**ls, revisited**

So far we've used the command line to navigate the filesystem.

We can do more with the command line to view directories and files. We can also use the command line to copy, move, and remove files and directories. Let's see how to do this.

You can reference the filesystem for this lesson [here](https://s3.amazonaws.com/codecademy-content/courses/learn-the-command-line/img/LCL-fileTrees-02.png). It is highly recommended for this lesson.

# ls -a

$ ls -a . .. .preferences action drama comedy genres.txt

1. The ls command lists all files and directories in the working directory.
2. The -a modifies the behavior of the lscommand to also list the files and directories starting with a dot (.). Files started with a dot are hidden, and don't appear when using ls alone.

The -a is called an option. Options modify the behavior of commands. Here we used ls -a to display the contents of the working directory in more detail.

In addition to -a, the ls command has several more options. Here are three common options:

* -a - lists all contents, including hidden files and directories
* -l - lists all contents of a directory in long format
* -t - order files and directories by the time they were last modified.

Let's practice using these options below.

**ls -l**

$ ls -l drwxr-xr-x 5 cc eng 4096 Jun 24 16:51 action drwxr-xr-x 4 cc eng 4096 Jun 24 16:51 comedy drwxr-xr-x 6 cc eng 4096 Jun 24 16:51 drama -rw-r--r-- 1 cc eng 0 Jun 24 16:51 genres.txt

The -l option lists files and directories as a table. Here there are four rows, with seven columns separated by spaces. Here's what each column means:

1. Access rights. These are actions that are permitted on a file or directory.
2. Number of hard links. This number counts the number of child directories and files. This number includes the parent directory link (..) and current directory link (.).
3. The username of the file's owner. Here the username is cc.
4. The name of the group that owns the file. Here the group name is eng.
5. The size of the file in bytes.
6. The date & time that the file was last modified.
7. The name of the file or directory.

# ls -alt

$ ls -alt drwxr-xr-x 4 cc eng 4096 Jun 29 12:22 . -rw-r--r-- 1 cc eng 0 Jun 29 12:22 .gitignore drwxr-xr-x 5 cc eng 4096 Jun 30 14:20 .. drwxr-xr-x 2 cc eng 4096 Jun 29 12:22 satire drwxr-xr-x 2 cc eng 4096 Jun 29 12:22 slapstick -rw-r--r-- 1 cc eng 14 Jun 29 12:22 the-office.txt

The -t option orders files and directories by the time they were last modified.

In addition to using each option separately, like ls -a or ls -l, multiple options can be used together, like ls -alt.

Here, ls -alt lists all contents, including hidden files and directories, in long format, ordered by the date and time they were last modified.

# cp I

cp frida.txt lincoln.txt

The cp command copies files or directories. Here, we copy the contents of **frida.txt** into **lincoln.txt**.

# cp II

cp biopic/cleopatra.txt historical/

To copy a file into a directory, use cp with the source file as the first argument and the destination directory as the second argument. Here, we copy the file **biopic/cleopatra.txt** and place it in the **historical/** directory.

cp biopic/ray.txt biopic/notorious.txt historical/

To copy multiple files into a directory, use cpwith a list of source files as the first arguments, and the destination directory as the last argument. Here, we copy the files **biopic/ray.txt**and **biopic/notorious.txt** into the **historical/**directory.

# Wildcards

cp \* satire/

In addition to using filenames as arguments, we can use special characters like \* to select groups of files. These special characters are called wildcards. The \* selects all files in the working directory, so here we use cp to copy all files into the **satire/** directory.

cp m\*.txt scifi/

Here, m\*.txt selects all files in the working directory starting with "m" and ending with ".txt", and copies them to **scifi/**.

# mv

The mv command moves files. It's similar to cpin its usage.

mv superman.txt superhero/

To move a file into a directory, use mv with the source file as the first argument and the destination directory as the second argument. Here we move **superman.txt** into **superhero/**.

mv wonderwoman.txt batman.txt superhero/

To move multiple files into a directory, use mvwith a list of source files as the first arguments, and the destination directory as the last argument. Here, we move **wonderwoman.txt**and **batman.txt** into **superhero/**.

mv batman.txt spiderman.txt

To rename a file, use mv with the old file as the first argument and the new file as the second argument. By moving **batman.txt** into **spiderman.txt**, we rename the file as **spiderman.txt**.

# rm

rm waterboy.txt

The rm command deletes files and directories. Here we remove the file **waterboy.txt** from the filesystem.

rm -r comedy

The -r is an option that modifies the behavior of the rm command. The -r stands for "recursive," and it's used to delete a directory and all of its child directories.

Be careful when you use rm! It deletes files and directories permanently. There isn't an undelete command, so once you delete a file or directory with rm, it's gone.

**Generalizations**

Congratulations! You learned how to use the command line to view and manipulate the filesystem. What can we generalize so far?

* Options modify the behavior of commands:
  + ls -a lists all contents of a directory, including hidden files and directories
  + ls -l lists all contents in long format
  + ls -t orders files and directories by the time they were last modified
  + Multiple options can be used together, like ls -alt
* From the command line, you can also copy, move, and remove files and directories:
  + cp copies files
  + mv moves and renames files
  + rm removes files
  + rm -r removes directories
* Wildcards are useful for selecting groups of files and directories