

How to compare two text files on Linux Terminal

If you want to see the difference between two text files, use the diff command. This tutorial will show you how to use diff commands on Linux and macOS.

If you want to see the difference between two text files, use the diff command. This tutorial will show you how to use diff commands on Linux and macOS.

1. Use document comparison feature in Word 2010
2. How to compare data on 2 Excel columns
3. How to filter duplicate data on 2 Excel sheets

The diff command compares two text files and creates a list of differences between the two files. To be more precise, it creates a list of changes in the first file to match the second file. Knowing this you will understand the diff command output more easily. The diff command is designed to find the difference between source code files and create output that can be read and operated on other programs such as patch commands.

Let's start analyzing two files. The order of the files on the command line determines the diff command file to be the first file and the second file. In the example below, alpha1 is the first file and alpha2 is the second file. Both files contain phonetic alphabet but the second file, alpha 2, has made some edits so that the two files are not the same.

We can compare files using the following command. Type diff, space, first file name, space, second file name and press **Enter** .

```
alpha1 diff alpha2
```



Read the output like? Each difference between the two files is listed in order in a column and labeled. This label contains numbers on both sides and middle letters, for example 4c4. The first number is the number of lines in

alpha1 and the second number is the number of lines in alpha2. The middle letter means the following:

1. c: The first file line needs to be changed to match the line in the second file.
2. d: The line in the first file needs to be deleted to match the second file.
3. a: Additional content should be added to the first file to match the second file.

For example, 4c4 means that the fourth line in the alpha1 file must be changed to resemble the fourth alpha2 file line. This is the first difference between the two files that diff finds.

Lines start with `<` reference to the first file and lines beginning with `>` reference to the second file. The **Delta** series says the word Delta is the fourth line content in the alpha1 file. Line `> Dave` said from Dave is the 4th line content on the alpha2 file. In short, you need to replace the word Detal from Dave on the fourth line in the alpha1 file so that the two files are the same.

The next change is 12c12, similarly, we can see that line 12 of the alpha1 file contains from Lima, but line 12 in alpha2 comes from Linux.


The third change tells us that one line has been deleted in the alpha2 file. The specified 21d20 label needs to delete line 21 in the first file so that the two files are the same. The `>` line indicates the line contents to be deleted from alpha1.

The fourth difference is 26a26.28. This label indicates that there are three more lines in the alpha2 file than the alpha1 file. Line numbers 26 and 28 are separated by commas to represent line ranges. In this example, the range from line 26 to line 28. You can understand this label needs to add three lines in alpha2 to alpha1 so that the two files are the same. The lines to add contain Quirk, Strange and Charm.

If you only want to know if the two text files are the same, use the `-s` option.

```
diff -s alpha1 alpha3
```


```
dave@howtogeek:~/work$ diff -s alpha1 alpha3
Files alpha1 and alpha3 are identical
dave@howtogeek:~/work$
```



You can use the `-q` option to see if two files are different.

```
diff -s alpha1 alpha3
```

```
dave@howtogeek:~/work$ diff -q alpha1 alpha2
Files alpha1 and alpha2 differ
dave@howtogeek:~/work$
```



And if there are different results, this option does not specify the difference between the two files.

The **-y** option uses another layout to display the differences between the two files. When using this option, you can use the **-w** option to limit the number of columns displayed for readable results. Below is the diff command to create the results displayed on both sides and limited to 70 columns.

```
diff -y -W 70 alpha1 alpha2
```

```
Juliet          Juliet
Kilo            Kilo
Lima            |    Linux
Mike            Mike
November       November
Oscar          Oscar
Papa           Papa
Quebec         Quebec
Romeo          Romeo
Sierra         Sierra
Tango          Tango
Uniform        <
Victor         Victor
Whiskey        Whiskey
X-ray          X-ray
Yankee         Yankee
Zulu           Zulu
               >    Quirk
               >    Strange
               >    Charm
```

dave@howtogeek:~/work\$

The first file on the command line is displayed on the left and the second file is displayed on the right. Lines in each file are displayed side by side. There are indicator characters along with these lines in alpha2 indicating changes, deletions or adding content.

1. |: Change line is in the second file.
- 2.
3. >: Added line in the second file and the first file is not available.

If you like to briefly summarize these changes, use the **suppress-common-lines option** so that the diff command only shows changes, adds or deletes.

```
diff -y -W 70 --suppress-common-lines alpha1 alpha2
```

```
dave@howtogeek:~/work$ diff -y -W 70 --suppress-common-lines alpha1 al
pha2
Delta          |    Dave
Lima           |    Linux
Uniform        <
               >    Quirk
               >    Strange
               >    Charm
```

dave@howtogeek:~/work\$



Add color to the command

Another utility called **colordiff** adds color to highlight diff output to help viewers see the difference.

To install this utility, use **apt-get** if you use Ubuntu or another Debian-based distribution. On other Linux distributions use that distro's package management tool.

```
sudo apt-get install colordiff
```

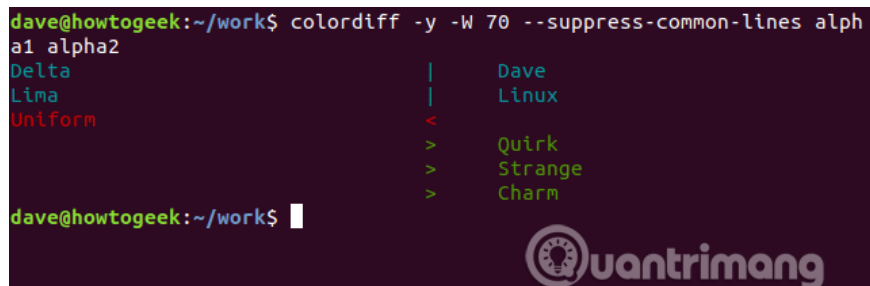
Use colordiff as diff.

```
dave@howtogeek:~/work$ colordiff --suppress-common-lines alpha1 alpha2
4c4
< Delta
---
> Dave
12c12
< Lima
---
> Linux
21d20
< Uniform
26a26,28
> Quirk
> Strange
> Charm
dave@howtogeek:~/work$
```



In fact, colordiff is a wrapper for diff, so you can still use diff options with colordiff.

```
dave@howtogeek:~/work$ colordiff -y -W 70 --suppress-common-lines alpha1 alpha2
Delta          |      Dave
Lima          |      Linux
Uniform       <
              >      Quirk
              >      Strange
              >      Charm
dave@howtogeek:~/work$
```



Provide some context to easily find differences

To find some changes between two text files, instead of listing them all, you can request diff to provide some specific context. There are two ways to do this and produce the same result, displaying some lines before and after each change line.

The first way is to use the **-c** option.

```
colordiff -c alpha1 alpha2
```

```
dave@howtogeek:~/work$ colordiff -c alpha1 alpha2
*** alpha1      2019-04-08 10:13:40.816588696 -0400
--- alpha2      2019-04-08 11:05:10.728774006 -0400
*****
*** 1,7 ***
Alpha
Bravo
Charlie
! Delta
Echo
Foxtrot
Golf
--- 1,7 ---
Alpha
Bravo
Charlie
! Dave
Echo
Foxtrot
Golf
*****
```



This diff output has a title, the title lists the two file names and its edit times. Have an asterisk (*) before the first file and dash (-) before the second file. Asterisks and dashes will be used to indicate the output of that file.

The line with an asterisk and 1.7 in the middle indicates changes in the alpha1 file from lines one to seven. The Delta word marked has changed. It has an exclamation point (!) Next to it and is red. You can see before and after that text there are three lines of text that remain unchanged, indicating the context of that line in the file.

The line with a dash with 1.7 in the middle indicates changes in the alpha2 file. Similar to the above, we see the word Dave in line 4 marked changes.

The three lines of text unchanged before and after each change are the default values. You can specify this line number if you want. To do so, use the -c option with the uppercase C and the number of lines you want to display.

```
colordiff -C 2 alpha1 alpha2
```

```
dave@howtogeek:~/work$ colordiff -C 2 alpha1 alpha2
*** alpha1      2019-04-08 10:13:40.816588696 -0400
--- alpha2      2019-04-08 11:05:10.728774006 -0400
*****
*** 2,6 ***
Bravo
Charlie
! Delta
Echo
Foxtrot
--- 2,6 ---
Bravo
Charlie
! Dave
Echo
Foxtrot
*****
*** 10,14 ***
Juliet
Kilo
! Lima
```



The second option is to use the -u option.

```
colordiff -u alpha1 alpha2
```

```
dave@howtogeek:~/work$ colordiff -u alpha1 alpha2
--- alpha1      2019-04-08 10:13:40.816588696 -0400
+++ alpha2      2019-04-08 11:05:10.728774006 -0400
@@ -1,7 +1,7 @@
Alpha
Bravo
Charlie
-Delta
+Dave
Echo
Foxtrot
Golf
@@ -9,7 +9,7 @@
India
Juliet
Kilo
-Lima
+Linux
Mike
November
Oscar
```



As with the previous option, you also have a title on the output, including the file name and the number of changes. The hyphen (-) is preceded by the name of alpha1 and the plus sign (+) before the alpha2 name. This tells us that dashes are used to indicate alpha1 and plus signs for alpha2. You can see on the figure above that the @ signs appear scattered in the list, it marks the beginning of the changes and which row of each file.

You can see the three lines before and after the highlighted line, indicating the context of the line changing. The line from the alpha1 file starts with a dash, the line from the alpha2 file starts with a plus sign. With this option, it only takes 8 lines to list the difference while the above option requires up to 15 lines.

Similar to the above option, you can use the -u option with uppercase U to provide the desired context line number.

```
colordiff -U 2 alpha1 alpha2
```

```
dave@howtogeek:~/work$ colordiff -U 2 alpha1 alpha2
--- alpha1      2019-04-08 10:13:40.816588696 -0400
+++ alpha2      2019-04-08 11:05:10.728774006 -0400
@@ -2,5 +2,5 @@
Bravo
Charlie
-Delta
+Dave
Echo
Foxtrot
@@ -10,5 +10,5 @@
Juliet
Kilo
-Lima
+Linux
Mike
November
@@ -19,5 +19,4 @@
Sierra
Tango
-Uniform
```

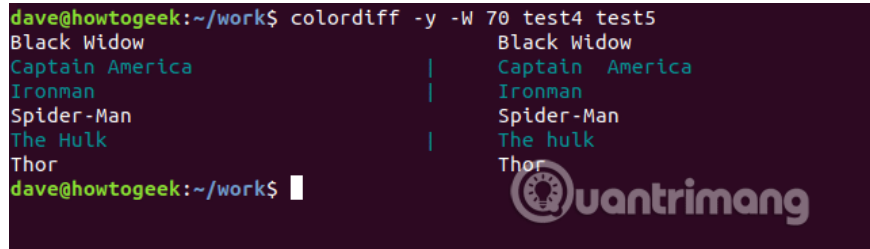


Remove white space and uppercase and lowercase letters

Try comparing two other files, test4 and test5 with 6 superheroes.

```
colordiff -y -W 70 test4 test5
```

```
dave@howtogeek:~/work$ colordiff -y -W 70 test4 test5
Black Widow
Captain America
Ironman
Spider-Man
The Hulk
Thor
dave@howtogeek:~/work$
```



The results showed that diff did not find any difference in the lines of Black Widow, Spider-Man and Thor. It marked changes in Captain America, Ironman and The Hulk.

So what are the differences in these lines? In the test5 Hulk is written in lowercase, h, and Captain America has an extra space between Captain and America. So what's in the Ironman line? You can't see with the naked eye the difference in this line on two files. Here they have white space.

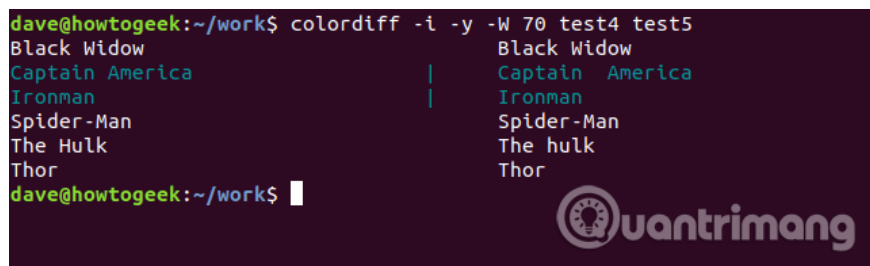
You can ignore specific differences between two files like:

1. -i: Ignore the difference in capitalization and lowercase.
2. -Z: Ignore whitespace.
3. -b: Ignore changes in the number of spaces.
4. -w: Ignore any whitespace changes.

Please ask diff to check those two files again, but this time ignore any differences in capitalization and lowercase.

```
colordiff -i -y -W 70 test4 test5
```


```
dave@howtogeek:~/work$ colordiff -i -y -W 70 test4 test5
Black Widow
Captain America
Ironman
Spider-Man
The Hulk
Thor
dave@howtogeek:~/work$
```



The Hulk and the hulk are considered the same and are not marked with lowercase h. Next try skipping whitespace with the following command:

```
colordiff -i -Z -y -W 70 test4 test5
```


```
dave@howtogeek:~/work$ colordiff -i -Z -y -W 70 test4 test5
Black Widow
Captain America | Black Widow
Ironman         | Captain America
Spider-Man      | Ironman
The Hulk       | Spider-Man
Thor           | The hulk
               | Thor
dave@howtogeek:~/work$
```



You can see the diff does not mark the different Ironman line, only the Captain America line. Now try requesting diff to ignore the difference in upper and lower case and ignore all white space issues.

```
colordiff -i -w -y -W 70 test4 test5
```

```
dave@howtogeek:~/work$ colordiff -i -w -y -W 70 test4 test5
Black Widow
Captain America
Ironman
Spider-Man
The Hulk
Thor
dave@howtogeek:~/work$
```



You can see the two files now have no differences.

The diff command has many options but mostly involves creating readable machine output. The options in the examples above will help you track the differences between the two files more easily.

I wish you all success!

You finished reading the article "**How to compare two text files on Linux Terminal**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.