

# Notes 4

---

- [Presentation](#)
- [article](#)

## File system

- **Description:** The way files are stored and organized.
- **Usage:** `/root directory`
- **Example:**
  - `/usr`: Contains most user utilities and applications
  - `ls`: list file and directories in the current directory
  - `/home`: Holds all the home directories for the users

## pathname

- **Description:** In a filesystem, every file has a **pathname** which indicates the location of the file in the filesystem.
  - **Usage:** `/root directory/current directory/parent directory`
  - **Example:**
    - `echo /home/$USER`
    - `/home/your_user_name/my_script`
    - `echo $HOME`

## Absolute path

- **Description:** the location of file starting a root of the file system
  - **Usage:** `root directory/home directory/parent directory/child directory`
  - **Example:**
    - `/home/john/Downloads/song.mp3`
    - `/home/maria53/Downloads/list.txt`
    - `cat /home/kt/abc.sql`

## Relative path

- **Description:** the location of a file starting from the current working directory or a directory that is located inside the current working directory.
  - **Usage:** `current directory-parent directory/child directory`
  - **Example:**

- Downloads/song.mp3
- Downloads/list.txt
- \$pwd
  - /home/kt

## The difference between your home directory and the home directory

- **Description:** ***Your home directory*** is specific to you and contains your personal files, while ***the home directory*** is a broader concept representing any user's personal file storage area.
  - **Usage:** `/home/username`
  - **Example:**
    - `your hoome:/home/maria53`
    - `home:/home`

## parent directory

- **Description:** a dirtectory containing one or more directories and files.
  - **Usage:** `root directory/current directory/parent directory/child directory`
  - **Example:**
    - `/home/user1`
    - `../File.txt`
    - `cd ..`

## child directory or subdirectory

- **Description:** a better name for this is a subdirectory or subfolder. This is a directory inside another directory.
  - **Usage:** `root directory/current directory/parent directory/child directory`
  - **Example:**
    - `home/downloads/music/music_1`
    - `.././pictures/picture_1`
    - `new_movies/horror/evil_dead`

## Bash special characters

- **Description:** Special characters are function like commands that tell the shell to perform a specific action without having to type the complete command. These special characters make working on the command line more efficiently. Here is short list to keep in mind.
- **Usage**

- perform specific action
- **Example:**
  - **(single period):** represents the current directory.
  - **.. (2 consecutive periods):** represents the parent directory.
  - **~ (tilde character):** expands the current users home directory. It is like a variable that the shell uses to store the absolute path of the user's home directory.
  - **/ (one forward slash):** as mentioned earlier, this is the root directory and the shortest path in the system. This is the beginning of the directory tree. There is nothing before it and everything after it.
  - **- (hyphen-minus):** is used to move to the previous current working directory.

## environment variables

- **Description:** Environment variables store values of a user's environment and can be used in commands in the shell. These values can be unique to the user's environment which makes them ideal when writhing commands that you want to use regales of which user is using the computer.
- **Usage** \$ option
  - **Example:**
    - **\$USER**= stores the current's user username
    - **\$HOME** = stores the absolute path of current's user home directory
    - **\$PWD** = stores the absolute path of the present working directory.
    - **\$OLDPWD** = stores the absolute path of the previous current working directory

## user defined variables

- **Description:** User-defined variables are fields which you define yourself in a program. They are used to store values or intermediate results obtained at some point in program processing for additional processing or display.
  - **Usage:**
    - Store Data
  - **Example:**
    - username: Maria
    - age: 20
    - gpa: 3.0

## Why do we need use \$ with variables in bash shell scripting?

- **Description:** The \$ signals to the shell that it should replace the variable name with its actual value
- **Usage**

- shell script, run command in terminal

- **Example:**

- `#!/bin/bash var1="devil" var2=23 echo $var1 $var2`
- `#!/bin/bashecho "The Current User: $USER" echo "The Current Shell:$SHELL" echo "The Current's User Home: $HOME" echo -e "The Path Var: \n $PATH"`