

# Additional file commands

Adapted from materials by Dr. Carrier



`cut` - Grab columns of delimited file

`cut` - Grab columns of delimited file

Delimited - Fields separated by a certain character

E.g., Commas, tabs, spaces

# cut - Grab columns of delimited file

Delimited - Fields separated by a certain character

E.g., Commas, tabs, spaces

Usage:

```
cut [options] [file]
```

# cut - Grab columns of delimited file

Delimited - Fields separated by a certain character

E.g., Commas, tabs, spaces

Usage:

```
cut [options] [file]
```

Important options

# cut - Grab columns of delimited file

Delimited - Fields separated by a certain character

E.g., Commas, tabs, spaces

Usage:

```
cut [options] [file]
```

Important options

```
--delimiter=X -> Set delimiter
```

# cut - Grab columns of delimited file

Delimited - Fields separated by a certain character

E.g., Commas, tabs, spaces

Usage:

```
cut [options] [file]
```

Important options

```
--delimiter=X -> Set delimiter
```

```
-f N -> Grab the Nth delimited field (start at 1)
```

# cut - Grab columns of delimited file

Delimited - Fields separated by a certain character

E.g., Commas, tabs, spaces

Usage:

```
cut [options] [file]
```

Important options

```
--delimiter=X -> Set delimiter
```

```
-f N -> Grab the Nth delimited field (start at 1)
```

```
--complement -> Ignore these fields instead
```



# cut - Grab columns of delimited file

Delimited - Fields separated by a certain character

E.g., Commas, tabs, spaces

Usage:

```
cut [options] [file]
```

Important options

`--delimiter=X` -> Set delimiter

`-f N` -> Grab the Nth delimited field (start at 1)

`--complement` -> Ignore these fields instead

Operates on stdin if no file given

tr - Translate characters

# tr - Translate characters

Usage:

```
tr [options] [set1] [set2]
```

Translates characters in set1 to those in set2

# tr - Translate characters

Usage:

```
tr [options] [set1] [set2]
```

Translates characters in set1 to those in set2

Note: Operates on stdin

# tr - Translate characters

Usage:

```
tr [options] [set1] [set2]
```

Translates characters in set1 to those in set2

Note: Operates on stdin

Replace 'a' with 'b':

```
tr a b < file
```

# tr - Translate characters

Usage:

```
tr [options] [set1] [set2]
```

Translates characters in set1 to those in set2

Note: Operates on stdin

Replace 'a' with 'b':

```
tr a b < file
```

Also has many useful sets:

```
tr [:upper:] [:lower:] < file
```

# tr - Translate characters

Usage:

```
tr [options] [set1] [set2]
```

Translates characters in set1 to those in set2

Note: Operates on stdin

Replace 'a' with 'b':

```
tr a b < file
```

Also has many useful sets:

```
tr [:upper:] [:lower:] < file
```

Can also delete characters:

```
tr -d '#' <file
```

# wc - Word count

Usage:

```
wc [file]
```



# wc - Word count

Usage:

```
wc [file]
```

Counts lines, words, bytes by default

# wc - Word count

Usage:

```
wc [file]
```

Counts lines, words, bytes by default

To narrow it to a certain field:

```
wc -l [file]
```

# wc - Word count

Usage:

```
wc [file]
```

Counts lines, words, bytes by default

To narrow it to a certain field:

```
wc -l [file]
```

**-w** for words **-b** for bytes

# wc - Word count

Usage:

```
wc [file]
```

Counts lines, words, bytes by default

To narrow it to a certain field:

```
wc -l [file]
```

**-w** for words **-b** for bytes

Operates on stdin if no file given