

Shell Basic Syntax and Common Commands

Basic Command Syntax

A basic Bash command typically follows this structure:

```
command [options] [arguments]
```

- **command:** The name of the program or utility you want to execute.
- **options:** Also known as flags or modifiers, these are used to alter the behavior of the command. They usually start with a dash (-) or double dash (--).
- **arguments:** These specify the input for the command, such as files, directories, or other data.

Examples

1. Listing Files:

```
ls -l /home/user
```

- **ls:** Command to list directory contents.
- **-l:** Option to use long listing format.
- **/home/user:** Argument specifying the directory.

2. Copying Files:

```
cp file1.txt file2.txt
```

- **cp:** Command to copy files.
- **file1.txt:** Source file (argument).
- **file2.txt:** Destination file (argument).

Common Options (Flags)

Options modify the behavior of commands. Some common options include:

- Single-letter options prefixed with a single dash (-):
 - `-a`: All (e.g., `ls -a` shows hidden files).
 - `-r`: Recursive (e.g., `cp -r` copies directories recursively).
- Word options prefixed with a double dash (--):
 - `--help`: Displays help information about the command.
 - `--version`: Shows the version of the command.

Arguments

Arguments are the inputs to commands. They can be filenames, directories, or other data. Multiple arguments are separated by spaces.

Quoting and Escaping

- **Double Quotes (")**: Preserve the literal value of all characters within the quotes, except for `$`, ```, and `\`.

```
echo "Hello, $USER"
```

- **Single Quotes (')**: Preserve the literal value of all characters within the quotes.

```
echo 'Hello, $USER'
```

- **Backslash (\)**: Escapes the following character, preserving its literal value.

```
echo Hello\ World
```

Variables

Variables are used to store data that can be reused in scripts.

- **Assigning Values:**

```
NAME="John"
```

- **Accessing Values:**

```
echo $NAME
```

Common Bash Commands

pwd : Path to working directory

ls : list all files / directories

ls -a : list all files / directories including hidden files / directories

clear : clear the window

cd : change directory

cd ~ : change to the current user directory

mkdir : make directory . Note: use quotation marks if your name for the directory or the file have spaces .

touch / (**new-item** for windows powershell or you can use a workaround like **echo ""> newFile.txt**) : create a new file.

Open / (**start** for windows powershell) : open a file or directory

cat : to output the content of a file.

man : to display the manual for the command . To exit type q

rm : remove a file .

rmdir : remove a directory .

cp : Copy files or directories from one location to another.

Example:

```
cp file1.txt /home/user
```

will copy "file1.txt" to the /home/user directory.

mv : Move or rename files or directories.

Example:

```
mv file1.txt file2.txt
```

will rename "file1.txt" to "file2.txt".

find : to search for a file

Example:

```
find . -name "*.txt"
```

grep : to search for a pattern in a file

Example:

```
grep "search_text" file1.txt
```

will search for "search_text" in "file1.txt".

. : current directory

.. : parent directory

Reference:

<https://www.gnu.org/savannah-checkouts/gnu/bash/manual/bash.html>