Digital Humanities und historisch-biografische Forschung

Im Geschichtskurs (33438) des Moduls ›Digital Humanities‹ werden fachtypische Quellengattungen mit verschiedenen digitalen Verfahren im Hinblick auf ihren Nutzen für die historische Biografieforschung untersucht. Es werden zum einen Grundkenntnisse in einschlägigen Computerprogrammen vermittelt, zum anderen sollen einfache Computerprogramme selbst geschrieben werden, um die digitale Quellen- und Methodenkritik zu internalisieren. Ebenfalls steht die kritische Reflexion über den Nutzen und Mehrwert von digitalen Verfahren für die historische Forschung im Vordergrund. In dieser Leseprobe erhalten Sie einen kleinen Einblick in die Kursinhalte des Geschichtskurses.

1. Die Digitalität des Biographischen

Um ein Grundverständnis für die Anwendung von digitalen Methoden in der historischen Biografieforschung zu erhalten, werden im ersten Kurskapitel zunächst die theoretischen Grundlagen erarbeitet.

Ausschnitt: Einleitung

Medien machen Kommunikation möglich und beeinflussen (steuern) kommunikative Prozesse. Diese Grundeinsicht, welche der kanadische Geisteswissenschaftler Herbert Marshall McLuhan auf die ein wenig überpointierte und vielfach kolportierte Formel „the Medium is the message“[1] brachte, lässt es sinnvoll und notwendig erscheinen, nach der Interdependenz von digitalem Format und biographischer Darstellung zu fragen. Dabei ist davon auszugehen, dass biographische Äußerungen stets in einem mehr oder weniger ausgeprägten kommunikativen Kontext erfolgen und dies für das digitale Medium ebenso gilt wie für das Buch, den Film oder das Interview. Es sollte nicht von vornherein zwischen biographischen und autobiographischen Darlegungen unterschieden werden, da auch letztere in aller Regel an Kommunikationsprozesse gebunden sind, indem sie die darin enthaltenen Werte und Narrative aufnehmen und verarbeiten bzw. danach streben, eigene Deutungsangebote zu etablieren.

Betrachten wir die Darstellungsformen von Biographischem im World Wide Web, fällt zunächst deren Formenreichtum und Vielschichtigkeit auf – ganz im Sinne des Netzes als eines „Emblems der Vielheiten“.[2] So stellt sich die internetbasierte Kommunikation

über biographische Themen, beispielsweise im Chat, Web- oder Videoblog, höchst unterschiedlich dar und bringt vielerlei getrennte und vermischte Erzählweisen hervor, die wiederum jeweils eigene kommunikative und sozialpsychologische Komponenten beinhalten.[3]

Hypertexte weisen als HTML-Dokumente spezifische Eigenschaften auf. Sie sind über Links miteinander verknüpft;[4] zugleich werden über diese Links unterschiedliche semiotische Einheiten wie Bild-, Audio- oder Videodateien verbunden.[5] Es ist zu unterscheiden zwischen geschlossenem und offenem Hypertext.[6] Während der geschlossene Hypertext das Internet vorrangig als Informationsmedium nutzt, erlaubt die offene Version den Eingriff der Nutzenden und lässt ein „dynamisches Gebilde entstehen".[7] Mittels der Einbettung von Hypertexten und Hypermediaprodukten in Hypertextnetze werden zudem Interaktivität und „Wreading" – eine Verschmelzung von „Lesen" und „Schreiben" – ebenso wie sprachliche und inhaltliche Vielfalt gefördert und Distanzkommunikation ermöglicht.[8] Weitere Differenzierungsmerkmale umfassen u. a. die Perspektive des Verfassenden, die vorgesehene Orientierung an den Nutzenden (einschließlich Verlinkungsart, Serviceleistungen und interaktive Angebote), die modulare Zuordnung, die narrative Strukturierung sowie die syntaktische Gestaltung.[9]

Die Präsentation biographischer Daten und ihre Verflechtung mit Hypertexten folgt meist der konventionellen Erwartungshaltung einer linearen Darstellung von Phasen im Sinne eines „sozial standardisierten Regelsystem[s] sinnvoller Sukzession",[10] welche nicht nur eine bestimmte thematische Reihenfolge bestimmt, sondern auch den Lebensweg vor allem bekannter Persönlichkeiten lexikalisch stereotypisiert. Biographische Hypertexte mit Dokumentarfunktion werden unterschiedlich aufbereitet. Die einfachste Darstellung präsentiert eine Auflistung mit reduzierter Syntax, die sich an konventionellen Formulierungsmustern orientiert, einem Laufbahnmodell folgt und somit als geschlossener, mehr oder wenig stark sequentierter Hypertext – vorwiegend nur spärlich verlinkt – klassifiziert wird.[11] In dieser Form gestaltete Lebensläufe bieten eine erste Orientierungsmöglichkeit und sind in Internetarchiven und entsprechenden Datenbanken, beispielsweise des Deutschen Historischen Museums,[12] zugänglich. Gleichzeitig kann ein spezieller Fokus die Lebensläufe thematisch eingrenzen, wie es im Rahmen des im Jahr 2003 abgeschlossenen Projekts „Biographien Dokumentation der Forschungsstelle Kulturimpuls"[13], das Personen vorstellt, welche die Geschichte der Anthroposophie im 20. Jahrhundert geprägt hatten, getan wird. **[…]**

2. MAXQDA: Digitale, inhaltliche Erschließung von Interview-Transkripten

Im Kapitel ›Interviews und MAXQDA‹ wird die Anwendung des Programms MAXQDA eingeübt, dass die inhaltliche und strukturelle Analyse von lebensgeschichtlichen Interviews ermöglicht.

Ausschnitt: Einführung

Das Institut für Geschichte und Biographie der Fernuniversität in Hagen verfügt im Archiv *Deutsches Gedächtnis* über einen Bestand von über 3 000 Erinnerungsinterviews, die als Videos oder als Tondokumente der Forschung zugänglich sind für lebensgeschichtliche oder erfahrungsgeschichtliche Projekte.

Ein Teil dieser Erinnerungsinterviews ist bereits transkribiert, ein großer Teil noch nicht. Wie wichtig es innerhalb eines Forschungsprojektes ist, mit (vollständigen) Texttranskriptionen zu arbeiten, darin gehen die Meinungen auseinander. Manche sind der Ansicht, dass vollständige Transkriptionen nur in begründeten Fällen unternommen werden sollten, etwa bei besonders aussagekräftigen Interviews. Für die anderen empfehlen sie, die wörtliche Transkription auf zentrale Passagen zu beschränken und ansonsten ausführliche Paraphrasen oder Zusammenfassungen anzufertigen, die über die angesprochenen Inhalte Auskunft geben.[1] Damit ist gemeint, dass die Abschnitte des Interviews, die nicht als zentral empfunden werden, überblicksartig schriftlich zusammengefasst werden, wobei immer ein Verweis auf die entsprechenden Stellen (Timecodes) im Audiofile erfolgt. So kann die Bearbeitenden diese ansteuern, wenn sie die in der Inhaltsangabe angesprochenen Themen als für die Forschungsfragen relevant erachten.

Neben dem zeitlichen Aspekt (für die Verschriftlichung von einer Stunde Interviewgespräch wird eine Transkriptionszeit von vier bis sechs Stunden veranschlagt)[2], wird für eine zeitsparende Teiltranskription auch das Argument ins Feld geführt, dass es ohnehin immer notwendig sei, die Tonpassagen zusätzlich zum Transkript anzuhören, um einen vollständigeren Eindruck über den Aussagegehalt der Interviewausschnitte zu gewinnen. Es genügt also nicht, sich lediglich auf den transkribierten Text zu stützen. Gleichwohl ist der transkribierte Text diejenige Form von Erinnerungsinterviews, mit der sich am einfachsten arbeiten lässt.[3] In dem Forschungsprojekt zur Geschichte der sowjetischen Sonderlager, von dem im Folgenden noch die Rede sein wird, wurden 75 % der rund 40 geführten Interviews transkribiert.[4]

Wie genau solche Transkriptionen erfolgen, kann also von Projekt zu Projekt unterschiedlich gehandhabt werden. In diesem Beitrag soll es um den Umgang mit bereits transkribierten Interviews gehen, also den wortwörtlichen Verschriftlichungen der Tonaufnahmen von Erinnerungsinterviews. Dazu, wie die Erinnerungsinterviews als *Audiofiles***[5]** weiter bearbeitet werden können, wird Ihnen der Beitrag von Almut Leh Hinweise geben.

In unserem Beitrag sollen Möglichkeiten aufgezeigt werden, vollständig transkribierte Interviewtexte zu erschließen, Sinnabschnitte zu markieren und zu annotieren und damit Schritt für Schritt Vorbereitungen für eine Thesenbildung zu treffen. Auch bei einer kleineren Anzahl von Interviews steht man vor der Frage, wie das erhobene Material – in der Regel lebensgeschichtliche, narrative Interviews – erschlossen werden soll.

Wenn man überlegt, dass solch ein Interview im Schnitt zwischen zwei und fünf Stunden (reine Sprechzeit) dauern kann, dann wird deutlich, dass hier unter Umständen über 100 Textseiten pro Interview zusammenkommen können. Wenn man darüber hinaus von einem Sample von 30–35 Interviews ausgeht, dann wird deutlich, wie viel Quellenmaterial sich hier ansammeln kann, das zugänglich gemacht werden muss: Welche Themen werden an welchen Stellen angesprochen? Wo werden welche Deutungen von den InterviewpartnerInnen transportiert? Welche Zusammenhänge werden hergestellt? **[…]**

Ausschnitt: Strukturierung von lebensgeschichtlichen Interviews

MAXQDA ist ein Programm zur digitalen Erfassung, Strukturierung und Kategorisierung von Transkripten und Audiodateien, die über vielfältige Formen der Visualisierung, der Annotationen und der systematischen Suche verfügen.

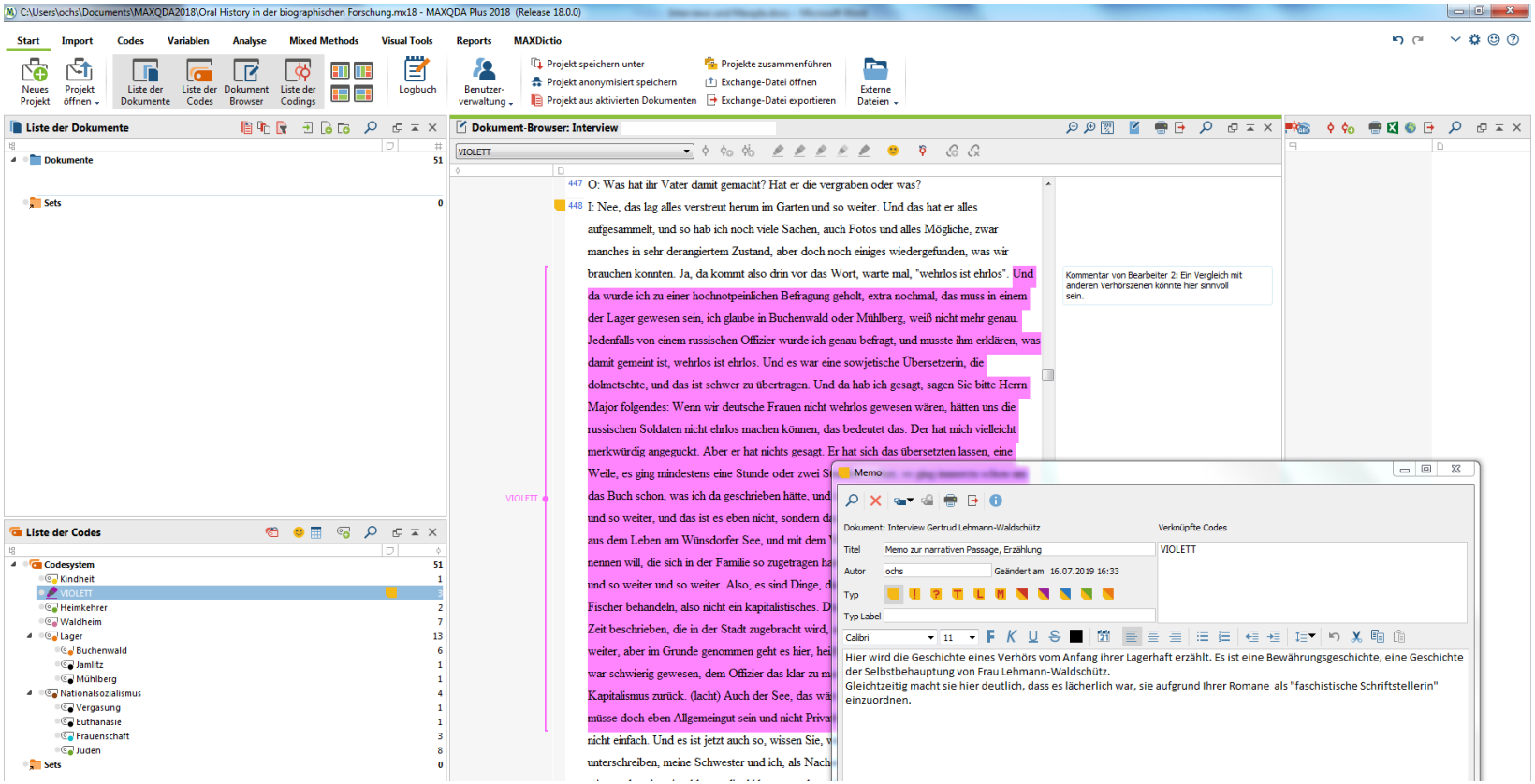
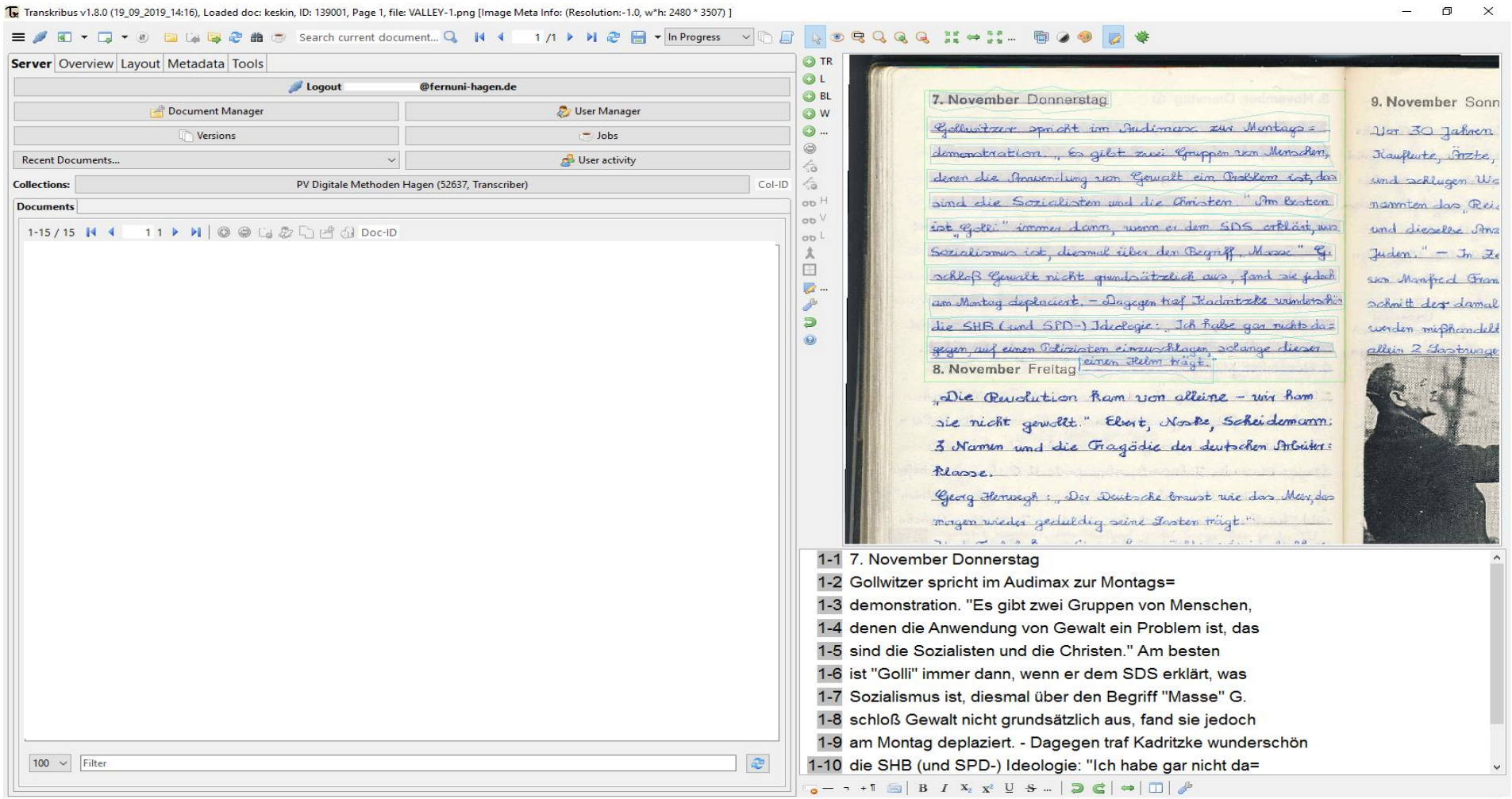


Abb. 1

Die **Memofunktion (**der kleine gelbe Merkzettel) ermöglicht uns noch **Annotationen**, die mit diesen (oder anderen) Textpassagen verknüpft werden und in denen Zusammenhänge näher erläutert und/oder erste Deutungen vorgenommen werden können. Rechts neben der Textspalte gibt es noch eine **Kommentarspalte**, in der nur kurze Bemerkungen festgehalten werden können. Sie ist auf Anhieb sichtbar und kann u. a. zur Verständigung bei der gemeinsamen Bearbeitung eines Transkriptes dienen. **[…]**

3. Transkribus: Handschriftenerkennung und Transkription

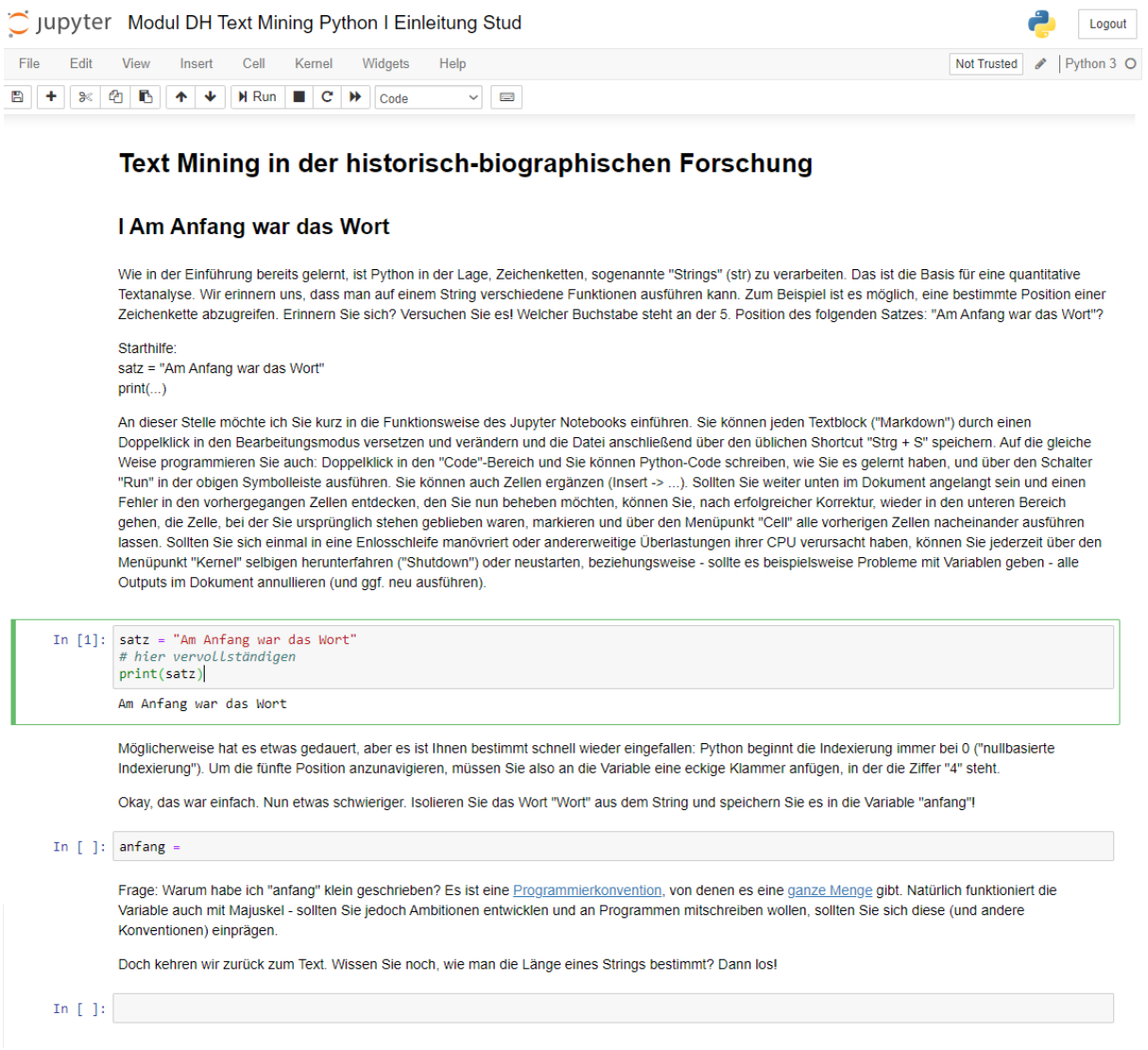
Ein Teil des Kurses widmet sich der digitalen Aufbereitung von historischen Quellen.Mit Transkribus werden Ausschnitte von historischen Quellen mithilfe von bereitgestellten Datenbanken zur Texterkennung und digitalen Wörterbüchern transkribiert. Somit wird durch eigene Anwendung erlernt, wie digitale Werkezeuge und Methoden die Geschichtsforschung unterstützen können und welche Zugänge zu historischen Quellen sich dadurch eröffnen.



4. Text-Mining: Quantitative Untersuchungsverfahren

Im Kapitel zum Text-Mining wird nähergebracht, wie quantitative Verfahren qualitative Forschungsansätze der Historiographie unterstützen oder sogar erst möglich machen. Hierfür lernen Sie die Grundlagen der Programmiersprache ›Python‹. Verwendet wird dazu die interaktive Entwicklungsumgebung von Jupyter-Notebook.

Ausschnitt: Text-Mining mit Python (Kapitel I.)



jupyter Modul DH Text Mining Python I Einleitung Stud

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

Text Mining in der historisch-biographischen Forschung

I Am Anfang war das Wort

Wie in der Einführung bereits gelernt, ist Python in der Lage, Zeichenketten, sogenannte "Strings" (str) zu verarbeiten. Das ist die Basis für eine quantitative Textanalyse. Wir erinnern uns, dass man auf einem String verschiedene Funktionen ausführen kann. Zum Beispiel ist es möglich, eine bestimmte Position einer Zeichenkette abzugreifen. Erinnern Sie sich? Versuchen Sie es! Welcher Buchstabe steht an der 5. Position des folgenden Satzes: "Am Anfang war das Wort"?

Starthilfe:
satz = "Am Anfang war das Wort"
print(...)

An dieser Stelle möchte ich Sie kurz in die Funktionsweise des Jupyter Notebooks einführen. Sie können jeden Textblock ("Markdown") durch einen Doppelklick in den Bearbeitungsmodus versetzen und verändern und die Datei anschließend über den üblichen Shortcut "Strg + S" speichern. Auf die gleiche Weise programmieren Sie auch: Doppelklick in den "Code"-Bereich und Sie können Python-Code schreiben, wie Sie es gelernt haben, und über den Schalter "Run" in der obigen Symbolleiste ausführen. Sie können auch Zellen ergänzen (Insert -> ...). Sollten Sie weiter unten im Dokument angelangt sein und einen Fehler in den vorhergegangen Zellen entdecken, den Sie nun beheben möchten, können Sie, nach erfolgreicher Korrektur, wieder in den unteren Bereich gehen, die Zelle, bei der Sie ursprünglich stehen geblieben waren, markieren und über den Menüpunkt "Cell" alle vorherigen Zellen nacheinander ausführen lassen. Sollten Sie sich einmal in eine Enlosschleife manövriert oder andererweitige Überlastungen ihrer CPU verursacht haben, können Sie jederzeit über den Menüpunkt "Kernel" selbigen herunterfahren ("Shutdown") oder neustarten, beziehungsweise - sollte es beispielsweise Probleme mit Variablen geben - alle Outputs im Dokument annullieren (und ggf. neu ausführen).

In [1]:
```
satz = "Am Anfang war das Wort"
# hier vervollständigen
print(satz)
```

```
Am Anfang war das Wort
```

Möglicherweise hat es etwas gedauert, aber es ist Ihnen bestimmt schnell wieder eingefallen: Python beginnt die Indexierung immer bei 0 ("nullbasierte Indexierung"). Um die fünfte Position anzunavigieren, müssen Sie also an die Variable eine eckige Klammer anfügen, in der die Ziffer "4" steht.

Okay, das war einfach. Nun etwas schwieriger. Isolieren Sie das Wort "Wort" aus dem String und speichern Sie es in die Variable "anfang"!

In [ ]:
```
anfang =
```

Frage: Warum habe ich "anfang" klein geschrieben? Es ist eine Programmierkonvention, von denen es eine ganze Menge gibt. Natürlich funktioniert die Variable auch mit Majuskel - sollten Sie jedoch Ambitionen entwickeln und an Programmen mitschreiben wollen, sollten Sie sich diese (und andere Konventionen) einprägen.

Doch kehren wir zurück zum Text. Wissen Sie noch, wie man die Länge eines Strings bestimmt? Dann los!

In [ ]:
```
```

Text Mining in der historisch-biographischen Forschung

II Auch digitale Mühlen mahlen langsam: das Preprocessing

Um Python zu füttern, müssen wir eine Textdatei einlesen. Diese wird uns von Transkribus in der Unicodekodierung "UTF-16" ausgegeben [siehe Grundlagen der Textformatierung im Kurs der Literaturwissenschaftler]. Eine entsprechend vorbereitete Datei finden Sie im Ordner, in dem sich auch die Jupyter Notebooks befinden, deshalb können Sie ohne weitere Pfadangabe die Funktion "open" zum Öffnen der Datei "tagebuch UTF-16BE.txt" nutzen. Sollte die Datei an einem anderen Ort liegen, suchen Sie diese und kopieren Sie sich die Pfadangabe in die Zwischenablage (Rechtsklick, Eigenschaften -> Ort). Nach folgenden Schemata wenden Sie die Funktion an:

In [9]:
```
# im selben Verzeichnis wie das Programm:
# open('Datei.txt', mode='r', encoding ='utf-16').read()
# in einem anderen Verzeichnis:
# text = open('C:\\hier\\der\\Pfad\\zur\\Datei.txt', mode='r', encoding ='utf-16').read()
```

Nutzen Sie die "print"-Funktion, um zu überprüfen, ob der Text korrekt eingelesen wurde.

In [ ]:
```

```

Nun folgt eine ganze Kaskade notwendiger Aufgaben, die man - ich sage es direkt - mit entsprechenden Programmerweiterungen (oder Webservices) einfach umgehen könnte. Wir werden diese Bibliotheken später noch ausprobieren; es widerspräche jedoch den propädeutischen Maßgaben des Kurses, diese - nun ja, nennen wir sie "Shadow Box" - an dieser Stelle schon zu lichten. Die Inhalte der Bibliotheken sind zwar quelloffen und transparent, jedoch mit Ihrem momentanen Wissen noch böhmische Dörfer. Deshalb entwerfen wir nun unser eigenes kleines Programm zur Vorverarbeitung von Textdateien - dem Preprocessing. Warum eigentlich? Und was muss genau geschehen?

Hier sind wir wieder beim Thema "Wissen" - wir alle wissen spätestens seit der ersten Klasse, was ein Wort ist. Weiß ein Computer das? Jein - nur, wenn wir es ihm beibringen. Wir müssen der Maschine also zunächst Wissen antrainieren, und das tun wir, indem wir den Text entsprechend aufbereiten: Satzzeichen müssen von den Wörtern gelöst werden, weil sie für den Computer genauso Zeichen sind wie die Buchstaben des Alphabets; die Groß- und Kleinschreibung muss je nach Anwendung angepasst werden, weil es beispielsweise zur Dopplung von Nicht-Substantiven in der Satzmitte und an Satzanfängen kommen würde - andererseits wäre für eine inhaltliche Analyse Großschreibung wieder notwendig - oder wie soll man das gebeugte Verb "macht" und das Substantiv "Macht" sonst auseinanderhalten?[1] Die Tokenisierung kennen Sie bereits - fangen Sie doch damit an: Zerlegen Sie unseren Text in Wörter, speichern Sie ihn in der Variablen "words" und lassen Sie sich diese ausgeben.

In [10]:
```
# words =
```

Das sieht abenteuerlich aus - und spätestens jetzt wissen Sie, warum wir noch einiges an Arbeit vor uns haben, bevor wir die wirklich interessanten Dinge ausprobieren können. Bitte finden Sie nun heraus, wofür das Zeichen '\n' steht - denn das findet sich überhaupt nicht in unserer ursprünglichen txt-Datei.