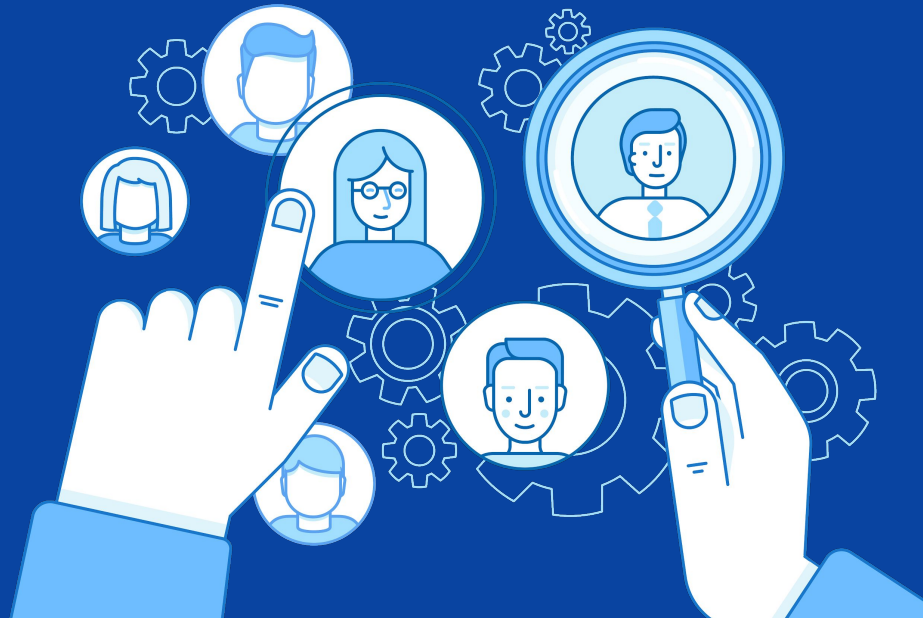# Digital Career Institute

Fundamentals - Creating & Manipulating & Installing

Goal of the Module

In this sub module, learners will be introduced to various methods to create or manipulate files through the terminal and basic markdown authoring.

- Use the terminal to copy, move and remove files and folders
- Understand the differences between using separate commands and using one command with flags
- Author content using basic common markdown syntax.

Topics

Use the Terminal for :

- creating directories & files
- file permissions
- Copying - deleting - moving - rename
- introduce package managers (brew)
- Introduce markdown

Digital Career Institute

DCI

Create Directories

```
$ mkdir OPTION <DIRECTORY>
```

Command description	Command and output

Example creation of a single directory using mkdir with the verbose -v option.	`$ mkdir -v films`
	`mkdir: created directory`
	`  \`films'`

Example creation of nested directory structure. The -p option will create any folders that don't already exist. This example combines mkdir with the -p option.	`$ mkdir -p /tv/shows/`

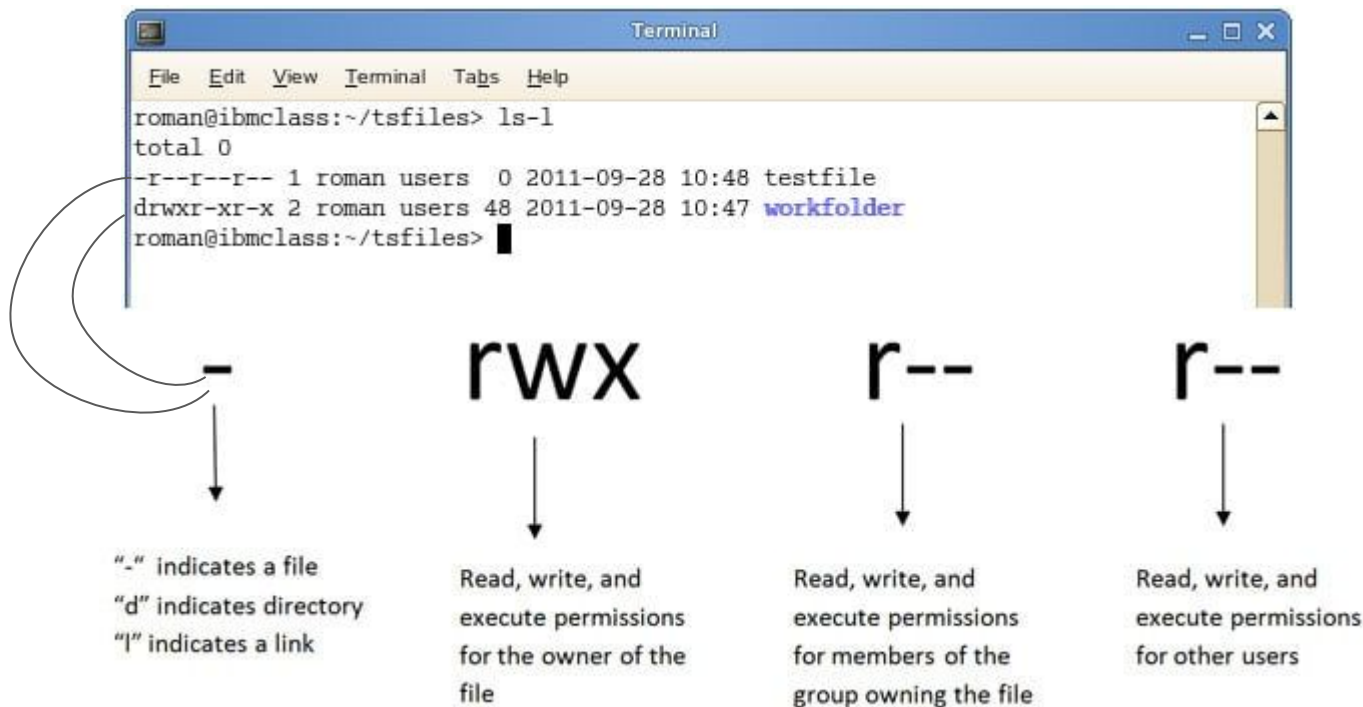| Option	Description
`-p, --parents`	If parent directories in the specified path do not exist, automatically generate them to accommodate (and do not generate an error).
`-v, --verbose`	Display the operation's execution step by step.
`-h, --help`	Display help information for this command.

Create files (touch)

```
touch OPTION <filename>
```

Command description	Command and output
Example creation a new file	`$ touch new-file`
Change only access time	`$ touch -a new-file`
Change only the modification time	`$ touch -m new-file`
Check file state (modification, creating time)	`$ stat new-file`

Create files (nano)

Command description	Command and output
Example creation a new file or edit the file if exist	`$ nano newfile`
To search for a text, type in the search term, and press Enter. The cursor will move to the first match. To move to the next match, press **Alt+w**.	Ctrl+w
Same the previous command but with replace	Ctrl+\
Start text selection then move the cursor to the end of the text that you want to select	Alt+a
End selecting	Ctrl+6
Copy selected text	Alt+6
Cut selected text	Ctrl+k
Past text	Ctrl+u
exit	Ctrl+x

File permissions

To show files permissions ls -l

rwx

r--

r--

"-" indicates a file
"d" indicates directory
"l" indicates a link

Read, write, and execute permissions for the owner of the file

Read, write, and execute permissions for members of the group owning the file

Read, write, and execute permissions for other users

File permissions

Command description	Command and output
To add a permission to group owners (w write, x execute, r read) and (g for group, o for other owner, u for users)	`$ chmod g+w filename`
To remove a permission to group owners	`$ chmod g-w filename`
give only read permission for everyone	`$ chmod a=r folder-name`
give read, write, and execute to everyone	`$ chmod ugo+rwx folder-name`

Copy

`$ cp [options] source dest`
Use this command to copy one or more files or directories.

Command description	Command and output
archive files	`$ cp -a`
force copy by removing the destination file if needed	`$ cp -f`
interactive - ask before overwrite	`$ cp -i`
link files instead of copy	`$ cp -l`
follow symbolic links	`$ cp -L`
no file overwrite	`$ cp -n`
recursive copy (including hidden files)	`$ cp -R`
update - copy when source is newer than dest	`$ cp -u`
verbose - print informative messages	`$ cp -v`

Copy

Examples

Command description	Command and output
Copy single file main.c to destination directory bak	`$ cp main.c bak`
Copy 2 files main.c and def.h to destination absolute path directory /home/usr/rapid/	`$ cp main.c def.h /home/usr/rapid/`
Copy all C files in current directory to subdirectory bak	`$ cp *.c bak`
Copy directory src to absolute path directory /home/usr/rapid/ :	`$ cp src /home/usr/rapid/`
Copy all files and directories in dev recursively to subdirectory bak	`$ cp -R dev bak`
Force file copy	`$ cp -f test.c bak`
Interactive prompt before file overwrite	`$ cp -i test.c bak` `cp: overwrite 'bak/test.c'? y`
Update all files in current directory - copy only newer files to destination directory bak	`$ cp -u * bak`

Remove

`$ rm   OPTION   file/directory`
Use this command to delete one or more files or directories.

Command description	Command and output
This is applicable if you have defined a list of separate <FILE> entries and one (or more) among them does not exist. The removal will continue without prompting you.	`$ rm -f`
Ask before removing files.	`$ rm -i`
This option performs a "quick-delete" to remove content recursively.	`$ rm -r`
Display the operation's execution step by step.	`$ rm -v`

Move

```
$ mv [options] source dest
```
Use this command to rename a single file, or move a single/multiple files.

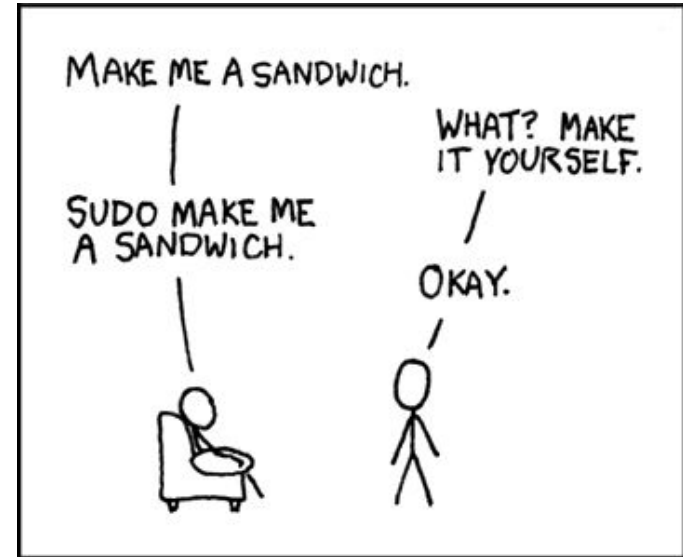| Command description	Command and output
Ask before overwriting existing files.	`$ mv -i`
Display the operation's execution step by step.	`$ mv -v`

Use mv to Renaming a file

In this example, the file is renamed and kept in the same directory:

```
$ mv -i /123456/test/movie.mp4   /123456/test/movie_old.mp4
```

What is `sudo`?

Have you ever tried to run a command in terminal only to be given "Access Denied?

- **`Sudo`**, the one command to rule them all. It stands for "super user do!"
- It is much better than logging in as root, or using the su "switch user" command.
- If you prefix "sudo" with any command, it will run that command with elevated privileges. Elevated privileges are required to perform certain administrative tasks

Introduce package managers (`brew`)

What Does Homebrew Do?

- Homebrew installs the software you need.
- Homebrew installs packages to their own directory and then symlinks their files into /usr/local.
- Homebrew won't install files outside its prefix and you can place a Homebrew installation wherever you like.

Example:
Install firefox using brew:

```
$ brew install --cask firefox
```

Introduce package managers (brew)

- Install brew
  ```
  $ /bin/bash -c "$(curl -fsSL
  https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)
  ```

- First update the formulae and Homebrew itself:
  ```
  $ brew update
  ```

- You can now find out what is outdated with:
  ```
  $ brew outdated
  ```

- Upgrade everything with:
  ```
  $ brew upgrade
  ```

- Upgrade only firefox using brew:
  ```
  $ brew upgrade firefox
  ```

Introduce markdown

Markdown is a lightweight markup language for creating formatted text using a plain-text editor

Headings

Markdown	Rendered Output
`# Heading level 1`	# Heading level 1
`## Heading level 2`	## Heading level 2
`### Heading level 3`	### Heading level 3
`#### Heading level 4`	#### Heading level 4
`##### Heading level 5`	##### Heading level 5
`###### Heading level 6`	###### Heading level 6

Bold

Markdown	Rendered Output
`I just love **bold text**.`	I just love **bold text**.
`I just love __bold text__.`	I just love **bold text**.
`Love**is**bold`	Love**is**bold

Introduce markdown

Italic

Markdown	Rendered Output
`Italicized text is the *cat's meow*.`	Italicized text is the *cat's meow*.
`Italicized text is the _cat's meow_.`	Italicized text is the *cat's meow*.
`A*cat*meow`	A*cat*meow

Bold and Italic

Markdown	Rendered Output
`This text is ***really important***.`	This text is ***really important***.
`This text is ___really important___.`	This text is ***really important***.
`This text is __*really important*__.`	This text is ***really important***.
`This text is **_really important_**.`	This text is ***really important***.
`This is really***very***important text.`	This is really ***very*** important text.

Blockquotes

To create a blockquote, add a > in front of a paragraph.

```
> Dorothy followed her through many of the beautiful rooms in her castle.
Dorothy followed her through many of the beautiful rooms in her castle.
```

> Dorothy followed her through many of the beautiful rooms in her castle.

Blockquotes with Multiple Paragraphs

```
> Dorothy followed her through many of the beautiful rooms in her castle.
>
> The Witch bade her clean the pots and kettles and sweep the floor and keep the fire fed with wood.
```

> Dorothy followed her through many of the beautiful rooms in her castle.
>
> The Witch bade her clean the pots and kettles and sweep the floor and keep the fire fed with wood.

Introduce markdown

Ordered Lists

Markdown	Rendered Output

`1. First item`
`2. Second item`
`3. Third item`
`4. Fourth item`
	1. First item
	2. Second item
	3. Third item
	4. Fourth item

`1. First item`
`2. Second item`
`3. Third item`
`    1. Indented item`
`    2. Indented item`
`4. Fourth item`
	1. First item
	2. Second item
	3. Third item
	    1. Indented item
	    2. Indented item
	4. Fourth item

Unordered Lists

Markdown	Rendered Output
`- First item` `- Second item`	• First item • Second item
`* First item` `* Second item`	• First item • Second item
`+ First item` `+ Second item`	• First item • Second item

`- First item`
`- Second item`
`- Third item`
`    - Indented item`
`    - Indented item`
	• First item
	• Second item
	• Third item
	    ○ Indented item
	    ○ Indented item

Introduce markdown

Code Blocks

Code blocks are normally indented four spaces or one tab. When they're in a list, indent them eight spaces or two tabs.

Images

```
![alternative text if image link is broken](image link)
```

Links

```
[link text](link)
```

```
[DCI](https://digitalcareerinstitute.org/)
```

output

[DCI](#)

Self Study

THANK
YOU

Contact Details
DCI Digital Career Institute gGmbH