

## How to use diff

diff is a unix program for comparing files. It expects two command-line arguments, the names of the two files you want to compare. So, for example, if you have two files, **a.txt** and **b.txt**, you can get diff to report any differences between these files by running it like this:

```
diff a.txt b.txt
```

If the two files have any differences, diff will report where they don't match, but you have to know how to read the diff output. Let's say files **a.txt** and **b.txt** contain the following:

```
a.txt
Roses are Red

Roses are red,
Violets are blue,
Sugar is sweet,
And so are you.
```

```
b.txt
Roses are Red

Roses are red,
Violets are blue,
Honey is sweet,
And so are you.
```

You can see that the 5<sup>th</sup> line is a little bit different. If you compare these files, you'll get the following output from diff. The 5c5 tells you that line 5 has changed between the two files and the next few lines show what this line looked like in a.txt (the arrow pointing left) and b.txt (the arrow pointing right).

```
diff a.txt b.txt
5c5
< Sugar is sweet,
---
> Honey is sweet,
```

In some cases, diff may explain the differences between two files as adding or removing lines (instead of just changing them). In the examples below, there's an extra line in the **b.txt** file that's not there in **a.txt**.

```
a.txt
Roses are Red

Roses are red,
Violets are blue,
Sugar is sweet,
And so are you.
```

```
b.txt
Roses are Red

Roses are red,
Violets are blue,
Sugar is sweet,
and bad for your teeth,
And so are you.
```

If you run `diff` on these two files, you'll get the following output. The `5a6` tells you a new line was added after line 5 (line 6) and then it reports what this new line looks like. `diff` uses a similar notation to show lines that are present in the first file but not in the second (but, it uses a `d` to indicate a deleted line instead of an `a` for an added line.).

```
diff a.txt b.txt
5a6
> and bad for your teeth,
```

If there's just a difference in whitespace, `diff` will still report it, but it may be non-obvious what the actual difference is. For example, I put an extra space right at the end of the first line in the `b.txt` file.

a.txt

```
Roses are Red
Roses are red,
Violets are blue,
Sugar is sweet,
And so are you.
```

b.txt

```
Roses are Red
Roses are red,
Violets are blue,
Sugar is sweet,
And so are you.
```

There's a space right here.

Running this example through `diff`, it will report that the first lines are different, but since it's just a difference in whitespace, it's hard to tell exactly what difference it's detecting.

```
diff a.txt b.txt
1c1
< Roses are Red
---
> Roses are Red
```

There's a trick you can use to help show differences like this. If you send the output of `diff` through the `cat` program, you can use the `-A` option to get `cat` to put a `$` at the end of every line and to show other invisible characters in the `diff` output. It will show a tab character as `^I` and it will show a carriage return character (part of line termination on Windows) as `^M`. For the two files above, the `$` at the end of each line makes it easier to see that there's an extra space at the end of the first line in `b.txt`.

```
diff a.txt b.txt | cat -A
1c1$
< Roses are Red$
---$
> Roses are Red $
```