Command Line and Version Control

Miklós Koren (koren.mk)

CODEDTHINKING.
by Koren

Tools

Git Bash
https://gitforwindows.org/

Dataset
Second at https://thnk.ng/day1

Video recording
https://thnk.ng/2023-10-12

Lessons from the Command Line

1. Everything is a file.
2. Almost everything is a text file.
3. The default app is not always the best.
4. Extensions are helpful, but not binding.
5. Useful to load and work with file in other folders.
6. Organize files logically into folders and subfolders.
7. Files and folders are the same whether accessed from the command line, or via apps.

Commands covered (1)

```
pwd
cd Downloads
cd workshop
ls
ls -l
ls -l -h
ls -lh
head ted-sample.csv
head Command Line and Version Control.md
head "Command Line and Version Control.md"
head Command\ Line\ and\ Version\ Control.md
head -n 1 "Command Line and Version Control.md"
head -n 10 "Command Line and Version Control.md"
head -n101 ted-sample.csv > ted100.csv
head -n101 ted-sample.csv » ted100.csv
```

Commands covered (2)

```
ls -l *.csv
ls -l *sv
mv ted100.csv ted200.csv
cp ted200.csv ted200-backup.csv
mv ted200.csv ted200.py
rm ted200.py
rm -i ted200-backup.csv
```

Version control

We all know the problem

paper1.tex → paper2.tex → paper_final.tex → paper_final2.tex

We can have a system

paper-2023-10-12.tex

Git is such a system

`git` is a tool to save snapshots of your files in a folder.

Lessons from Version Control

1. Also useful on its own (use case: backup on a server without internet). But even better with GitHub, a collaborative website.
2. Commits are not automatic. You decide what to save and have to a give a commit message.
3. Merge conflicts are hard. But they are relatively rare.
4. Use cases for branches:
   - keep work separate to avoid intermittent conflicts
   - review someone else's work
   - experiment with big changes to code

Git commands covered

```
git init
ls -al
git status
git add ted200-backup.csv
git status
git config --global user.name "Miklos Koren"
git config --global user.email "foo@bar.com"
git commit -m "Add 200-line version of TED"
git log
head -n100 ted200-backup.csv > ted100-backup.csv
mv ted100-backup.csv ted200-backup.csv
git diff
git commit -m "Remove 102 lines from ted200-backup.csv"
git add ted200-backup.csv
git commit -m "Remove 102 lines from ted200-backup.csv"
git add "Command Line and Version Control.md"
git commit -m "Add workshop notes"
git diff
git commit -am "Add another lesson the command line part"
```