



# The ABC of Computational Text Analysis

## #4 Introduction to the Command-line

Alex Flückiger  
18 March 2021

Faculty of Humanities and Social Sciences  
University of Lucerne

# Recap last Lecture

- Successful Installation? 
- Scripting   
*automate, document, reproduce*
- Any questions?

# Outline

- learn principles of the shell 🏛️
- perform shell commands 🐎
- solving exercises 🏗️

# Starting a Shell

## macOS

- open `Terminal`
- shell type: `zsh`

## Windows

- open `Ubuntu 20.04 LTS`
- shell type: `Bash`
- ~~open Windows Command Prompt~~

# Bourne-again Shell

## Bash

- offers many built-in apps
- shell prompt

*USER@HOSTNAME : ~\$*

- home directory

*~ refers to /home/USER*

- case-sensitive
- no feedback

# Unix Philosophy

Build small programs that *do one thing*  
and *do it well.* 🧐

# Getting started in a Shell

generic components

```
command -a --long_argument FILE      # non-working example command
```

run command + help

```
echo "hello world"    # print some text  
echo --help           # get help for any command (e.g., echo)  
man echo              # get help for any command (e.g., echo)
```

# Structure of a File System

- hierarchical file system

*tree-like*

- absolute path

*/home/alex/KED2021/slides/KED2021\_01.html*

- relative path from current directory

*KED2021/slides/KED2021\_01.html*

*works across systems*

- common directories

*. current dir*

*.. parent dir*

*~ home dir*

- find your files on Windows

*/mnt/c/Users/YOUR\_USERNAME/*

*shortcut with documents*

```
.
├── data
├── materials
│   └── installation.pdf
├── README.md
└── slides
    ├── images
    │   ├── ai.jpg
    │   └── zotero.png
    ├── KED2021_01.html
    ├── KED2021_01.md
    ├── KED2021_02.html
    └── KED2021_02.md
```



# Navigating in a File System

list content

```
pwd                # show absolute path of current directory

ls                 # list content of current directory
ls -lh            # list with more information
ls dirname         # list content of directory dirname

cd ..              # change directory to go folder up
cd dir/subdir      # go to folder dir/subdir (two folders down)
```

open in file manager (GUI)

```
open .             # open path in finder (macOS)
explorer.exe .     # open Windows Explorer from WSL Ubuntu (Windows)
nautilus .         # open path in file manager (Ubuntu)
```

# Open Files


## show within Shell

```
more text.txt           # print content (space to scroll)
head text.txt           # print first 10 lines of file
tail -5 text.txt        # print last 5 lines of file
```

## show with default application (GUI)

```
open text.txt           # macOS
wslview text.txt        # WSL Ubuntu (Windows)
xdg-open text.txt       # Ubuntu
```

# Useful Key Actions

- autocompletion: `TAB`
- get last command: 
- scrolling: `SPACE`
- interrupt `CTRL + C`
- quit: `q` or `CTRL + D`

# Creating, Moving and Copying

create files and directories

```
touch test.txt      # create a new file  
mkdir data          # make a new directory
```

copy files

```
cp test.txt /other/.      # copy file, keep its name  
mv test.txt /other/new_name.txt # move or rename a file
```

# Removing Files

Watch out, there is no recycle bin. No way back!

```
rm old.txt           # remove a file  
rm -r old_data       # remove a folder with all its files
```

# Wildcards

placeholders to match ...

- any single character: `?`
- any sequence of characters: `*`

```
mv data/*.txt new_data/.    # move txt-files from to another subfolder
cp *.txt files/.            # copy all txt-files in a single folder
```

# Searching

collect certain files only

```
ls *.txt           # list all files with the suffix .txt (in current directory)
```

find specific files

```
# search on filename
find /path/to/dir -name "fname" # find a file in specific directory
locate -i pattern_1 pattern_2   # global search of files/folders

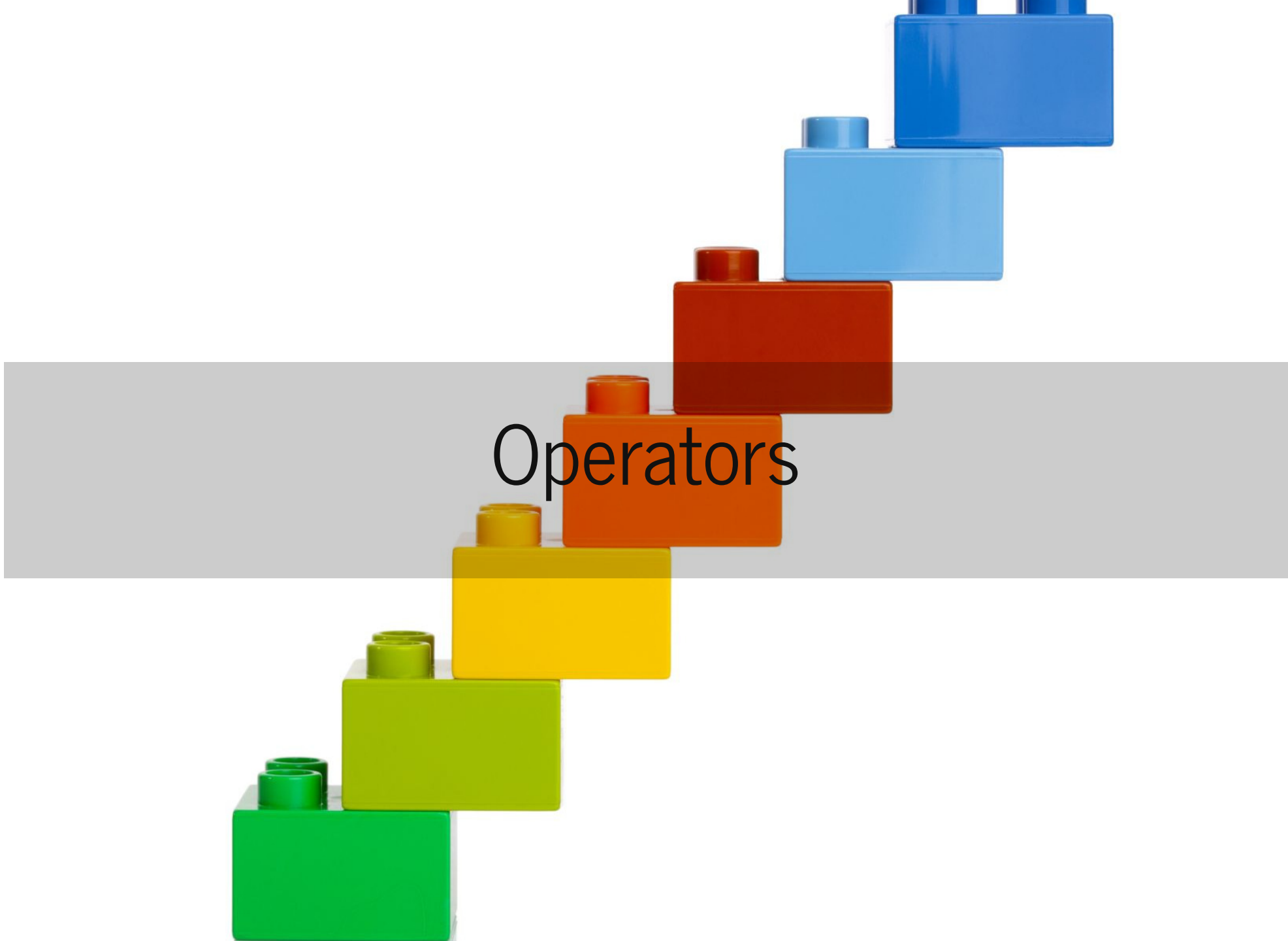
# search on content
grep -r 'x'                     # find files in any subfolder containing x
```

# 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 ...
```





# Combining Commands

shell operators to ...

- stream to next command: `|` (pipe)
- redirect into file (overwrite): `>`
- append to existing file: `>>`

```
echo 'line 1' > test.txt      # write into file
more test.txt | tail -1      # pass output to next command
```

[Learn more about operators](#)

# Merging Files

```
cat part_1.txt part_2.txt      # concatenate multiple files  
cat *.txt > all_text.txt      # merge all txt into a single one
```

# Scripting

Example script: `find_all_pdf.sh`

```
#!/bin/sh

echo "This is a list of all PDFs on my computer:"
locate -i /home/*.pdf
```

- file with one command per row
- start script with Shebang

`#!/bin/sh`

- execute with

`bash SCRIPTNAME.sh`

# Conventions 🙏

- no spaces/umlauts in names  
*alphanumeric, underscore, hyphen, dot*
- files have a suffix, folders not  
*text\_1.txt vs. texts*
- descriptive file organization  
*SOURCE/YEAR/speech\_party\_X.txt*
- separate data from scripts
- never change the raw data

# Organizing Code

- Git to track file changes
- GitHub hosting platform

Get course repository

```
# get an initial copy of the course material  
git clone https://github.com/aflueckiger/KED2021.git  
  
# update your local copy continuously  
cd KED2021  
git pull
```





Questions?

# Assignment #1

- get/submit via OLAT  
*starting tonight*  
*deadline: 26 March 2020, 23:59*
- discuss issues on OLAT forum
- ask friends for support, not solutions



# In-class: Exercises I

1. If you have not cloned the course repository from Github yet, do this now.
2. Create a new directory called `tmp` in the course directory `KED2021`.
3. Check out the `touch` command. The `man` command is your friend.
4. Use `touch` to create a new file called `advice_for_programmers.txt` in `tmp`.
5. Write the following content into that file, one line at a time using operators:

```
How about making programming a little more accessible? Like:  
from human_knowledge import solution
```

6. Make sure that the content was written into that file with `more`.

# In-class: Exercises II

1. Navigate up and down in in your filesystem using `cd` and list the respective files per directory with `ls`. Where can you find your personal documents? Print the absolute path with `pwd`.  
A hint to Windows users as they are working in a Ubuntu subsystem, have a look at: `/mnt/c/Users`
2. Read `man ls` and write an `ls` command that lists your documents ordered  
*by recency (time)*  
*by size*
3. Use the `|` and `>` operators to write the 3 “last modified” files in your documents folder into a file called `last-modified.txt` on your desktop (desktop is also a directory). It is a single command performing multiple operations, one after another.

# Additional Resources

useful primers on Bash

- [The Programming Historian](#)
- [DigitalOcean](#)