

# Introduction to Digital History

Ina Serif



# Table of contents

<b>Welcome</b>	<b>1</b>
<b>1 Was ist digital history?</b>	<b>3</b>
<b>2 shell</b>	<b>5</b>
<b>Appendices</b>	<b>7</b>
<b>A Appendix</b>	<b>9</b>
<b>B Further Ressources</b>	<b>11</b>



# Welcome

Der vorliegende Guide begleitet die Einführungskurse im Fach Geschichte an der Universität Basel und soll einen ersten Einblick in den Bereich digital history geben. Die epochen- und areaspezifischen Inhalte der verschiedenen Einführungskurse sollen dabei berücksichtigt werden, die Verweise auf verschiedene digital-history-Projekte aus unterschiedlichen Epochen mit der Zeit also anwachsen.

Der Guide wird für die Teilnehmer:innen der Einführungskurse von einer Präsenzsitzung begleitet, bietet aber hoffentlich auch unabhängig davon einen Mehrwert. Für Kommentare, Anregungen oder Beschwerden freue ich mich über eine Nachricht.

Der Guide ist in zwei Teile gegliedert: Die Kapitel 1–5 sollen eine erste Übersicht über “digital history” bieten und den Blick auf Neuerungen und Veränderungen richten, die sich in den Geschichtswissenschaften aus der Nutzung digitaler Methoden ergeben. Der anschließende Praxisteil zeigt an einem konkreten Beispiel die Anwendung verschiedener Techniken auf, die sich (nicht nur) für Historiker:innen bei der Arbeit mit Quellenmaterial anbieten. Der Praxisteil verfolgt dabei zwei Ziele: Zum einen sollen Hemmungen bei der Arbeit mit dem Computer, die über die Nutzung als elektronische Schreibmaschine hinausgeht, abgebaut werden. Zum zweiten sollen ein grundlegendes Verständnis dafür hergestellt werden, welche Möglichkeiten computergestützte Analysen bieten und wie diese in der historischen Arbeit eingesetzt werden können.

Die Übersicht soll möglichst knapp gehalten werden – es gibt zahlreiche ausführliche Grundlagenwerke, weswegen viele Themen nur kurz angeschnitten, dafür aber mit weiterführenden Verweisen versehen werden. Dasselbe gilt für den Praxisteil: Weiterreichende Anleitungen, Tutorials oder Onlinekurse werden an entsprechender Stelle verlinkt. Vollständigkeit wird an keiner Stelle beansprucht; Hinweise auf weitere Online-Angebote nehme ich gerne auf.



# Chapter 1

## Was ist digital history?

Über die Antwort zu dieser Frage kann man ausgiebig diskutieren; als Teilgebiet der digital humanities, der digitalen Geisteswissenschaften, kann die folgende aktuelle und pragmatische Definition von Blaney et al. hilfreich sein:

Digital humanities, in our view, is a question of approach: if you are actively and critically using digital tools to aid your work in researching, teaching or learning, you are probably doing digital humanities. We would encourage anyone to learn to program if they are interested in doing so, but we do not see it as a defining characteristic of work in digital humanities.<sup>1</sup>

---

<sup>1</sup>Blaney, Jonathan; Winters, Jane; Milligan, Sarah u. a.: Doing digital history: a beginner's guide to working with text as data, Manchester 2021 (IHR research guides), S. 6.





# Chapter 2

## shell

Shell 101

Es gibt zwei Arten, um mit dem Computer zu interagieren bzw. ihn zu nutzen: über ein **G**raphical **U**ser **I**nterface (GUI) oder, etwas direkter, über die Kommandozeile<sup>1</sup>. Um eine Datei “Bild1.jpg” im Ordner “Bilder” zu löschen, öffnet man den Explorer (Windows) oder den Finder (Mac), klickt sich zum Ordner “Bilder”, macht einen Rechtsklick auf die zu löschende Datei “Bild1.jpg”, klickt “In den Papierkorb legen” oder zieht die Datei mit der Maus direkt dorthin. Man kann dieselbe Aktion als Kommando eintippen: Man öffnet eine Powershell (Windows) oder das Terminal (Mac), navigiert mit Texteingabe zum entsprechenden Ordner, bspw. `cd Dokumente/Bilder` und gibt dort das Kommando `rm "Datei.jpg"` ein, das mit der Entertaste ausgeführt wird.

```
(base) serina00@dg-19-mac-02 Bilder % rm "Datei.jpg"
```

Die beiden Vorgehensweisen unterscheiden sich dabei in drei Punkten:

1. Das Kommando `rm` ist endgültig, die Datei ist ohne Übergangszeit im Papierkorb gelöscht.
2. Das Kommando lässt sich relativ simpel auf eine Vielzahl von Dokumenten anwenden, wobei ganz unterschiedliche Bedingungen beachtet werden können, und es lässt sich mit anderen Kommandos verbinden.
3. Terminal sieht k3wl aus.

Bevor wir den zweiten – und für unsere Arbeit hilfreichsten – Unterschied genauer anschauen, kurz zur Kommandozeile:

---

<sup>1</sup>Kommandozeile/command line, bash, shell, prompt finden sich oft als synonym genutzte Begriffe für Command Line Interfaces. Auf UNIX-basierten Betriebssystemen wie Mac OS und Linux ist das Terminal als Interface weit verbreitet; für Details: [https://en.wikipedia.org/wiki/Command-line\\_interface#History](https://en.wikipedia.org/wiki/Command-line_interface#History). Windows-nutzer:innen kommen mit der Powershell ganz gut zurecht, es empfiehlt sich eventuell die Installation von Cygwin oder MinGW, um mit einem UNIX-basierten Interface arbeiten zu können.

In einem Terminal/Shell können Befehle/Programme ausgeführt werden, die auf der Datenstrukturebene stattfinden – wie beispielsweise das Löschen einer Datei, `rm Dateiname.xyz`, oder das Erstellen eines Ordners, `mkdir NeuerOrdner`. Oder aber Operationen auf Dateninhaltsebene – wie beispielsweise das Suchen eines bestimmten Begriffs in einer Textdatei, `grep Begriff Textdatei.txt`, oder das Auszählen mehrerer Begriffe und das Speichern des Ergebnisses in einer neuen Datei, `grep -Ec "Begriff1|Begriff2" Textdatei.txt > Ergebnisse.txt`.

Woher weiss Ihre Shell aber, was sie ausführen soll, wenn Sie `rm` oder `grep` eintippen? Es gibt zahlreiche Shell-Programme, die bereits auf Ihrem System vorinstalliert sind, und mit denen Sie vieles tun können – öffnen Sie Ihre Shell und tippen Sie `date` ein: Das aktuelle Datum mit Uhrzeit erscheint. Ihre Shell sucht nach dem ersten Argument, dem Befehl, im Filesystem des Computers, und wenn sie fündig wird, führt sie eine Aktion mit den entsprechenden Parametern aus.

tmi: Wenn Sie `echo $PATH` im Terminal eingeben, sehen Sie eine Auflistung der Orte, an denen nach Befehlen gesucht wird. Tippen Sie `which date` ein und drücken Sie ‘Enter’, um zu sehen, wo das Programm “date” in Ihrem Computer liegt.

Wenn Sie einen Befehl eintippen, den es nicht gibt bzw. für den es noch kein installiertes Programm auf Ihrem Computer gibt, bekommen Sie eine simple Fehlermeldung:

```
(base) serina00@dg-19-mac-02 EK-dh % nonsense
command not found: nonsense
```

Das aktuelle Datum wird Ihnen wahrscheinlich auch in Ihrer Toolbar angezeigt, und einen neuen Ordner können Sie per Rechtsklick erstellen, dazu brauchen Sie das Terminal nicht unbedingt. Um einen Begriff in einem Textdokument zu finden und alle Vorkommen zu zählen, können Sie das Dokument öffnen, `Strg-F` drücken, den Begriff eingeben und das Ergebnis sehen. Wenn Sie nach mehreren Begriffen suchen wollen, müssen Sie dieselbe Aktion zweimal ausführen: `Strg-F`, Begriff 2.

Wenn Sie wissen möchten, wie häufig Arthur Dent, Zaphod Beeblebrox, Slartibartfast und Mrs Enid Kapelsen in den sechs Bänden von “The Hitchhiker’s Guide to the Galaxy” genannt werden, müssen Sie, wenn Sie den Volltext heruntergeladen haben, für jeden Namen eine Suche ausführen, mit `Strg-F`. Bei der Suche nach Personen mit Vor- und Nachnamen wie “Arthur Dent” suchen Sie nach “Arthur”, nach “Dent” und nach “Arthur Dent” und ziehen alle “Arthur Dent”-Treffer von den übrigen Treffern ab, um nichts doppelt zu zählen und ihre Suchergebnisse zu verfälschen. Die Trefferzahl all Ihrer Suchanfragen schreiben Sie in ein neues Dokument.

Sie können dasselbe auch mit dem Terminal machen und einige der Built-in-Programme nutzen: Sie bewegen sich mit `cd`, `change directory`, in den Ordner

(directory), in dem Ihre Textdatei liegt:

```
(base) serina00@dg-19-mac-02 serina00 % cd Documents/progr/EK_dh/Hitchhiker
```

(Um zu prüfen, was dort liegt, können Sie im Terminal `ls` (für `list`) eingeben, bzw. in der Powershell `dir` (für `directory`))

```
(base) serina00@dg-19-mac-02 Hitchhiker % ls hitchhiker_fulltext
```

Mit einer Zeile können Sie die in einem Texteditor ausgeführten Suchvorgänge mit `grep` (Global Regular Expression Print) ausführen und die Ergebnisse mit `>` in eine neue Datei schreiben:



## Appendix A

# Appendix

---

API	<b>A</b> pplication <b>P</b> rogramming <b>I</b> nterface: a facility offered by a web resource which allows search queries independent of a <b>GUI</b> , often performed using scripts
bash	default program that runs in the <b>command line</b>
big data	huge amount of data, identifiable through repeated freezing of your standard program when opening a file
born digital data	data which originated in a digital form
CLI	<b>C</b> ommand <b>L</b> ine <b>I</b> nterface, text interface that allows interaction with the computer; see also <b>bash</b>
CMS	<b>C</b> ontent <b>M</b> anagement <b>S</b> ystem
Console	See <b>CLI</b>
Crowdsourcing	projects that include the active participation of the public to generate content, transcribe sources etc.
csv	<b>c</b> omma <b>s</b> eparated <b>v</b> alues, a structured text format, using commas as separators between columns
distant reading	quantitative approach to huge amounts of texts, using computational methods to search for interpretable patterns
GUI	<b>G</b> raphical <b>U</b> ser <b>I</b> nterface
HTML	<b>H</b> ypertext <b>M</b> arkup <b>L</b> anguage, a structured text format, like the format this guide is written in, to render documents in a browser
Jupyter notebook	web application/interactive coding environment that runs in a browser; let's you create and share code ( <a href="https://jupyter.org">https://jupyter.org</a> )
machine readable	transformation of, for example, text into a data format that is processable by a computer
OCR	<b>O</b> ptical <b>C</b> haracter <b>R</b> ecognition, process of transforming text on an image into a data format

---

OS	<b>O</b> perating <b>S</b> ystem
OSS	<b>O</b> pen <b>S</b> ource <b>S</b> oftware
Regular Expression terminal	syntax for search and replace text using patterns (instead of exact matches) See <b>CLI</b>

---

## Appendix B

### Further Ressources

- MIT Computer Science Department: 1-hour-lecture on the Shell (video)
- Programming Historian: Introduction to the Bash Command Line (self-learning lesson)
- Programming Historian: Introduction to the PowerShell (self-learning lesson)
- datacamp course:Introduction to Shell (interactive self-learning lesson)
- Jeroen Janssens: Data Science at the command line (book)

Blaney, Jonathan; Winters, Jane; Milligan, Sarah u. a.: Doing digital history: a beginner's guide to working with text as data, Manchester 2021 (IHR research guides).

