

---

# Weiterentwicklung eines WordPress-Plugins zur gamifizierten Spendenverteilung: Gutenberg-Integration und Plugin-Architektur am Beispiel von “Charigame” in Kooperation mit der elancer-team GmbH

Bachelorarbeit zur Erlangung des akademischen Grades  
*Bachelor of Science*  
im Studiengang Medieninformatik  
an der Fakultät für Informatik und Ingenieurwissenschaften  
der Technischen Hochschule Köln

vorgelegt von: Christian Krenn  
chrstian.krenn@smail.th-koeln.de

eingereicht bei: Prof. Dr. Hoai Viet Nguyen  
Zweitgutachter\*in: Tobias Derksen

Gummersbach, 03.09.2025

## **Kurzfassung/ *Abstract***

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>IV</b>
<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>Glossary</b>	<b>VI</b>
<b>Abkürzungsverzeichnis</b>	<b>VII</b>
<b>Symbolverzeichnis</b>	<b>VIII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Problemstellung und Motivation . . . . .	1
1.2 Zielsetzung der Arbeit . . . . .	1
1.3 Aufbau der Arbeit . . . . .	1
<b>2 Theoretische Grundlagen</b>	<b>3</b>
2.1 WordPress als Content-Management-System . . . . .	3
2.2 Plugin-Entwicklung mit WordPress . . . . .	3
2.3 Gutenberg-Editor: Konzept und technische Grundlagen . . . . .	3
2.4 Gamification im Kontext digitaler Anwendungen . . . . .	3
2.5 Überblick über Spendenverteilung und Charity-Plattformen . . . . .	3
<b>3 Projektkontext</b>	<b>6</b>
3.1 Das Projekt Charigame . . . . .	7
3.2 Anforderungen der elancer-team GmbH . . . . .	7
3.3 Ausgangszustand des bestehenden Plugins . . . . .	8
<b>4 Konzeption und Design</b>	<b>18</b>
4.1 Anforderungsanalyse . . . . .	18
4.1.1 Funktionale Anforderungen . . . . .	18
4.1.2 Nicht-funktionale Anforderungen . . . . .	19
4.1.3 Stakeholder-Analyse (elancer-team GmbH) . . . . .	19
4.2 Plugin-Architektur . . . . .	20
4.2.1 Modularer Aufbau und Komponenten . . . . .	20
4.2.2 Datenmodell und Datenbankstruktur . . . . .	20

4.2.3	API-Design und Schnittstellen . . . . .	21
4.3	Gutenberg-Integration . . . . .	21
4.3.1	Block-Design und -Funktionalität . . . . .	21
4.3.2	Editor-Integration . . . . .	21
<b>5</b>	<b>Implementierung</b>	<b>22</b>
5.1	Entwicklungsumgebung und Tools . . . . .	22
5.1.1	Plugin-Softwarearchitektur . . . . .	22
5.1.2	Backend-Entwicklung . . . . .	23
	<b>Literatur</b>	<b>24</b>
	<b>Anhang</b>	<b>27</b>

## Tabellenverzeichnis

3.1	Eine einfache Tabelle . . . . .	10
-----	---------------------------------	----

## Abbildungsverzeichnis

3.1	Vielleicht handelt es sich hierbei um Kunst? . . . . .	9
3.2	Eine schöne Grafik, die im Quellcode erzeugt wird! . . . . .	10
3.3	Beispiel für ein listing . . . . .	12

## Glossar

**Hund** Behaartes, vierbeiniges Säugetier. Bester Freund des Menschen. 13

## Abkürzungsverzeichnis

**SVM** support vector machine. 13



## Symbolverzeichnis

$\vec{F}$  Kraft, vektorielle Größe. 13

# 1 Einleitung

## 1.1 Problemstellung und Motivation

## 1.2 Zielsetzung der Arbeit

## 1.3 Aufbau der Arbeit

Das vorliegende Dokument kann als Muster und Anleitung für wissenschaftliche Abschlussarbeiten verwendet werden. Es beruht ursprünglich auf einem Leitfaden, den Prof. Dr. Stephan Freichel als Prüfungsausschussvorsitzender für die Studiengänge B.Sc. Logistik und M.Sc. *Supply Chain and Operations Management* an der Fakultät für Fahrzeugsysteme und Produktion erstellt hat.

Der Text in dieser Vorlage beschreibt allgemeine formale Anforderungen, insbesondere zum Inhaltsverzeichnis, zum Einfügen von Quellenverweisen und zum Erstellen eines Literaturverzeichnisses.

Die Vorlage kann fachübergreifend als Musterdatei für Abschlussarbeiten an der TH Köln verwendet werden. Allerdings müssen Sie dann unbedingt klären, ob sie den Konventionen in Ihrem Studienfach entspricht.

Für die Möglichkeit eines fachunabhängigen Gebrauchs wurde das Dokument von Maria-Anna Worth (i. R.) und Susanne Neuzerling (Hochschulreferat Planung und Controlling) inhaltlich überarbeitet und modifiziert. Eine erneute Überarbeitung und Aktualisierung erfolgte durch Andreas Bissels (Schreibzentrum). Frau Katharina Bata hat die Dokumentvorlage in LaTeX erstellt. Zuletzt wurde diese Vorlage 2023 von André Ulrich und Jan Salmen überarbeitet und um diverse Hinweise speziell zur Gestaltung mit L<sup>A</sup>T<sub>E</sub>X ergänzt (siehe Kapitel 3).

Zur Vorbereitung auf Ihre Abschlussarbeit empfehlen wir Ihnen die Angebote des Schreibzentrums der Kompetenzwerkstatt<sup>1</sup>; hierzu gehören sowohl eine Schreibberatung als auch Schreibkurse. Das Schreibzentrum ist Ihre Anlaufstelle an der TH Köln in Fragen rund um das wissenschaftliche Schreiben.

---

<sup>1</sup><https://www.th-koeln.de/schreibzentrum>

Sichern Sie diese Dokumentvorlage bitte zweifach auf Ihrem Rechner: Einmal, um weiterhin auf den hier dargestellten Inhalt zugreifen zu können, und ein zweites Mal, um sie mit Ihrer eigenen Abschlussarbeit zu überschreiben.

*Bitte beachten Sie: Die Vorlage ersetzt nicht die spezifischen Vorgaben der jeweiligen Prüfungsausschüsse. Sollte es in Ihrem Fach besondere formale Vorgaben geben, so gelten diese.*

Köln, August 2023

## **2 Theoretische Grundlagen**

### **2.1 WordPress als Content-Management-System**

### **2.2 Plugin-Entwicklung mit WordPress**

### **2.3 Gutenberg-Editor: Konzept und technische Grundlagen**

### **2.4 Gamification im Kontext digitaler Anwendungen**

### **2.5 Überblick über Spendenverteilung und Charity-Plattformen**

In diesem Kapitel finden Sie grundlegende Hinweise zum formalen Aufbau Ihrer Arbeit.

**Reihenfolge** Eine wissenschaftliche Arbeit besteht in der Regel aus den folgenden Teilen:

1. Deckblatt
2. Kurzfassung/Abstract (optional)
3. Inhaltsverzeichnis
4. Abbildungs- und Tabellenverzeichnis (auch am Ende üblich)
5. Abkürzungsverzeichnis (auch am Ende üblich)
6. Einleitung
7. Hauptteil
8. Zusammenfassung/Fazit
9. Literaturverzeichnis
10. Anhänge (optional)

## 11. Erklärung

**Deckblatt** Das Deckblatt beinhaltet: Titel der Arbeit, Art der Arbeit, Verfasser\*in, Matrikelnummer, Abgabetermin, Betreuer\*in sowie Zweitgutachter\*in. Das Deckblatt wird bei Arbeiten, die länger sind als 15 Seiten, bei der Seitenanzahl zwar mitgezählt, jedoch nicht nummeriert. **Inhaltsverzeichnis** Wir empfehlen eine Dezimalgliederung wie in diesem Dokument angelegt. Werden innerhalb eines Kapitels Unterüberschriften verwendet, müssen mindestens zwei vorhanden sein: wo ein 2.1 ist, muss es ein 2.2 geben.

Das Inhaltsverzeichnis enthält immer die Seitenangaben zu den aufgelisteten Gliederungspunkten; es wird dabei aber selbst nicht im Inhaltsverzeichnis aufgelistet. Die Seiten, die das Inhaltsverzeichnis selbst einnimmt, können römisch gezählt werden.

Für eine Abschlussarbeit ist eine Gliederungstiefe von wenigstens drei Ebenen üblich. In der Regel werden nur bis zu vier Ebenen vorne im Inhaltsverzeichnis abgebildet. Hier sollten Sie aber unbedingt die Gepflogenheiten in Ihrem Fach berücksichtigen und ggf. in Erfahrung bringen. **Abbildungsverzeichnis und Tabellenverzeichnis** Abbildungen und Tabellen werden in entsprechenden Verzeichnissen gelistet. In dieser Vorlage erscheinen sie direkt nach dem Inhaltsverzeichnis. Dann können die entsprechenden Seiten römisch gezählt werden. Die Verzeichnisse können jedoch auch am Ende der Arbeit vor oder hinter dem Literaturverzeichnis stehen. Dann werden sie regulär mit Seitenzahlen versehen. **Abkürzungsverzeichnis** Die Zahl der Abkürzungen sollte übersichtlich bleiben. Das Abkürzungsverzeichnis enthält lediglich wichtige fachspezifischen Abkürzungen in alphabetischer Reihenfolge, insbesondere Abkürzungen von Organisationen, Verbänden oder Gesetzen. Gängige Abkürzungen wie „u. a.“, „z. B.“, „etc.“ werden nicht aufgenommen.

Zur technischen Umsetzung mit L<sup>A</sup>T<sub>E</sub>X vergleiche auch Abschnitt 3.3. **Literaturverzeichnis** Das Literaturverzeichnis wird alphabetisch nach Autorennamen geordnet. Es enthält alle im Text zitierten Quellen – und nur diese. Mehrere Schriften einer Person werden nach Erscheinungsjahr geordnet. Schriften derselben Person aus einem Erscheinungsjahr müssen Sie selbst unterscheidbar machen. In den Ingenieurwissenschaften wird zusätzlich häufig ein Nummern- oder Autorenkürzel dem Namen in eckigen Klammern vorangestellt. Mehr hierzu und weitere wichtige Regeln des Zitierens lernen Sie in den E-Learning-Kursen des Schreibzentrums<sup>1</sup> kennen.

Zur Verwaltung der verwendeten Literatur eignen sich entsprechende Softwaretools wie Citavi oder Zotero, die mit verschiedenen Textverarbeitungsprogrammen kompatibel sind. **Rechtschreibung, Grammatik** Achten Sie bei der Abgabe Ihrer Arbeit auf ein einwandfreies Deutsch bzw. Englisch. Wenn Fehler die Lesbarkeit beeinträchtigen, kann sich dies durchaus negativ auf die Note auswirken. Nutzen Sie daher unbedingt

---

<sup>1</sup>[https://ilu.th-koeln.de/goto.php?target=cat\\_52109&client\\_id=thkilu](https://ilu.th-koeln.de/goto.php?target=cat_52109&client_id=thkilu)

die Rechtschreibprüfung Ihres Textverarbeitungsprogramms, auch wenn diese nicht alle Fehler erkennt.

Für alle, die sich bei diesem Thema unsicher fühlen, empfehlen wir die E-Learning-Kurse des Schreibzentrums<sup>2</sup>. Wenden Sie sich ggf. auch an die Beauftragte für Studierende mit Beeinträchtigung<sup>3</sup>. **Umfang der Arbeit** Alle Fächer nennen verbindliche Angaben zu Unter- und Obergrenzen, die in der Regel eingehalten werden müssen. Verzeichnisse und Anhänge werden dabei in aller Regel nicht mitgezählt. In Einzelfällen – insbesondere bei empirischen Arbeiten – können abweichende Vereinbarungen mit der Betreuungsperson getroffen werden.

---

<sup>2</sup>[https://ilu.th-koeln.de/goto.php?target=cat\\_52109&client\\_id=thkilu](https://ilu.th-koeln.de/goto.php?target=cat_52109&client_id=thkilu)

<sup>3</sup>[https://www.th-koeln.de/studium/studieren-mit-beeintraechtigung\\_169.php](https://www.th-koeln.de/studium/studieren-mit-beeintraechtigung_169.php)

### 3 Projektkontext

Mit  $\text{\LaTeX}$  ist es verhältnismäßig einfach, Dokumente zu erstellen, die professionellen Ansprüchen genügen. Ein entscheidender Vorteil ist, dass der Nutzer fast nur den Inhalt beisteuert, während die korrekte äußere Form dann automatisch erzeugt wird.  $\text{\LaTeX}$  basiert auf  $\text{\TeX}$ , das von Donald Knuth entwickelt wurde [5]. Einige weitere Vorteile gegenüber gängiger Textverarbeitung:

**Frei/Plattformunabhängig:** Bei  $\text{\LaTeX}$  handelt es sich um freie Software. Es wird kein proprietärer Editor benötigt, um  $\text{\LaTeX}$ -Dokumente zu schreiben. Tatsächlich können die Dokumente auf *jedem* Rechner mit *jedem* beliebigen Editor bearbeitet werden.

**Reines Textformat:** Der Quelltext – die `tex`-Datei – ist ein reines Textformat. Dadurch eignen sich  $\text{\LaTeX}$ -Dokumente auch hervorragend zur Versionskontrolle mit beispielsweise git. Dies wiederum ermöglicht eine effiziente Zusammenarbeit mehrerer Autor\*innen.

**Aufteilen großer Dokumente:** Der Quelltext großer Dokumente, wie beispielsweise von Projektarbeiten, kann auf mehrere Dateien aufgeteilt werden. So können beispielsweise mehrere Personen an jeweils einem eigenen Kapitel arbeiten. Aufgrund der beiden oberen Punkte wird es auch nicht zu Kompatibilitätsproblemen kommen.

**Trennen von Layout/Inhalt:** Mit  $\text{\LaTeX}$  kann man explizit das Layout für das gesamte Dokument festlegen – oder die verwendete Dokumentklasse kümmert sich implizit darum. Zeitgemäße Textverarbeitung bietet mit Formatvorlagen zwar entsprechende Funktionalitäten; aber durch den programmatischen Ansatz mit  $\text{\LaTeX}$  kann noch genauer Einfluss auf das Layout genommen werden. Anschließend kann die ganze Konzentration auf das Schreiben gelegt werden.

**Professionelles Ergebnis:** Ein mit  $\text{\LaTeX}$  erzeugtes Dokument schaut professioneller aus, als ein entsprechendes, mit Textverarbeitung erzeugtes Dokument. Das gilt vor allem für mathematiklastige Dokumente. Aber auch andere Dokumente können von einem einheitlichen Layout, gleichmäßigem Grauwert des Fließtexts, stimmigeren Seitenumbrüchen und hochwertigen Vektorgraphiken profitieren – um nur mal einige Punkte zu nennen.

**Vielseitig einsetzbar:** Mit  $\text{\LaTeX}$  können nicht nur „einfache“ Dokumente erzeugt werden. Es existieren unzählige Dokumentklassen, die beispielsweise auch das Erstellen von Präsentationen oder Postern ermöglichen.

In den folgenden Abschnitten 3.1 bis 3.3 wird auf diverse Aspekte eingegangen, die Sie beim Erstellen Ihres Dokuments berücksichtigen sollten.

## 3.1 Das Projekt Charigame

Sie definieren in Ihren  $\text{\TeX}$ -Dokumenten, was Ihre Inhalte sind (Text mit Gliederung, Bilder, Tabellen, Literaturverweise, ...) und wie diese jeweils grob aussehen sollen (z. B. Platzierung von Abbildungen mittels *Gleitumgebungen*, vgl. Abschnitt 3.3).

Beim Erstellen des endgültigen Dokuments wendet  $\text{\LaTeX}$  „unter der Haube“ eine ganze Menge Regeln an, die festlegen, wie das alles bestmöglich umgesetzt werden kann. Diese Regeln betreffen z. B. den Anteil von Text und Bildern pro Seite, Abstände innerhalb von Zeilen, aber auch Sonderfälle wie das Vermeiden einzelner Zeilen eines Abschnitts alleine auf einer Seite (sog. „Schusterjungen“ oder „Hurenkinder“).

Im Ergebnis kann es also passieren, dass z. B. Ihre Abbildungen „springen“, während Sie an Ihrem Text arbeiten. Das hat im Zweifel alles seine Richtigkeit und kann im Notfall am Ende noch optimiert werden.

In diesem Zusammenhang ist zu vermeiden, in den Gestaltungsprozess einzugreifen, indem z. B. manuell Zeilenumbrüche („`\newline`“ oder „`\\`“) eingefügt werden oder Abstände. Ausnahmen bitte nur in begründeten Fällen wie in dieser Vorlage bei der Gestaltung des Deckblatts.

Weitere Infos dazu, wie Sie mit dieser Vorlage hier weiterarbeiten können, finden Sie in Abschnitt 3.3.

## 3.2 Anforderungen der elancer-team GmbH

Wir nutzen in dieser Vorlage das Kapitel („`\chapter`“) als höchste Gliederungsebene. Danach kommen Abschnitte („`\section`“) und Unterabschnitte („`\subsection`“). Diese drei Ebenen werden nummeriert und erscheinen im Inhaltsverzeichnis. Falls Sie Ihren Text weiter gliedern wollen, gibt es noch den „`\paragraph`“-Befehl.

Bitte beachten Sie, dass im Text nie zwei Überschriften direkt aufeinander folgen sollten. Nach einer Überschrift kommt immer erst etwas Text (siehe z. B. die Kapitelanfänge hier auf Seite 3 und Seite 6). Für weitere Hinweise vgl. Abschnitt 2.5.



### 3.3 Ausgangszustand des bestehenden Plugins

Stellen im Text, an denen ein neuer Absatz beginnen soll, können im Quellcode durch „`\par`“ markiert werden. Wie diese Absätze im fertigen Dokument genau aussehen, wird durch den Parameter „`\parskip`“ in der Dokumentenklasse bestimmt – dazu mehr in Abschnitt 3.3. Das ist ein großer Vorteil von  $\text{\LaTeX}$ : Der Stil kann jederzeit für das gesamte Dokument einfach verändert werden.

Hinweis: Sie erhalten das gleiche Verhalten auch, wenn Sie im Quellcode statt des „`\par`“-Befehls eine leere Zeile stehen lassen. Vielleicht gefällt Ihnen das sogar noch besser. **Silbentrennung** Die automatische Silbentrennung in  $\text{\LaTeX}$  funktioniert grundsätzlich gut. Es kann aber immer mal kleinere Probleme und erwünschtes Verhalten geben. Wenn Sie die Trennung für ein bestimmtes Wort beeinflussen möchten, können Sie mit dem „`\hyphenation`“-Befehl manuell die erlaubten Trennstellen spezifizieren. So kann man insbesondere erreichen, dass bestimmte Wörter nie getrennt werden, was z. B. für Eigennamen unerwünscht sein könnte.

Zum Beispiel werden Wörter, die einen Bindestrich enthalten, *nur* dort getrennt, das kann dazu führen, dass Zeilen nicht richtig dargestellt werden können, was zu einer Warnung führt (siehe Abschnitt 3.3). In solche Fällen müssten Sie im Quellcode manuell zusätzlich Trennstellen angeben. **Aufzählungen** Nutzen Sie die Umgebungen

- „`\begin{itemize}`“ ... „`\end{itemize}`“
- „`\begin{enumerate}`“ ... „`\end{enumerate}`“
- „`\begin{description}`“ ... „`\end{description}`“
- „`\begin{labeling}`“ ... „`\end{labeling}`“

um schöne Listen zu erstellen. Auch hier gilt, dass das genaue Aussehen im Dokument global eingestellt wird, das können Sie jederzeit verändern, dazu mehr in Abschnitt 3.3. **Abbildungen** Wenn jemand Ihre fertige Arbeit in die Hände bekommt, kann es gut sein, dass sie/er zunächst grob durchblättert, dabei kaum Text liest, aber die Abbildungen anschaut. Aus dieser Erfahrung entstammt die „Regel“, dass man die wichtigsten Punkte der Arbeit auf diese Weise verstehen können sollte.

Abbildungen stehen nie alleine, sondern werden durch die Unterschrift (*caption*) beschrieben. Dabei sollte alles enthalten sein, was notwendig ist, um die Abbildung zu verstehen. Nur in Ausnahmefällen muss man in der Unterschrift auf den Text verweisen. Umgekehrt muss auf jede Abbildung mindestens ein Mal im Text verwiesen werden, dazu siehe auch Abschnitt 3.3.

In den folgenden beiden Abschnitten wird zwischen Bildern (in Abschnitt 3.3) und Vektorgrafiken (in Abschnitt 3.3) unterschieden, da es sich um ganz unterschiedliche Techniken handelt, die jeweils passend genutzt werden sollten.

Denken Sie daran, dass nicht alle Menschen alle Farben gleich gut sehen können. Etwa 10 % der Männer in Deutschland sind beispielsweise von einer Rot-Grün-Schwäche betroffen. Vielleicht wird Ihre Arbeit auch auf einem Schwarz-Weiß-Drucker gedruckt. Daher sollten Sie Abbildungen im besten Fall so gestalten, dass sie auch ohne Farben verständlich sind. **Bilder** Bilder können Sie mit „`\includegraphics`“ einbinden. Es reicht (und wird sogar empfohlen!), den Dateinamen ohne Endung und ohne Pfad anzugeben. Beim Kompilieren werden alle Verzeichnisse durchsucht, die im „`\graphicspath`“ angegeben sind. In aller Regel soll ein Bild nicht alleine im



Abbildung 3.1: Vielleicht handelt es sich hierbei um Kunst?

Dokument erscheinen, sondern in einer Umgebung, die die automatische Nummerierung sicherstellt, eine Bildunterschrift hinzufügt und schließlich ermöglicht, dass die Abbildung an einer optimalen Stelle platziert wird (daher auch die Bezeichnung „Gleitumgebung“. In diesem Fall ist das die „`\figure`“-Umgebung.

Für die Umgebung stellen wir ein, wo sie auftauchen darf (dazu siehe auch Abschnitt 3.1). Dabei steht **t** für ganz oben auf der Seite, **b** für ganz unten und **h** für „hier“, was also die Positionierung innerhalb des Texts meint. Falls Sie mal Platz sparen müssen, sind **t** und **b** zu bevorzugen.

Achtung: Viele Inhalte wie Formeln, Code, Diagramme, Visualisierung von Daten, usw. sollten *nicht* als Bild eingefügt werden, sondern in einer passenden Form. Dazu siehe den folgenden Abschnitt über Vektorgrafiken. **Vektorgrafiken** Einfache Abbildungen (z. B. Koordinatensysteme, Ablaufdiagramme, usw.) müssen Sie nicht als Bild einfügen. Stattdessen können diese im Quellcode direkt erzeugen können. Dafür bietet sich das mächtige „**tikz**“-Paket an.

Ein Vorteil ist, dass Ihr Dokument so kleiner bleibt. Aber auch, dass die Abbildungen i. d. R. hübscher aussehen. Das gilt insbesondere beim Betrachten am Bildschirm, da

sich Vektorgrafiken beliebig skalieren lassen. Das erlaubt es Ihnen sogar, Ihre Daten,

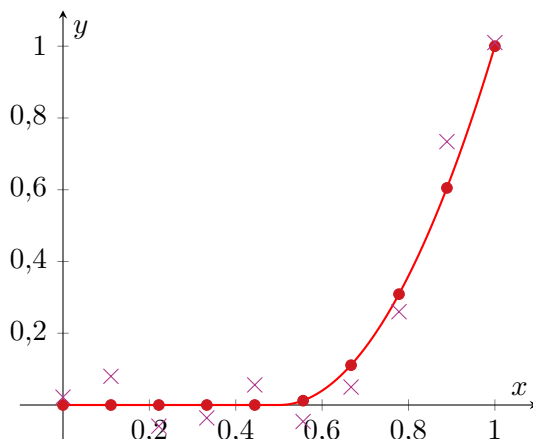


Abbildung 3.2: Eine schöne Grafik, die im Quellcode erzeugt wird!

z. B. aus Experimenten, separat zu halten und entsprechende Abbildungen dynamisch daraus zu generieren. Siehe dazu das Beispiel in Abbildung 3.2.

Sie finden ganz viel Beispiele zu TikZ natürlich im Internet. Außerdem gibt es ein aktuelles Buch [15]. **Tabellen** Grundsätzlich werden Tabellen in  $\text{\LaTeX}$  mit der „`\tabular`“-Umgebung gebaut. Das ist dann nur die Tabelle selbst, ohne Nummerierung und ohne Bildunterschrift. Das Prinzip ist also das gleiche wie bei Abbildungen (s. o.): Erst die Umgebung (hier „`\table`“), darin die Tabelle selbst. Vielleicht sind die

Überschrift links	Überschrift rechts
1	2222
10	222
100	22

Tabelle 3.1: Eine einfache Tabelle

Befehle `rowcolor` oder `multicolumn` irgendwann für Sie nützlich. Es gibt noch viele weitere Pakete, die helfen, noch hübschere Tabellen zu gestalten, beispielhaft seien hier nur `array`, `booktabs` und `tabularx` genannt. **Abbildungs- und Tabellenverzeichnis** Mit  $\text{\LaTeX}$  lassen sich Abbildungs- und Tabellenverzeichnis automatisch erstellen. Dabei tauchen alle Einträge entsprechend auf, für die Sie *Gleitumgebungen* korrekt angelegt haben (siehe Abschnitt 3.3 und 3.3).

Für diese Verzeichnisse wird standardmäßig der Text aus der `caption` übernommen. Dabei kommt es immer wieder vor, dass diese Beschreibung zu lang ist. Dafür kann

mit in der `caption` in eckigen Klammern optional eine kürzere Version angeben. Dazu siehe auch Abschnitt 4.1.3. **Formeln** Eine der größten Stärken von L<sup>A</sup>T<sub>E</sub>X ist, dass man viele Möglichkeiten hat, Formeln einfach und schön aufzuschreiben. Das „`amsmath`“-Paket ist in diesem Zusammenhang besonders beliebt, weil es ganz viele Möglichkeiten bietet. Hier nur ein kleines Beispiel mit der „`align`“-Umgebung:

$$\sum_{i=1}^n i = 1 + 2 + \dots + (n-1) + n \quad (3.1)$$

$$= (1+n) + (2+(n-1)) + \dots \quad (3.2)$$

$$= \frac{1}{2} \cdot (n+1) \quad (3.3)$$

Aber auch einfache Formeln im Text wie  $x \in \mathbb{N}$  sind natürlich möglich. Ein häufiger Fehler dabei ist, dass der „Mathe-Modus“ im Text vergessen wird: Wir sprechen über den  $x$ -Wert und *nicht* den x-Wert.

Ebenso häufig gibt es den Fehler auch andersrum, also dass Text im Mathe-Modus geschrieben wird.:  $a_{falsch} = 42$ , aber  $a_{richtig} = 42$ , vielleicht auch  $a_{\text{richtig}} = 42$ .

Funktionen wie  $\sin$  werden automatisch gut dargestellt, wie in

$$\sin \alpha = \left( \frac{a}{c} \right) \quad (3.4)$$

Die Klammern wurden hier nur eingefügt, um den entsprechenden Mechanismus zu demonstrieren: automatisch wachsende Klammern!

Manchmal wollen Sie einen eigenen „Operator“ benutzen, der optisch gleich aussehen soll. Genau dafür ist der „`\DeclareMathOperator`“-Befehl da, damit kann man fehlende Funktionen wie etwa  $\operatorname{sgn}(x)$  hinzufügen.

Es empfiehlt sich, allen mathematischen Symbole, die Sie in Ihrer Arbeit benutzen, im Quellcode sprechende Namen zu geben, das geht am einfachsten mit dem „`\newcommand`“-Befehl. Dann können Sie jederzeit anpassen, wie Sie den Gewichtsvektor  $\mathbf{w}$  im gesamten Dokument darstellen wollen oder wie die imaginäre Einheit  $i$  mit  $i^2 = -1$  aussehen soll. Das setzt natürlich voraus, dass Sie die Bezeichnungen konsequent nutzen. Dadurch wird aber auch Ihr Quellcode besser lesbar! **Quellcode, Pseudocode** Soll in der Abschlussarbeit ein Ausschnitt vom Quelltext dargestellt werden, so ist die naheliegende Idee, einfach einen Screenshot davon aufzunehmen und via `\includegraphics` als Abbildung einzufügen. Allerdings entpuppt sich diese Idee als schlecht, sobald das fertige Dokument näher herangezoomt wird: Sofort verpixelt der dargestellte Quelltext. L<sup>A</sup>T<sub>E</sub>X bietet hierfür jedoch eine elegantere Alternative: Das Paket `listings` zum Darstellen von Quelltext – direkt im Quelltext des L<sup>A</sup>T<sub>E</sub>X-Dokuments oder aber direkt aus einer externen Datei ausgelesen.

Mithilfe der Umgebung `lstlisting` lässt sich der Quelltext direkt im  $\text{\LaTeX}$ -Dokument eingeben. Mit dem Befehl `\lstinputlisting{<Datei>}` lässt sich der Quelltext aus einer externen  $\langle\text{Datei}\rangle$  auslesen und darstellen. Achtung: Der Pfad zu  $\langle\text{Datei}\rangle$ , relativ zum  $\text{\LaTeX}$ -Dokument, muss hierbei angegeben werden.

Außerdem kann das Layout des im fertigen Dokument dargestellten Quelltexts beeinflusst werden. Hierfür existiert ein *key-value-Interface*, über welches mithilfe spezieller *keys* Einfluss auf Dinge wie beispielsweise die zu verwendende Schriftart oder die Hintergrundfarbe genommen wird. Dazu wird der Befehl `\lstdefinestyle{<Stil>}` verwendet. Dabei ist  $\langle\text{Stil}\rangle$  eine Liste mehrerer Paare der Form  $\langle\text{key}\rangle=\langle\text{value}\rangle$ , welche jeweils durch ein Komma voneinander getrennt werden. Ein Beispiel ist in der Präambel dieser Vorlage, in der Datei `definitions.tex` zu finden. Für nähere Informationen sei an dieser Stelle auf die Dokumentation des Paktes verwiesen. Ein Beispiel für solch ein mit der Umgebung `lstlisting` erzeugten Quelltext ist in Abbildung 3.3 gegeben. **Weitere Verzeichnisse** Mithilfe des Pakets `glossaries` lassen sich weitere

<pre>\lstdefinestyle{myLaTeX}{   language=TeX,   basicstyle=\footnotesize\ttfamily,   frame=single,   backgroundcolor=\color{gray!10}, }</pre>	<pre>\begin{lstlisting} \lstdefinestyle{myLaTeX}{   language=TeX,   basicstyle=\footnotesize\ttfamily,   frame=single,   backgroundcolor=\color{gray!10}, } \end{lstlisting}</pre>
------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Abbildung 3.3: Beispiel für ein listing, welches mithilfe der Umgebung `lstlisting` erstellt worden ist. Links ist das fertige Listing zu sehen, rechts ist der entsprechende Quelltext dargestellt, der zu ebenjener Ausgabe führt. Zufälligerweise handelt es sich um einen Ausschnitt desjenigen Stils, der in dieser Vorlage verwendet wird.

Verzeichnisse erzeugen. Ein Glossar sowie ein Abkürzungs- oder Symbolverzeichnis lassen sich direkt erzeugen. Außerdem können auch weitere Verzeichnisse definiert werden. Wer das komplette Potential von `glossaries` ausschöpfen möchte, benötigt Perl auf dem Rechner sowie das Perl-Skript `makeglossaries`. Allerdings existiert auch eine „eingedampfte“ Variante mit etwas eingeschränkter Funktionalität, welche komplett ohne Perl und externes Skript auskommt. Hierzu sei auf die Dokumentation des Pakets verwiesen.

Durch die Option `toc` beim Laden von `glossaries` erscheinen die zusätzlichen Verzeichnisse auch im Inhaltsverzeichnis. Wird weiterhin das Paket `hyperref` verwendet, so sind die im Text ausgegebenen Einträge dieser Verzeichnisse Links, die direkt in das entsprechende Verzeichnis führen. Die Verzeichnisse selbst können dann durch den

Befehl `\printglossaries` an der gewünschten Stelle im Dokument ausgegeben werden. Auch hier wird für weiterführende Informationen wieder auf die Dokumentation des Pakets verwiesen.

**Glossar erstellen** Ein Glossar kann ohne weitere Vorkehrungen direkt verwendet werden. Ein Eintrag im Glossar kann dann über den Befehl `\newglossaryentry{⟨Label⟩}{⟨Spezifikation⟩}` definiert werden. Dabei ist `⟨Spezifikation⟩` eine *key-value*-Liste. Die wichtigsten *keys* sind `name` und `description`, über welche der Name und die Beschreibung des zu definierenden Eintrags festgelegt werden. Über den Befehl `\gls{⟨Label⟩}` kann dann der zuvor definierte Begriff im Text ausgegeben werden. Beispiel gefällig? Der Hund ist der beste Freund des Menschen.

**Abkürzungsverzeichnis erstellen** Um ein Abkürzungsverzeichnis verwenden zu können, muss `glossaries` mit der Option `acronym` geladen werden. Eine Abkürzung kann dann in der Präambel über den Befehl `\newacronym{⟨Label⟩}{⟨Abkürzung⟩}{⟨Ausgeschrieben⟩}` definiert werden. Über den Befehl `\gls{⟨Label⟩}` kann dann die zuvor definierte Abkürzung im Text ausgegeben werden. Dabei stellt L<sup>A</sup>T<sub>E</sub>X dann automatisch sicher, dass die Abkürzung bei der ersten Erwähnung im Text ausgeschrieben wird. Bei allen späteren Erwähnungen wird dann nur noch die Abkürzung ausgegeben. Beispiel gefällig? Das ist eine support vector machine (SVM). Und dort ist gleich noch eine SVM.

**Symbolverzeichnis erstellen** Um ein Symbolverzeichnis verwenden zu können, muss `glossaries` mit der Option `symbols` geladen werden. Ein Symbol kann dann in der Präambel über den Befehl `\newglossaryentry{⟨Label⟩}{⟨Spezifikation⟩}` definiert werden. Der Befehl ist also genau derselbe wie beim Glossar. Zusätzlich muss für `⟨Spezifikation⟩` noch `type=symbols` angegeben werden. Über den Befehl `\gls{⟨Label⟩}` kann dann das zuvor definierte Symbol im Text ausgegeben werden. Beispiel gefällig? Die Kraft  $\vec{F}$  ist gemäß der folgenden Gleichung definiert:

$$\vec{F} = m \cdot \vec{a}$$

**Verweise** Setzen Sie in Ihrem Quellcode Marken mit dem „`\label`“-Befehl. Aus der Platzierung geht hervor, auf welche Nummerierung sich die Marke bezieht, also etwa Gliederungsebene (siehe Abschnitt 3.2), Tabelle (siehe Abschnitt 3.3), Abbildung (siehe Abschnitt 3.3) oder Gleichung (siehe Abschnitt 3.3). Alle genannten werden nämlich separat nummeriert, das kann am Anfang etwas gewöhnungsbedürftig sein.

Auf die markierten Stellen können Sie dann mit dem „`\ref`“-Befehl verweisen, wobei der eben nur die passende Nummer liefert. Die passende Bezeichnung, z. B. „Abbildung“, müssten Sie dann selbst ergänzen. Daher haben wir hier das Paket `cleveref` eingebunden, das uns den zuletzt genannten Schritt automatisiert.

Auf jede Abbildung und jede Tabelle muss im Text verwiesen werden, es dürfen keine nummerierten Umgebungen einfach „in der Luft hängen“. Im Gegensatz dazu müssen Abschnitte und Gleichungen nicht alle explizit referenziert werden. Sie können aber Ihren Leser\*innen helfen, wenn Sie über sinnvolle Verweise nachdenken. **Besondere Abstände und Zeichen** An Leerzeichen kann grundsätzlich ein Zeilenumbruch (oder sogar Seitenumbruch) erfolgen. In manchen Fällen möchte man das vermeiden, u. a., weil Zeilen nicht mit Zahlen beginnen sollten. Ein typisches Beispiel ist „Lange Straße 123“. Hier benötigt man hinter „Straße“ ein sog. geschütztes Leerzeichen, das mit einer Tilde erzeugt wird.

Genauso wie Gleitumgebungen optimal verteilt werden (vgl. Abschnitt 3.1), werden auch horizontale Abstände zwischen Wörtern und Sätzen in  $\text{\LaTeX}$  in jeder Zeile dynamisch angepasst. Dabei werden alle Punkte als Satzende interpretiert. Bei Abkürzungen wie „z. B.“ sieht das nicht schön aus, der Leerzeichen-Abstand ist zu groß. Hier muss in der Mitte manuell ein halbes Leerzeichen erzeugt werden mit „\,“. Wenn Ihnen das Tippen solcher Konstrukte zu umständlich erscheint, können Sie sich eigene Kommandos wie „\zb“ definieren. Zum Definieren eigener Befehle vgl. Abschnitt 3.3.

Auch bei waagerechten Strichen gibt es, wie bei Leerzeichen, unterschiedliche Längen. Für Gedankenstriche – solche hier – oder wenn „bis“ gemeint ist (wie in 14:00–16:00), reicht der einfache Bindestrich (Minuszeichen) nicht aus, das passende Zeichen wird in  $\text{\LaTeX}$  einfach durch ein Doppel-Minus erzeugt.

Problematisch im Quellcode sind alle Zeichen, die in  $\text{\LaTeX}$  eine Funktion haben: Prozentzeichen %, kaufmännisches Und &, Unterstrich \_, geschweifte Klammern { ... } sind typische Beispiele. Diese müssen im Quelltext mit einem *Backslash* eingegeben werden, sonst erhält man Fehlermeldungen. **Wahl der Grundschriftart** Standardmäßig verwendet  $\text{\LaTeX}$  für den Fließtext eine serifenbehaftete Schriftart. Für gedruckte Arbeiten sind serifenbehaftete Schriften vorteilhaft, weil die Serifen die Grundlinie betonen und somit das Auge beim Rücksprung am Zeilenende zum Beginn der nächsten Zeile unterstützt. Außerdem führen die unterschiedlichen Strichstärken zu eindeutigeren Wortbildern und unterstützen somit den Leseprozess.

Wird solch ein Dokument jedoch an einem alten Monitor mit geringer Auflösung betrachtet, so kann es sein, dass die feinen Serifen nicht mehr vernünftig dargestellt werden. In solch einem Fall kann es vorteilhaft sein, eine serifenlose Schrift zu verwenden. Auch kann es sein, dass serifenlose Schriften aus Gründen der Barrierefreiheit bevorzugt werden.

In diesem Fall kann mit dem Befehl `\renewcommand{\familydefault}{\sfdefault}` eine serifenlose Schrift als Grundschriftart festgelegt werden. Wer `lualatex` zum Kompilieren sowie das Paket `fontspec` verwendet, kann außerdem auf alle verfügbaren

Schriften zugreifen. Ist auf dem Rechner die Schriftart Arial vorhanden, so kann mit dem Befehl `\setsansfont{Arial}` die Schriftart Arial als serifenlose Schrift festgelegt werden. Am Ende der Datei `definitions.tex` sind die beiden besagten Zeilen zu finden und müssen bei Bedarf nur auskommentiert werden.

**Metadaten für den pdf-Betrachter** Manche pdf-Betrachter können zusätzliche Metadaten, wie Name des Autors, Titel des Dokuments (auch abweichend vom Namen der Datei) oder Schlüsselbegriffe anzeigen. Mit dem Paket `hyperref` lassen sich diese Metadaten mit dem Befehl `\hypersetup{⟨Einstellungen⟩}` konfigurieren. Dabei ist `⟨Einstellungen⟩` eine *key-value*-Liste. Die wesentlichsten *keys* sind `pdfauthor`, `pdftitle` und `pdfkeywords`. Die Bedeutung dieser *keys* ist selbsterklärend.

Außerdem werden Links im Dokument (bei Verwendung des Pakets `hyperref`) in manchen pdf-Betrachtern als farbige Kästchen hervorgehoben. Diese farbigen Kästchen erscheinen natürlich nicht im gedruckten Dokument. Sie dienen lediglich als Hilfe, dass man nicht „auf gut Glück“ mit dem Cursor über das Dokument fahren muss, bis man den Link gefunden hat. Wenn die farbigen Kästchen stören, so können diese in `\hypersetup` mit `hidelinks` deaktiviert werden.

**Wechsel zwischen ein- und doppelseitigem Layout** Diese Vorlage ist für ein einseitiges Layout optimiert. Dabei sind die linken und rechten Ränder jeweils gleich groß auf allen Seiten, neue Kapitel beginnen unmittelbar auf der nächsten Seite. Wird das fertige pdf-Dokument am Computer betrachtet, sieht das genau richtig aus. Soll das Dokument hingegen doppelseitig ausgedruckt werden, so kann das Layout noch etwas angepasst werden: Typischerweise sind die inneren Ränder dann etwas schmaler als die äußeren Ränder. Und neue Kapitel beginnen jeweils auf einer neuen, rechten Seite – was zu einzelnen Vakantseiten zwischen den Kapiteln führen kann. Für doppelseitig ausgedruckte Dokumente sieht das dann besser aus.

Um das doppelseite Layout zu aktivieren, genügt es bereits, die Auskommentierung der Option `twoside` im optionalen Argument von `\documentclass` zu entfernen.

**Kompilieren** Das Erstellen (Kompilieren) von großen Dokumenten mit  $\text{\LaTeX}$  kann verhältnismäßig lange dauern. Da man i. d. R. nur an wenigen Stellen gleichzeitig arbeitet, kann es daher sinnvoll sein, übrige Teile auszukommentieren. Das geht besonders leicht, wenn man Text in getrennte Dateien auslagert und mit dem „`\input`“- und / oder „`\include`“-Befehl einbindet. So bleibt auch das Hauptdokument übersichtlich.

Grundsätzlich sollte das Ziel sein, dass Ihr Dokument ohne Warnungen kompiliert. Am besten kümmert man sich regelmäßig darum, entsprechende Probleme zu beheben.

Eine typische Warnung ist „*Reference ... undefined*“. Vielleicht verschwindet sie beim nochmaligen Erstellen, denn erst dann sind ggf. neue Positionen bekannt. Wenn



diese Warnung bleibt, muss das Problem unbedingt behoben werden, sonst haben Sie irgendwo Fragezeichen im Text stehen.

Warnungen, die sich auf zu volle Boxen beziehen, sind teilweise schwieriger zu verstehen und / oder zu beheben. Im **draft**-Modus (vgl. Abschnitt 3.3) werden die zugehörigen Stellen genau markiert, das kann eine große Hilfe sein. Gegen zu lange Zeilen hilft teilweise, Trennstellen zu markieren (vgl. Abschnitt 3.3). Sonst muss ggf. ein Satz minimal umformuliert werden.

Der **draft**-Modus hat drüber hinaus den Vorteil, dass das Kompilieren schneller geht (s. o.), dafür werden für Abbildungen nur Platzhalter eingefügt. **Diese Vorlage** In dieser Vorlage wird KOMA-Script verwendet, eine „Sammlung von Klassen und Paketen für L<sup>A</sup>T<sub>E</sub>X“<sup>1</sup>, die insbesondere das Erstellen von deutschen Texten mit den entsprechenden üblichen typographischen Standards unterstützt.

Dokumente mit L<sup>A</sup>T<sub>E</sub>X zu erstellen ist ganz ähnlich wie Programmieren. Ein Beispiel: Überall dort, wo ein Absatz entstehen soll, haben wir in unserem „Quellcode“ den Befehl „`\par`“ benutzt. Was dieser Befehl genau tut, wird durch dessen Implementierung festgelegt. Und diese ergibt sich hier sozusagen aus dem Parameter `parskip` der Dokumentklasse.

Andere Einstellungen, die direkt in der Dokumentklasse erfolgen können, betreffen z. B. die Schriftgröße, die Bindungskorrektur (BCOR) und die Größe von Überschriften (`headings`). Im Prinzip kann man auch die Größe der Ränder mit dem DIV-Parameter beeinflussen, davon wird aber abgeraten. Die Ränder werden automatisch so eingestellt, dass Zeilen eine Länge haben, die gut zu lesen ist.

Hier haben wir außerdem die Option `twoside` gewählt, für beidseitigen Druck. Daher sind die Ränder außen auf geraden und ungeraden Seiten unterschiedlich. Falls Sie Ihr Dokument am Ende einseitig drucken wollen, stellen Sie das bitte um.

Nach der Festlegung der Dokumentklasse haben wir in der sog. Präambel einige Pakete eingebunden. Zum Beispiel das `sclayer-scrpage`-Paket, mit dem wir das Aussehen der Fuß- und Kopfzeile definieren können. Diese Vorlage wurde so eingerichtet, dass in den Kopfzeilen einer Doppelseite oben links immer die aktuelle Kapitel-Überschrift und oben rechts die aktuelle Abschnitt-Überschrift angezeigt wird (siehe `definitions.tex`).

In der vorliegenden Vorlage werden einige Pakete eingebunden. Im Folgenden wird die Funktion der wichtigsten davon kurz erläutert.

**fontspec** Erlaubt die freie Wahl der Schriftart (funktioniert aber nur bei Kompilation mit `lualatex`)

---

<sup>1</sup><https://komascript.de>

**babel** Erlaubt das Umstellen der Standard-Sprache auf Deutsch

**selnolig** Sorgt für automatisch korrekt gesetzte Ligaturen (funktioniert aber nur bei Verwendung von **fontspec** und somit auch **lualatex**, übernimmt automatisch die Spracheinstellung von **babel**)

**microtype** Optimiert das Aussehen des Textes (Satzspiegel)

**csquotes** Sorgt für automatisch korrekt gesetzte Anführungszeichen (übernimmt automatisch die Spracheinstellung von **babel**)

**tikz und pgfplots** Damit können Abbildungen direkt im Quellcode erzeugt werden, vgl. Abschnitt 3.3

**hyperref** Anklickbare Links im PDF

**biblatex** Verbesserte Quellenangaben und -verzeichnis

**amsmath und amssymb** Große Erweiterung der Möglichkeiten, mathematische Inhalte darzustellen

**listings** Zur Darstellung von Quellcode, vgl. Abschnitt 3.3

**cleveref** Vereinfacht das Einfügen von Verweisen (siehe Abschnitt 3.3)

**glossaries** Komfortables Erstellen eines Abkürzungsverzeichnis

## 4 Konzeption und Design

Fremdes Gedankengut muss immer kenntlich gemacht werden. Vor allem muss es überprüfbar und auffindbar sein. Hierzu dient die Technik des Zitierens und Belegens.

Verschiedene Fachrichtungen und Studiengänge folgen spezifischen Zitierkonventionen. Wichtige Hinweise zu gängigen Zitationssystem und -stilen finden Sie in den E-Learning-Kursen des Schreibzentrums<sup>1</sup>.

### 4.1 Anforderungsanalyse

Wörtlich übernommene Textpassagen werden durch Anführungszeichen unten und oben („...“) kenntlich gemacht.

Wenn Ihr Zitat bereits ein Zitat enthält, müssen Sie die „doppelten Anführungszeichen“ im Text durch ‚einfache Anführungszeichen‘ ersetzen. Dazu vergleiche auch Abschnitt 3.3.

#### 4.1.1 Funktionale Anforderungen

Für alle Zitate muss ein Quellenverweis erstellt werden. Der Quellenverweis ist eine im Zitationsstil festgelegte Kurznotation, die auf die vollständige Literaturangabe im Literaturverzeichnis verweist. Quellenverweise können *entweder* im laufenden Text (anglo-amerikanische Zitierweise bzw. Harvard-Stil) *oder* über eine Fußnote am unteren Ende der Seite *oder* in einer Endnote am Ende des gesamten Textes erfolgen. Hier gelten unterschiedliche fachliche Konventionen, die unbedingt beachtet werden müssen.

Entscheidet man sich für Kurzverweise im Textfluss oder für Endnoten, kann der Fußnotenbereich für Kommentare und für Verweise auf Stellen im eigenen Text genutzt werden.

*Beachten Sie die Positionen von Hochzahl und Satzzeichen:*

---

<sup>1</sup>[https://ilu.th-koeln.de/goto.php?target=cat\\_52109&client\\_id=thkilu](https://ilu.th-koeln.de/goto.php?target=cat_52109&client_id=thkilu)

Wenn das wörtliche Zitat selbst mit einem Punkt endet, steht dieser vor dem beendenden Anführungszeichen. Die Hochzahl folgt dann direkt danach ohne Leerzeichen. Ein Punkt für den eigenen Satz entfällt dann.

Wenn das wörtliche Zitat nicht mit einem Punkt endet gibt es zwei Fälle: Steht das Zitat mitten im eigenen Satz, folgt die Hochzahl direkt nach den Anführungszeichen. Steht das Zitat am Ende des eigenen Satzes, notiert man zuerst die Anführungszeichen, dann den eigenen Satzpunkt und erst dann die Hochzahl.

Damit die Quellenverweise auch bei mehreren gleichlautenden Kurztiteln oder Jahreszahlen eines Autors eindeutig bleiben, erhalten die Jahreszahlen einen zusätzlichen kleinen Buchstaben.

#### 4.1.2 Nicht-funktionale Anforderungen

Textverarbeitungsprogramme machen das Verweisen über *derselbe* und *ebenda* heute überflüssig, da man nicht mehr gezwungen ist, die gleichen Angaben wieder und wieder abzutippen. Sollten Sie sich (zum Beispiel auf Anraten Ihres Prüfenden) dennoch für dieses Verfahren entscheiden, empfehlen wir dringend, *ebenda* und *derselbe* etc. *erst in der redaktionellen Endphase* einzufügen, weil die Bezüge erst dann klar sind.

#### 4.1.3 Stakeholder-Analyse (elancer-team GmbH)

Bei Zitaten aus zweiter Hand, sog. Sekundärzitaten, übernimmt man ein Zitat eines Autors oder einer Autorin, ohne sich in der Originalquelle über den Sinnzusammenhang informiert zu haben. Zitate aus zweiter Hand sind nur zulässig, wenn die Originalquelle nicht beschaffbar ist. Bei allgemein zugänglicher wissenschaftlicher Literatur sind Zitate aus zweiter Hand unbedingt zu vermeiden. Ist es nicht möglich, ein Zitat mit dem Originaltext zu vergleichen, dann notiert man zitiert nach (es folgt die Quelle, der man das Zitat entnommen hat) oder zitiert in. Dies kommt zum Beispiel vor, wenn man wissenschaftsgeschichtliche Sachverhalte aus Lehrbüchern zitiert. **Abbildungen und Tabellen zitieren** Hier gelten die gleichen Richtlinien wie für Textzitate, d. h. auch hier gibt es getreue und abgeänderte Übernahmen. Beim originalgetreuen Zitat können Sie eine Abbildung z. B. in PowerPoint oder einem Zeichen-/Vektorprogramm genau nachbilden. Beim Scannen ist das Copyright zu berücksichtigen; es ist nur dann erlaubt, wenn der Autor bzw. der Verlag die – zumeist kostenpflichtige – Erlaubnis dazu erteilt hat. Der Quellenverweis ist in diesem Fall wie beim wörtlichen Zitat zu gestalten. Bei eigenen Veränderungen, muss dem Quellenverweis ein Zusatz angehängt werden (z. B. *mit geringfügigen Veränderungen, mit eigenen Berechnungen*, usw.).

Weil Grafiken häufig übernommen werden, empfiehlt es sich, eigene Grafiken oder Bebilderungen auch als solche zu kennzeichnen.

In Tabellen- und Abbildungsunterschriften wird die Quelle immer direkt unter der Abbildung angegeben und nicht in einer Fuß- oder Endnote. Ins Abbildungs- und Tabellenverzeichnis gehören die Quellenangaben allerdings nicht. Dazu siehe Abschnitt 3.3.

## 4.2 Plugin-Architektur

Nachfolgend wird die Gestaltung des obligatorischen Literaturverzeichnisses erläutert.

### 4.2.1 Modularer Aufbau und Komponenten

Im Literaturverzeichnis werden alle verwendeten Schriften in alphabetischer Reihenfolge nach Autorennamen gelistet. Das Literaturverzeichnis muss alle im Text zitierten Quellen beinhalten, aber auch keine darüber hinaus gehenden. Sind Werke nicht nur in gedruckter Form, sondern auch elektronisch publiziert, sollte die Literaturangabe der Druckfassung folgen. Es sei denn, das digitale Dokument hat eine feste Dokumentennummer (DOI). Wurde im Textteil in den Quellenverweisen statt des Titels ein Buchstabenkürzel verwendet, so ist diese Angabe auch im Literaturverzeichnis kenntlich zu machen.

Mehrere Werke, die ein Autor innerhalb eines Jahres veröffentlicht hat, müssen differenziert werden. Für die Kurztitel im Text sind die Jahreszahlen daher durch angehängte Kleinbuchstaben zu unterscheiden. Die Jahreszahl in der Quellenangabe bleibt jedoch unverändert ohne angehängte Buchstaben. Die Reihenfolge von a, b, c etc. richtet sich nach der Reihenfolge der Quellenverweise. Ob Vornamen abgekürzt oder ausgeschrieben werden, hängt von den Konventionen Ihres Faches ab. Bleiben Sie jedoch einheitlich.

### 4.2.2 Datenmodell und Datenbankstruktur

Die Reihenfolge in der Notation der Literaturangabe hängt vom Dokumententyp und von fachlichen Konventionen ab. Bei Monografien sieht sie also anders aus als bei Zeitschriftenaufsätzen, in der Chemie wieder anders als in den Geisteswissenschaften oder der Mathematik. Auch für die *Interpunktion* gibt es keine einheitlichen Vorgaben.

Die Daten einer Literaturangabe entnehmen Sie bei Büchern dem so genannten Titelblatt. Dies ist nicht der Buchdeckel, sondern ein bedrucktes Blatt am Buchbeginn mit den wichtigsten werk- und buchidentifizierenden Angaben. Zunächst werden Verfasser oder Herausgeber zusammen mit dem Titel genannt, darauf folgen Erscheinungs- und Druckvermerke wie Verlag, Ort und Erscheinungsjahr. Es gibt zwar eine DIN-Regel für die Titelbeschriftung, die aber sehr unterschiedlich ausgelegt wird.

In den E-Learning-Kursen des Schreibzentrums<sup>2</sup> finden Sie weitere Informationen zu den bibliographischen Angaben für verschiedene Publikationstypen. Die hier folgende Literaturliste ist nur ein Beispiel für ein mögliches Format.

### **4.2.3 API-Design und Schnittstellen**

## **4.3 Gutenberg-Integration**

### **4.3.1 Block-Design und -Funktionalität**

### **4.3.2 Editor-Integration**

---

<sup>2</sup>[https://ilu.th-koeln.de/goto.php?target=cat\\_52109&client\\_id=thkilu](https://ilu.th-koeln.de/goto.php?target=cat_52109&client_id=thkilu)

## 5 Implementierung

Fremdes Gedankengut muss immer kenntlich gemacht werden. Vor allem muss es überprüfbar und auffindbar sein. Hierzu dient die Technik des Zitierens und Belegens.

Verschiedene Fachrichtungen und Studiengänge folgen spezifischen Zitierkonventionen. Wichtige Hinweise zu gängigen Zitationssystem und -stilen finden Sie in den E-Learning-Kursen des Schreibzentrums<sup>1</sup>.

### 5.1 Entwicklungsumgebung und Tools

Wörtlich übernommene Textpassagen werden durch Anführungszeichen unten und oben („...“) kenntlich gemacht.

Wenn Ihr Zitat bereits ein Zitat enthält, müssen Sie die „doppelten Anführungszeichen“ im Text durch ‚einfache Anführungszeichen‘ ersetzen. Dazu vergleiche auch Abschnitt 3.3.

#### 5.1.1 Plugin-Softwarearchitektur

Für alle Zitate muss ein Quellenverweis erstellt werden. Der Quellenverweis ist eine im Zitationsstil festgelegte Kurznotation, die auf die vollständige Literaturangabe im Literaturverzeichnis verweist. Quellenverweise können *entweder* im laufenden Text (anglo-amerikanische Zitierweise bzw. Harvard-Stil) *oder* über eine Fußnote am unteren Ende der Seite *oder* in einer Endnote am Ende des gesamten Textes erfolgen. Hier gelten unterschiedliche fachliche Konventionen, die unbedingt beachtet werden müssen.

Entscheidet man sich für Kurzverweise im Textfluss oder für Endnoten, kann der Fußnotenbereich für Kommentare und für Verweise auf Stellen im eigenen Text genutzt werden.

*Beachten Sie die Positionen von Hochzahl und Satzzeichen:*

---

<sup>1</sup>[https://ilu.th-koeln.de/goto.php?target=cat\\_52109&client\\_id=thkilu](https://ilu.th-koeln.de/goto.php?target=cat_52109&client_id=thkilu)

Wenn das wörtliche Zitat selbst mit einem Punkt endet, steht dieser vor dem beendenden Anführungszeichen. Die Hochzahl folgt dann direkt danach ohne Leerzeichen. Ein Punkt für den eigenen Satz entfällt dann.

Wenn das wörtliche Zitat nicht mit einem Punkt endet gibt es zwei Fälle: Steht das Zitat mitten im eigenen Satz, folgt die Hochzahl direkt nach den Anführungszeichen. Steht das Zitat am Ende des eigenen Satzes, notiert man zuerst die Anführungszeichen, dann den eigenen Satzpunkt und erst dann die Hochzahl.

Damit die Quellenverweise auch bei mehreren gleichlautenden Kurztiteln oder Jahreszahlen eines Autors eindeutig bleiben, erhalten die Jahreszahlen einen zusätzlichen kleinen Buchstaben.

### 5.1.2 Backend-Entwicklung

Textverarbeitungsprogramme machen das Verweisen über *derselbe* und *ebenda* heute überflüssig, da man nicht mehr gezwungen ist, die gleichen Angaben wieder und wieder abzutippen. Sollten Sie sich (zum Beispiel auf Anraten Ihres Prüfenden) dennoch für dieses Verfahren entscheiden, empfehlen wir dringend, *ebenda* und *derselbe* etc. *erst in der redaktionellen Endphase* einzufügen, weil die Bezüge erst dann klar sind.



## Literatur

Wie in Abschnitt 4.2.1 erläutert, werden im Literaturverzeichnis ausschließlich die Quellen angegeben, auf die im Rahmen einer Arbeit tatsächlich verwiesen wird. Bitte prüfen Sie also das Literaturverzeichnis Ihrer Arbeit immer dahingehend, ob alle zitierten Quellen – und nur diese – erfasst wurden. Dies trifft auf das nun folgende Verzeichnis *nicht* zu; die meisten der hier aufgeführten Quellen werden in dieser Vorlage nicht zitiert. Es handelt sich lediglich um ein Beispiel für ein Literaturverzeichnis mit Literaturempfehlungen zum wissenschaftlichen Schreiben.

- [1] Volker Ahrens. *Abschlussarbeiten richtig gliedern in Naturwissenschaften, Technik und Wirtschaft*. 2. Auflage. Zürich: vdf, 2018.
- [2] Doris Berger. *Wissenschaftliche Arbeiten in den Wirtschafts- und Sozialwissenschaften. Hilfreiche Tipps und praktische Beispiele*. Wiesbaden: Gabler, 2010.
- [3] Georg Bruhn und Gertrude Hirsch Hadorn. *Textanalyse in den Wissenschaften. Inhalte und Argumente analysieren und verstehen*. 4., durchgesehene und aktualisierte Auflage. Zürich: vdf Hochschulverlag, 2021.
- [4] Urs Dahinden, Sabrina Sturzenegger und Alessia Neuron. *Wissenschaftliches Arbeiten in der Kommunikationswissenschaft*. 2. korrigierte und aktualisierte Auflage. Bern: Haupt, 2014.
- [5] Donald E. Knuth. *The T<sub>E</sub>X Book*. Addison-Wesley Professional, 1986.
- [6] Hans Ebel und Claus Bliefert. *Bachelor-, Master- und Doktorarbeit. Anleitungen für den naturwissenschaftlich-technischen Nachwuchs*. 4. Auflage. Weinheim: Wiley-VCH, 2009.
- [7] Umberto Eco. *Wie man eine wissenschaftliche Abschlußarbeit schreibt*. 13. Auflage. Wien: facultas, 2010.
- [8] Helga Esselborn-Krumbiegel. *Von der Idee zum Text. Eine Anleitung zum wissenschaftlichen Schreiben*. 5., aktualisierte Auflage. Paderborn: Schöningh, 2017.
- [9] Andrea Frank, Stefanie Haacke und Swantje Lahm. *Schlüsselkompetenzen: Schreiben in Studium und Beruf*. Studtgart, Weimar: Metzler, 2007.

- [10] Melanie Fröhlich, Christiane Henkel und Anna Surmann. *Zusammen schreibt man weniger allein. (Gruppen-)Schreibprojekte gemeinsam meistern*. Opladen, Toronto: Budrich, 2017.
- [11] Karl-Heinz Göttert. *Kleine Schreibschule für Studierende*. 2. Auflage. München: Fink, 2002.
- [12] Andreas Hirsch-Weber und Stefan Scherer. *Kleine Schreibschule für Studierende*. Stuttgart: Eugen Ulmer, 2016.
- [13] Matthias Karmasin und Rainer Ribing. *Die Gestaltung wissenschaftlicher Arbeiten. Ein Leitfaden für Facharbeit/VWA, Seminararbeiten, Bachelor-, Master-, Magister- und Diplomarbeiten sowie Dissertationen*. 10., überarbeitete und aktualisierte Auflage. Wien: facultas, 2019.
- [14] Ulrike Kipman, Ulrike Leopold-Wildburger und Thomas Reiter. *Wissenschaftliches Arbeiten 4.0. Vortragen und Verfassen leicht gemacht*. 3. Auflage. Berlin: Springer, 2018.
- [15] Stefan Kottwitz. *LaTeX Graphics with TikZ*. Packt, 2023.
- [16] Bruno P. Kremer. *Vom Referat bis zur Abschlussarbeit. Naturwissenschaftliche Texte perfekt produzieren, präsentieren und publizieren*. 5., aktualisierte und erweiterte Auflage. Berlin: Springer Spektrum, 2018.
- [17] Otto Kruse. *Lesen und Schreiben. Der richtige Umgang mit Texten im Studium*. 3. überarbeitete und erweiterte Auflage. Konstanz, München: UKV/Lucius, 2018.
- [18] Stefan Kühtz. *Wissenschaftlich formulieren. Tipps und Textbausteine für Studium und Schule*. 6., aktualisierte und erweiterte Auflage. Paderborn: Schöningh, 2021.
- [19] Ulrike Lange. *Fachtexte lesen, verstehen, wiedergeben*. 2. überarbeitete Auflage. Paderborn: Schöningh, 2018.
- [20] Jürg Niederhauser und Dudenredaktion. *Die schriftliche Arbeit. Für Schule, Hochschule und Universität*. 3. Auflage. Berlin: Dudenverlag, 2019.
- [21] Frank Norbert. *Handbuch Wissenschaftliches Schreiben. Eine Anleitung von A bis Z*. Paderborn: Schöningh, 2019.
- [22] Monika Oertner, Ilona St. John und Gabriele Thelen. *Wissenschaftlich Schreiben. Ein Praxisbuch für Schreibtrainer und Studierende*. Paderborn: Schöningh, 2014.
- [23] Lydia Prexl. *Alles, was Ingenieur:innen über Deutsch wissen müssen*. München: UKV, 2021.
- [24] Lydia Prexl. *Mit digitalen Quellen arbeiten. Richtig zitieren aus Datenbanken, E-Books, YouTube und Co*. 3. aktualisierte und überarbeitete Auflage. Paderborn: Schöningh, 2019.

- [25] Jan Schaller. *Papierlos studieren. Wissenschaftlich arbeiten in digitalen Zeiten*. Opladen, Toronto: Budrich, 2020.
- [26] Ulrike Scheuermann. *Schreibdenken. Schreiben als Denk- und Lernwerkzeug nutzen und vermitteln*. 3., durchgesehene Auflage. Opladen, Toronto: Budrich, 2016.
- [27] Kirsten Schindler. *Klausur, Protokoll, Essay. Kleine Texte optimal verfassen*. 3., durchgesehene Auflage. Paderborn u. a.: Schöningh, 2011.
- [28] Kirsten Schindler. *Wissenschaftliches Arbeiten ... leicht verständlich*. 8., überarbeitete und erweiterte Auflage. München: UKV, 2022.
- [29] Mona Stier. „Jenseits der Hausarbeit: Die vielen kleinen Textsorten des Hochschulalltags“. In: *JoSch. Journal der Schreibberatung* 9 (2015), S. 82–87.
- [30] Judith Wolfsberger. *Frei geschrieben. Mut, Freiheit & Strategie für wissenschaftliche Abschlussarbeiten*. 4., bearbeitete Auflage. Wien, Köln, Weimer: Böhlau Verlag, 2016.
- [31] Christian Wymann. *Der Schreibzeitplan: Zeitmanagement für Schreibende*. 2., überarbeitete Auflage. Opladen, Berlin, Toronto: Budrich, 2021.

## **Anhang**

## Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Anmerkung: In einigen Studiengängen findet sich die Erklärung unmittelbar hinter dem Deckblatt der Arbeit.

---

Ort, Datum

---

Unterschrift