Bash

- Count words in a file

  ```
  wc –w filename
  ```

- Convert a text file to uppercase

  ```
  cat file | tr '[:lower:]' '[:upper:]'
  ```

- Sort lines

  ```
  sort file
  ```

- Find and replace

  ```
  sed –i 's/old/new/g' file
  ```

- List duplicate lines

  `sort file | uniq -d`
- List unique lines

  `sort file | uniq`
- Line count

  `wc -l filename`

- Shuffle a text

  ```
  shuf file
  ```

- Turn a text into a list of lines

  ```
  cat file | tr ' ' '\n'
  ```

- Given a text file, frequency analysis

  ```
  cat file | tr " " "\n" | sort | uniq -c | sort -nr | head -n5
  ```

Lets create fake image files `touch {1..10}.png`

```
for file in *.png; do
    echo "$file"
done
```

We can go through files as easily as this.

We can now mass rename!

```
for file in *.png; do
    mv "$file" "${file%.jpg}_$(date +%Y%m%d).jpg"
done
```

cut

Let's say we have a data as follows:

```
tugberk, 80
hande, 100
ahmet, 40
```

If we want to get the grades we can say:

`cat file | cut -d ", " -f2`

We can later sort it and get the highest grade.

But how can i get the name of the person with the highest grade?

`cat file | sort -t, -k2 | head -n1 | cut -d, -f1`

grep

- grep -o will give you how many times that word appears in a file.
- `grep –r pattern` searches recursively in the current dir.
- `grep –c pattern file` counts how many times pattern occurs
- `grep –n pattern file` shows lines with numbers along the matching lines
- `grep –v pattern file` shows lines that do NOT match the pattern

tr

- `tr 'a' 'b' < file` replaces all occurrences of a with b.
- `tr -d 'a'` removes all occurrences of **a**
- `tr -s ' '` squeezes multiple spaces into single.
- `tr 'a-z' 'A-Z'` case conversion
- `tr -c 'set1' 'set2'` replaces characters NOT in set1.

sed

- `sed 's/find/replace/g'`` replace all occurrences
- `sed -i` to edit inplace
- `sed '2s/find/replace` only change 2nd line
- `sed /pattern/d` deletes line matching the pattern.