Commands to explore the file system

- pwd
  - Displays the Absolute path of the current working directory
  - To use, type *pwd*
  - pwd: prints the AP of the current working directory
- cd
  - Changes the current working directory. Moves you from one directory to another.
  - By default, will always take you to your home directory.
  - cd + (name of directory)
  - cd: takes you home
  - cd ~: expand to the AP of the user's home directory
  - cd/usr/share/themes: Takes you to the user's themes directory
- ls
  - Used to list files and directories.
  - By default will list the current directory when no other directory is specified
  - ls + (option) + (directory)
  - ls --help: shows all options of the ls command
  - ls -A: Lists all files including hidden ones in current directory
  - ls -t: lists a directory sorted by last time modified
  - ls -lhgGr Documents/: Long lists a directory excluding group and owner info, with human readable file sizes and sorts them in reverse order

Definitions

- File system: Similar to an upsidown tree, the paths and branches of your system, starting from THE ROOT and going down into each folder to every file.
- Pathname: Also called the file path, it's the location of a file in your computer, can either be absolute or relative.
- Absolute Path: The location of a file starting from the root of the entire file system. They can be used from anywhere in the file system to navigate. A command with an absolute path will always find the file because it will always start at the root, or beginning, of the file system. Always longer to type than a relative path.
- Relative Path: The location of a file starting from a child directory of the current working directory or from within the current directory. Typing a relative path is always faster, but it can't work from anywhere in the file system, they're restricted to the directory they are in. To use a relative path and make it work, the file must be reachable from the current directory and onward. Shows a better understanding of Linux and shows you have a good mental image of the file system you are in.
- THE HOME DIRECTORY Vs. YOUR HOME DIRECTORY
  - THE HOME DIRECTORY is the absolute home of the file system on the computer. Think of it as the entire apartment building. YOUR HOME DIRECTORY is like your single apartment in the apartment building. It's all the files and folders in the current user's file system. The user has complete control and can make all the decisions in their home directory, but for THE HOME DIRECTORY, that's overseen by the admin of the building. It's the home directory of all the

user home directories. The home directory contains all user directories and if the admin wishes, can alter them how they need to. A user cannot go into someone else's files, but the admin can.

- Parent Directory: A directory that contains one or more directories and files.
- Child Directory: Also known as a subdirecttory; A directory inside of another directory, similar to a child and their parent, they go together with Parent Directory.
- Bash Special Characters:
    - . (single period): represents the current directory
    - .. (two consecutive periods): represents the parent directory
    - ~ (tilde): expands the current users' home directory. A variable the shell uses to store the AP (absolute path) of the users' home directory. ~/Downloads is the same as /home/rdeida/Downloads
    - / (one forward slash): the root directory and the shortest path in the file system. The beginning of the directory tree, nothing is before it, but everything is after it.
    - - (hyphen): used to switch to the previous current working directory
    - # (number sign): Used for single line comments in shell scripting
    - ! (exclamation mark): repeats a command from from the history. !5 would repeat the 5th command in the command history. Typing in *history* shows the command history
    - !! (two exclamation marks): repeats the last command. *!!* will repeat the previous command while *sudo !!* will repeat the last command but with an added *sudo* at the beginning. Useful for when sudo is not added when trying to do admin tasks.
- Environment Variables
    - A variable is a place to store data in programming. Think of it as a box with a label on it. Whatever label you give that box, the box will store it.
    - An environment variable stores values of a user's environment and can be used in commands. To see environment variables type *env* in the shell. Some examples are:
        - $USER - Stores current user's name
        - $HOME - stores AP of the current user's home directory
        - $PWD - Sores tha absolute path of the present working directory
        - $OLDPWD - Sores tha absolute path of the previous working directory
- User Defined Variables
    - Variables a user can create in a script that exist only there to store info. Used to temporarily store data for said script. Such as defining a name, age, value, or anything from a user input.
        - Must be a string up to 20 letters, digits, or underscore characters
        - Cannot start with a number
        - Case sensitive
            - val1 does not equal Val1
        - Assigned using an equals sign with no spaces
            - name='Romelo'
        - Shells stores all values as text strings
            - Bash is pretty much untyped
        - The $ defines an environment variable, making it important when typing them out. If we need to use the $ in any part of the script, we escape it with \