Terminal Kung-Fu

13 MAY 2015 on Tools (/tag/tools/), Programming Principles (/tag/programming-principles/), Level 1 (/tag/level-1/), Lesson (/tag/lesson/)

What is the terminal?

When launched, it provides a line interface to control the underpinnings of the UNIX-based operating system. In short, it lets you control your computer like hackers from the movies!

Why use it?

Until now, you've been used to using a mouse and monitor to interact with your computer. However, some systems you'll want to use don't have a graphical user interface and many of the applications you will use in this class are only run (or better run) via the terminal.

What can I do with the terminal?

There's a lot you can do with it! You can create files, delete them, make folders, run applications, and perform complex operations all via the command line. Here are some examples of a few:

- **`pwd`** tells you what directory you are currently in.
- **`ls`** lists the files and folders in the current directory.
- **`cd`** changes what directory you're in. You can use **/** to get to the root directory quickly.
- **`mkdir`** makes a new folder in the current directory.
- **`touch`** creates a new file with the name and extension given.
- **`rm`** deletes a file or folder. Use it with -r to force it to delete a folder even if there are still objects in the folder you are trying to delete.
- **`mv`** either *moves* or *renames* a file or folder, depending on how you use it
- To *rename* a file, you would do `mv <old-file-name> <new-file-name>`, without the `<` and `>`. For example:

```
mv myOldFile.js myAwesomeNewFile.js
```

- To move a file or folder to another directory, you would do `mv <file-to-move> <place-to-move-it-to>`, again without the `<` and `>`. For example:

```
mv fileToMove.js dev/

#(Moves this fileToMove.js file into the dev folder)
```

OR

```
mv fileToMove.txt ../../my-folder/things-i-save/notes/

#(Moves the file back two directories, forward into the my-folder directo
```

- `cp [file to be copied] [where to put copy]` copies the files that you name.

- `cp -r [new directory name]` will copy a directory.

- `nano` will open the nano text editor. If we say `nano [file that already exists]` we will be editing that file, but if we say `nano [new file that doesn't exist yet]` we will create a new file and start editing it.

    - At the bottom of Nano, we can see that if we hold down control and push "x" we will exit. While we exit, it asks us to push `y` to save, and then `enter` to confirm the file name.
- `help` gives a list of useful commands such as the one given in this list.

What complex operations can one do in the terminal?

Now that you know how to do some of the most basic operations in the terminal, you are prepared to deal with more complex programs like Git, SSH-ing into a server, editing files directly in terminal, etc. Another great resource that is best accessed through the terminal is called a package manager. In this class you will learn to use NPM, which is a package manager that will give you access to powerful tools to use within your own work.

(/author/erik/)

Erik Slack (/author/erik/)

Share this post

Read more posts (/author/erik/) by this author.

Search for another post

SEARCH!

(/intro-to-javascript/)

READ THIS NEXT

Intro to JavaScript

History of Javascript JavaScript was originally created in 10 days by Brendan Eich who was working at Netscape. The...

(/string-manipulation/)

YOU MIGHT ENJOY

Exercise - String Manipulation

The purpose of this exercise is to get you comfortable using the terminal with node, npm, and your text...