

cat

What it does: Shows you what's inside files or sticks them together.

How to use it: `cat [OPTION]... [FILE]...`

Examples:

1. Peek into a single file:

```
cat file.txt
```

2. Stick two files together and see the result:

```
cat file1.txt file2.txt
```

3. Stick all `.txt` files together and save to a new file:

```
cat *.txt > all_together.txt
```

tac

What it does: Like `cat`, but flips everything backwards.

How to use it: `tac [OPTION]... [FILE]...`

Examples:

1. Read from bottom to top:

```
tac file.txt
```

2. Reverse and combine two files:

```
tac file1.txt file2.txt > reversed.txt
```

3. Reverse a bunch of files and save it:

```
tac *.txt > all_reversed.txt
```

head

What it does: Snags the top part of files to show you.

How to use it: `head [OPTION]... [FILE]...`

Examples:

1. Grab the first 10 lines:

```
head file.txt
```

2. Sneak peek the first 20 lines:

```
head -n 20 file.txt
```

3. Show the first 5 lines of multiple files:

```
head -n 5 file1.txt file2.txt
```

tail

What it does: Grabs the last bit of files.

How to use it: `tail [OPTION]... [FILE]...`

Examples:

1. Show the end:

```
tail file.txt
```

2. Keep an eye on the end of a growing file:

```
tail -f file.txt
```

3. Get the last 20 lines:

```
tail -n 20 file.txt
```

cut

What it does: Chops out sections from each line.

How to use it: `cut OPTION... [FILE]...`

Examples:

1. Snip the first column:

```
cut -f1 file.txt
```

2. Get a slice of characters:

```
cut -c1-5 file.txt
```

3. Choose multiple fields:

```
cut -f1,3 file.txt
```

sort

What it does: Puts things in order.

How to use it: `sort [OPTION]... [FILE]...`

Examples:

1. Alphabetize a list:

```
sort file.txt
```

2. Order numbers properly:

```
sort -n file.txt
```

3. Reverse the order:

```
sort -r file.txt
```

WC

What it does: Counts lines, words, and bytes.

How to use it: `wc [OPTION]... [FILE]...`

Examples:

1. Just the basics:

```
wc file.txt
```

2. Count lines only:

```
wc -l file.txt
```

3. Show the longest line length:

```
wc -L file.txt
```

diff

What it does: Shows differences between files.

How to use it: `diff [OPTION]... FILES`

Examples:

1. Spot the changes:

```
diff file1.txt file2.txt
```

2. Ignore case when comparing:

```
diff -i file1.txt file2.txt
```

3. Show changes side by side:

```
diff -y file1.txt file2.txt
```

grep

What it does: Searches for specific patterns in files.

How to use it: `grep [OPTION]... PATTERN [FILE]...`

Examples:

1. Search for a string in a file:

```
grep "search term" file.txt
```

2. Count occurrences of a string in a file:

```
grep -c "search term" file.txt
```

3. Search in all `.txt` files in a directory:

```
grep "search term" *.txt
```