

Bash Date – Format Options and Examples

In this tutorial, we will learn the available format options for date command and how to format a date in Bash Scripting, with examples.

Bash Date

To format Bash Date to a required one, bash shell provides date command along with many format options.

Bash Date Command

Following is the syntax of date command

```
$ date
```

Format Bash Date with Options

As already said, you can format the Bash Date. And the format you wish may contain spaces as well.

Date command accepts options if provided any

```
$ date +<format-option><format-option>
```

To format date with spaces, use the syntax

```
$ date '+<format-option><format-option> <format-option>'
```

List of Bash Date Formatting Options

Following are the list of available options for date command :

Format option	Part of Date	Description	Example Output
date +%a	Weekday	Name of weekday in short (like Sun, Mon, Tue, Wed, Thu, Fri,	Mon

Format	Part of Date	Description	Example
date +%A	Weekday	Name of weekday in full (like Sunday, Monday, Tuesday)	Monday
date +%b	Month	Name of Month in short (like Jan, Feb, Mar)	Jan
date +%B	Month	Month name in full short (like January, February)	January
date +%d	Day	Day of month (e.g., 01)	04
date +%D	MM/DD/YY	Current Date; shown in MM/DD/YY	02/18/18
date +%F	YYYY-MM-DD	Date; shown in YYYY-MM-DD	2018-01-19
date +%H	Hour	Hour in 24-hour clock format	18
date +%I	Hour	Hour in 12-hour clock format	10
date +%j	Day	Day of year (001..366)	152
date +%m	Month	Number of month (01..12) (01 is January)	05
date +%M	Minutes	Minutes (00..59)	52
date +%S	Seconds	Seconds (00..59)	18
date +%N	Nanoseconds	Nanoseconds (000000000..999999999)	300231695
date +%T	HH:MM:SS	Time as HH:MM:SS (Hours in 24 Format)	18:55:42
date +%u	Day of Week	Day of week (1..7); 1 is Monday	7
date +%U	Week	Displays week number of year, with Sunday as first day of week (00..53)	23
date +%Y	Year	Displays full year i.e. YYYY	2018
date +%Z	Timezone	Time zone abbreviation (Ex: IST, GMT)	IST

You may use any of the above-mentioned format options (first column) for the date command in the aforementioned syntax.

Examples

Bash Date Format MM-DD-YYYY

To format date in MM-DD-YYYY format, use the command `date +%m-%d-%Y`.

Bash Script

```
#!/bin/bash
```

```
d=`date +%m-%d-%Y`
```

```
echo $d # 12-30-2017
```

Output

```
11-20-2020
```

Please observe the upper and lower case letters : %m for month, %d for day and %Y for year. %M would mean minutes.

Bash Date Format MM-YYYY

To format date in MM-YYYY format, use the command `date +%m-%Y` .

Bash Script

```
#!/bin/bash
d=`date +%m-%Y`
echo $d # 12-2017
```

Output

```
11-20-2020
```

Bash Date Format: Weekday DD-Month, YYYY

To format date in MM-DD-YYYY format, use the command `date +%m-%d-%Y` .

Bash Script

```
#!/bin/bash
d=`date +%A %d-%B, %Y`
echo $d # Saturday 30-December, 2017
```

Output

```
11-20-2020
```

Conclusion

Concluding this [Bash Tutorial – Bash Date](#), we have learned to use date command, the list of Bash Date

Concluding this [Bash Tutorial - Bash Date](#), we have learned to use date command, the list of Bash Date Format options available with the date command and some of the examples demonstrating the usage of the format options.

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 Builtin 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