

# 08. Deleting, Copying, Moving

## Commands table

### 01. What matters in this section

### 02. Deleting Files with rm

### 03. Deleting Folders with -d and -r

-d

-r

### 04. Moving Files with mv

### 05. Moving Folders with mv

### 06. Renaming

### 07. copying

## Commands table

command	application
<code>rm &lt;filename/s&gt;</code>	remove file/s passed as an argument
<code>rm -d / rmdir &lt;dirname&gt;</code>	remove empty directory
<code>rm -r &lt;dirname&gt;</code>	remove directories and their contents recursively
<code>mv &lt;filename&gt; &lt;name_to_change&gt;</code>	rename file with the name passed as an argument
<code>mv &lt;filename&gt; path/changed_file_name</code>	move and rename file with same command

## 01. What matters in this section

- all commands are useful in the section. They are:

1. `rm`

2. `rm -d & rm -r`

3. `mv`

4. `cp`

## 02. Deleting Files with rm

- we use `rm <filename>` to remove the files.
- with `rm` command, It doesn't go to the trash from where we can get those files back. Once a file is removed, it's removed.
- we can remove multiple files with single command of rm using multiple arguments.
  - example: `rm <file1> <file2> <file3>`

## 03. Deleting Folders with -d and -r

### -d

- we can use `rm -d` `rm --dir` `rmdir` commands to remove files and empty directory

### -r

- we can use `rm -r` or `rm -R` to remove directories and their contents recursively.
  - It keeps on traversing children's directories and deleting files and directories.
- It's very important to make sure we are sure to use the command while using. Data can't be recovered once it's gone.
- we can also use options like `-i` or `-I` called interactive which prompts before each time after removing 3 files which is also a bit

secure way if you are just learning

- can also be used with plain `rm` commands.

## 04. Moving Files with mv

- we use `mv <file_to_move> <destination_to_move>` to move files. The destination path can be both absolute and relative
- to move the file to directly home path, we can do one do it with Tilda. i.e `mv <filename> ~` will move file to home directory
- If we don't use any argument for `destination_to_move`
- we can move multiple files with mv commands by passing with a sequence of spaces. The last argument must be a path for the directory to move the file into
  - example: `mv file1 file2 file3 ~/pank/movehere` will move all files into `home/pank/movehere`

## 05. Moving Folders with mv

- while moving folders make sure that destination exists else it will move the folder and rename it in that location

## 06. Renaming

- We can rename only one source at one time using `mv <filename> <name_to_change>` in the same directory.
- We can rename both files and directories with it. While renaming files, make sure that the name is not already present. else, it will move into that directory
- We can move and rename files or folders with a single command using the same way

- example:

1. `mv <filename> ~/pank/movehere/changed_file_name` will move file to the directory and also rename it
2. `mv <foldername> ~/pank/movehere/changed_folder_name` will move folder to the directory and also rename it

## 07. copying

- we can copy file with `cp <filename> <path_to_copy>`
- same works with folders
- while copying non-empty folders we need to use recursive option
- example: `cp <dirname> <path_to_copy> -r`