COMMAND LINE PRACTICE

The command line may seem difficult at first, but once you learn a few basic commands you will have the ability to run a wide variety of bioinformatics software. It's just like learning another (programming) language!

If you are brand new to the Command Line, we suggest you start by completing Tutorials 1–4, 8, and 10–14 on this website: https://linuxjourney.com/lesson/the-shell.

> Important notes for following this tutorial:
>
> - Text with a gray background in `monospace font` represents commands to type in
> - Colored text surrounded by **< >** is something you will have to replace with your own username

(0) Setting up the command line on your computer

Option A: I am working on an *Apple* or *Linux* computer

You're all set! Just open the application called "Terminal" on your computer and proceed to the next section.

Option B: I am working on a *Windows 10* computer

Follow these instructions to install Windows Subsystem for Linux (WSL) on your computer: https://www.windowscentral.com/install-windows-subsystem-linux-windows-10. Follow the instructions under section "How to install Windows Subsystem for Linux using Settings." When you get to the step asking you which Linux distribution you want to install, select "Ubuntu."

Then launch your new "Ubuntu" application and proceed to the next section.

Option C: I am working on a *Windows 7* computer

There isn't a built-in way to run Linux commands on older versions of Windows, so we recommend practicing on a Linux, Apple, or Windows 10 computer, if possible. If you need to use an older Windows computer, you have a few options for Command Line practice:

Option C-1: Practice Linux commands online using https://repl.it/languages/bash or a similar website. You will not have access to files on your computer, but this is a quick and easy way to practice the most basic commands. Work in black area on the right side of the screen and proceed to the next section.

Option C-2: Download Cygwin from this website: https://cygwin.com/. To do this, click on the setup-x86_64.exe file to install it. Once it downloads, double click on the file to start the installation procedure. Once it completes, you can open the "Cygwin" application to use a Linux command line on your computer.

(1) Command line practice

1. Use the `pwd` command to figure out your current location within the computer.

   Type the following into your terminal window and then press **Enter**:

   ```
   pwd
   ```

   You should see something like (may differ slightly on Windows computers, and may look very different if you are using an online terminal):

   ```
   /home/<username>
   ```

2. Use the `cd` command to navigate to the Desktop.

   On a Mac or Linux machine, type the following and then press **Enter**:

   ```
   cd ~/Desktop
   ```

   On a Windows machine, type the following and then press **Enter**:

   ```
   cd /mnt/c/Users/<username>/Desktop
   ```

   Remember that punctuation matters! And don't hesitate to use the **tab** key to help you complete paths.

   Note: If you are practicing in an online terminal (e.g., https://repl.it/languages/bash) you will not have access to the files on your computer, so you will not be able to move to the Desktop. Proceed to the next step.

3. Use the `ls` command to see what's in your current location.

   Type the following into your terminal window and then press **Enter**:

   ```
   ls
   ```

   You should see a list of files and folders. These are all the items that are in your Desktop. This should match what you see on the Desktop of the computer you are using. If you are using an online terminal, there may be nothing listed, since you will be on an empty online computer (this is normal).

4. Use the `mkdir` command to make a new folder in your current location.

   Type the following into your terminal window and then press **Enter**:

   ```
   mkdir mytestfolder
   ```

   If you are practicing locally (and not online), you should see a new folder appear on your Desktop.

5. Use the `ls` command to look at the files and folders in your current location again.

   Type the following into your terminal window and then press **Enter**:

   ```
   ls
   ```

   You should see the same output as in #3 above, except this time there should be a folder called "mytestfolder" in this list. The command line is just another way to navigate the files and folders of your computer! (If you are practicing online, you are working on a test computer that has files and folders just like yours).

6. Use the `ls` command to look inside the new folder you just made.

   Type the following into your terminal window and then press **Enter**:

   ```
   ls mytestfolder
   ```

   Try out the **tab** key to help you complete the name of the folder ("mytestfolder"). By entering the name of a folder after the `ls` command, you are telling the computer to look in that location. When you don't provide the name of a folder after the command, the computer will look in your current location by default.

   When you type this command, nothing will happen! You will just see the next line of the command line (ending with **$**). This is because this folder is empty. You just made this brand new folder, so there is nothing in it.

7. Create a new text file in your favorite text editor.

   Leave the terminal window for a moment (don't close it) and open up your favorite text editor (e.g., TextEdit, Notepad++, gedit, etc.). Type "Hello World" in your file and then save it as **myfile.txt**. Make sure you save it inside your new folder ("mytestfolder") which is located on your Desktop. No need to do anything special, just used File>Save to save the file as you normally would.

   Note: If you are practicing in an online terminal (e.g., https://repl.it/languages/bash) you will not have access to the files on your computer. Type the following command into your computer to create a new file on the online computer using the command line:

   ```
   (only type the command below if you are practicing online)

   echo "Hello World" > mytestfolder/myfile.txt
   ```

8. Use the command line to find the file you just created.

   Go back to the terminal window (double check that you are still in the Desktop; if online, in the same folder you were originally) and type the following into your terminal window and then press **Enter**:

   ```
   ls mytestfolder
   ```

The result of this command was blank before, but now there is a file inside of this folder! You should see the name of your file ("myfile.txt") listed after you press Enter.

9. Use the `less` command to look inside the file you just created.

   Type the following into your terminal window and then press **Enter**:

   ```
   less mytestfolder/myfile.txt
   ```

   This command tells the computer to look inside the file that is located inside "mytestfolder." You should see a new screen containing the contents of your file ("Hello World"). Remember: if you forgot the exact name of your file, you can start typing "my" and then press **tab** to autocomplete the rest of the file name.

   Press **q** to go back to the regular terminal screen.

   If you are practicing online, you might see this instead:

   ```
   bash: less: command not found
   ```

   This means that the `less` command is not installed on that online computer. Don't worry – there are other ways to look inside your files on the command line. Proceed to the next step.

10. Use the `more` command to look inside the file you just created.

    Type the following into your terminal window and then press **Enter**:

    ```
    more mytestfolder/myfile.txt
    ```

    This command tells the computer to look inside the file that is located inside "mytestfolder." You should see a new screen containing the contents of your file ("Hello World"). Remember: if you forgot the exact name of your file, you can start typing "my" and then press **tab** to autocomplete the rest of the file name. This command should work on most online command lines as well.

11. Can you think of a different way to look inside your file?

    You could first move into the "mytestfolder" directory and view your file from there. Try typing the following commands into your terminal window and press **Enter** after each command:

    ```
    cd mytestfolder

    more myfile.txt
    ```

    Take a moment to play around with some of the commands you have already learned (specifically, `ls` and `pwd`) to better understand your current location.

12. Go up one directory.

Sometimes you might want to go "up" one directory, i.e., to move to the folder that contains the folder that you are currently in. There is a handy shortcut for this. Type the following into your terminal and then press **Enter**:

```
cd ..
```

Take a moment to use `ls` and `pwd` to better understand your current location.

13. Use the `cp` command to copy the file you created.

    Let's make a copy of "myfile.txt" and put it in a new folder called "mytest_2". Which of the following options will get you the result you want?

    Option A:

    ```
    mkdir mytest_2

    cp mytestfolder/myfile.txt copy_of_myfile.txt
    ```

    Option B:

    ```
    mkdir mytest_2

    cp mytestfolder/myfile.txt mytest_2/copy_of_myfile.txt
    ```

    In both options, you are first making a new folder in your current location called "mytest_2", and in both options the name of your copied file is "copy_of_myfile.txt". So what's the difference?

    In Option A, you are not providing a location for "copy_of_myfile.txt", so by default the file will be copied to your current location (the Desktop, if you are working in a local terminal window). In Option B, you are telling the computer to put "copy_of_myfile.txt" inside the folder you made called "mytest_2". So Option B is the correct answer! Type the text of Option B into your terminal and press **Enter** after each command to execute it.

    Note: if you are having trouble with the above command, make sure you are typing all file and folder names correctly. The **tab** key can be very helpful for this!

14. Use the `ls` command to understand your current file structure.

    Type the following into your terminal window and press **Enter** after each command:

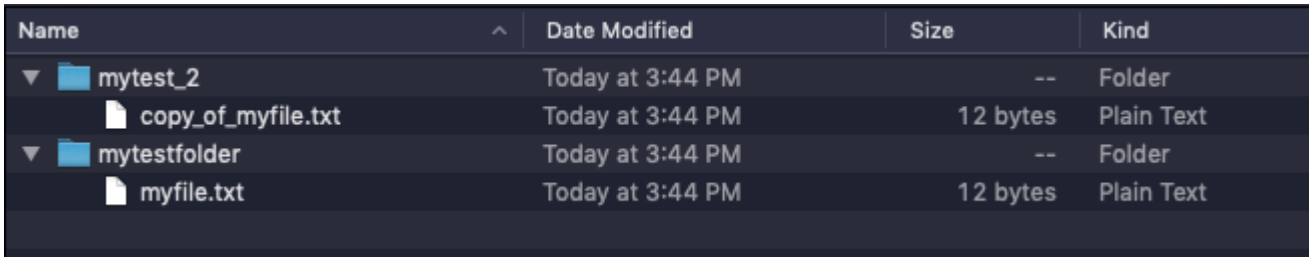
    ```
    ls

    ls mytestfolder

    ls mytest_2
    ```

    Do you understand how your files are organized? If you are working in a local terminal window, take a moment to leave the terminal, open up a File Finder window, and confirm that everything looks as you expect.

    Inside the Desktop folder, you should see something like this (in addition to any other files and folders you already had on the Desktop):



15. Use the `*` to select multiple files and folders.

    What if you want to find all the files and folders that start with "my"? Type the following into your terminal window and press **Enter**:

    ```
    ls my*
    ```

    The asterisk means "match anything", so your computer should return a list of all files and folders inside your current directory that start with "my."

16. Use the `*` to find all text files in a folder.

    More commonly, you may want to perform a command on all files of the same type, e.g., text files. Type the following command to list (the function of the `ls` command) all text files inside "mytestfolder":

    ```
    ls mytestfolder/*.txt
    ```

    You should see the file that you created, since it ends in ".txt":

    ```
    mytestfolder/myfile.txt
    ```

    The asterisk still means "match anything". In this case, we are matching any file name, as long as the extension of the file is ".txt", meaning it is a text file.

17. Finally, use the `rm` command to remove "copy_of_myfile.txt".

    Type the following into your terminal window and then press **Enter**:

    ```
    rm mytest_2/copy_of_myfile.txt
    ```

    BE VERY CAREFUL WHEN USING THE `rm` COMMAND. There is no undo command, so once you delete a file on the command line, it is gone forever. Avoid using this command unless you absolutely have to.

    After running the above, use `ls` to check inside of "mytest_2" – there should be nothing there!

18. This all seems really simple – why am I using the command line instead of clicking around on my computer?

> Great question. The commands you used today are very basic, but the Linux command line can be used to perform infinitely complex tasks on your computer. You can even write a list of commands (called a script) and execute all of them at once with just one simple command. More importantly, a lot of bioinformatics software can only be run from the command line, so it is a useful skill to be able to execute commands and run programs, even if you are not yet able to write programs yourself.

There are many, many resources online (and in print) to teach you how to use the Linux command line. We encourage you to keep practicing these commands and to ask the instructors if you have questions!