

# View, Navigate, Create, Change

Using the terminal

1<sup>st</sup> week

2<sup>nd</sup> week

Overview

Versioning  
(Branches)

View, Navigate,  
Create, Change

Publishing

Installation

Collaboration

Versioning  
(Basics)

Review

# Using the terminal Viewing & Navigating

# DCI The Terminal

Open the terminal – what do you see?

- You are always working in some specific directory
- You use it with text commands
- Try out the first command

```
$ ls
```

# DCI The Linux Filesystem

Files and folders (directories) exist in a hierarchy.

Each file and folder has a unique path, folders are separated by `/`.

`/tmp/notes.txt`

File **notes.txt** inside the **tmp** folder

`/home/dci/cv.pdf`

File **cv.pdf** inside **dci**, inside **home**

These are examples **absolute** (full) paths.

# DCI The Linux Filesystem

## Notable directories

<code>/</code>	the root folder
<code>/etc</code>	configuration files
<code>/var</code>	log files, other variable files
<code>/home</code>	home folders
<code>/home/dci</code>	home folder for user dci
<code>~</code>	shortcut to <i>your</i> home folder

# DCI The Linux Filesystem

## Notable directories

/	the root folder
/etc	configuration files
/var	log files, other variable files
/home	ome folders
/home/dci	home folder for user dci
~	shortcut to <i>your</i> home folder

## Try

```
$ ls /  
$ ls /var  
$ ls /etc  
$ ls /home
```

# DCI The Linux Filesystem

Commands can have *options* (sometimes called *flags* or *parameters*)

```
$ ls --help      # show help for ls
$ ls -l          # show long (detailed) listing
```

# DCI The Linux Filesystem

On Linux files starting with a “.” are considered hidden files

## Try

```
$ ls -l          # long listing
$ ls -a          # show hidden files
$ ls -la         # long listing, show hidden
$ ls -lah        # long listing, show hidden, human readable sizes
```



# DCI The Linux Filesystem

**Relative paths** are paths relative to the **current working directory**.

Check the current working directory:

```
$ pwd
```

Use “.” to refer to the current directory and “..” for the parent directory.

```
$ ls .           # contents of current folder
$ ls ..          # contents of parent folder
$ ls Documents   # contents of "Documents" folder
$ ls ./Documents # . means current folder
$ ls ../Documents # .. means parent folder
```

# DCI The Linux Filesystem

Change the current working directory with **cd**

```
$ cd ..           # go to parent folder
$ cd ~           # go to home folder
$ cd             # go to home folder
$ cd Documents   # go to "Documents" folder
$ cd ~/Downloads # go to the "Downloads" folder in your home
```

What next?

```
$ less file.txt # open a text file for viewing (q to quit)
```

Awesome terminal feature: **tab autocomplete**

# At the core of the lesson

- Issue commands in the terminal
- Your home folder is ~
- Absolute and relative paths

## Useful commands

```
$ ls          # list
$ cd          # change directory
$ pwd        # print working directory
$ less       # view text file
```

## Command options like -help

```
$ ls --help    # show help for ls
$ cd Downloads # change directory
```

# Using the terminal

## Creating & Manipulating

# DCI Working with files

We will have **lots** of files and folders!

Be organized and systematic from the start

```
$ mkdir projects          # make projects directory
$ cd projects            #
$ mkdir test-project      # make test-project directory
$ cd test-project        #
$ pwd                    # print working directory
$ touch plan.txt         # create empty file plan.txt
$ ls                     #
```

# DCI Working with files

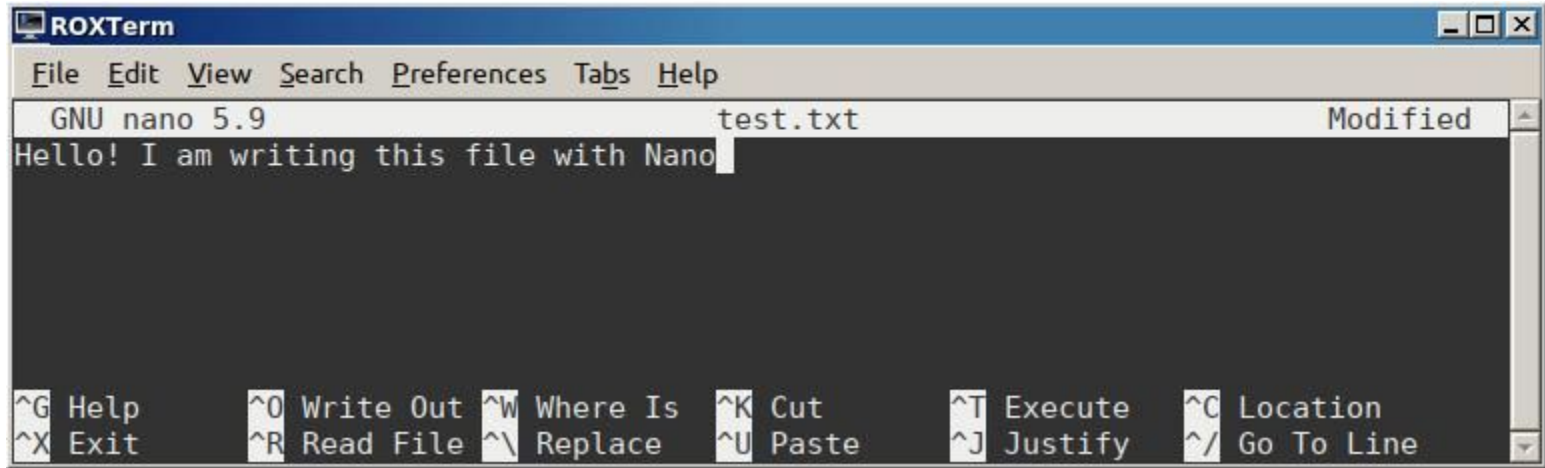
## Practice commands with me!

```
$ ls -la           # long listing, show hidden
$ ls --help        # help for ls
$ mkdir --help     # help for mkdir
$ mkdir projects   # make projects folder
$ cd projects      # change to projects
$ pwd              # print working directory
$ less file.txt    # view text file
$ touch plan.txt   # create empty file plan.txt
$ man ls           # show manual for ls (q to quit!)
```

So many... let's find a *cheat sheet*!

# DCI Working with files

```
$ nano test.txt      # introducing the nano text editor
```



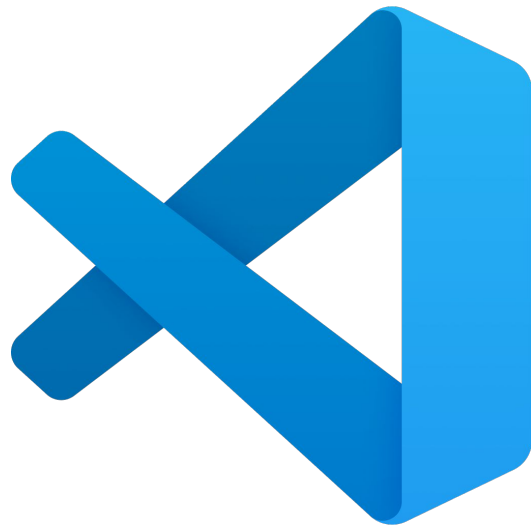
**^X** means press Ctrl+x

**M-C** means press Alt-c

# DCI Working with files

## VS Code

- Lightweight but powerful IDE
  - Integrated Development Environment
- Super popular
- Supports many languages
- Extensible
- Has a terminal *built in* ❤️

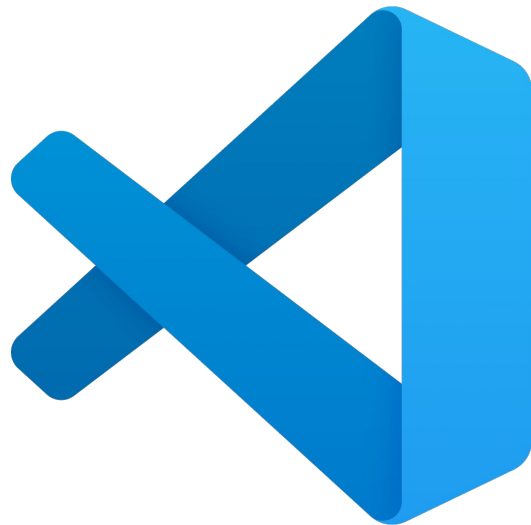




# DCI Working with files

## VS Code

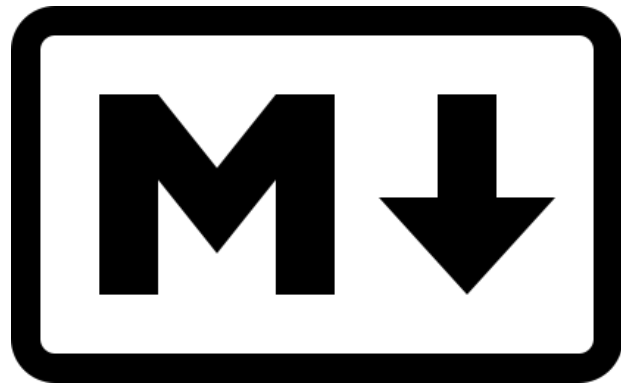
- One of our most important tools
- Learn it well
- Train usage in your free time
- We will practice with **VS Code** + **Markdown**



# DCI Working with files

**Markdown** files have the **.md** extension

- Simple markup language
  - Not a programming language
  - Clear *syntax*
- Understandable as plain text
- Can be *rendered*



[Markdown in action](#)

(main documentation file for VS Code)

## DCI Working with files

```
$ cd ~/projects  
$ mkdir markdown-test  
$ code markdown-test
```



# DCI Working with files

```
# Main heading
## Second level heading
### Third level heading

Normal, bold, italic, `highlighted` text!

1. List item
2. List item

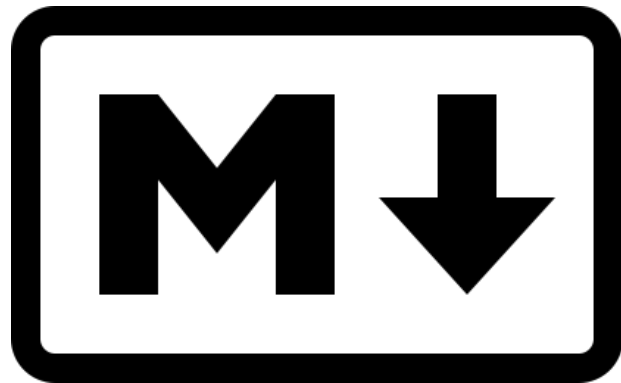
- List item
- List item

> This here is a quote

[A link](https://www.google.com)

![An image](https://placekitten.com/100 "Cat")

```html
<main>
  <p>HTML inside Markdown!</p>
</main>
```
```



So many... let's find a *cheat sheet*!

# DCI Working with files

## Deleting

```
$ rm test.txt           # delete file
$ rm *.pdf              # delete all .pdf files
$ rm -r old-project     # delete folder
$ rm -rf old-project    # force delete folder (careful!)
```

The \* (asterisk, wildcard) matches multiple files!

# DCI Working with files

## Moving

```
$ mv test.txt newname.txt
```

```
# move file (rename)
```

```
$ mv my-project ~/backup
```

```
# move folder to ~/backup
```

# DCI Working with files

## Copying

|                          |                              |
|--------------------------|------------------------------|
| \$ cp test.txt test2.txt | # copy file                  |
| \$ cp -r project backup  | # copy folder                |
| \$ cp --help             | # show help for cp           |
| \$ cp * ~/backup         | # copy all files to ~/backup |
| \$ cp -r * ~/backup      | # copy all files & folders   |

# At the core of the lesson

- Visual Studio Code is our main editor
- Markdown is a markup language
- The terminal can do a lot
  - Edit text
  - Create and manage files
  - Create and manage folders
  - Start programs



# Installation

file system, package manager, version control

1<sup>st</sup> week

2<sup>nd</sup> week

Overview

Versioning  
(Branches)

View, Navigate,  
Create, Change

Publishing

Installation

Collaboration

Versioning  
(Basics)

Review