Open/Display Content of a File

This lesson will illustrate various ways to display content and word count of files.

wc

Definition:

`wc` (word count) is used to get word count, newline count, byte and characters count in the files specified in its input.

Syntax:

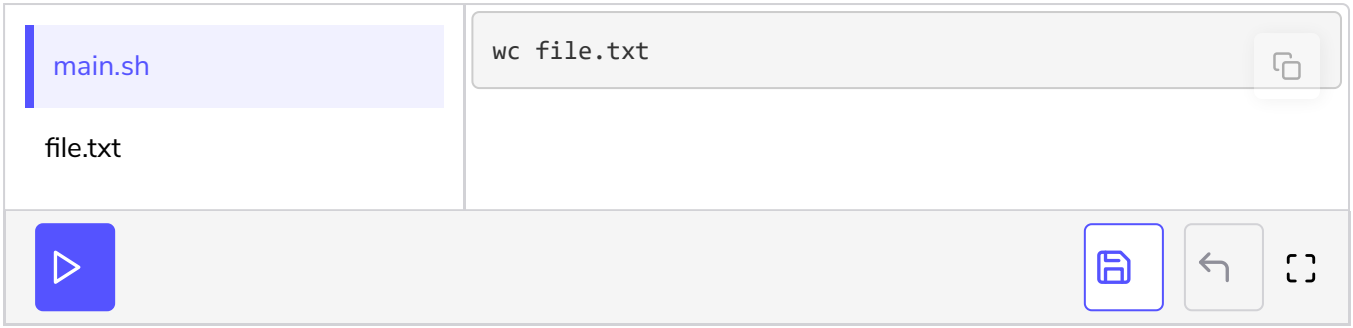
```
wc [option] [file]
```

Options:

Option	Description
-m	Print the character counts.
-c	Print the byte counts.
-l	Print the newline counts.
-w	Print the word counts.
-L	Print the length of the longest line.

Examples:

Execute the following example. Here, **6** is the number of lines, **9** is the number of words, and **65** is the number of characters.



cat

Definition:

`cat` commands — short for concatenation — can be used for multiple purpose. The three main uses of `cat` are: concatenation, read file and new file creation. Cat is considered one of the most reliable methods to open a file in a read mode and print its content.

Syntax:

Concatenation:

```
cat [option] [file_name(s)]
```

File Reading:

```
1. cat [file_name(s)] // To print contents of file
```

```
2. cat [file_name] > [file_name] // To redirect content to another file
```

```
3. cat [file_name] | less // To filter the content to be displayed with pi
ping
```

File Creation:

```
1. cat > [new_file_name]  // To create new file or overwrite if file alrea
dy exists
```

```
2. cat >> [existing_file_name] // To preserve previous file if it alread
y exists by appending any new text
```

Operators:

Operator	Meaning
`>`	*Redirection Operator*. Redirects the contents of one file to another
`>>`	*Append Operator*. Appends the content of one file at the end of the other file. Used to prevent overwritting issues.
`|`	*Piping Operator*. Pipes the content of a file if its too large to display

Options:

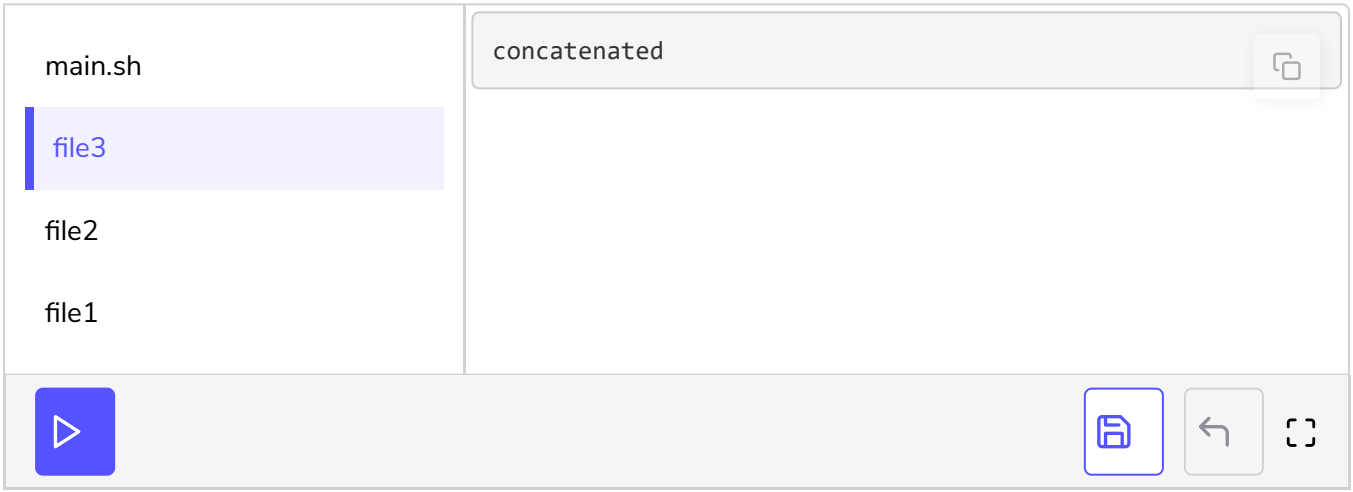
Option	Meaning
-n	Display the contents of file with line numbers
-E	Concatenate '$' at the end of each line of file
-T	Replaces `tab` as `^I`
-v	To show non-printable characters on command line
-A	Combination of v,E and T

Examples:

Examples:

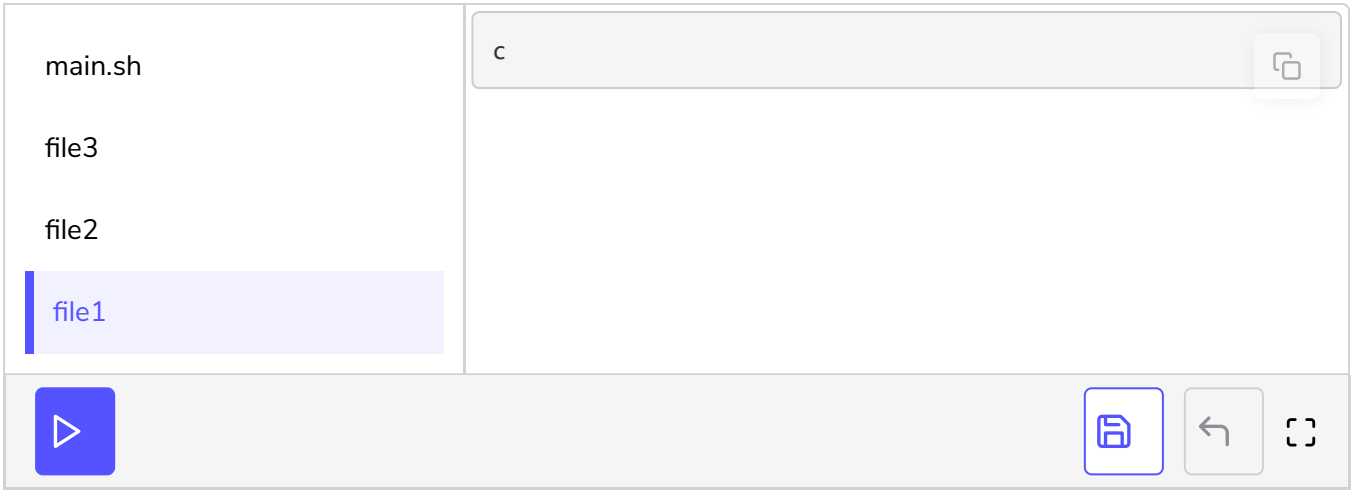
1. To concatenate the content of three files and then redirect the output to a fourth file:

```
cat file1 file2 file3 > file4
```



2. To pipe the content of three files given as arguments and then sort them in alphabetical order in a new file

```
cat file1 file2 file3 | sort > file4
```



> **Piping?**
>
> You might have heard this term on various platforms. Piping, in general, is a very broad topic itself but we would limit our discussion here by only considering its use in file manipulation domain. It's a common technique used in Linux to redirect or chain the content to another destination. The destination could be another program, another file or even an output device such as printer. This is oftenly used when the content of a file are to large so we are bound to pipe the content and

direct them somewhere else.