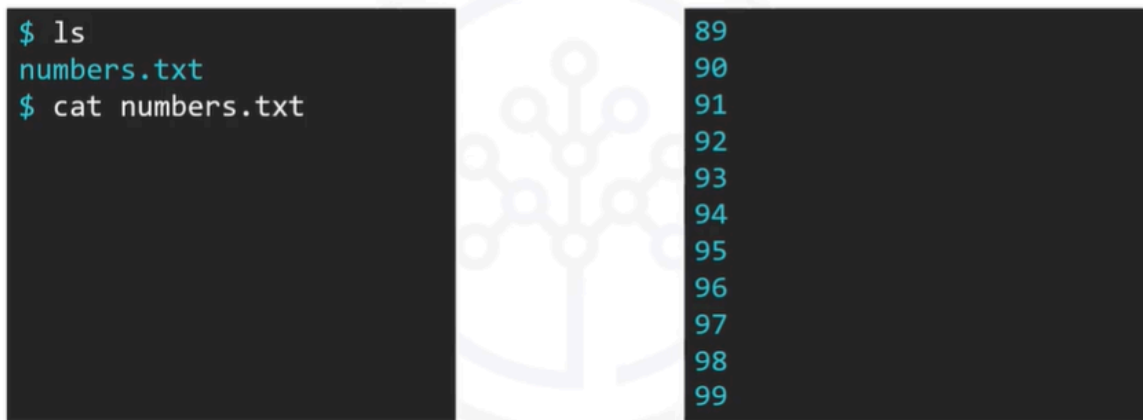


Viewing File Content

Viewing your file all at once

`cat` (catenate) – Print entire file contents

A terminal window with a dark background. On the left, the commands '\$ ls' and '\$ cat numbers.txt' are entered. On the right, the output of the cat command is shown, displaying a list of numbers from 89 to 99, one per line. A faint, light-colored logo is visible in the background of the terminal window.

```
$ ls
numbers.txt
$ cat numbers.txt
89
90
91
92
93
94
95
96
97
98
99
```

- There are several useful commands for viewing file content. To begin with, you can use the `cat` command to print the entire file to standard output. Suppose your current directory contains a single file called `numbers.txt`, which you can see by entering the `LS` command. To print the contents of this file to standard output, you can type `cat numbers.txt`, which produces the output shown here consisting of the numbers 89 through to 99. You can see that the output takes up the entire terminal window. The file is much longer than the twelve lines you see here. For this reason, you might not always want to use `cat` to view contents. Thankfully, there are alternative commands for cases such as these.

Viewing your file page-by-page

`more` – Print file contents page-by-page

```
$ more numbers.txt
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8
```

Viewing your file page-by-page

`more` – Print file contents page-by-page

```
$ more numbers.txt
```

space bar

```
9  
10  
11  
12  
13  
14  
15  
16  
17
```

Viewing your file page-by-page

`more` – Print file contents page-by-page

```
$ more numbers.txt
```

q + enter

```
$
```

- The `more` command allows you to view a file's content in a page by page format. By entering `more numbers.txt`, you see the numbers 0-8 on page 1 as shown. By page we just mean the current terminal window. If you expand your terminal window vertically, you also increase the page size. When you click the space bar, you will see the next page which displays the numbers 9-17. Entering `Q` quits the `more` program and returns you to the command prompt.

Viewing the first 10 lines

`head` – Print first 10 lines of file

```
$ head numbers.txt
0
1
2
3
4
5
6
7
8
9
```

Viewing the first N lines

`head` – Print first N lines of file

```
$ head -n 3 numbers.txt
0
1
2
```

Viewing the last 10 lines

`tail` – Print last 10 lines of file

```
$ tail numbers.txt
90
91
92
93
94
95
96
97
98
99
```

Viewing the last N lines

`tail` – Print last N lines of file

```
$ tail -n 3 numbers.txt
97
98
99
```

Counting lines, words, and characters

`WC` (word count) – Count characters, words, lines

```
$ cat pets.txt
cat
cat
cat
cat
dog
dog
cat
```

```
$ wc pets.txt
7 7 28 pets.txt
$ wc -l pets.txt
7 pets.txt
$ wc -w pets.txt
7 pets.txt
$ wc -c pets.txt
28 pets.txt
```

- You can use the `WC` command to count the number of characters, words, or lines in your file. Imagine you have a file called `pets.txt`. Entering `cat pets.txt` shows that the file contains the word `cat` or the word `dog` on each line.
- By entering `WC pets.txt`, you get the result `7 7 28 pets.txt`. Which means that your file contains 7 lines, 7 words, and 28 characters. But 7 times 3 is 21 so why does `WC` see 28 characters?
 - Because it also counts new line characters. You can't see them there, but there are seven new line characters one of which represents end of file.
- To view only line count, you can use the `-l` option which returns `7 pets.txt`. Similarly to view only word count, you can use the `-w` option, and to view only character count, you can use the `-c` option.