Remove a File in Bash

Here, you will learn how to delete files in a shell.

rm

Definition:

`rm` is used to delete files and directories. By default, it is only set to remove files and not directories for safety. It basically unlinks the file name with the data stored in so that it cannot be accessed anymore. Before deleting any file, you must have permissions to delete it. You can delete multiple files at once. This command is almost similar to `rmdir`. The only difference is that `rmdir` is used to delete empty directories only. You can also delete symbolic links to a file, in this way, only link is removed but the file remains unaffected.

Syntax:

```
rm [options] [file_name(s)]
```

Options:

To indicate `rm` the end of options, `--` double dashes are used. This is very useful when a directory's name starts with `-`

Option	Meaning
-f	Means *force*. This option tells `rm` to force delete all the specified files without showing any message
-i	Means *interactive*. This option prompts the user for confirmation before deleting any file

-r or -R	Means *recursive*. It is used to recursively delete diectories by first emptying them and then removing them one by one

Example:

1. Delete multiple files at once

`rm file_1 file_2 file_3`

```
touch file_1 file_2 file_3 file_4 #Create 4 files
echo "Current Directory Files:"
ls

rm file_1 file_2 file_3
echo "Updated Directory Files:"
ls
```

2. Delete a directory named “my_dir”

`rm -r my_dir`

```
mkdir my_dir my_dir_2
echo "Current Directory Structure:"
ls

rm -r my_dir
echo "Updated Directory Structure:"
ls
```