

# Other Commonly Used Bash Commands

Get to learn some useful and important bash commands that are necessary in order to interact with your machine efficiently and get your day-to-day tasks done.

## 1. echo

### Definition:

`echo` is the built-in command in bash and C-shells that simply prints its arguments on the console or terminal.

### Syntax:

```
echo [option(s)] [string(s)]
```

### Options:

Options	Meanings
-n	Do not output a trailing newline.
-e	Enable interpretation of backslash escape sequences.
-E	Disable interpretation of backslash escape sequences.
-help	Display a help message and exit.

### Example:

- To print a statement “I like to code.” on the console

```
echo I like to code.
```

```
echo I like to code.
```



- To print a statement having words in each consecutive line

```
echo -e 'Every \nword \nin \nnew \nline.'
```

Here, `\n` represents a “new line”, and `-e` is enabling for the command line to interpret the backslash.

```
echo -e 'Every \nword \nin \nnew \nline.'
```



## 2. clear

### Definition:

The clear command is used to remove all previous commands from *console*. It neither accepts “options” nor “arguments” (input files). After it is executed, all there is left on the command line is the *command prompt* on the upper left corner.

### Syntax:

```
clear
```

## 3. sleep

### Definition:

Sleep command pauses for some time as specified by the **NUMBER**

sleep command pauses for some time as specified by the **NUMBER**

## Syntax:

```
sleep NUMBER[SUFFIX]
```

The **SUFFIX** may be:

- *s* for seconds (the default)
- *m* for minutes
- *h* for hours
- *d* for days

## Example:

To sleep for 4 hours:

```
sleep 4h
```

To sleep for 5 days:

```
sleep 5d
```

To sleep for 5.5 seconds:

```
sleep 5.5
```

```
sleep 5.5  
ls
```

