

# CLI - Understanding paths in Bash

In Bash, a "path" typically refers to a filesystem path, which is used to specify the location of a file or directory in the filesystem hierarchy. Understanding how to use paths effectively is a fundamental skill when working in the Linux shell. Here's a step-by-engine tutorial on understanding and manipulating paths in Bash.

## 1. Understanding Path Types

There are two types of paths in Bash:

- **Absolute paths** : These begin with the root directory (`/`) and provide a complete address of a file or directory from the root of the filesystem.

Example for absolute path:

```
/home/user/Documents/file.txt.
```

- **Relative paths** : These are relative to the current working directory. They do not begin with a slash.

Examples for relative paths include:

- `Documents/file.txt` (relative to the current directory)
- `./Documents/file.txt` (also relative to the current directory, where `.` represents the current directory)
- `../sibling_folder/file.txt` (uses `..` to represent the parent directory)

## 2. Viewing the Current Path

To find out your current directory, use the `pwd` command, which stands for "print working directory".

```
pwd
```

## 3. Changing Directories

To change the current directory, use the `'cd'` command.

- To go to the home directory:

```
cd ~
```

- To go up one directory level:

```
cd ..
```

- To enter into a specific directory (using absolute path):

```
cd /path/to/directory
```

#### 4. Listing Files and Directories

Use the `ls` command to list the contents of a directory.

- List files in the current directory:

```
ls
```

- List files in a specific directory:

```
ls /path/to/directory
```

- List all files, including hidden ones:

```
ls -a
```

Understanding and using paths correctly in Bash can significantly enhance your efficiency and effectiveness in handling files and directories. Practice different commands such as `mv`, `cp`, `rm` to build your proficiency in managing your Linux filesystem from the Bash shell.