

# Bash Echo – Syntax & Examples

## Bash Echo Command

**Bash Echo** is a command in bash shell that writes its arguments to standard output.

Whatever you see in the terminal is because of echo command being executed by other programs.

Following are the topics we shall go through in this tutorial :

- [Syntax](#)
- [Simple Echo Example](#)
- [Example-1](#) – Echo without trailing newline
- [Example-2](#) – Echo interpreting backslash escaped characters
- [Example-3](#) – Echo without interpretation

## Syntax

Following is the syntax of echo command

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

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

## Options Available with Echo Command

Following are the options available with echo command :

Option	Description	Example
-n	Trailing outline is omitted	echo -n "Learn Bash"
-E	Disable interpretation of backslash escaped characters	echo -E "Learn\nBash"
-e	Enable interpretation of backslash escaped characters	echo -e "Learn\nBash"

## Examples

Terminal is a bash shell program. It can execute bash commands directly in the terminal or you may create a bash script file and run the file in terminal.

## Example – Simple Echo Example

Directly on Terminal

```
arjun@arjun-
VPCEH26EN:~/bash$
arjun@arjun-VPCEH26EN:~/bash$ echo "Learn Bash"
Learn Bash
```

Using Script File

```
bash-echo-example
#!/bin/bash
#!/bin/bash
echo "Learn Bash"
```

When you run above bash script in Terminal

```
Bash Script Output
arjun@arjun-
VPCEH26EN:~/bash$
arjun@arjun-VPCEH26EN:~/bash$ ./bash-echo-example
Learn Bash
```

## Example – Echo without trailing newline

Directly on Terminal

```
arjun@arjun-
VPCEH26EN:~/bash$
arjun@arjun-VPCEH26EN:~/bash$ echo -n "Learn Bash"
Learn Bash arjun@arjun-VPCEH26EN:~/bash$
```

Using Script File

```
bash-echo-example
#!/bin/bash
```

```
#!/bin/bash

echo -n "Learn Bash"
```

When you run above bash script in Terminal

Bash Script Output

```
arjun@arjun-VPCEH26EN:~/bash$ ./bash-echo-example
Learn Bash arjun@arjun-VPCEH26EN:~/bash$
```

## Example – Echo command interpreting backslash escaped characters

Directly on Terminal

```
arjun@arjun-VPCEH26EN:~/bash$ echo -e "Learn\nBash"
Learn
Bash
```

Using Script File

bash-echo-example

```
#!/bin/bash

echo -e "Learn Bash"
```

When you run above bash script in Terminal

Bash Script Output

```
arjun@arjun-VPCEH26EN:~/bash$ ./bash-echo-example
Learn
Bash
```

## Example – Echo without interpretation of escaped characters

Directly on Terminal

```
arjun@arjun-VPCEH26EN:~/bash$  
arjun@arjun-VPCEH26EN:~/bash$ echo -E "Learn\nBash"  
Learn\nBash
```

Using Script File

```
bash-echo-example  
#!/bin/bash  
  
#!/bin/bash  
  
echo -E "Learn Bash"
```

When you run above bash script in Terminal

```
Bash Script Output  
arjun@arjun-VPCEH26EN:~/bash$  
arjun@arjun-VPCEH26EN:~/bash$ ./bash-echo-example  
Learn\nBash
```

## Conclusion :

In this [Bash Tutorial](#) – **Bash Echo**, we have learnt the syntax of echo command, options available with echo command, with example Bash Script Files.

### Bash Shell Scripting

- [Bash Tutorial](#)
- [Bash Script Example](#)
- [Bash File Extension](#)
- [Bash Echo](#)
- [Bash Comments](#)
- [Bash Variable](#)
- [Bash Command Line Arguments](#)

↳ [Bash Read User Input](#)

↳ [Bash Read Password](#)

↳ [Bash Date Format](#)

↳ [Bash Sleep](#)

## Operators

↳ [Bash Arithmetic Operators](#)

## Conditional Statements

↳ [Bash If](#)

↳ [Bash If Else](#)

↳ [Bash Else If](#)

↳ [Bash Case](#)

## Loops

↳ [Bash For Loop](#)

↳ [Bash While Loop](#)

↳ [Bash Until Loop](#)

## Strings

↳ [Bash String Manipulation Examples](#)

↳ [Bash String Length](#)

↳ [Bash If String Equals](#)

↳ [Bash Split String](#)

↳ [Bash SubString](#)

↳ [Bash Concatenate String](#)

↳ [Bash Concatenate Variables to Strings](#)

## Functions

↳ [Bash Function](#)

↳ [Bash Override Built-in Commands](#)

## Arrays

↳ [Bash Array](#)

## Files

↳ [Bash Write to File](#)

↳ [Bash Read File](#)

↳ [Bash Read File line by line](#)

↳ [Bash If File Exists](#)

┆ Bash If File is Directory

┆ Bash If File is Readable

## Bash Others

┆ Bash Check if variable is set