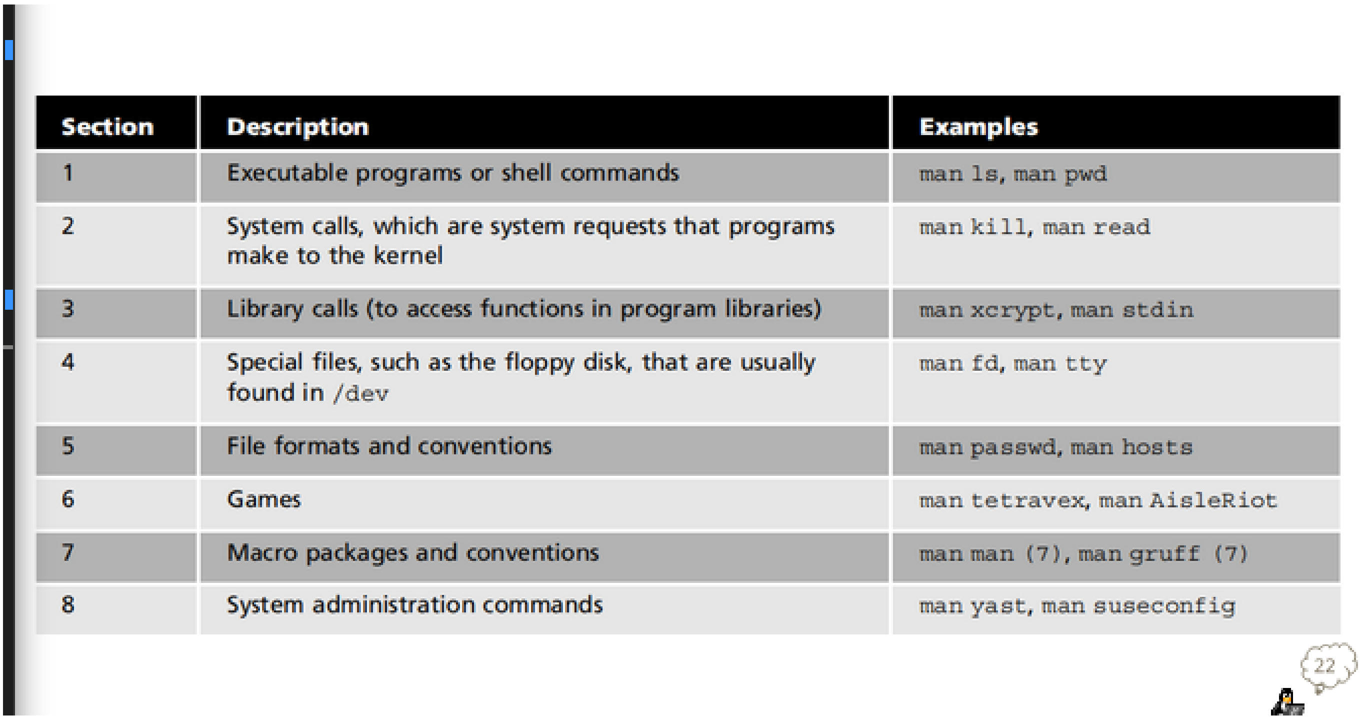
10

Copying files and dirextories Examples



Examples of copying files and directories

- To copy a file
  - `cp Downloads/wallpapers.zip Pictures/`
- To copy a directory with absolute path
  - `cp -r ~/Downloads/wallpapers ~/Pictures/`
- To copy the content of a directory to another directory
  - `cp Downloads/wallpapers/* ~/Pictures/`
- To copy multiple files in a single command
  - `sudo cp -r script.sh program.py home.html assets/ /var/www/html/`



Section	Description	Examples
1	Executable programs or shell commands	man ls, man pwd
2	System calls, which are system requests that programs make to the kernel	man kill, man read
3	Library calls (to access functions in program libraries)	man xcrypt, man stdin
4	Special files, such as the floppy disk, that are usually found in /dev	man fd, man tty
5	File formats and conventions	man passwd, man hosts
6	Games	man tetravex, man AisleRiot
7	Macro packages and conventions	man man (7), man gruff (7)
8	System administration commands	man yast, man suseconfig

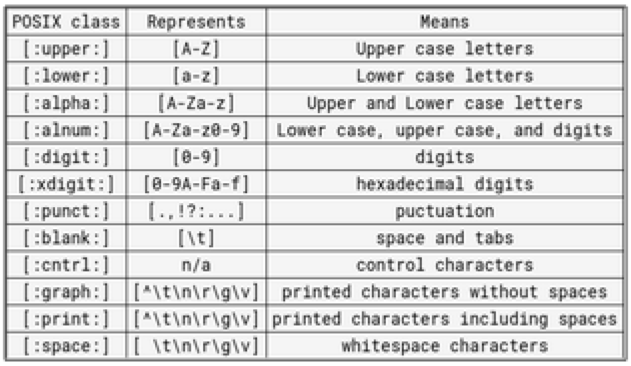
Wildcard

The * Wildcard

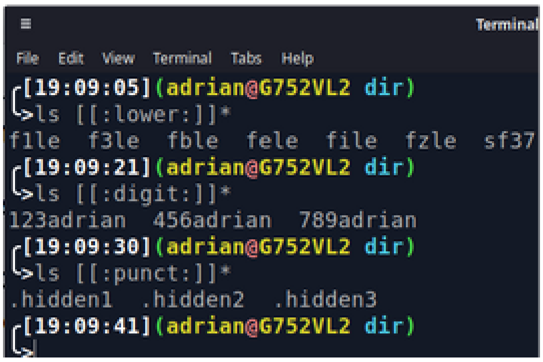
- The main wildcard is a star, or asterisk (*) character.
- A star alone matches anything and nothing and matches any number of characters.
- For example, `ls *.txt` will match all files that end in .txt regardless of the size of the file name.
- Examples of when to use the * wildcard:
  - When you want to list all 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 with Regex Character Classes

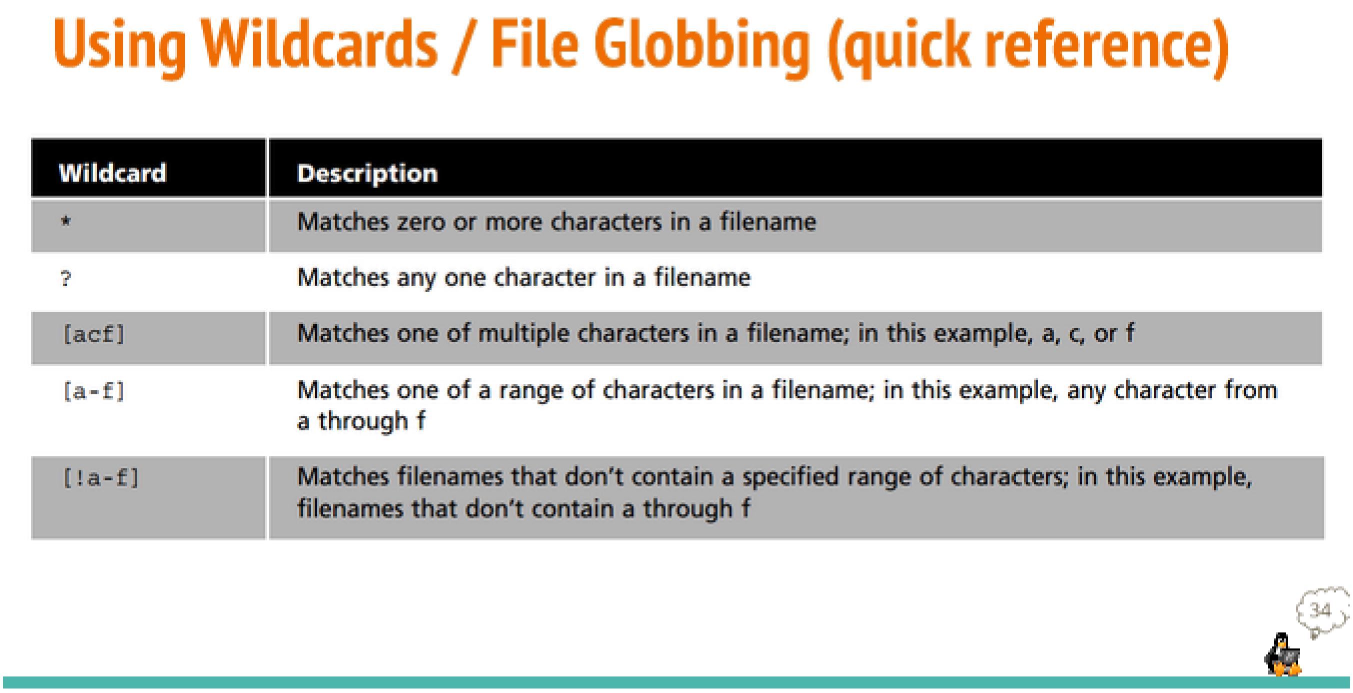
You can use POSIX or Character Classes with the [] wildcard



POSIX class	Represents	Means
[:upper:]	[A-Z]	Upper case letters
[:lower:]	[a-z]	Lower case letters
[:alpha:]	[A-Za-z]	Upper and Lower case letters
[:alnum:]	[A-Za-z0-9]	Lower case, upper case, and digits
[:digit:]	[0-9]	digits
[:xdigit:]	[0-9A-Fa-f]	hexadecimal digits
[:punct:]	[.,!?:...]	puctuation
[:blank:]	[\t]	space and tabs
[:cntrl:]	n/a	control characters
[:graph:]	[^\t\n\r\g\v]	printed characters without spaces
[:print:]	[^\t\n\r\g\v]	printed characters including spaces
[:space:]	[\t\n\r\g\v]	whitespace characters



```
┌[19:09:05](adrian@G752VL2 dir)
└>ls [[:lower:]]*
f1le  f3le  fble  fele  file  fzle  sf37
┌[19:09:21](adrian@G752VL2 dir)
└>ls [[:digit:]]*
123adrian  456adrian  789adrian
┌[19:09:30](adrian@G752VL2 dir)
└>ls [[:punct:]]*
.hidden1  .hidden2  .hidden3
┌[19:09:41](adrian@G752VL2 dir)
└>
```



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