KED2024-04

Expansion

Batch processing with expansion

```
touch text_{a..c}.txt
# is equivalent to
touch text_a.txt text_b.txt text_c.txt

mkdir {2000..2005}{a..c}
# is equivalent to
mkdir 2000a 2000b 2000c 2001a 2001b 2001c ...
```

Writing a runnable Script

Example script: `TODO.sh`

```
#!/bin/sh

echo "This is my homework."
```

- file with suffix `.sh`

  one command per row

  `#` precedes comments

- start script with Shebang `#!/bin/sh`
- execute with `bash SCRIPTNAME.sh`

The beauty of scripting is automation. ⚡

Assignment #1 ✍️

- get/submit via OLAT
  - starting tonight
  - deadline: 23 March 2024, 23:59
- discuss issues on OLAT forum
- ask friends for support, not solutions

KED2024-05

Lowercasing

Reduce word forms

```
echo "ÜBER" | tr "A-ZÄÖÜ" "a-zäöü"  # fold text to lowercase
```

Removing and Replacing Symbols

```
echo "3x3" | tr -d "[:digit:]"      # remove all digits
cat text.txt | tr -d "[:punct:]"    # remove punctuation like .,:;?

tr "Y" "Z"                          # replace any Y with Z
```

Standard Preprocessing

Save a preprocessed document

```
# lowercase, no punctuation, no digits
cat speech.txt | tr "A-ZÄÖÜ" "a-zäöü" | \
tr -d "[:punct:]" | tr -d "[:digit:]" > speech_clean.txt
```

Join Lines

```
cat test.txt | tr -s "\n" " "   # replace newlines with spaces
```

Trim Lines

```
cat -n text.txt          # show line numbers
sed "1,10d" text.txt     # remove lines 1 to 10
```

Check Differences between Files

sanity check after modification

```
# show differences side-by-side and only differing lines
diff -y --suppress-common-lines text_raw.txt text_proc.txt
```

Split Files

```
# splits file at every delimiter into a standalone file
csplit huge_text.txt  "/delimiter/" {*}
```

Where there is a shell,
there is a way. 

KED2024-07

Conversion of DOCX

Use case: news articles from Nexis

- `pandoc` to convert many file formats
- download as single articles in `.docx` on Nexis

```
# convert docx to txt
pandoc infile.docx -o outfile.txt

### Install first with
brew install pandoc     # macOS
sudo apt install pandoc # Ubuntu
```

Basics of Batch Processing

perform the same operation on many files

```
# loop over all txt files
for file in *.txt; do

    # indent all commands in loop with a tab

    # rename each file
    # e.g. a.txt -> new_a.txt
    mv $file new_$file

done
```

Perform OCR for many PDF

```
for FILEPATH in *.pdf; do
    # convert pdf to image
    convert -density 300 $FILEPATH -depth 8 -strip \
    -background white -alpha off temp.tiff

    # define output name (remove .pdf from input)
    OUTFILE=${FILEPATH%.pdf}

    # perform OCR on the tiff image
    tesseract -l deu temp.tiff $OUTFILE

    # remove the intermediate tiff image
    rm temp.tiff

done
```

Configure ImageMagick

Only Windows Ubuntu: Paste the following in your command-line

```
# disable security policy for Windows
sudo sed -i '/<policy domain="coder" rights="none" pattern="PDF"/d'

# increase memory limits
sudo sed -i -E 's/name="memory" value=".+"/name="memory" value="8Gi
sudo sed -i -E 's/name="map" value=".+"/name="map" value="8GiB"/g'
sudo sed -i -E 's/name="area" value=".+"/name="area" value="8GiB"/g
sudo sed -i -E 's/name="disk" value=".+"/name="disk" value="8GiB"/g
```

#LifeHack: Make a PDF searchable

Use case: scanned book chapters

```
# output searchable pdf instead of txt
convert -density 300 -depth 8 -strip -background white -alpha off -
file_in.pdf temp.tiff

tesseract -l deu temp.tiff file_out pdf
```

Conversion of digitalized PDFs

use-case: historical party programmes

1. extract image from PDF + improve contrast
2. run optical character recognition (OCR) on the image

```
# convert scanned pdf to tiff, control quality with parameters
convert -density 300 -depth 8 -strip -background white -alpha off \
infile.pdf temp.tiff

# run OCR for German ("eng" for English, "fra" for French etc.)
tesseract -l deu temp.tiff file_out

### Install first with
brew install imagemagick            # macOS
sudo apt-get install imagemagick    # Ubuntu
```