The **uniq** command in Linux is a command-line utility that reports or filters out the repeated lines in a file.   
In simple words, **uniq** is the tool that helps to detect the adjacent duplicate lines and also deletes the duplicate lines. **uniq** filters out the adjacent matching lines from the input file(that is required as an argument) and writes the filtered data to the output file.

**Syntax of uniq Command :** 

//...syntax of uniq...//

**$uniq [OPTION] [INPUT[OUTPUT]]**

The syntax of this is quite easy to understand. Here, **INPUT** refers to the input file in which repeated lines need to be filtered out and if INPUT isn’t specified then **uniq** reads from the standard input. **OUTPUT** refers to the output file in which you can store the filtered output generated by **uniq** command and as in the case of INPUT if OUTPUT isn’t specified then **uniq** writes to the standard output.

Now, let’s understand the use of this with the help of an example. Suppose you have a text file named kt.txt which contains repeated lines that needs to be omitted. This can simply be done with uniq.

//displaying contents of kt.txt//

**$cat kt.txt**

I love music.

I love music.

I love music.

I love music of Kartik.

I love music of Kartik.

Thanks.

Now, as we can see that the above file contains multiple duplicate lines. Now, lets’ use uniq command to remove them: 

//...using uniq command.../

**$uniq kt.txt**

I love music.

I love music of Kartik.

Thanks.

/\* with the use of uniq all

the repeated lines are removed\*/

As you can see that we just used the name of the input file in the above uniq example and as we didn’t use any output file to store the produced output, the uniq command displayed the filtered output on the standard output with all the duplicate lines removed.

**Note: uniq** isn’t able to detect the duplicate lines unless they are adjacent to each other. The content in the file must be therefore sorted before using uniq or you can simply use **sort -u** instead of uniq command.

**Options For uniq Command:** 

1. **-c – -count :** It tells how many times a line was repeated by displaying a number as a prefix with the line.
2. **-d – -repeated :** It only prints the repeated lines and not the lines which aren’t repeated.
3. **-D – -all-repeated[=METHOD] :** It prints all duplicate lines and METHOD can be any of the following:
   * **none :** Do not delimit duplicate lines at all. This is the default.
   * **prepend :** Insert a blank line before each set of duplicated lines.
   * **separate :** Insert a blank line between each set of duplicated lines.
4. **-f N – -skip-fields(N) :** It allows you to skip N fields(a field is a group of characters, delimited by whitespace) of a line before determining the uniqueness of a line.
5. **-i – -ignore case :** By default, comparisons done are case sensitive but with this option case insensitive comparisons can be made.
6. **-s N – -skip-chars(N) :** It doesn’t compare the first N characters of each line while determining uniqueness. This is like the -f option, but it skips individual characters rather than fields.
7. **-u – -unique :** It allows you to print only unique lines.
8. **-z – -zero-terminated :** It will make a line end with 0 bytes (NULL), instead of a newline.
9. **-w N – -check-chars(N) :** It only compares N characters in a line.
10. **– – help :** It displays a help message and exit.
11. **– – version :** It displays version information and exit.

**Examples of uniq with Options**

**1. Using -c option :** It tells the number of times a line was repeated.

//using uniq with -c//

**$uniq -c kt.txt**

3 I love music.

1

2 I love music of Kartik.

1

1 Thanks.

/\*at the starting of each

line its repeated number is

displayed\*/

**2. Using -d option :** It only prints the repeated lines.

//using uniq with -d//

**$uniq -d kt.txt**

I love music.

I love music of Kartik.

/\*it only displayed one

duplicate line per group\*/

**3. Using -D option :** It also prints only duplicate lines but not one per group.

//using -D option//

**$uniq -D kt.txt**

I love music.

I love music.

I love music.

I love music of Kartik.

I love music of Kartik.

/\* all the duplicate lines

are displayed\*/

**4. Using -u option :** It prints only the unique lines.

//using -u option//

**$uniq -u kt.txt**

Thanks.

/\*only unique lines are

displayed\*/

**5. Using -f N option :** As told above, this allows the N fields to be skipped while comparing the uniqueness of the lines. This option is helpful when the lines are numbered as shown in the example below:

//displaying contents of f1.txt//

**$cat f1.txt**

1. I love music.

2. I love music.

3. I love music of Kartik.

4. I love music of Kartik.

//now using uniq with -f N option//

**$uniq -f 2 f1.txt**

1. I love music.

3. I love music of Kartik.

/\*2 is used cause we needed to

compare the lines after the

numbering 1,2.. and after dots\*/

**6. Using -s N option :** This is similar to -f N option but it skips N characters but not N fields.

//displaying content of f2.txt//

**$cat f2.txt**

#%@I love music.

^&(I love music.

\*-!@thanks.

#%@!thanks.

//now using -s N option//

**$uniq -s 3 f2.txt**

#%@I love music.

\*-!@thanks.

#%@!thanks.

/\*lines same after skipping

3 characters are filtered\*/

**7. Using -w option :** Similar to the way of skipping characters, we can also ask uniq to limit the comparison to a set number of characters. For this, -w command-line option is used.

//displaying content of f3.txt//

**$cat f3.txt**

How it is possible?

How it can be done?

How to use it?

//now using -w option//

**$uniq -w 3 f3.txt**

How

/\*as the first 3 characters

of all the 3 lines are same

that's why uniq treated all these

as duplicates and gave output

accordingly\*/

**8. Using -i option :** It is used to make the comparison case-insensitive.

//displaying contents of f4.txt//

**$cat f4.txt**

I LOVE MUSIC

i love music

THANKS

//using uniq command//

**$uniq f4.txt**

I LOVE MUSIC

i love music

THANKS

/\*the lines aren't treated

as duplicates with simple

use of uniq\*/

//now using -i option//

**$uniq -i f4.txt**

I LOVE MUSIC

THANKS

/\*now second line is removed

when -i option is used\*/

**9. Using -z option :** By default, the output uniq produces is newline terminated. However, if you want, you want to have a NULL terminated output instead (useful while dealing with uniq in scripts). This can be made possible using the -z command line option.

Syntax: 

//syntax of using uniq

with -z option//

**$uniq -z file-name**