Command line basics

- What the hell is "the shell"?
- Why should I use one?
- Getting started!
- References

```
#knitr::opts_chunk$set(engine.opts = list(bash = "-l"))
```

Hi everyone!

This will be a very soft introduction to the command-line.

~~I am assuming you are using either a Mac or a Linux machine.~~ I assume nothing! We are using a web-based command line!

What the hell is "the shell"?

- From: Software Carpentry Foundation*

Humans and computers commonly interact in many different ways, such as through a keyboard and mouse, touch screen interfaces, or using speech recognition systems. The most widely used way to interact with personal computers is called a graphical user interface (GUI). With a GUI, we give instructions by clicking a mouse and using menu-driven interactions.

While the visual aid of a GUI makes it intuitive to learn, this way of delivering instructions to a computer scales very poorly. Imagine the following task: for a literature search, you have to copy the third line of one thousand text files in one thousand different directories and paste it into a single file. Using a GUI, you would not only be clicking at your desk for several hours, but you could potentially also commit an error in the process of completing this repetitive task. This is where we take advantage of the Unix shell. The Unix shell is both a command-line interface (CLI) and a scripting language, allowing such repetitive tasks to be done automatically and fast. With the proper commands, the shell can repeat tasks with or without some modification as many times as we want. Using the shell, the task in the literature example can be accomplished in seconds. Now you know what it is! It is a computer program that allows you to "talk" to your computer and give it tasks. But you need to speak its language!

In a Unix or Linux based machines, which includes Macs (they use Unix in the background with a fluffy interface for the user), the command line or shell is easy to access. On a Mac, you can do so by going to the app "Terminal".

Opening a terminal window

For this introductory activity, we will use the terminal pane you opened in our B216_S23 binder.

In your life, know that you can access a terminal in any computer:

- Mac or other Unix based systems: In your personal or lab computer, go to the "Terminal" application. Tip: if you don't know where to find it, open Finder and search for it.
- Windows: not ideal but possible. I don't know how to do it though. Good luck!

Let's get familiar with the terminal!

Remember, the terminal is a way for you to talk to your computer. Or, in this case, a virtual space.

Note: a directory is what you ordinarily call a "folder" in a computer. For example, on my mac I have a `Documents` folder where I have a `B216_S23` folder. Those are both directories, or subdirectories. The ultimate direcotry is the one that contains all the subdirectories! On Mac/Unix machines, that's given by

```
/
```

Once you open a terminal, run the "path to working directory" command:

```
pwd
```

Q1. What does this show you? (I.e., what is the output of this command) Type your answer below.

Now check what's in that directory using the "list" command:

```
ls
```

Q2. What do you think the above command does?

Notice a resemblance between the output of your command line and the left sidebar panel? That's because the terminal is a different way to access things! Instead of clicking on things through a user interface, you type commands.

Why should I use one?

It may not seem worth your while to run a program ONCE using the command-line, but when you find yourself contemplating manually repeating a computational task 100 times, the appeal becomes clear!

- high action-to-keystroke ratio.
- support for automating repetitive tasks
- capacity to access networked machines.

In bioinformatics, there are often user friendly versions of software (though not always), but there is most certainly a command line version! Why?

- the use of memory is more efficient
- when dealing with huge files, the command line is the only way!

Moving on!

After you typed `ls` into your terminal, you saw something in your output. Some of those things are directories. Let's have a look by using the "change directory" command:

```

```

And then check what directory you're currently in with the path to working directory command:

```
pwd
```

Knowing that your downloads folder (on a Mac) has the following path: /Users/your username/Downloads

Type a command that will take you there without having to type the entire path above

```
#type your command in your terminal
```

Now check where you are by using the "pwd" command:

```
#type your command in your terminal
```

Great! Now let's go back to the Desktop folder:

```
#type your command in your terminal
```

Now, we are going to create a subdirectory(a subfolder) within your desktop. Let's call it "B216". The command to create that is "make directory", which is executed by typing:

```
mkdir B216
```

Next, we are going to move into that directory. Remember, you are already in the Desktop directory, and the B216 directory lives inside it, so all you have to do is use the "cd" command followed by the name of the subdirectory. Try it!

```
#type your command in your terminal
```

Great! Now, let's list the files inside this directory. Use the "list" commands, which ix executed by typing "ls":

```
#type your command in your terminal
```

You shouldn't see anything in there, because you just created this - it's brand new! But to check that this works, list the files that are currently in your "Downloads" folder. (if you're anything like your professor, there will be about 2,000 files in there). You will do this by typing "ls" followed by the path to your Downloads folder:

```
#type your command in your terminal
```

Finally, let's go back to your Desktop folder. There are three ways to do this. Two you already know (type the path or the shortcut path we saw above). The third one involves using "..".

Your directories have an internal structure. "Desktop" is a subdirectory of "Users/" and "B216" is a subdirectory of "Desktop". Whenever you are in a subdirectory, you can go back to the one "above" it by simply typing "../".

Try the three different commands, followed by "pwd" to check that they will all get you to where you want to go:

```
#type your command in your terminal and write it down here
cd ../ # this command means "go back to the directory above this one in the hierarchy
#check where you are by typing pwd:
```

References

https://swcarpentry.github.io/shell-novice/01-intro/index.html

The end!