

Biomedical Wearable Technologies  
for Healthcare and Wellbeing

# Terminal primer

---

A.Y. 2022-2023

Giacomo Cappon



# Intro

---

- **Terminal** is the software that allows to send UNIX Shell commands to the operating system through the command line.
- This very brief primer lists some useful UNIX Shell commands that you will need to use during this course in order to work with GIT.
- UNIX (Mac and Linux) users have **Terminal** already installed.
- Windows users must install and use **git-bash** instead of Terminal since the latter is not available for Windows (to install it, follow step 4b in the “**Setup the development environment**” guide you can find in the moodle page of the course).
- In the following, and during the course, I will always refer to both Terminal and git-bash as “Terminal”

# UNIX Shell - Directories

---

- When you open the Terminal, it should start in your home directory. This is your individual space on the system for your files. You can find out the name of your current working directory using the `pwd` (parent working directory) command:

`pwd`

```
cappe@MacBook-Pro-di-Giacomo ~ % pwd
/Users/cappe
cappe@MacBook-Pro-di-Giacomo ~ %
```

- No matter where in the directory structure you are, you can always get back to your home directory by using the `cd` (change directory) command without any arguments:

`cd`

```
cappe@MacBook-Pro-di-Giacomo arrow % pwd
/Users/cappe/Repos/arrow
cappe@MacBook-Pro-di-Giacomo arrow % cd
cappe@MacBook-Pro-di-Giacomo ~ % pwd
/Users/cappe
cappe@MacBook-Pro-di-Giacomo ~ %
```

# UNIX Shell - Directories

---

- To create a folder you can use the `mkdir` (make directory) command followed by the name of the directory you want to create:

`mkdir primer`

```
cappe@MacBook-Pro-di-Giacomo ~ % mkdir primer  
cappe@MacBook-Pro-di-Giacomo ~ % █
```

- You can remove an empty subdirectory with the `rmdir` (remove directory) command (but don't do this right now):

`rmdir primer`

```
cappe@MacBook-Pro-di-Giacomo ~ % rmdir primer  
cappe@MacBook-Pro-di-Giacomo ~ % mkdir primer  
cappe@MacBook-Pro-di-Giacomo ~ % █
```

(Note: if you do remove "primer", please create it again since all examples will refer to it.)

# UNIX Shell - Directories

---

- To move into a directory you can use the `cd` command this time followed by the name of the directory you want to move into:

`cd primer`

```
cappe@MacBook-Pro-di-Giacomo ~ % pwd
/Users/cappe
cappe@MacBook-Pro-di-Giacomo ~ % cd primer
cappe@MacBook-Pro-di-Giacomo primer % pwd
/Users/cappe/primer
cappe@MacBook-Pro-di-Giacomo primer % █
```

# UNIX Shell - Directories

---

- To move into a directory you can use the `cd` command followed by the name of the directory you want to move into, e.g.:

`cd primer`

```
cappe@MacBook-Pro-di-Giacomo ~ % pwd
/Users/cappe
cappe@MacBook-Pro-di-Giacomo ~ % cd primer
cappe@MacBook-Pro-di-Giacomo primer % pwd
/Users/cappe/primer
cappe@MacBook-Pro-di-Giacomo primer % █
```

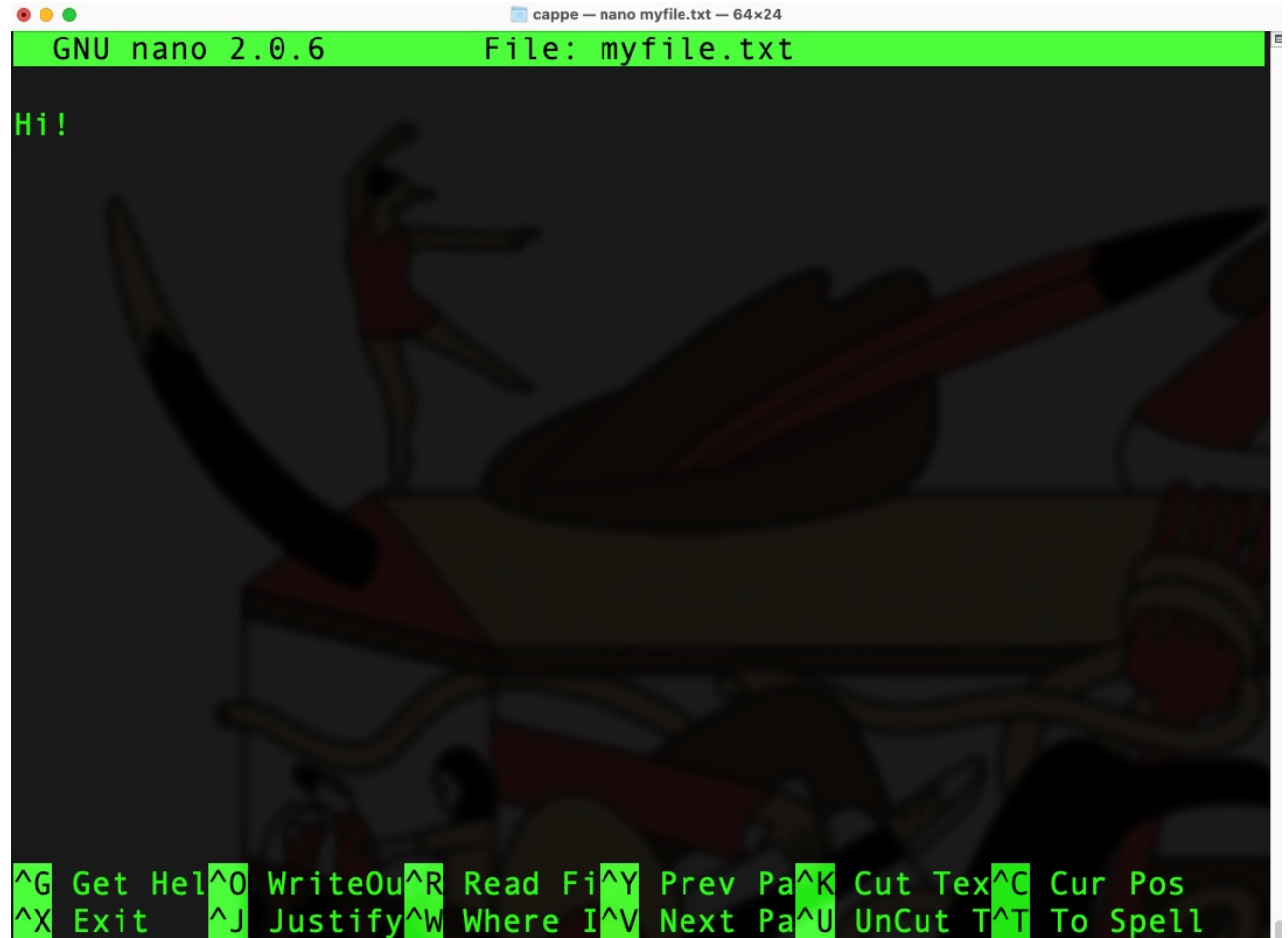
- To move back into the parent folder use “`..`” as directory name:

`cd ..`

```
cappe@MacBook-Pro-di-Giacomo primer % pwd
/Users/cappe/primer
cappe@MacBook-Pro-di-Giacomo primer % cd ..
cappe@MacBook-Pro-di-Giacomo ~ % pwd
/Users/cappe
cappe@MacBook-Pro-di-Giacomo ~ %
```

# UNIX Shell - Files

- To create a file (for example a .txt file) you can use the nano text editor, e.g.:  
`nano myfile.txt`
- A blank window will open where you can write and edit the myfile.txt file.
- To exit, use “CTRL+X” on your keyboard. You will be asked to save the changes you made to the file:
  - Use “Y” on your keyboard to save and exit
  - Use “N” on your keyboard to exit without saving
- If you choose “Y”, the editor will ask you to choose a file name. Choose it and press enter to exit.

A screenshot of the GNU nano 2.0.6 text editor. The window title bar shows "cappe - nano myfile.txt - 64x24". The editor's status bar at the top is green and displays "GNU nano 2.0.6" and "File: myfile.txt". The main editing area has a dark background with a faint, stylized illustration of a person writing on a large scroll. The text "Hi!" is written in green on the first line. At the bottom, a green status bar lists various keyboard shortcuts: ^G Get Hel, ^O WriteOu, ^R Read Fi, ^Y Prev Pa, ^K Cut Tex, ^C Cur Pos, ^X Exit, ^J Justify, ^W Where I, ^V Next Pa, ^U UnCut T, and ^T To Spell.

# UNIX Shell - Files

---

- To list the content of a directory, use the `ls` (list) command:

`ls`

- If you want to also see all the details of each file and also the hidden folders, use the `-la` option:

`ls -la`

```
cappe@MacBook-Pro-di-Giacomo primer % ls
myfile.txt
cappe@MacBook-Pro-di-Giacomo primer % ls -la
total 8
drwxr-xr-x  3 cappe  staff   96 Mar  2 09:21 .
drwxr-xr-x+ 63 cappe  staff 2016 Mar  2 09:21 ..
-rw-r--r--  1 cappe  staff   4 Mar  2 09:18 myfile.txt
cappe@MacBook-Pro-di-Giacomo primer %
```

- Note: I won't cover what are those details since this is not necessary for the course. If you want to know more about it, you can find some useful resources at the end of the document.

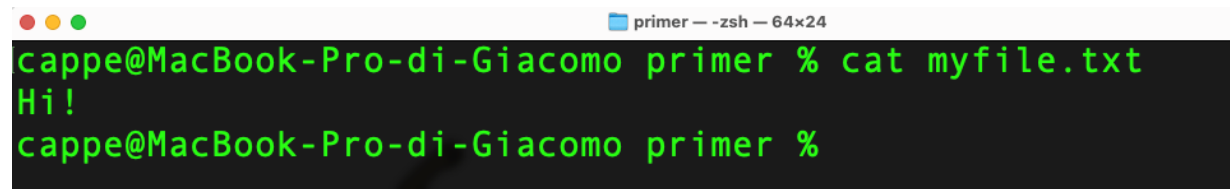


# UNIX Shell - Files

---

- To show the content of a file use the `cat` command followed by the name of the file you want to show, e.g.:

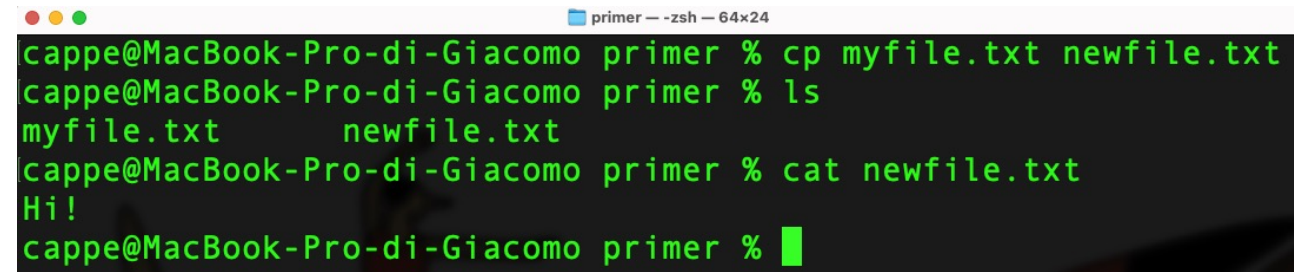
```
cat myfile.txt
```

A terminal window titled 'primer --zsh-- 64x24' showing the command 'cat myfile.txt' being executed. The output is 'Hi!'. The prompt is 'cappe@MacBook-Pro-di-Giacomo primer %'.

```
cappe@MacBook-Pro-di-Giacomo primer % cat myfile.txt
Hi!
cappe@MacBook-Pro-di-Giacomo primer %
```

- To copy a file into another use the `cp` (copy) command followed by the name of the source file and the name of the destination file, e.g.:

```
cp myfile.txt newfile.txt
```

A terminal window titled 'primer --zsh-- 64x24' showing the command 'cp myfile.txt newfile.txt' being executed. Then 'ls' is run, showing 'myfile.txt' and 'newfile.txt'. Finally 'cat newfile.txt' is run, showing 'Hi!'. The prompt is 'cappe@MacBook-Pro-di-Giacomo primer %'.

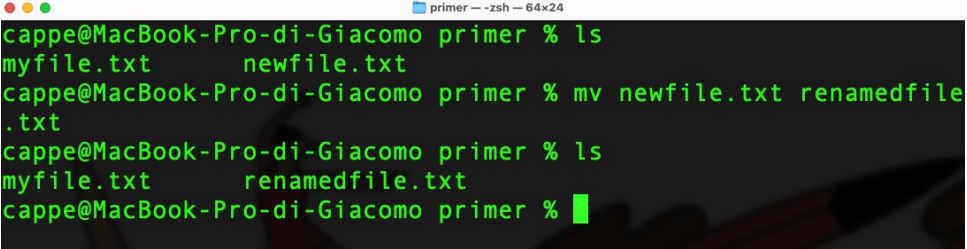
```
cappe@MacBook-Pro-di-Giacomo primer % cp myfile.txt newfile.txt
cappe@MacBook-Pro-di-Giacomo primer % ls
myfile.txt      newfile.txt
cappe@MacBook-Pro-di-Giacomo primer % cat newfile.txt
Hi!
cappe@MacBook-Pro-di-Giacomo primer %
```

# UNIX Shell - Files

---

- To rename a file use the `mv` (move) command followed by the name of the file you want to rename and the new file name, e.g.:

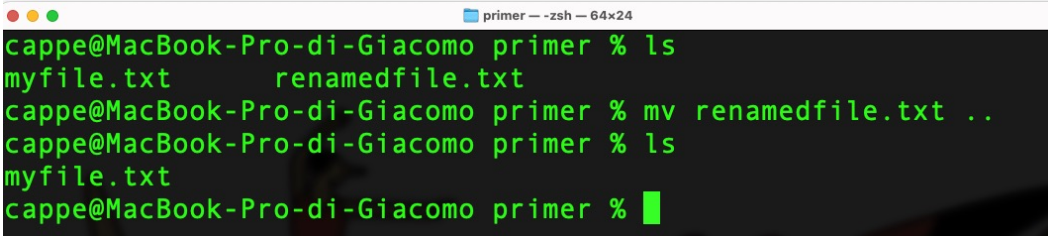
```
mv newfile.txt renamedfile.txt
```

A terminal window titled 'primer -- zsh -- 64x24' showing the process of renaming a file. The user 'cappe' is at 'MacBook-Pro-di-Giacomo' in the 'primer' directory. The first command is 'ls', showing 'myfile.txt' and 'newfile.txt'. The second command is 'mv newfile.txt renamedfile.txt'. The third command is 'ls', showing 'myfile.txt' and 'renamedfile.txt'.

```
cappe@MacBook-Pro-di-Giacomo primer % ls
myfile.txt      newfile.txt
cappe@MacBook-Pro-di-Giacomo primer % mv newfile.txt renamedfile
.txt
cappe@MacBook-Pro-di-Giacomo primer % ls
myfile.txt      renamedfile.txt
cappe@MacBook-Pro-di-Giacomo primer %
```

- To move a file into another directory, use the `mv` command followed by the name of the file you want to move and the name of the destination directory, e.g. (this moves the file into the parent directory):

```
mv renamedfile.txt ..
```

A terminal window titled 'primer -- zsh -- 64x24' showing the process of moving a file to its parent directory. The user 'cappe' is at 'MacBook-Pro-di-Giacomo' in the 'primer' directory. The first command is 'ls', showing 'myfile.txt' and 'renamedfile.txt'. The second command is 'mv renamedfile.txt ..'. The third command is 'ls', showing only 'myfile.txt'.

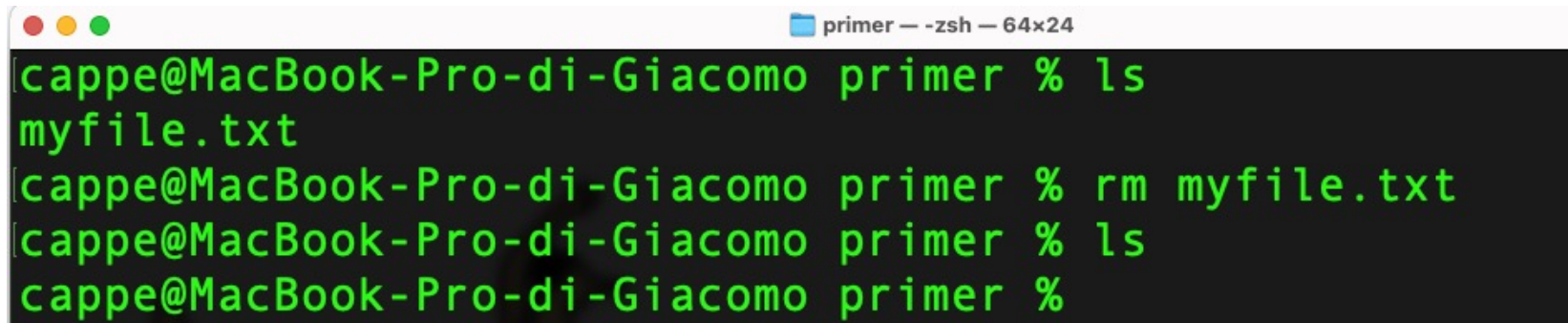
```
cappe@MacBook-Pro-di-Giacomo primer % ls
myfile.txt      renamedfile.txt
cappe@MacBook-Pro-di-Giacomo primer % mv renamedfile.txt ..
cappe@MacBook-Pro-di-Giacomo primer % ls
myfile.txt
cappe@MacBook-Pro-di-Giacomo primer %
```

# UNIX Shell - Files

---

- To remove a file use the `rm` (remove) command followed by the name of the file you want to remove, e.g.:

```
rm myfile.txt
```

A terminal window titled "primer — zsh — 64x24" with a dark background and green text. It shows a sequence of commands and their outputs. The user 'cappe' is at the 'MacBook-Pro-di-Giacomo' machine, in the 'primer' directory. The commands and outputs are: 1. 'ls' followed by 'myfile.txt'. 2. 'rm myfile.txt'. 3. 'ls' followed by a blank line. 4. A new prompt line.

```
cappe@MacBook-Pro-di-Giacomo primer % ls  
myfile.txt  
cappe@MacBook-Pro-di-Giacomo primer % rm myfile.txt  
cappe@MacBook-Pro-di-Giacomo primer % ls  
cappe@MacBook-Pro-di-Giacomo primer %
```

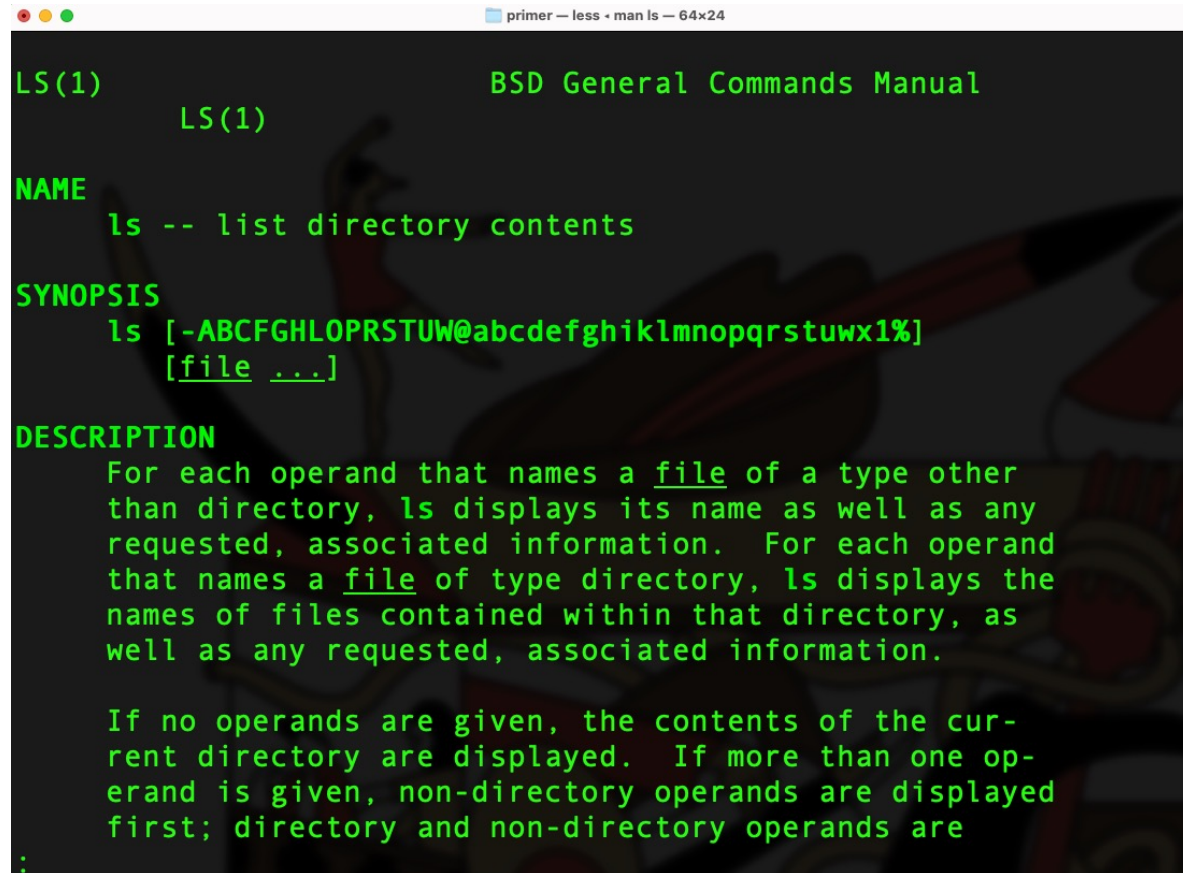
# UNIX Shell - Help

---

- To show the online help for a specific command use the man (manual) command, e.g. (this will show the online help for the ls command):

```
man ls
```

- To exit press “q” on the keyboard



```
LS(1) BSD General Commands Manual
LS(1)
NAME
ls -- list directory contents
SYNOPSIS
ls [-ABCFGHLOPRSTUW@abcdefghijklmnopqrstuvwxyz1%]
[file ...]
DESCRIPTION
For each operand that names a file of a type other
than directory, ls displays its name as well as any
requested, associated information. For each operand
that names a file of type directory, ls displays the
names of files contained within that directory, as
well as any requested, associated information.

If no operands are given, the contents of the cur-
rent directory are displayed. If more than one op-
erand is given, non-directory operands are displayed
first; directory and non-directory operands are
:
```

# UNIX Shell - Summary

---

`nano myfile.txt`

*text edit file "myfile.txt"*

`ls`

*list files in current directory*

`ls -la`

*long format listing with also hidden things*

`cat myfile.txt`

*view contents of text file "myfile.txt"*

`more myfile.txt`

*paged viewing of text file "myfile.txt"*

`less myfile.txt`

*scroll through text file "myfile"*

`cp srcfile destfile`

*copy file "srcfile" to new file "destfile"*

`mv oldname newname`

*rename (or move) file "oldname" to "newname"*

`rm myfile.txt`

*remove file "myfile.txt"*

`mkdir dirname`

*make new directory called "dirname"*

`cd dirname`

*move into "dirname"*

`rmdir dirname`

*remove (empty) directory "dirname"*

`pwd`

*display current working directory*

`man command`

*display man page for "command"*

# Useful resources

---

- UNIX Shell Crash Course by swcarpentry
  - [https://www.youtube.com/watch?v=8c1BL5b47kg&ab\\_channel=Geek%27sLesson](https://www.youtube.com/watch?v=8c1BL5b47kg&ab_channel=Geek%27sLesson)
- UNIX Shell docs by swcarpentry
  - <https://swcarpentry.github.io/shell-novice/>