Introduction to the Command Line

What Is a Shell?

A **shell** is a computer program that presents a **command line interface (CLI)**. Instead of clicking buttons in a graphical user interface (GUI), you control your computer by typing commands. Many scientific tools require the command line, and it allows automation, customization, and efficient processing of many files.

Starting Point: The Home Directory

When you open a terminal, you usually start in your **home directory**, which is your personal space on the computer.

To print your current location in the filesystem:

```
pwd
```

Exploring the File System

The filesystem is organized like an inverted tree, with the **root directory** at the top.

List what is in the current directory:

```
ls
```

Commands often accept **flags**, which modify their behavior:

```
ls -F
```

Check available flags using the manual:

```
man ls
```

Press `q` to quit the manual.

Moving Around: Changing Directories

Change directory:

```
cd Desktop
```

Check where you are and what's around you:

```
pwd
ls
```

Move further along a path:

```
cd shell-lesson
pwd
ls
```

List directories and files:

```
ls -F
```

Go back to the parent directory:

```
cd ..
```

Return to the home directory:

```
cd
```

Return to the previous directory:

```
cd -
```

Some invalid commands:

```
cd ---
cd --
```

You can move through multiple directories at once:

```
cd Desktop/shell-lesson
```

Use **Tab completion** to save time and avoid typing errors.

Creating and Removing Directories

Create a new directory:

```
mkdir test
```

Enter and leave the directory:

```
cd test
ls
cd ..
```

Remove a directory:

```
rm test      # error
rm -r test   # correct
```

Viewing Files

Display the entire file:

```
cat 829-0.txt
```

Show only the beginning:

```
head 829-0.txt
head -n 15 829-0.txt
```

Show only the end:

```
tail 829-0.txt
```

Scroll interactively:

```
less 829-0.txt
```

Press `q` to exit.

Command History

Show recent commands:

```
history
```

Use the **up arrow** to repeat recent commands.

Renaming and Moving Files

Rename a file:

```
mv 829-0.txt gullivers_travels.txt
```

Show that the contents are unchanged:

```
head gullivers_travels.txt
```

Use **wildcards**:

```
head -n 3 *.txt
```

Create a directory and move files into it:

```
mkdir gutenberg
mv gullivers_travels.txt gutenberg
```

List the contents of another directory:

```
ls gutenberg
```

Move multiple files:

```
mv *.txt gutenberg
```

Copy files:

```
cp gutenberg/*.txt .
```

Echo and Redirecting Output

Print a message:

```
echo "Repeat this line"
```

Test a command:

```
head -n 1 *.txt
```

Redirect output to a file:

```
head -n 1 *.txt > book_titles.txt
```

View the file:

```
less book_titles.txt
```

Chaining Commands with Pipes

Count lines in a file:

```
wc -l gullivers_travels.txt
```

Combine commands using a pipe (`|`):

```
head -n 20 gullivers_travels.txt | wc -l
```

This sends the output of the first command into the second.

Summary

This lesson introduced: - Navigating the filesystem with `pwd`, `ls`, and `cd` - Creating and deleting directories - Viewing and manipulating files - Using wildcards - Copying, moving, and renaming files - Redirecting output and chaining commands - Using history and tab completion

Practice these commands to build confidence. The command line is a powerful tool once you get used to it!