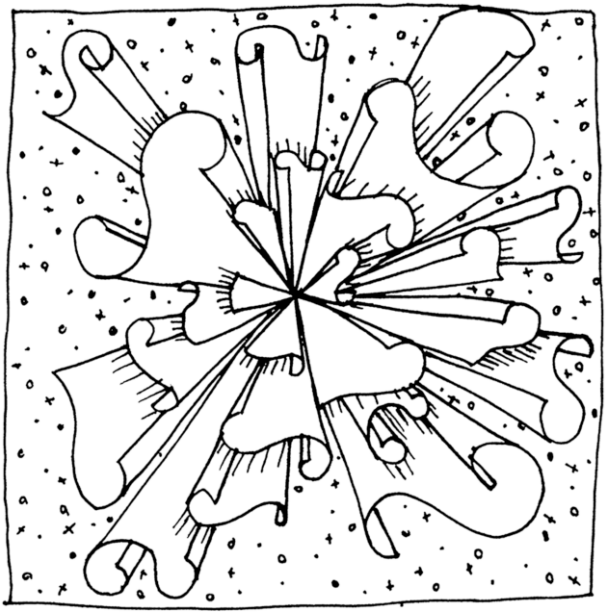
Datenformate

3



J. Jünger, C. Gärtner, *Computational Methods für die Sozial- und Geisteswissenschaften*, https://doi.org/10.1007/978-3-658-37747-2_3

Zusammenfassung

Das Kapitel führt in Datenformate ein, die Ihnen bei der Arbeit mit Computational Methods begegnen werden. Sie lernen, wie die wichtigsten Datenformate aufgebaut sind, welche Arten von Datenbanken es gibt und worin der Unterschied zwischen Datenformaten und Datenmodellen besteht.

Im Online-Repositorium unter https://github.com/strohne/cm finden Sie begleitend zum Kapitel weitere Materialien, auf die wir im Text mit ☛ verweisen.

Schlüsselwörter

Datentypen · Markdown · Zeichenkodierung · CSV · HTML und XML · JSON · SQL und NoSQL · Datenmodell · Resource Description Framework (RDF) · Semantic Web

Das Wort ‚Daten' ist eines der meistverwendeten Wörter in diesem Buch. Etymologisch leitet sich die Bezeichnung von dem lateinischen ‚datum' ab, was als ‚gegeben' übersetzt werden kann (Kluge 2002, S. 181). Als Datum werden in der ursprünglichen Bedeutung die Zeit- und Ortsangaben auf Schriftstücken bezeichnet (Kluge 2002, S. 181). Diese Verwendungsweise findet sich auch heute noch, die Extension des Begriffs, das heißt der Umfang als Datum bezeichneter Gegenstände, hat sich aber stark ausgeweitet. Je nach Perspektive werden darunter etwa Zahlen, Beobachtungen oder Bits verstanden (Ballsun-Stanton 2010). Vor allem das letzte Verständnis, Bits als Kodierung von Information, findet sich im Kontext der Informatik wieder. In diesem Sinne ist der Begriff sogar in einer ISO-Norm[1] standardisiert und wird dort definiert als „A reinterpretable representation of information in a formalized manner suitable for communication, interpretation, or processing" (ISO/IEC 2382-1, siehe ISO 2015). Ein wichtiges Merkmal ist dabei, dass es sich um Repräsentationen von etwas handelt. Daten stehen zum Beispiel für das Verhalten von Menschen oder allgemeiner für Informationen.[2] Dabei kann

[1] Die International Organization for Standardization (ISO) erarbeitet internationale Normen für viele Bereiche, unter anderem informationstechnische Normen.

[2] Die Verbindung von Daten, Informationen und Wissen wird häufig als Pyramide modelliert, wobei von einem zunehmenden Bedeutungsgehalt ausgegangen wird. Aus Sicht der Informatik bestehen Analysen daraus, dass Informationen aus Daten extrahiert werden, auf denen dann handlungsleitendes Wissen basiert. Diese Vorstellung ist jedoch zu hinterfragen, da Informationen aus sozialwissenschaftlicher Sicht nicht in Daten enthalten sein können, sondern lediglich zugeschrieben werden. Zur DIKW-Pyramide (data, information, knowledge, wisdom) siehe Ackoff (1989) und Rowley (2007).

es verschiedene Repräsentationen für die gleichen Dinge geben, das heißt, Daten können unterschiedlich angeordnet bzw. formatiert sein (☛ *Repositorium*). In diesem Kapitel werden zunächst verschiedene Datentypen und dann verbreitete Datenformate vorgestellt. Für eine Repräsentation in Tabellenform eignet sich das CSV-Format. Sollen große Datenbestände verwaltet werden, kommen häufig SQL-Datenbanken zum Einsatz, in denen mehrere Tabellen enthalten sein können. Im Prinzip kann fast jede Datenstruktur in Tabellenform gebracht werden, das ist aber nicht immer gut handhabbar. Eine höhere Flexibilität in Bezug auf die Strukturierung bieten HTML und XML für Textdaten oder JSON für Daten, die aus einzelnen Werten zusammengesetzt sind. Auch solche flexiblen Datenbestände werden bei entsprechendem Umfang in Datenbankmanagementsystemen vorgehalten, die dann als NoSQL-Datenbanken bezeichnet werden.

Die genannten Datenformate geben kaum Vorgaben darüber, welche Bedeutung die Daten haben. Die Bedeutung wird erst mit Datenmodellen festgelegt, in denen die Verwendung bestimmter Datenelemente inhaltlich definiert ist. So kann man definieren, dass zu einer Personenbeschreibung ein Name, ein Geburtsdatum und ein Wohnort gehören. Insbesondere im World Wide Web besteht der Bedarf, solche Merkmale zu standardisieren, um die Datenbestände verschiedener Anbieter verknüpfen zu können. Die Idee wird Semantic Web genannt und das zugrundeliegende Datenmodell nennt sich Ressource Description Framework (RDF). Dieses Modell – und verschiedene dazugehörige Datenformate – werden am Ende des Kapitels vorgestellt.

3.1 Datentypen

Daten sind in ihrer kleinsten Einheit Zeichen, die von Computern verarbeitet werden. Diese Zeichen haben verschiedene Datentypen, sie sind zum Beispiel Zahlen (integer, double, float), Zeichenketten (string, character) oder Wahrheitswerte (boolean). Menschen können meistens auf den ersten Blick erkennen, um welchen Datentyp es sich handelt, Computern muss das hingegen erst gesagt werden. So könnte ein Programm eine 13 sowohl als die ganze Zahl 13, als die Dezimalzahl 13,0 oder als eine Abfolge von einzelnen Buchstaben „1“ und „3“ erkennen. Diese Unterscheidung ist wichtig, da je nach Datentyp unterschiedliche Operationen möglich sind. So kann man mit Zahlen beispielsweise rechnen. Zeichenketten wie das Wort „Datum“ kann man zwar nicht multiplizieren oder dividieren, stattdessen könnte man in dem Wort aber beispielsweise nach Vokalen suchen.

Einzelne Werte mit solchen grundlegenden, atomaren Datentypen werden mathematisch als **Skalare** bezeichnet. Atomare Datentypen können wiederum zu komplexeren Datentypen zusammengesetzt werden. Eine Liste gleichartiger Elemente wird **Vektor** genannt, zum Beispiel eine Liste mit Datumsangaben. Fasst man mehrere solcher Vektoren zusammen, entsteht eine **Matrix** (Abb. 3.1). Für viele datenanalytische Fragestellungen eignen sich Matrizen, wenn etwa jede Zeile für eine Person steht und in den Spalten die Eigenschaften der Personen festgehalten sind. Eine Matrix setzt sich also aus einer Liste mit Vektoren zusammen, je nach Perspektive aus einer Liste mit Zeilen oder aus einer Liste mit Spalten.

Matrizen im engeren Sinn enthalten nur einen einzigen Datentyp, zum Beispiel nur Jahreszahlen (Ganzzahl) oder nur Geburtsorte (Zeichenkette). Um mehrere Datentypen gleichzeitig zu erfassen, kommen in verschiedenen Programmiersprachen noch eine ganze Reihe weiterer komplexer Datentypen vor. Sehr flexibel sind Diktionäre, die Name-Wert-Paare in Listen erfassen und tief verschachtelt sein können, das heißt ein Wert kann wiederum ein Diktionär sein. Für die Datenanalyse wird zudem meist mit Tabellen gearbeitet, die ähnlich wie eine Matrix aus Zeilen und Spalten bestehen, in den verschiedenen Spalten aber ganz unterschiedliche Datentypen enthalten können. Beispiele für atomare und komplexe Datentypen in den Programmiersprachen R und Python finden sich in Tab. 3.1.

Immer wenn etwas nicht wie erwartet funktioniert, empfiehlt es sich, die Datentypen zu überprüfen. Besonders wichtig ist es, bei der Programmierung und Datenanalyse zu unterscheiden, wann es sich um Zeichenketten (mit Anführungszei-

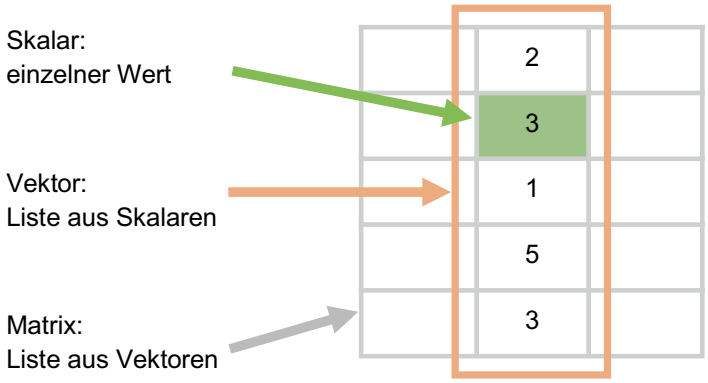


Abb. 3.1 Aufbau von Matrizen. (Quelle: eigene Darstellung)

chen), Zahlen (ohne Anführungszeichen) oder festgelegte Bezeichnungen bzw. Eigennamen von Datenobjekten (ohne Anführungszeichen) handelt. Dabei können nicht beliebige Anführungszeichen eingesetzt werden, sondern es ist in jeder Sprache festgelegt, wann einfache oder doppelte Zeichen verwendet werden müssen. Insbesondere typografische Anführungszeichen „“ wie sie in Textprogrammen automatisch bei der Eingabe entstehen – in Deutschland erst Anführungszeichen unten, dann oben – funktionieren nicht, darauf sollte beim Kopieren von Daten und Quelltexten geachtet werden.

Tab. 3.1 Beispiele für atomare und zusammengesetzte Datentypen

Datentyp	**Beschreibung**	**Beispiel** und *Bezeichnung* in R	**Beispiel** und *Bezeichnung* in Python
Atomare Datentypen			
Zahlen	Ganze Zahlen (numeric)	1884L *integer*	1884 *int*
Dezimal-zahlen	Zahlen mit Nachkommastellen (numeric)	1.6 *double*	1.6 *float*
Wahrheits-werte	Logische Werte (boolean)	TRUE, FALSE *logical*	TRUE, FALSE *bool*
Zeichen-ketten	Ketten von Buchstaben, Zahlen oder Sonderzeichen (string)	"Anna" *character*	"Anna" *str*
Zusammengesetzte Datentypen			
Vektor	Liste von Elementen des gleichen Typs	c(1,2,3)	np.array([1,2,3])
Liste	Ansammlung von Elementen mit unterschiedlichen Typen	list('Anna',1884)	['Anna',1884]
Tupel	Zusammensetzung von Elementen mit unterschiedlichen Typen	tuple('Anna',25)	('Anna',25)
Diktionär	Name-Wert-Paare (dict, dictionary)	list('Name'='Anna', 'Jahr'=1884)	{'Name': 'Anna', 'Jahr': 1884}
Matrix	Liste von Vektoren bzw. zweidimensionales Array	matrix(1:4, nrow=2)	np.array([1,2],[3,4])
Tabelle	Datensatz in Tabellenform, bestehend aus Zeilen, Spalten und Zellen	*data.frame* *tibble*	*pd.DataFrame*

Quelle: Eigene Darstellung

3.2 Textformate (MD)

Viele der im Folgenden besprochenen Datenformate sind einfache Textdateien, die mit einem Editor wie Atom oder Notepad++ geöffnet und bearbeitet werden können (siehe Kap. 1). So wie verschiedene natürliche Sprachen – Deutsch, Englisch, Farsi – durch eine Syntax, Vokabulare und eine bestimmte Aussprache geprägt sind, legen die Datenformate jeweils fest, welche Zeichen wie kodiert werden. Damit ein Computerprogramm die Dateien auch versteht, sind über alle Datenformate hinweg zwei Besonderheiten zu beachten.

Erstens sind nicht alle Zeichen sichtbar. Als **Zeilenumbruch** werden in Textdateien unter Windows zwei unsichtbare Zeichen verwendet, das Carriage Return (CR; Wagenrücklauf) gefolgt vom Line Feed (LF). Diese Zeichen sind Steuerzeichen und simulieren eine Schreibmaschine, bei der am Ende einer Zeile zunächst das Papier nach rechts geschoben wurde, sodass sich die Schreibposition am Zeilenanfang befindet. Dann wurde die Walze mit dem Papier zu einer neuen Zeile weitergedreht.

Mitunter werden Zeilenumbrüche durch ihre ASCII[3]-Werte `#13` für Carriage Return und `#10` für Line Feed dargestellt, manchmal auch durch die Escape-Sequenzen `\r` für Carriage Return und `\n` für Line Feed. In Unix-Systemen wie macOS oder Linux wird ausschließlich der Line Feed verwendet. In älteren Mac-OS-Varianten war dagegen das Carriage Return gebräuchlich. Die verschiedenen Möglichkeiten sind eine häufige Fehlerquelle beim Einlesen von Datensätzen. Im Zweifelsfall empfiehlt es sich, alle Zeilenumbrüche zunächst mit einem Texteditor durch ein einzelnes Line-Feed-Zeichen zu ersetzen. Um in einem Texteditor zu sehen, welche Zeichen in einer Datei als Zeilenumbruch fungieren, können Sie zum einen die Steuerzeichen einblenden, häufig steht dafür eine mit dem Absatzsymbol ¶ gekennzeichnete Schaltfläche zur Verfügung. Die Zeilenumbrüche sollten nun am Ende jeder Zeile sichtbar markiert sein. Zum anderen wird in Texteditoren üblicherweise das Format der Zeilenumbrüche in der Statusleiste aufgeführt und kann mit einem Klick geändert werden.[4]

[3] Der American Standard Code for Information Interchange ist im Jahr 1963 entstanden und umfasst 128 Zeichen, die bis heute die Grundlage für fast alle computerbasierten Zeichensysteme bilden.

[4] In Notepad++ und Atom ist beispielsweise in der unteren Leiste neben der Zeichenkodierung der Zeilenumbruch eingeblendet – „Windows (CR LF)“ oder „Macintosh (CR)“. Mit einem Links- bzw. Rechtsklick können Sie das Format ändern. Alternativ ist in Atom die Funktion Line Ending Selector: Convert To […] über die Kommando-Palette (Tastenkombination Strg bzw. Cmd + Shift + P) erreichbar.

Zweitens muss die Kodierung der verschiedenen lokalen und internationalen Zeichen – arabische Schriftzeichen, chinesische Glyphen, deutsche Umlaute, französische Akzente oder auch Emojis – beachtet werden. Als **Zeichenkodierung** (engl. *encoding*) wird, wenn keine Umlaute oder andere Sonderzeichen enthalten sind, normalerweise das ASCII-Format verwendet. Für Umlaute oder andere Sonderzeichen kommen je nach Sprache unterschiedliche Erweiterungen dieses Formats zur Anwendung, die zum Teil mit verwirrenden Bezeichnungen versehen sind. Besonders verbreitet sind ISO-8859-1 (auch ANSI genannt) sowie Windows-1252 (auch Latin-1 oder CP1252 genannt).

Mit ANSI lassen sich bis zu 256 verschiedene Zeichen darstellen. Immer mehr setzt sich allerdings Unicode durch. Dieser Standard ist mit der Idee verbunden, alle Zeichen der Welt zu erfassen, enthält zum Beispiel auch Emojis und fortlaufend werden Anträge auf neue Emojis gestellt (Unicode 2021). Um diese Vielfalt vollständig abzubilden, werden mehrere Bytes benötigt, das entsprechende UTF-16-Format verwendet deshalb zwei Bytes und UTF-32 sogar vier Bytes je Zeichen. Als platzsparende Variante kommt meistens UTF-8 zum Einsatz und das aus gutem Grund: Mit UTF-8 lässt sich nicht nur jedes bislang über Unicode standardisierte Zeichen sehr platzsparend kodieren. Es ist zudem abwärtskompatibel, denn wenn keine Sonderzeichen benötigt werden, dann ist die Kodierung identisch zum ASCII-Format.[5]

Bei UTF-8 wird für die ASCII-Zeichen ein Byte verwendet, für alle weiteren Zeichen dagegen zwei Bytes. Immer wenn zwei Bytes verwendet werden, gibt es wiederum verschiedene Möglichkeiten, welches Byte als erstes angegeben ist. Im Zweifelsfall wird dies durch eine Byte Order Mark (BOM) gekennzeichnet.[6] Dabei handelt es sich um ein Zeichen ganz am Anfang der Textdatei. Einige CSV-Dateien enthalten dieses Zeichen, andere nicht. Werden Sonderzeichen also in einer geöffneten Datei nicht richtig angezeigt – wenn etwa anstelle von „Hüte“ ein „HÃ¼te“ erscheint – lohnt es sich, die Zeichenkodierung zu überprüfen und gegebenenfalls mit einem Texteditor umzustellen.

In Textdateien können beliebige Inhalte festgehalten werden, nicht nur vorstrukturierte Daten, sondern zum Beispiel auch eigene Notizen. Diese Dateien können unkompliziert mit anderen geteilt werden, da nahezu auf jedem Computer

[5] Für ASCII werden sieben Bit verwendet, ein achtes Bit kennzeichnet dann, ob ASCII oder ein erweiterter Zeichensatz wie ANSI oder UTF-8 vorliegt. Deshalb sind alle ASCII-Zeichen im ANSI-Zeichensatz und auch in UTF-8 enthalten.

[6] Bei UTF-16 unterscheidet man deshalb UTF-16LE (little endian) und UTF-16BE (big endian), damit sind die beiden unterschiedliche Reihenfolgen der zwei Bytes gemeint.

mindestens ein Programm zum Bearbeiten von Textdateien vorhanden ist. Allerdings sind darin normalerweise keine Formatierungen wie Fettungen oder Kursivierungen möglich. Um dennoch zumindest grundlegende Formatierungen vornehmen zu können, hat sich in vielen Bereichen das Format Markdown etabliert:

- Überschriften beginnen mit einer Raute:
  ```
  # Überschrift 1
  ## Überschrift 2
  ```
- Aufzählungen werden durch Bindestriche realisiert:
  ```
  - Erster Punkt
  - Zweiter Punkt
  ```
- Text wir kursiv geschrieben, indem er in Sternchen eingeschlossen wird:
  ```
  *kursiver Text*
  ```
- Für fetten Text werden zwei Sternchen verwendet:
  ```
  **fetter Text**
  ```
- Code wird in Backticks eingeschlossen, um darin zum Beispiel Sternchen oder Bindestriche verwenden zu können. Backticks sind rückwärtsgerichtete Anführungszeichen und auf der Tastatur etwas versteckt:
  ```
  `Quellcode im Text`
  ```
- Soll Quellcode über mehrere Zeilen gehen, dann werden zu Beginn und am Ende jeweils drei Backticks gesetzt:
  ````
  ```
  Code über
  mehrere Zeilen.
  ```
  ````

Absätze werden durch eine Leerzeile erzeugt und aufpassen sollte man bei Zeilenumbrüchen. Denn ein einzelner Zeilenumbruch wird nicht immer als Zeilenumbruch dargestellt. Zeilenumbrüche in Aufzählungen funktionieren beispielsweise nur dann, wenn die vorangegangene Zeile mit zwei Leerzeichen endet und der folgende Text mit zwei Leerzeichen eingerückt wird. Darüber hinaus sind viele weitere Formatierungen zur Darstellung von Links, Bildern oder Tabellen möglich. Eine Übersicht bieten Cheatsheets, die in großer Zahl im Internet zu finden sind.[7]

Während einfache Textdateien meistens mit der Endung *.txt* abgespeichert werden, sind Markdown-Dateien an der Endung *.md* zu erkennen. Typischerweise fin-

[7] Siehe zum Beispiel Cone (2022; https://www.markdownguide.org/cheat-sheet/).

det sich beispielsweise in Git-Repositorien immer eine Datei *readme.md*, in der eine Einführung in das Repositorium gegeben wird. Auf der Webseite des Repositoriums werden die Markdown-Formatierungen dann automatisch in HTML umgeformt, sodass im Browser echte Listen, Kursivierungen oder Überschriften erscheinen. Schauen Sie sich einmal den Quelltext der readme-Datei im ☛ Repositorium und die Darstellung auf der Webseite an!

3.3 Tabellenformate (CSV)

Eines der am meisten verbreiteten Formate, um tabellarische Daten zu speichern, ist das CSV-Format. Hierbei handelt es sich um eine Textdatei, in der jede Zeile einen Fall enthält, zum Beispiel einen Kommentar auf einer Online-Plattform oder Angaben zu einer Person in einem Roman. Innerhalb einer Zeile sind die Werte mit einem Komma getrennt. Die Abkürzung des Dateiformats steht dementsprechend für Comma Separated Values. In Abb. 3.2 ist ein Auszug aus einer solchen Datei abgebildet, in der Tweets mit einer ID und zugehörigen Kennwerten aufgelistet sind. Die Spaltennamen sind in der ersten Zeile angegeben. Hier wird auch eine Besonderheit sichtbar: Die Liste der Hashtags ist in der zweiten Zeile in Anführungszeichen gesetzt, damit sie als ein Wert zählt, obwohl darin ebenfalls ein Komma enthalten ist.

Es gibt allerdings etliche Varianten dieses Formats und man muss im Einzelfall prüfen, womit man es zu tun hat. Die Dateien unterscheiden sich zum einen wie alle Textdateien durch die verwendeten Zeilenumbrüche und durch die Zeichenkodierung (siehe oben). Dazu kommen zwei Besonderheiten von CSV-Dateien:

- Als **Trennzeichen** zwischen den Werten sind neben Kommata auch Semikola, Leerzeichen, Tabulator oder seltener die Pipe (senkrechter Strich `|`) oder die Raute (Doppelkreuz `#`) gebräuchlich. Je nach Voreinstellung des Systems kommen Microsoft Office-Programme meistens besser mit Tabulatoren zurecht, andere Statistikprogramme vor allem mit Kommata und Semikola.

```
id,name,from,favorites,replies,retweets,hashtags
6,eaduenergy,Forschungslabor Eadu,64,0,1,"sternzerstörer,werft"
7,eaduenergy,Forschungslabor Eadu,3,0,,todesstern
8,eaduenergy,Forschungslabor Eadu,30,0,6,kyber
```

Abb. 3.2 Auszug aus einer CSV-Datei. (Quelle: eigene Darstellung)

Soll in einem Feld das Trennzeichen selbst verwendet werden, dann muss dieses markiert werden, damit dadurch keine neue Spalte beginnt. Häufig werden in diesem Fall die Werte in doppelte Anführungszeichen gesetzt. Dadurch verschiebt sich das Problem, denn auch Anführungszeichen können in den Werten selbst vorkommen. Dieses Problem wird wiederum anders gelöst und als Maskieren bezeichnet: Entweder wird den Anführungszeichen ein Backslash vorangestellt oder die Anführungszeichen werden verdoppelt. Soll wiederum ein solches Maskierungszeichen wie der Backslash verwendet werden, wird es nochmals maskiert, das heißt verdoppelt. Diese Form des Maskierens von reservierten Zeichen wird in vielen anderen Sprachen wie R oder Python ähnlich umgesetzt.

- Die **Spaltenbezeichnungen** sind häufig in der ersten Zeile angegeben, aber nicht immer. Dann müssen die Spaltennamen beim Einlesen zusätzlich angegeben werden.

Sollten Sie CSV-Dateien in Programme einlesen, um diese weiterzuverarbeiten, können Konvertierungsprobleme auftreten. Dies passiert, wenn das Programm von einem anderen CSV-Dialekt ausgeht, als ihn die Datei aufweist. In diesem Fall versuchen Sie zunächst herauszufinden, ob Sie Einstellungen beim Einlesen des Datensatzes vornehmen können, etwa um das Trennzeichen festzulegen. Um einschätzen zu können, welcher ‚Dialekt' in einer CSV-Datei eingesetzt wird, kann man sie mit einem Texteditor öffnen (siehe Kap. 1; ☛ *Repositorium*). Wichtig ist dabei, dass man im Texteditor auch die unsichtbaren Steuerzeichen einblendet. Sonst lassen sich Tabulatoren und Leerzeichen oder auch die verschiedenen Zeilenumbruchszeichen nicht voneinander unterscheiden. Mit einem Texteditor können Sie das Format der Datei ggf. ändern, indem Sie beispielsweise alle Tabulatoren durch Semikola ersetzen. Auch bei anderen Dateiformaten lohnt es sich in der Regel, sie einmal mit einem Texteditor zu betrachten.

3.4 Auszeichnungssprachen (HTML und XML)

Um in einem fortlaufenden Text einzelne Abschnitte zu markieren und zu formatieren, eignen sich Auszeichnungssprachen. Häufig verwendete Auszeichnungssprachen sind die Hypertext Markup Language (HTML) oder die Extensible Markup Language (XML). Die Grundprinzipien von Auszeichnungssprachen sind meist ähnlich. So geht es zunächst darum, allgemein die gesamte Struktur sowie einzelne Merkmale in Texten zu annotieren. Umgesetzt wird dies durch das Umklammern einzelner Textteile.

Im Detail unterscheiden sich die verschiedenen Auszeichnungssprachen in ihrer syntaktischen Umsetzung und in den Anwendungsfällen. HTML ist die Standard-

sprache, mit der Webseiten geschrieben werden. Soll beispielsweise ein Wort fett dargestellt werden, so wird es mit einem <b>-Tag umschlossen. Der Name des Tags ist in spitzen Klammern angegeben, der Buchstabe „b“ steht in diesem Fall für „bold“ (deutsch *fett*). Direkt vor der Textstelle wird ein öffnendes Tag gesetzt und direkt danach ein schließendes Tag. Der Unterschied zwischen diesen beiden Varianten besteht darin, dass schließende Tags nach der ersten spitzen Klammer einen Schrägstrich / enthalten:

```
Dieser Satz enthält ein <b>fett</b> gedrucktes Wort.
```

Nur öffnende Tags können mit weiteren Attributen versehen werden. Solche Attribute werden abgetrennt mit einem Leerzeichen hinter dem Namen des Tags angegeben. Auf den Namen des Attributs folgt ein Gleichheitszeichen, danach in einfachen oder doppelten Anführungszeichen der Wert. Attribute sind also Name-Wert-Paare. So kann ein Tag mit einem eindeutigen Bezeichner (ID) versehen werden, damit es von anderen gleichnamigen Tags unterscheidbar ist. Häufig sind auch `class`-Attribute anzutreffen, die für die Gestaltung im Web eine besondere Rolle spielen (siehe Abschn. 4.1).

```
Dieser Satz enthält zwei <b id="wort1" class="gruen">fett
</b> gedruckte Wörter mit <b id="wort2" class="gruen">
unterschiedlichen</b> IDs und der gleichen Klasse.
```

Durch Tags markierte Bereiche können auch verschachtelt werden, etwa um einen Absatz zu markieren und darin Wörter hervorzuheben. Absätze werden durch <p>-Tags markiert. Allerdings dürfen sich die Bereiche nicht überkreuzen. Wenn also eine Fettung im nächsten Absatz fortgesetzt werden soll, muss sie zunächst geschlossen und dann nach dem öffnenden Absatz-Tag wieder neu geöffnet werden:

```
<p>Dieser Absatz endet mit <b>fett gedruckten Wörtern
</b></p><p><b>Dieser Absatz</b> beginnt mit fett
gedruckten Wörtern.</p>
```

Die Möglichkeit zur hierarchischen Strukturierung ist eine besondere Stärke von Auszeichnungssprachen und macht sie flexibel einsetzbar. Außerdem erlaubt der HTML5-Standard in bestimmten Fällen auch alleinstehende Tags, die genauso aussehen wie öffnende Tags und nicht wieder geschlossen werden müssen. Damit wird der Quelltext übersichtlicher, etwa wenn Bilder eingebunden werden:

```
<img src="chewbacca.png" alt="Chewbacca">
```

Sprachen wie HTML legen darüber hinaus auch den Aufbau von Dokumenten fest. Ein HTML5-Dokument beginnt immer mit der Angabe des Dokumententyps und dem <html>-Element. Darin sind ein <head>-Element für nicht sichtbare allgemeine Angaben und ein <body>-Element mit den sichtbaren Inhalten der Seite enthalten. Das Grundgerüst sieht wie folgt aus:

```
<!DOCTYPE html>
<html>
  <head>
     <title>Seitentitel</title>
  </head>
  <body>
     Seiteninhalte
  </body>
</html>
```

Auch das Vokabular ist standardisiert, das heißt die Tags haben eine festgelegte Bedeutung. So sind im HTML5-Standard unter anderem folgende Tags festgelegt:[8]

- <h1>, <h2> bis <h6> kennzeichnen Überschriften erster, zweiter bis sechster Ordnung.
- <p> kennzeichnet Absätze.
- kennzeichnet Hervorhebungen.
- <table> kennzeichnet eine Tabelle, <tr> eine Tabellenzeile und <td> eine Zelle innerhalb einer Tabellenzeile.
- <img> wird für Bilder verwendet.
- <a> steht für Anchor und wird zum Setzen von Links verwendet.
- <meta> steht für Metanagaben wie das Erstelldatum oder der Titel einer Seite. Diese Angaben sind im Kopfbereich des Quelltextes eingebettet und nicht sichtbar.

Für die Formatierung von Webseiten sind noch weitere Elemente verbreitet, die keine vorgegebene Bedeutung haben. In Verbindung mit IDs und Klassen-Attributen werden diese Elemente dann mittels Cascading Style Sheets (CSS) optisch formatiert oder mittels JavaScript um interaktive Funktionen erweitert:

[8] Der HTML-Standard wird durch das W3C festgelegt und ist dort auch dokumentiert, siehe W3Schools (2022a; https://www.w3schools.com/tags/default.asp).

- `<span>` kennzeichnet Textbereiche, in der Regel kurze Phrasen oder einzelne Wörter.
- `<div>` kennzeichnet Blöcke, zum Beispiel den Kopfbereich oder den Fußbereich einer Seite.

Um einen Eindruck davon zu gewinnen, wie eine Webseite aufgebaut ist, können Sie sich den Quelltext im Browser ansehen. Hierzu kann in den meisten Browsern mit der rechten Maustaste ein Kontextmenü aufgerufen werden, in dem die Funktion SEITENQUELLTEXT ANZEIGEN oder ähnlich angeboten wird. Der gesamte Quelltext ist häufig jedoch eher unübersichtlich. Zudem interessiert in der Regel nur ein ganz bestimmter Ausschnitt aus dem Quelltext. Um schnell zu erfassen, wie ein bestimmter Teil einer Seite umgesetzt wurde, lässt sich eine Entwicklerkonsole öffnen. In der Konsole kann gezielt ein Ausschnitt aus dem Quelltext anvisiert werden. Dazu klicken Sie im Browser mit der rechten Maustaste auf ein Element, zum Beispiel ein Eingabefeld, und wählen dann ELEMENT UNTERSUCHEN (Abb. 3.3). Alternativ können Sie die Entwicklerkonsole in vielen Browsern mit

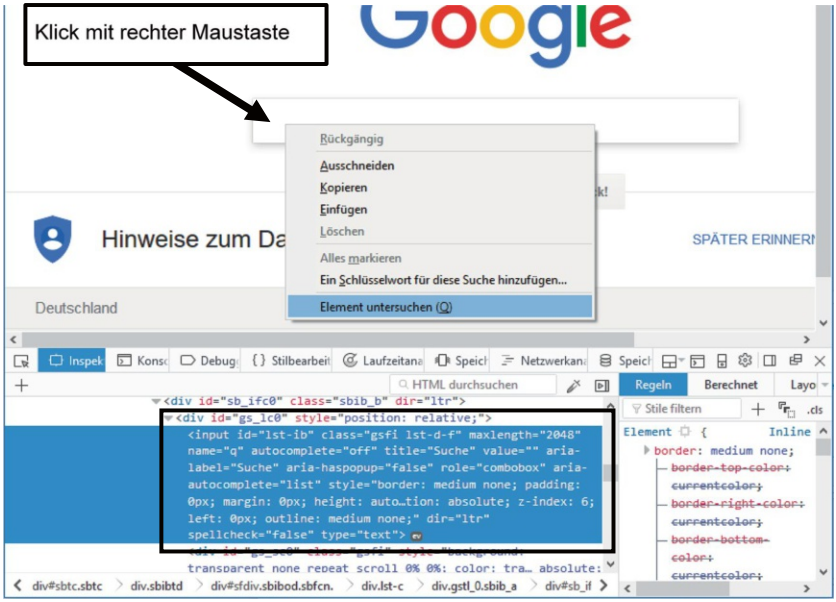


Abb. 3.3 Der Quelltext von Webseiten im Browser Firefox (Entwicklerkonsole). (Quelle: eigene Darstellung)

der Taste F12 erreichen. Sie können dort auch Texte oder Attribute verändern, probieren Sie es einmal aus!

Der Browser arbeitet nicht direkt mit dem Quelltext, sondern liest ihn in ein sogenanntes Document Object Model (DOM) ein. Dieser Vorgang heißt parsen und spielt im Zusammenhang mit automatisierter Datenerhebung eine Rolle (siehe Kap. 7). Das DOM bildet den Quelltext in einer Baumstruktur bestehend aus Knoten ab. Die Tags, Attribute und der Text werden dazu in Knoten umgewandelt. Dieses DOM bzw. die enthaltenen Knoten können durch Sprachen wie JavaScript verändert werden, um eine Webseite interaktiv zu gestalten.

Eine weitere Auszeichnungssprache, die vor allem als Datenbankformat eingesetzt wird, stellt die Extensible Markup Language (XML) dar.[9] Im Grunde funktioniert XML ähnlich wie HTML, auch mithilfe dieser Sprache werden Texte strukturiert. Im Gegensatz zu HTML ist XML allerdings in der inhaltlichen Ausgestaltung flexibler, was speziell an bestimmte Anwendungsfälle angepasste Datenstrukturen erlaubt. Die Flexibilität ist vor allem dadurch begründet, dass in XML fast beliebige Namen für Tags und Attribute verwendet werden können. Hier ist hauptsächlich die Syntax festgelegt, das heißt die Struktur aus Tags, Attributen und Text. Außerdem unterscheidet sich XML insofern von HTML, dass in XML-Dokumenten alle Tags immer geschlossen werden. Ein Tag kann bei Bedarf geöffnet und sofort wieder geschlossen werden, indem ein Schrägstrich vor die schließende spitze Klammer gestellt wird:

```
<img src="chewbacca.png" alt="Chewbacca" />
```

Viele dieser Auszeichnungssprachen liegen in unterschiedlichen Versionen vor. Während XML nur die allgemeine Struktur vorgibt, ist die Bedeutung der Elemente in weitergehenden Standards festgelegt. Zum Beispiel baut das TEI-Format,[10] ein in den Geisteswissenschaften verbreiteter Standard zur Aufbereitung von Texten, auf XML auf. Im TEI-Format ist beispielsweise festgelegt, dass Sätze oder Nebensätze mit bestimmten Tags wie `<li>` oder `<cl>` versehen werden, sodass die Satzstruktur rekonstruiert werden kann. Auch RSS-Feeds, in denen zum Beispiel Nachrichtenseiten die aktuellen Meldungen anbieten, sind ein XML-Format. Ebenso enthalten einige Microsoft-Office-Dateien eine Sammlung von XML-Dokumenten mit einer festgelegten Struktur (OpenDocument, Office Open XML),

[9] Auch der XML-Standard ist von W3C dokumentiert und einsehbar unter W3C (2013; https://www.w3.org/TR/xml/).

[10] Siehe Text Encoding Initiative (2022; https://tei-c.org/).

die zu einer Datei zusammengepackt und komprimiert sind.[11] Auch für die Annotation von Trainingskorpora für das Machine Learning (siehe Kap. 8) wird XML eingesetzt. Im Oxford Text Archive (siehe Kap. 2) sind zum Beispiel Personen in Dokumenten als sogenannte Named Entities gekennzeichnet.

3.5 Objektdatenformate (JSON)

Innerhalb von Programmiersprachen werden Daten häufig als Objekte repräsentiert, die Eigenschaften mit bestimmten Werten haben. Eine Nutzerin könnte dann beispielsweise als Objekt erfasst sein, welches die Eigenschaft „Name“ mit dem Wert „Eliza“ hat. Werte können auch komplexe Datentypen umfassen – insbesondere Listen, wenn jemand mehrere Namen hat – oder die Werte selbst sind wiederum Objekte, wenn ein Namensobjekt zum Beispiel die Eigenschaften „Vorname“ und „Nachname“ enthält.

Mit derartigen Datenstrukturen lassen sich viele Bereiche der Welt innerhalb von Programmen modellieren. Dafür werden Listen, Name-Wert-Paare und elementare Datentypen wie Zahlen, Zeichenketten und Datumstypen benötigt. Ein Datenformat für solche Datenstrukturen ist JSON (JavaScript Object Notation).[12] Dieses Format wird im Web – dem Ursprungskontext der Programmiersprache JavaScript – zum Beispiel für APIs (Programmierschnittstellen) oder für interaktiv nachgeladene Inhalte verwendet.[13] Mittlerweile hat sich JSON in vielen weiteren Bereichen etabliert. So lassen sich zum Beispiel mit der Programmiersprache Python (siehe Abschn. 5.2) unter Verwendung der *json*-Bibliothek die internen Datenstrukturen als JSON abspeichern oder laden.[14]

In JSON werden Listen in eckige Klammern eingefasst, wobei die Elemente mit Kommata getrennt werden. Eine Sammlung von Name-Wert-Paaren wird als Dictionary bezeichnet und ebenfalls mit Kommata getrennt, aber in geschweiften Klammern zusammengefasst. Der Name (auch Schlüssel, engl. *key*) wird links von

[11] Zum Auspacken einer docx-Datei kann ein normales ZIP-Programm wie 7zip verwendet werden. Anschließend lassen sich die Inhalte im Texteditor ansehen. Probieren Sie es einmal aus!

[12] Eine Referenz zum JSON-Format wird von der Internet Engineering Task Force (IETF) herausgegeben, siehe Bray (2014; https://tools.ietf.org/html/rfc7159).

[13] Die zum interaktiven Nachladen verwendeten XMLHttpRequests vermitteln entgegen dem Namen nicht nur XML, sondern auch andere Datenformate wie JSON.

[14] Achtung bei der Verwendung von Anführungszeichen: JSON sieht ähnlich aus wie Dictionaries in Python (Tab. 3.1), allerdings dürfen hier anders als in Python nur doppelte Anführungszeichen verwendet werden.

einem Doppelpunkt angegeben, der Wert rechts davon. Zahlenwerte werden angegeben wie sie sind. Die Schlüssel und Zeichenketten werden in doppelte Anführungszeichen gesetzt:

```
[
    {
      "Name": "Eliza",
      "Typ" : "Bot",
      "Geburtsjahr" : 1966
    },
    {
      "Name": {
        "vorname": "Joseph",
        "nachname": "Weizenbaum"
      },
      "Typ": "Mensch",
      "Geburtsjahr": 1923
    }
]
```

Ein Vorteil gegenüber proprietären[15] Formaten zur Repräsentation von Objekten besteht darin, dass JSON sowohl maschinen- als auch menschenlesbar ist. Wie auch CSV-Dateien und HTML- oder XML-Dateien können JSON-Dateien mit einem Texteditor geöffnet und bearbeitet werden. Die Lesbarkeit ist vor allem dann gut, wenn die Hierarchie der Struktur durch Einrückungen sichtbar gemacht wird. Das gilt natürlich auch für XML und HTML. Einige Texteditoren stellen Funktionen bereit, um sinnvoll einzurücken. Dieser Vorgang und die entsprechenden Funktionen nennen sich im Englischen *pretty print*.

3.6 Datenbanken (SQL und NoSQL)

Jede Sammlung von CSV-, HTML- oder JSON-Dateien kann man bereits als Datenbank im weiteren Sinn begreifen. Wenn von einer Datenbank die Rede ist, meint man damit aber meistens ein Datenbank Management System (DBMS). Ein sol-

[15] Der Ausdruck „proprietär“ bedeutet, dass ein Hersteller ein Datenformat ohne Rücksicht auf Standards und häufig auch ohne öffentliche Dokumentation speziell für seine eigenen Produkte entwickelt hat. Proprietäre Software ist vereinfacht gesagt das Gegenstück zu Open-Source-Software.

ches System ist mehr als die Zusammenstellung von Daten. Dazu gehört in der Regel eine Software, die Funktionen zur effizienten Verwaltung (Speichern, Abfragen, Lastverteilung) und auch zum Sicherstellen der Datensicherheit (Transaktionen, Verschlüsselung) bereitstellt. Diese Funktionen sind einerseits für den Umgang mit großen Datenbeständen und andererseits für die Koordination gleichzeitiger Zugriffe nötig.

Im Allgemeinen unterscheidet man relationale und nichtrelationale Datenbanken. Tendenziell eignen sich relationale Datenbanken immer dann, wenn das Schema der erfassten Daten festgelegt ist und tabellarische Daten schnell aus der Datenbank ausgelesen werden sollen. Nichtrelationale Datenbanken sind dahingegen besonders dann von Vorteil, wenn vermehrt veränderliche Datenstrukturen oder Objekte in die Datenbank geschrieben werden. Vom Datenbankmanagementsystem hängt auch ab, über welche Sprache Daten in die Datenbank geschrieben oder aus dieser gelesen werden. Besonders bei relationalen Datenbanken kommt die Abfragesprache SQL (Structured Query Language) zum Einsatz. Nichtrelationale Datenbanken werden auch NoSQL-Datenbanken genannt, beispielsweise werden Netzwerke in Graphendatenbanken erfasst und über SPARQL oder die Cypher Query Language abgefragt (siehe Kap. 4).

Relationale Datenbanken legen die Daten in Tabellen ab, die untereinander über IDs verknüpft sind (engl. *join*). In einer Tabelle sind dann beispielsweise Blogartikel abgelegt und in einer anderen die Kommentare der Nutzer:innen zu einem Artikel. Jeder Artikel und jeder Kommentar erhalten eine ID, das heißt eine eindeutige Nummer.[16] In der Tabelle mit den Kommentaren wird zusätzlich die ID des dazugehörigen Artikels als sogenannter Fremdschlüssel abgelegt (siehe auch Abb. 3.4).

Man unterscheidet dabei vor allem drei Arten von Beziehungen:

- Bei 1:1-Beziehungen gehört zu einem Datensatz immer genau ein Datensatz aus einer anderen Tabelle. Zum Beispiel wird einer Person immer genau eine Postadresse zugeordnet.
- Bei 1:n-Beziehungen oder n:1-Beziehungen können zu einem Datensatz mehrere Datensätze aus einer anderen Tabelle gehören. Ein:e Verkäufer:in kann zum Beispiel mehrere Produkte verkaufen. Bei der Umsetzung erhält jede:r Verkäufer:in eine eindeutige ID (= Schlüssel) und diese ID wird in einer Spalte der Produkttabelle (= Fremdschlüssel) vermerkt.

[16] Es können theoretisch auch nichtnumerische Merkmale verwendet werden (z. B. Hashwerte). Die Verwendung von Nummern ist allerdings platzsparend und effizient, da Computer Zahlen gut verarbeiten können.

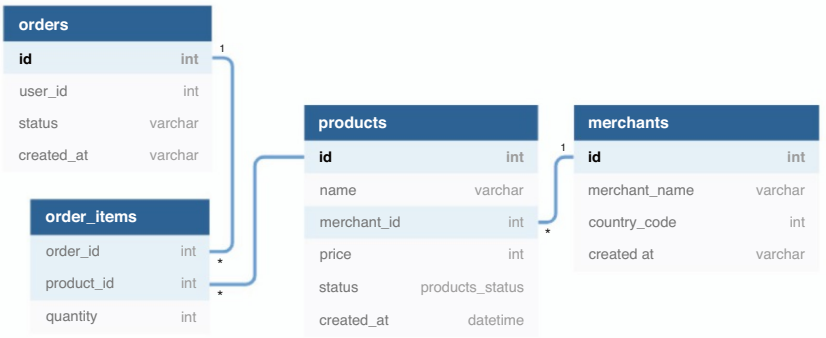


Abb. 3.4 Auszug aus der relationalen Datenstruktur eines Shop-Systems. Jeder Kasten steht für eine Tabelle, darin sind die Spalten aufgeführt. Die Verknüpfung erfolgt über IDs. Zum Beispiel wird in der Tabelle für die Elemente im Warenkorb (order_items) die ID des Produkts vermerkt. (Quelle: Holistics.io (2022; https://dbdiagram.io/))

- Bei n:m-Beziehungen werden mehreren Datensätzen aus einer Tabelle mehrere Datensätze aus einer anderen Tabelle zugeordnet. Zum Beispiel kann eine Bestellung mehrere Produkte umfassen, gleichzeitig kann ein Produkt mehrfach von unterschiedlichen Käufer:innen bestellt werden. Diese Struktur wird meistens über drei Tabellen abgebildet: eine Produkttabelle, eine Bestelltabelle und eine Warenkorbtabelle. Erst indem die Warenkorbtabelle zu den anderen beiden Tabellen in 1:n-Beziehungen steht, stehen die Produkt- und Bestelltabelle zueinander in einer n:m-Beziehung.

Die Struktur der Tabellen – vor allem Name der Tabellen, Name und Datentyp der Spalten – ist in der Regel festgelegt. Diese Struktur kann zwar verändert werden, betrifft dann aber den gesamten Datenbestand. Eine relationale Datenbank muss also den damit abgebildeten Gegenstand (z. B. Weblogs oder Online-Shops) vollständig modellieren.

Eines der am weitesten verbreiteten Datenbankmanagementsysteme ist MariaDB. Es ist als Open-Source-Software verfügbar, eine kommerzielle Lizenzierung erlaubt das kompatible MySQL.[17] Für den Datenaustausch verwendet man bei SQL-Datenbanken häufig sogenannte Dumps. Dabei handelt es sich um Textdateien, in denen die SQL-Befehle zum Erzeugen der Datenbank, der Tabellen und

[17] MariaDB (MariaDB Foundation 2022; https://mariadb.org/) ist in eine Abspaltung von MySQL (Oracle 2021; https://www.mysql.com/), beide Varianten sind meist austauschbar.

```
-- Struktur der Tabelle pages
CREATE TABLE IF NOT EXISTS `pages` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `created` datetime DEFAULT CURRENT_TIMESTAMP,
  `title` text COLLATE utf8_unicode_ci,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-- Daten der Tabelle pages
INSERT INTO `pages` (`id`, `created`, `title`) VALUES
(7, '2018-09-08 13:59:05', 'Kontakt'),
(8, '2018-09-08 13:59:05', 'Startseite'),
(9, '2018-09-08 13:59:05', 'Impressum')
```

Abb. 3.5 Auszug aus dem SQL-Dump eines Content-Management-Systems für Webseiten. (Quelle: eigene Darstellung)

der Inhalte aufgelistet sind (Abb. 3.5). Die Befehle können dann in einem Datenbankmanagementsystem ausgeführt werden, um eine Datenbank einzuspielen (siehe Abschn. 4.1.4). Da diese Dateien sehr groß werden können, sind sie meistens mit einem zip-Programm gepackt.

Dagegen ist bei nichtrelationalen Datenbanken die Struktur bzw. das Datenbankschema üblicherweise nicht festgelegt und wird durch die konkreten Datensätze bestimmt. Diese Systeme sind meistens für spezielle Einsatzzwecke optimiert. Für einen sehr schnellen und einfachen Datenzugriff können etwa Name-Werte-Listen verwendet werden. Beispiele dafür sind Google Bigtable[18] oder Redis.[19] Da die die Namen der Werte flexibel handhabbar sind und nicht von vornherein festgelegt werden müssen, können später immer wieder neue Datensorten aufgenommen werden. Das ist beispielsweise hilfreich, wenn sich erst später herausstellt, dass für eine:n Nutzer:in nicht nur der Link zum Twitter-Profil, sondern auch zum Instagram-Profil erfasst werden soll.

Weisen die Datensätze noch komplexere Strukturen auf, eignen sich dokumentenorientierte Datenbanken wie MongoDB,[20] in welcher Daten intern als JSON abgelegt werden, oder eXist[21] für XML-Dokumente. Ähnliche Systeme kommen für Volltextsuchmaschinen zum Einsatz, zum Beispiel bei Elasticsearch.[22] Gra-

[18] Siehe Google (2022a; https://cloud.google.com/bigtable).

[19] Siehe Redis (2022; https://redis.io/).

[20] Siehe MongoDB (2022; https://mongodb.com/).

[21] Siehe Meier (2003; http://www.exist-db.org/).

[22] Siehe Elastic (2022; https://elastic.co/elasticsearch/).

phenorientierten Systeme wie Neo4j verwalten wiederum besonders gut Netzwerkdaten.[23] Relationale und nichtrelationale Konzepte lassen sich auch mischen, indem flexible Datenformate wie JSON oder XML in den Spalten einer relationalen Datenbank abgelegt werden. Eine Übersicht über die aktuell verbreiteten Datenbanksysteme und Erläuterungen zum Aufbau finden Sie bei DB-Engines.[24]

3.7 Datenmodelle (RDF)

In Bezug auf Daten*formate* haben sich Standards wie CSV, JSON oder XML und HTML herausgebildet. Auch bei den Daten*banken* genießen einzelne Systeme wie MySQL hohe Popularität. Dennoch ist das Daten*modell* von Anwendung zu Anwendung sehr verschieden. Formal versteht man unter einem Datenmodell die Abbildung der Prozesse und Sachverhalte eines Anwendungsbereichs (auch Domäne genannt) auf die Datenstrukturen. Ein Datenmodell umfasst somit beispielsweise die Bezeichnungen der Datenfelder und die Beziehungen zwischen Tabellen. Datenmodelle sind zunächst abstrakte, konzeptionelle Vorstellungen, wie bestimmte Sachverhalte mithilfe von Daten modelliert werden sollen, beispielsweise das Zusammenspiel von Nutzer:innen, Posts und Kommentaren auf einer Social-Networking-Site oder die Struktur einer historisch-kritischen Edition. Die konkrete Ausgestaltung des Modells geschieht dann über die Erfassung der Daten, die mit festgelegten Datenformaten in einer Datenbank abgelegt werden (Abb. 3.6). In der Regel wird für jede Anwendung – sei es eine Smartphone-App oder ein wissenschaftliches Datenanalyseprojekt – ein eigenes Datenmodell entwickelt. Zur Dokumentation von Datenmodellen werden Sprachen wie die Unified Modeling Language (UML) eingesetzt.[25]

Besonders bei der Arbeit mit großen Datenbeständen oder wenn Daten veröffentlicht werden sollen, gewinnt ein einheitliches Datenmodell an Bedeutung. Ziel ist es dabei, Daten nach einem festgelegten System zu erfassen und abzuspeichern, damit diese auch von anderen verwendet werden können. Das lässt sich anhand von Personendatensätzen verdeutlichen: Das Geburtsdatum einer Person kann unter unterschiedlichen Bezeichnungen wie `Geburtstag`, `Geburtsdatum`, `birthday` oder `birthdate` erfasst sein. Für Menschen ist leicht zu erkennen, dass diese Daten den gleichen Sachverhalt repräsentieren. Computer können die Bedeutung nicht erkennen, sodass sich verschiedene Datensätze nicht ohne Weiteres verbinden lassen.

[23] Siehe Neo4J (2022; https://neo4j.com/).

[24] Siehe Solid IT (2022; https://db-engines.com/).

[25] Siehe OMG (2022; https://uml.org/).

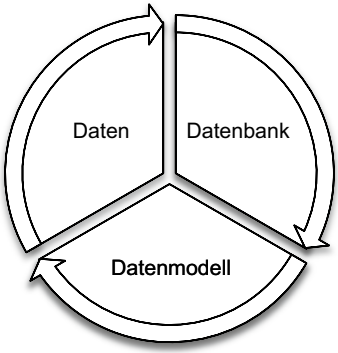


Abb. 3.6 Zusammenhang zwischen Daten, Datenbanken und Datenmodellen. (Quelle: eigene Darstellung)

Ein Lösungsansatz für diese Herausforderung stellt das Resource Description Framework (RDF) dar (siehe W3C 2014). Es umfasst drei wesentliche Konzepte:

1. **Aussagen** setzen sich aus drei Elementen zusammen: einem Subjekt, einem Prädikat und einem Objekt. Ein Geburtsdatum lässt sich wie folgt formulieren: Ada (Subjekt) ist geboren (Prädikat) am 10. Dezember 1815 (Objekt).[26] Im Prinzip lassen sich alle Daten als solche Tripel formalisieren.[27] Mehrere Aussagen lassen sich als Netzwerke bzw. Graphen auffassen. An einem Datum sind zum Beispiel mehrere Personen geboren, sodass verschiedene Personen über das Datum zumindest formal in Beziehung zueinanderstehen.
2. Die Elemente eines Tripels können über **Vokabulare** eindeutig bezeichnet werden. Vokabulare beinhalten festgelegte Ausdrücke für bestimmte Kategorien wie Geburtstage. Ein verbreitetes Vokabular zur Erfassung von Personen ist etwa Friends of a Friend (FOAF), in welchem Prädikate wie Namen, Adressen oder Bekanntschaftsbeziehungen standardisiert sind. Ein Geburtsdatum würde hier mit dem Schlüssel `foaf:birthday` erfasst werden. Weitere Vokabulare werden von schema.org oder DBPedia gepflegt, wobei ein Geburtsdatum über die Schlüssel `schema:birthDate` bzw. `dbo:birthDate` angegeben wird.[28]

[26] Diese Struktur stimmt nicht zwangsläufig mit den grammatischen Kategorien der Linguistik überein.

[27] Die Umformung von Datensätzen in diese Struktur kann über das Umformen vom Wide- in das Long-Format erreicht werden (siehe auch Kap. 4).

[28] Siehe Brickley und Miller (2014; http://xmlns.com/foaf/spec), Google et al. (2022; https://schema.org) und DBPedia Association (2021; http://dbpedia.org).

Eine Zusammenstellung untereinander verbundener Begriffe, beispielsweise dass Personen über ein Geburtsdatum und ein Sterbedatum verfügen, wird Ontologie genannt, wobei das Begriffssystem mit Sprachen wie der Web Ontology Language (OWL) formalisiert wird.[29]

3. Bezeichnungen werden entweder in der Form von Zeichenketten (= Literale) oder als eindeutige **Uniform Resource Identifier (URI)**[30] angegeben. Die Präfixe `foaf`, `schema` und `dbo` sind lediglich Abkürzungen, der vollständige URI für `foaf:birthday` lautet beispielsweise `http://xmlns.com/foaf/0.1/birthday`. Selbst wenn verschiedene Dienste für die gleichen Bedeutungen nicht die gleichen Bezeichnungen verwenden, können die Vokabulare ineinander übersetzt werden.[31]

Das Resource Description Framework bringt sehr unterschiedliche Datenmodelle auf den kleinsten gemeinsamen Nenner. Aus dieser Kleinteiligkeit ergibt sich der Nachteil, dass die Daten kaum noch durch Menschen erfasst werden können. Das ist aber auch nicht Ziel dieser Technologie. Vielmehr geht es darum, Daten in maschinenlesbarer Form mit Bedeutung zu versehen und daraus neue Aussagen abzuleiten – sodass nicht Menschen, sondern Maschinen die Daten effizient verknüpfen oder lesen können. Die Daten können dazu in Triple Stores – eine Form nichtrelationaler Datenbanken – abgelegt und über Abfragesprachen wie SPARQL abgefragt werden. Dabei werden Aussagen mit Platzhaltern und Bedingungen formuliert. Eine Abfrage kann zum Beispiel festlegen, dass die Subjekte vom Typ Wissenschaftler:in sein sollen: `?person rdf:type dbo:Scientist`. Mit einer Kombination verschiedener Tripel lässt sich erfragen, welche Wissenschaftler:innen heute Geburtstag haben (Abb. 3.7).

---

[29] Eine Visualisierung von Begriffsnetzwerken bietet beispielsweise WebVOWL (Link et al. 2019; http://vowl.visualdataweb.org/webvowl.html) und über das W3C Wiki (W3C 2021; https://www.w3.org/wiki/Search_engines) können entsprechende Vokabulare gefunden werden. Die Basic Formal Ontology (BFO) stellt darüber hinaus ein abstraktes Begriffssystem zur Verfügung, das möglichst universell einsetzbar ist (Ruttenberg 2020; http://basic-formal-ontology.org/).

[30] Eine spezielle Form von URIs sind Uniform Resource Locators (URLs, siehe Abschn. 2.1), das heißt Webadressen wie https://de.wikipedia.org/wiki/Ada_Lovelace. Hiermit wird eine Webseite eindeutig identifiziert. Durch die Angabe „https://“ ist zudem klar, dass auf diese Resource über das Web (https-Protokoll) zugegriffen werden kann. International Resource Identifiers (IRIs) sind dagegen eine Erweiterung von URIs um Sonderzeichen und Zeichen anderer Sprachen.

[31] Die Vokabulare selbst werden über Metasprachen wie die Web Ontology Language (OWL) oder RDF Schema (RDFS) beschrieben und lassen sich darüber untereinander verbinden.

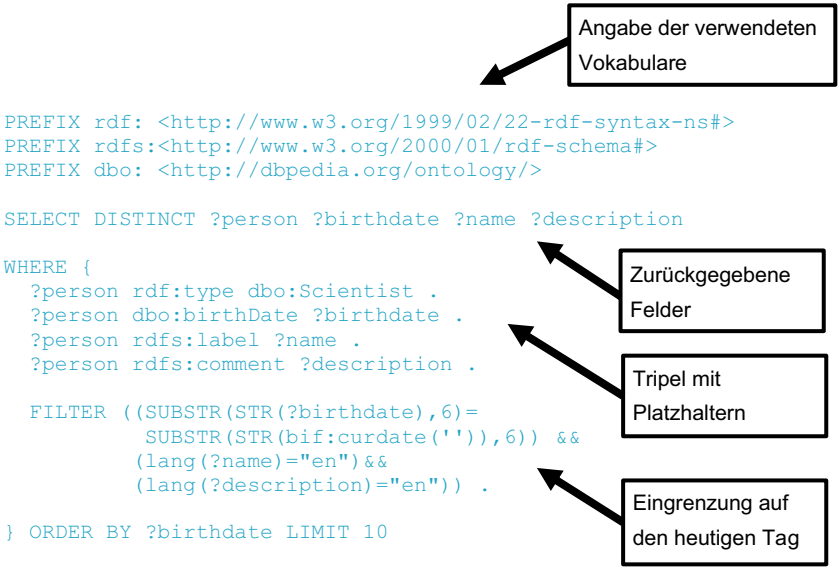


Abb. 3.7 SPARQL-Abfrage. Die Abfrage gibt bis zu zehn Wissenschaftler:innen zurück, die heute Geburtstag haben. Dazu werden drei Vokabulare (rdf, rdfs, dbo) verwendet. Sie können diese Abfrage unter OpenLink Software (2022; http://dbpedia.org/sparql) ausprobieren. Datengrundlage sind Wikipedia-Artikel. (Quelle: Eigene Darstellung auf Grundlage von Sack und Koutraki (2017))

Im Web frei verfügbare RDF-Daten werden als Linked Open Data (LOD) bezeichnet. Insbesondere in den Digital Humanities werden Datenbestände zunehmend als Linked Open Data verfügbar gemacht und mit anderen Datenbeständen verknüpft (Abb. 3.8). Im Kontext wissenschaftlicher Projekte eignet sich RDF erstens zur strukturierten Datenerfassung, etwa um systematisch bestimmte Wissensbereiche zu erfassen. Diese sogenannten Knowledge Graphs gruppieren sich um Entitäten wie Personen, Orte oder aber auch Softwares[32] und erfassen die Beziehungen zwischen den Entitäten in Form von Verwandtschaften, Einbindungen in Organisationen oder Erwähnungen in Texten. Zweitens können diese Daten abgefragt werden, um etwa Stichproben von Klöstern oder Schauspieler:innen zu ziehen (für ein Beispiel siehe Burggraaff und Trilling 2020).

Umfangreiche Datenbestände finden sich vor allem bei DBPedia und Wikidata. DBPedia extrahiert Daten aus Wikipedia-Artikeln. Umgekehrt werden struktu-

[32] Siehe zum Beispiel Schindler et al. (2021; https://data.gesis.org/somesci/) für die Erfassung von Softwares, die in wissenschaftlichen Publikationen erwähnt werden.

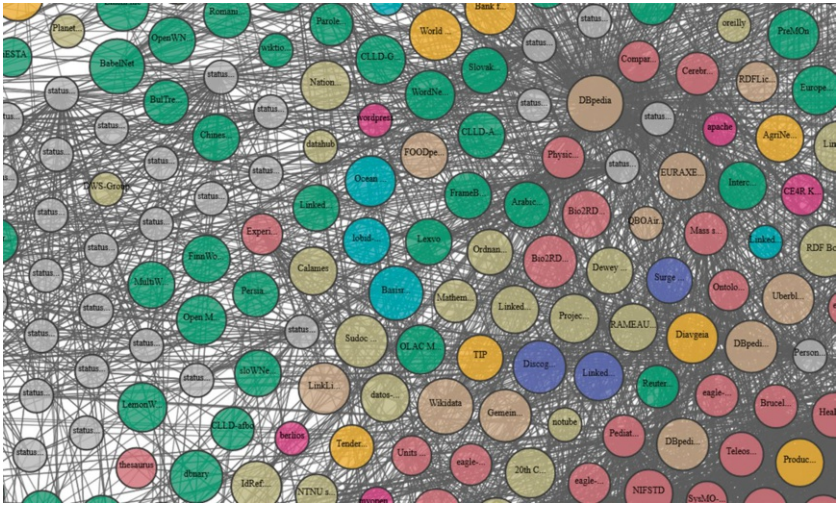


Abb. 3.8 The Linked Open Data Cloud. In der Linked Open Data Cloud wird erfasst, wie die Vokabulare und Datensätze verschiedener Anbieter zusammenhängen. (Quelle: McCrae (2021; https://lod-cloud.net))

rierte Daten aus Wikidata in die Wikipedia eingebunden. Geografische Datenbestände werden zum Beispiel durch GeoNames veröffentlicht.[33] RDF definiert dabei nicht, in welchem Format diese Daten abgespeichert werden – für diese sogenannte Serialisierung wird auf Formate wie JSON-LD, XML/RDF oder Textformate wie Turtle[34] aufgebaut. Viele Anbieter stellen mehrere Formate zur Verfügung (☛ *Repositorium*).

Im Web findet man RDF-kompatible Daten nicht nur als eigenständige Dateien, sondern auch direkt eingebettet in die Quelltexte von Webseiten (Abb. 3.9). RDF ist damit eine wichtige Grundlage für das Semantic Web,[35] womit das Ziel verfolgt wird, Daten miteinander zu verknüpfen und mit maschinenlesbarer Bedeutung zu versehen. Webseitenbetreiber können auf diese Weise strukturiert und standardi-

[33] Siehe Wick (2022; http://www.geonames.org/).

[34] Einige Formate gibt es in unterschiedlicher Komplexität: Turtle ist eine Untermenge von Notation3 und eine Übermenge von N-Triples.

[35] Eine Anlaufstelle zum Verständnis des Semantic Web ist W3C (2019; https://www.w3.org/2001/sw/wiki/). Zur Einbindung strukturierter Daten in Suchmaschinen siehe auch Google (2022b; https://developers.google.com/search/docs/advanced/structured-data/intro-structured-data).


```
<div itemscope itemtype="http://data-vocabulary.org/Movie">
  …
  <h1 itemprop="name">
     Star Wars: Der Aufstieg Skywalkers
  </h1>
   …
  <time itemprop="startDate"
   datetime="2019-12-18T17:00:00+01:00">17:00</time>
  …
</div>
```

Abb. 3.9 Einbettung von strukturierten Daten in Webseiten mit Microdata. Das HTML-Element eines Films ist hier mit dem leeren Attribut `itemscope` markiert, alle untergeordneten Angaben beziehen sich darauf. Eigenschaften werden durch das `itemprop`-Attribut ausgezeichnet. Die Ausprägungen sind in den Elementen enthalten. (Quelle: CineStar (2019))

siert die Öffnungszeiten eines Unternehmens, Geburtsdaten historischer Personen, das Kinoprogramm, Angaben zu den Autor:innen einer Webseite und vieles mehr bereitstellen. Schaut man sich den Quelltext einer Webseite genauer an, begegnen einem insbesondere folgende Formate:

- In den Metadaten einer Webseite finden sich häufig Angaben mit standardisierten Vokabularen wie Dublin Core,[36] unter anderem für die Beschreibung der Autor:innen oder des Änderungsdatums einer Seite. Mit dem Open Graph Protocol[37] werden in den Metatags zudem Daten erfasst, die beispielsweise in Messengern beim Teilen von Links eine Vorschau der Webseite erzeugen.

[36] Siehe DCMI (2022; https://www.dublincore.org/).

[37] Siehe Facebook (2020; https://ogp.me/).

- Die bestehende HTML-Struktur kann um spezielle Attribute erweitert werden. RFDa[38] definiert Attribute wie `vocab`, `typeof` und `property` zur Einbettung von RDF-Daten. Eine Alternative ist Microdata,[39] das ebenfalls Attribute festlegt, die in den HTML-Code eingefügt werden. Benötigt werden vor allem `itemscope`, `itemprop` und `content`.
- Mikroformate wie hCard (Orte, Kontaktinformationen …) und hReview (Bewertungen von Büchern, Musik, Restaurants …) sind auf bestimmte Datensorten spezialisiert.[40] Sie verwenden die in HTML ohnehin definierten Standardattribute `class`, `rel` und `rev`. Mikroformate sind schnell zu erlernen und eignen sich damit für einfache Anwendungsfälle.

Einen Eindruck der Daten, die in eine Webseite eingebettet sind, geben Tools wie JSON-LD Playground.[41] Über Browser-Plugins[42] lassen sich die in eine Webseite eingebundene Daten auch direkt während des Surfens anzeigen und erkunden. Diese Daten können in der Regel in verschiedenen Formaten (= Serialisierungen von RDF) heruntergeladen werden. Ohne den Einsatz spezieller Tools können in Webseiten eingebettete Daten für wissenschaftliche Analysen über Webscraping (siehe Abschn. 7.1) ausgelesen werden.

Die verschiedenen Technologien und Datenbestände des Semantic Web erscheinen zunächst recht heterogen und unübersichtlich. Gute Anlaufpunkte für solche unübersichtlichen Themen sind sogenannte Awesome Lists – eine Zusammenstellung relevanter Themenaspekte bietet die Liste „Awesome Semantic Web".[43] Dort finden Sie Verweise auf entsprechende wissenschaftliche Zeitschriften und erhalten Anregungen, inwiefern sich mit dieser Technologie wissenschaftliche Analysen durchführen lassen – das Resource Description Framework eröffnet eine Unmenge

[38] Eine Einführung in RDFa findet sich unter W3C (2015; https://www.w3.org/TR/xhtml-rdfa-primer/). Weitere Informationen siehe auch RDFa (2022; http://rdfa.info/).

[39] Microdata ist im HTML-Standard definiert, siehe Hickson et al. (2022; https://html.spec.whatwg.org/#microdata).

[40] Die Standardisierung findet über ein Wiki statt, siehe Microformats (2022; http://microformats.org/).

[41] Siehe JSON-LD (2022; https://json-ld.org/playground/). Ein weiteres Tool ist der HTML Structured Data Extractor (Herman 2012; https://www.w3.org/2012/sde/).

[42] Zum Beispiel der OpenLink Structured Data Sniffer (OpenLink Software 2021; https://osds.openlinksw.com/). Öffnen Sie beispielsweise die Seite der BBC mit klassischen Comedy-Programmen (https://www.bbc.co.uk/programmes/) und inspizieren Sie die Datensammlung über das Plugin.

[43] Siehe Semantalytics (2022; https://github.com/semantalytics/awesome-semantic-web).

an neuen Fragestellungen und Ansätzen zur Erfassung, Verknüpfung und Analyse von Wissen.

Übungsfragen

1. Was unterscheidet 1983 von „1983“ und warum ist dieser Unterschied wichtig?
2. Was legt die Zeichenkodierung einer Datei fest?
3. Mit welchen Mitteln können Markdown-Dateien formatiert werden?
4. Welche Trennzeichen werden in CSV-Dateien eingesetzt?
5. Wie finden Sie heraus, welches Trennzeichen in einer CSV-Datei verwendet wird? Überprüfen Sie die Beispiele im ☛ *Repositorium*!
6. Worin unterscheiden sich die Formate CSV, JSON und XML?
7. Was sind Unterschiede zwischen relationalen und nichtrelationalen Datenbanken?
8. Suchen Sie sich eine Datenbank (siehe Abschn. 2.3) und bestimmen Sie, ob es sich hierbei um eine relationale oder eine nichtrelationale Datenbank handelt!
9. Wie ist ein RDF-Tripel aufgebaut (☛ *Repositorium*)?
10. Was ist das Semantic Web?

Weiterführende Literatur

Russell, M. A. (2014). *Mining the social web. Data mining Facebook, Twitter, Linkedin, Google+, GitHub, and more.* (2. Aufl.). Sebastopol: O'Reilly.

Segaran, T., Evans, C. & Taylor, J. (2009). *Programming the Semantic Web.* Sebastopol: O'Reilly.

Silberschatz, A., Korth, H. F. & Sudarshan, S. (2020). *Database system concepts.* New York: McGraw-Hill.

Literatur

Ackoff, R. (1989). From data to wisdom. Presidential address to ISGSR, June 1988. *Journal of Applied Systems Analysis*, 16, 3–9.

Ballsun-Stanton, B. (2010). Asking about data: Experimental philosophy of Information Technology. In *5th International Conference on Computer Sciences and Convergence Information Technology, ICCIT 2010* (S. 119–124). Piscataway: Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/ICCIT.2010.5711041

Bray, T. (Internet Engineering Task Force IETF.). (2014). *The JavaScript Object Notation (JSON) Data Interchange Format*. Zugriff am 24.05.2022. https://datatracker.ietf.org/doc/html/rfc7159

Brickley, D. & Miller, L. (2014). *FOAF Vocabulary Specification 0.99*. Zugriff am 24.05.2022. http://xmlns.com/foaf/spec/

Burggraaff, C. & Trilling, D. (2020). Through a different gate: An automated content analysis of how online news and print news differ. *Journalism*, *21*(1), 112–129. https://doi.org/10.1177/1464884917716699

CineStar. (2019). *Star Wars: Der Aufstieg Skywalkers*. Zugriff am 01.06.2022. https://www.cinestar.de/kino-greifswald/film/star-wars-episode-ix

Cone, M. (2022). *Markdown Cheat Sheet. A quick reference to the Markdown syntax*. Zugriff am 24.05.2022. https://www.markdownguide.org/cheat-sheet/

DBpedia Association. (2021). *DBpedia*. Zugriff am 24.05.2022. https://www.dbpedia.org/

DCMI: Dublin Core Metadata Innovation. (2022). *Dublin Core*. Zugriff am 30.05.2022. https://www.dublincore.org/

Elastic. (2022). Elasticsearch (Version 8.2) [Computer software]. https://www.elastic.co/elasticsearch

Facebook. (2020). *The Open Graph protocol*. Zugriff am 02.07.2020. https://ogp.me/

Google. (2022a). Cloud Bigtable [Computer software]. https://cloud.google.com/bigtable

Google. (2022b). *Google Search Central. Understand how structured data works*. Zugriff am 30.05.2022. https://developers.google.com/search/docs/advanced/structured-data/intro-structured-data

Google, Microsoft, Yahoo & Yandex. (2022). *Schema.org*. Zugriff am 24.05.2022. https://schema.org/

Herman, I. (W3C Semantic Web, Hrsg.). (2012). *HTML Structured Data Extractor to RDF*. Zugriff am 30.05.2022. https://www.w3.org/2012/sde/

Hickson, I., Pieters, S., van Kesteren, A., Jägenstedt, P. & Denicola, D. (WHATWG, Hrsg.). (2022). *HTML. Living Standard*. Zugriff am 30.05.2022. https://html.spec.whatwg.org/multipage/

Holistics.io. (2022). *dbdiagram.io. Draw Entity-Relationship Diagrams, Painlessly*. Zugriff am 01.06.2022. https://dbdiagram.io/

ISO/IEC 2382. (2015). Information technology – Vocabulary.

JSON-LD. (2022). *JSON-LD Playground*. Zugriff am 30.05.2022. https://json-ld.org/playground/

Kluge, F. (2002). *Etymologisches Wörterbuch der deutschen Sprache* (24., durchges. und erw. Aufl.). Berlin: de Gruyter.

Link, V., Lohmann, S., Marbach, E., Negru, S. & Wiens, V. (2019). WebVOWL: Web-based Visualization of Ontologies (Version 1.1.7) [Computer software]. http://vowl.visualdata-web.org/webvowl.html

MariaDB Foundation. (2022). MariaDB Server. The open source relational database (Version 10.9.0) [Computer software]. https://mariadb.org/

McCrae, J. P. (Insight Centre for Data Analytics, Hrsg.). (2021). *The Linked Open Data Cloud*. Zugriff am 01.06.2022. https://lod-cloud.net/

Meier, W. (2003). eXist: An open source native XML database. In G. Goos, J. Hartmanis, J. van Leeuwen, A. B. Chaudhri, M. Jeckle, E. Rahm et al. (Hrsg.), *Web, Web-Services, and Database Systems* (S. 169–183). Berlin: Springer. https://doi.org/10.1007/3-540-36560-5_13

Microformats. (2022). *Welcome to the microformats wiki!* Zugriff am 30.05.2022. https://microformats.org/wiki/Main_Page

MongoDB. (2022) (Version 4.0.8) [Computer software]. https://www.mongodb.com/

Neo4j. (2022). Neo4j (Version 4.4.7) [Computer software]. https://neo4j.com/
OMG. (2022). *Unified Modeling Language (UML)*. Zugriff am 24.05.2022. https://www.uml.org
OpenLink Software. (2021). *The OpenLink Structured Data Sniffer (OSDS)*. Zugriff am 30.05.2022. https://osds.openlinksw.com/
OpenLink Software. (2022). *SPARQL Query Editor*. Zugriff am 01.06.2022. https://dbpedia.org/sparql/
Oracle. (2021). MySQL (Version 8.0) [Computer software]. https://www.mysql.com/
RDFa. (2022). *Linked Data in HTML*. Zugriff am 30.05.2022. http://rdfa.info/
Redis. (2022). Redis (Version 7.0) [Computer software]. https://redis.io/
Rowley, J. (2007). The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science*, *33*(2), 163–180. https://doi.org/10.1177/0165551506070706
Ruttenberg, A. (2020). *Basic Formal Ontology (BFO)*. http://basic-formal-ontology.org/
Sack, H. & Koutraki, M. (2017). *Information Service Engineering,* openHPI. https://open.hpi.de/courses/semanticweb2017/
Schindler, D., Bensmann, F., Dietze, S. & Krüger, F. (2021). *SoMeSci – A 5 Star Open Data Gold Standard Knowledge Graph of Software Mentions in Scientific Articles*. Zugriff am 30.05.2022. https://data.gesis.org/somesci/
Semantalytics. (2022). *Awesome Semantic Web. A curated list of various semantic web and linked data resources.* Zugriff am 30.05.2022. https://github.com/semantalytics/awesome-semantic-web
Solid IT. (2022). *DB-Engines. Informationen zu relationalen und NoSQL Datenbankmanagementsystemen*. Zugriff am 24.05.2022. https://db-engines.com/
Text Encoding Initiative. (2022). *Text Encoding Initiative*. Zugriff am 24.05.2022. https://tei-c.org/
Unicode. (2021). *Submitting Emoji Proposals*. Zugriff am 24.05.2022. https://unicode.org/emoji/proposals.html
W3C. (2013). *Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation*. Zugriff am 24.05.2022. https://www.w3.org/TR/xml/
W3C. (2014). *RDF 1.1 Primer. W3C Working Group Note*. Zugriff am 24.05.2022. https://www.w3.org/TR/rdf11-primer/
W3C. (2015). *RDFa 1.1 Primer – Third Edition. Rich Structured Data Markup for Web Documents. W3C Working Group Note*. Zugriff am 30.05.2022. https://www.w3.org/TR/xhtml-rdfa-primer/
W3C. (2019). *Semantic Web*. Zugriff am 30.05.2022. https://www.w3.org/2001/sw/wiki/Main_Page
W3C. (2021). *Search engines*. Zugriff am 24.05.2022. https://www.w3.org/wiki/Search_engines
W3Schools. (2022a). *HTML Element Reference*. Zugriff am 24.05.2022. https://www.w3schools.com/tags/default.asp
Wick, M. (Unxos GmbH, Hrsg.). (2022). *GeoNames*. Zugriff am 24.05.2022. https://www.geonames.org/