



FHDW Paderborn

**Studienarbeit im Modul Weiterführende
Datenbankkonzepte**

Exemplarisches Beispiel für eine Demonstration
von Quarto

Vorgelegt von:

Friedrich Fleißigerstudent

Fürstenallee 3-6

33102 Paderborn

Studiengang:

Angewandte Informatik (B.Sc.)

Prüfer:

Prof. Dr. Weis-Alles

Prof. Dr. Besser

Eingereicht am:

2024-05-67

Gendererklärung

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

Inhaltsverzeichnis

Gendererklärung	II
Inhaltsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	IV
Glossar	V
1 Einleitung	1
1.1 Zielsetzung	1
1.2 Stand	1
2 Grundlagen Quarto	2
2.1 Schrift	2
2.1.1 Schriftgrößen	2
2.1.2 Schrifttypen	2
2.1.3 Schrift Ausrichtung	3
2.2 Abbildungen	3
2.3 Tabellen	3
2.4 Zitate	4
2.5 Glossar und andere Verzeichnisse	4
2.6 Listen	5
2.7 Quelltext	5
2.8 Anhänge	7
3 Installation	8
3.1 Quarto etc.	8
3.2 Python	8
3.2.1 Installation	8
3.2.2 Problembehebung	8
3.3 Verwendung der Vorlage	9
3.4 Test der Silbentrennung	10
4 Zusammenfassung	11
A Erster Anhang	12
B Zweiter Anhang	13
C Struktur der Vorlage	14

Quellen	15
Ehrenwörtliche Erklärung	16

Abbildungsverzeichnis

Abbildung 1: Eine Unterschrift	3
Abbildung 3: Eine Gerade in Polarkoordinaten	7

Tabellenverzeichnis

Tabelle 2: Buchstaben	4
Tabelle A1: Alle Zahlen von 1..9	13

Glossar

FHDW Fachhochschule der Wirtschaft

Gummy Dies ist ein etwas längerer Glossareintrag, der eigentlich *Dummy* heißen sollte, aber wegen der alphabetischen Reihenfolge anders benannt werden musste. Ob die Glossareinträge in der Datei `Glossar.tex` durch eine zusätzliche Leerzeile im Quelltext getrennt werden ist für Quarto bzw. \LaTeX gleichgültig. Eine zusätzliche Leerzeile könnte aber die Lesbarkeit verbessern.

VDI Verein deutscher Ingenieure

1 Einleitung

1.1 Zielsetzung

Dies ist eine Vorlage zum Erstellen von Bachelorarbeiten, Praxisarbeiten und Studienarbeiten an der FHDW mit Quarto und dem Satzsystem \LaTeX . Diese Vorlage ist eine Adaption der FHDW- \LaTeX -Vorlage mit Anregungen von Lars-Andre Scheiperpeter und wurde von und Ulrich Reus erstellt. Diese Vorlage darf, so wie sie ist, frei verwendet und modifiziert werden. Haftung für Fehler, verlorene Zeit oder in Folge einer Verwendung der Vorlage verursachte Schäden wird nicht übernommen.

Die in der Vorlage verwendeten \LaTeX -Pakete und Styles sind sehr gut dokumentiert. Bezüglich tiefergehender Informationen zu den einzelnen Paketen sei auf die jeweilige Dokumentation verwiesen, die standardmäßig mit den jeweiligen Paketen installiert wird.

Alternativ zur PDF-Ausgabe kann auch ein HTML-Dokument generiert werden. Da diese Vorlage allerdings auf den Textsatz per \LaTeX optimiert ist, müssen beim Erscheinungsbild der Web-Darstellung möglicherweise Abstriche in Kauf genommen werden.

1.2 Stand

Diese Vorlage wurde im Mai 2024 zuletzt bearbeitet. Die verwendeten Software-Produkte wurden in den folgenden Versionen verwendet

- RStudio 2023.12.1+402
- Quarto 1.4.554
- Python 3.10.3

2 Grundlagen Quarto

Quarto-Dokumente werden in Markdown geschrieben. Diese Auszeichnungssprache zeichnet sich durch gute Lesbarkeit und einfache Erlernbarkeit aus. Zahlreiche Markdown-Editoren wie Obsidian oder RStudio ermöglichen die sofortige Darstellung des Markdown-Codes in einer der entgültigen Ausgabe ähnlichen Form.

Zahlreiche \LaTeX -Kommandos können in Markdown in gleicher Weise wie in \LaTeX benutzt werden. Allerdings wird dies dann nur beim Rendern in das PDF-Format und nicht bei der HTML-Ausgabe umgesetzt.

2.1 Schrift

2.1.1 Schriftgrößen

Die folgenden Beispiele für Schriftgrößen werden nur bei der PDF-Ausgabe unterstützt, da sie mit \LaTeX -Befehlen realisiert werden. In der HTML-Ausgabe werden dies Befehle ignoriert und der Text in Standardschrift ausgegeben.

Das ist kleine Schrift

Das ist normale Schrift

Das ist große Schrift

Das ist größere Schrift

Das ist Fußnoten Schrift

2.1.2 Schrifttypen

Das ist ein fetter Text

Das ist ein kursiver Text

Das ist ein unterstrichener Text

DAS IST EIN IN KAPITÄLCHEN GESETZTER TEXT

Das ist ein normaler Text

2.1.3 Schrift Ausrichtung

Quote Text (Der gesamte Text innerhalb der Umgebung wird von beiden Seiten eingerückt)

In einer Fußnote...¹

2.2 Abbildungen

Hier ist mit Abbildung 1 ein Bild mit einer Unterschrift eingebunden. Bei der Quellenangabe mit dem L^AT_EX-Befehl `\source` muss eine Leerzeile als Abstand eingehalten werden, da es sonst zu Fehlern bei der Erzeugung der Bildumgebung und des Querverweises kommt.



Abbildung 1: Eine Unterschrift

Quelle: Herr Mustermann

In Abbildung 2a sehen Sie das Logo der FHDW in Form eines Blocks mit zwei Bildern. Wenn Abbildung 2a erst auf der folgenden Seite zu sehen sein sollte, dann liegt das daran, dass auf dieser Seite nicht mehr genug Platz war. der verbleibende Platz auf der Seite kann dann unter Umständen mit dem folgenden Text aufgefüllt werden.



(a)
Das LOGO

Quelle: www.fhdw.de

2.3 Tabellen

In Tabelle 2. sind einige Buchstaben dargestellt

¹können zusätzliche Ergänzungen, Präzisierungen, Textverweise usw. eingeführt werden.

Tabelle 2: Buchstaben

Col1	Col2	Col3
A	B	C
E	F	G
A	G	G

Quelle: Eigene Darstellung

2.4 Zitate

Literaturarbeit ist ein zentraler Bestandteil wissenschaftlicher Arbeiten. Es wird erwartet, dass alle Quellen ordnungsgemäß angegeben und zitiert werden. Zur Verwaltung der Literatur empfiehlt sich die Verwendung einer entsprechenden Software, z.B. Zotero, für das es in Quarto umfangreiche Unterstützung gibt.

Zu Demonstrationszwecken wird hier ein vergleichendes Zitat angegeben.² Und hier ein „wörtliches Zitat“³. Dies wiederum ist ein Zitat einer Quelle mit mehreren Autoren.⁴ Ein Zitat im Fließtext ist zu sehen bei Füller et al. (2012), der diese Form des Quellennachweises offenbar gerne nutzt.

2.5 Glossar und andere Verzeichnisse

Ein Glossar beinhaltet Erklärungen von Abkürzungen und Begriffen, die in der Arbeit verwendet werden. Wenn kein Glossar benötigt wird, so muss Eintrag `glossar` im Kopf der Hauptdatei auf `false` gesetzt werden, damit die dazugehörige Seite nicht angezeigt wird. Das gleiche gilt auch für den Sperrvermerk und die Gendererklärung.

Die Glossareinträge selbst müssen in der Datei `Glossar.tex` zwischen den Kommentarseiten `%1. ----` und `% 2. ----` eingetragen werden. Ein neuer Eintrag besteht aus einer Zeile, die Abkürzung und Erläuterung enthält, die durch das Zeichen `&` getrennt werden. Die Zeile wird beendet durch die Zeichenfolge `\\`.

²Vgl. Maslennikov (2011), S.5 ff.

³Meier (2010), S. 13 f.

⁴Vgl. Hocking et al. (2010), S.32 ff.

2.6 Listen

Eine einfache Liste mit Punkten in Markdown:

- Punkt 1
- Punkt 2
 - Unterpunkt 2.1
 - UP 2.2
- Punkt 3

Eine einfache Liste mit Nummern in Markdown:

1. Punkt 1
2. Punkt 2
3. Punkt 3

Eine einfache Liste mit römischen Nummern in Form von LaTeX-Code:

- I. Punkt 1
- II. Punkt 2
- III. Punkt 3

2.7 Quelltext

Quarto-Dokumente können zusätzlich zum Text Code-Blöcke enthalten. In diesen Code-Blöcken können z.B. Daten aufbereitet und analysiert werden um dann in Form von Tabellen oder Diagrammen im Dokument zu erscheinen.

Wenn Sie auf die Schaltfläche **Render** klicken, wird ein Dokument erstellt, das sowohl den Inhalt als auch die Ausgabe des eingebetteten Codes enthält. Sie können Code wie folgt einbetten:

```
1 + 1
```

2

Es kann sowohl R-Code als auch Python-Code eingebettet werden. Der Code kann spezielle Kommentare enthalten, die die Verarbeitung und die Ausgabe der Code-Zelle steuern.

```
#| echo: false
2 * 2
```

Die Option `echo: false` im obigen Code-Block schaltet z.B. das Drucken des Codes ab. Im gerenderten Dokument wird dann nur die Ausgabe des Code-Blocks angezeigt.

Es gibt auch die Möglichkeit Code-Abschnitte in das Dokument aufzunehmen, die nur dargestellt, aber selbst nicht ausgeführt werden sollen. Listing 1 zeigt eine solche Python-Code Zelle, die nicht ausgeführt wird⁵:

Listing 1 Python: Anzahl CPU auslesen

```
import os
import sys
print(f'Dieser Rechner verfügt über {os.cpu_count()} CPU kernels.')
```

In Abbildung 3 ist ein Beispiel eines Polardiagramms aus Listing 2 zu sehen. Der Code ist doppelt vorhanden, um das Code-Listing hübsch darzustellen, wie in der Quarto Dokumentation⁶ beschrieben. der zweite Code-Block erstellt dann das Diagramm und wird selbst nicht ausgegeben:

Listing 2 Python: Beispiel Polardiagramm mit pyplot

```
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(
    subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()
```

⁵Matthes (2023)

⁶<https://quarto.org/docs/authoring/cross-references.html#code-listings>

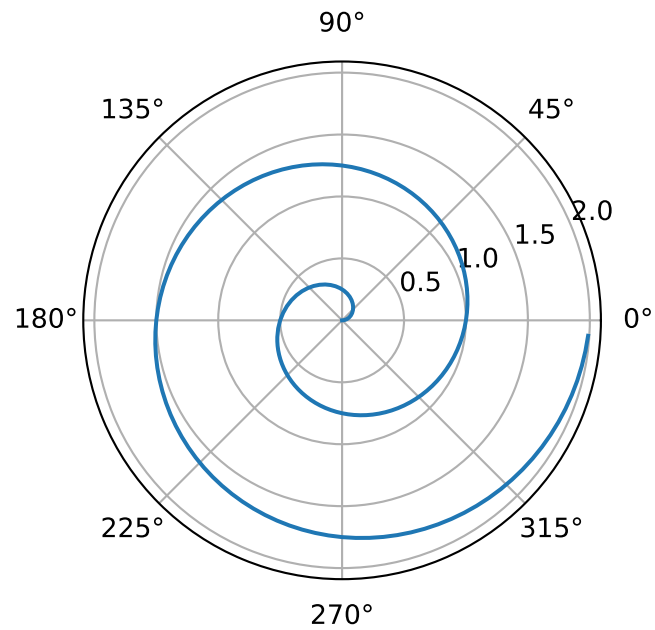


Abbildung 3: Eine Gerade in Polarkoordinaten

2.8 Anhänge

Eine Studienarbeit oder Thesis enthält in vielen Fällen auch einen Anhang, in dem zusätzliches Material untergebracht werden kann, das für den Textteil zu umfangreich, für den interessierten Leser aber dennoch nützlich ist, wie z.B. Tabelle A1.

3 Installation

3.1 Quarto etc.

Zur Installation folgen Sie bitte der Anleitung auf den Seiten von Quarto.org. Als Editor für Quarto empfiehlt sich R-Studio. Im Prinzip ist aber jeder Texteditor geeignet.

3.2 Python

3.2.1 Installation

Unter Windows empfiehlt sich die Installation von Python (z.B. Version 3.10) in Form von Miniconda, einer Verwaltungsumgebung für Python-Installationen⁷. In einem Cmd-Fenster⁸ kann dann die Funktionsfähigkeit getestet werden und die gewünschte Python-Version incl. der benötigten Pakete installiert werden:

```
conda create -n pq310 python=3.10.12

conda activate pq310
conda install numpy pandas matplotlib
conda install jupyter
```

Die so erzeugte Python-Umgebung `pq310` sollte dann auch in den Optionseinstellungen von RStudio aktiviert werden.

3.2.2 Problembeseitigung

Bei der Einrichtung von RStudio kann es wegen unterschiedlicher Python-Installationen zu Problemen wegen fehlender Python-Module kommen z.B.: `ModuleNotFoundError: No module named 'nbformat'`. Um zu ermitteln, welche Python-Installation von RStudio verwendet wird, führen Sie innerhalb Ihres Quarto-Dokuments den folgenden Code aus, der das verwendete Python-Executable und die Python-Version anzeigt.

⁷(„Miniconda — Miniconda Documentation“, o. J.)

⁸Unter Windows Startmenü: “Anaconda Prompt (miniconda3)”

```
import sys
print(sys.executable)
print(sys.version)
```

Es muss sichergestellt werden, dass in der Terminal-Umgebung, in der Quarto auf der Kommandozeile ausgeführt wird bzw. weitere Module installiert werden, die selbe Python-Umgebung genutzt wird, wie in RStudio. Dies kann z.B. mit folgenden Befehlen in der Terminal-Umgebung überprüft werden:

```
conda info --envs
python --version
```

In dieser Umgebung müssen dann fehlende Module installiert werden z.B.:

```
pip install nbformat nbclient
```

3.3 Verwendung der Vorlage

Wenn Sie dieses Dokument lesen, dann haben Sie vermutlich schon die Vorlage bei GitHub auf Ihren Rechner geladen. Falls nicht, dann sollten Sie dies jetzt tun.

Die weitere Vorbereitung besteht aus folgenden Schritten:

1. Für Ihre eigene Arbeit kopieren Sie die drei Dateien `MeineArbeit.qmd`, `MeineArbeit01.qmd` sowie `MeineArbeit02.qmd` und benennen Sie diese Kopien individuell z.B. `Thesis2024UR.qmd` etc.
2. Die Kopie von `MeineArbeit.qmd` (im Beispiel also `Thesis2024UR.qmd`) ist dabei die Hauptdatei, deren Namen in die Datei `_quarto.yml` im Abschnitt `project:render:` eingetragen werden muss
3. Am Anfang der Hauptdatei befindet sich das sog. Yaml-Frontmatter, in dem zahlreiche weitere Konfigurationsoptionen zu finden sind. Diese müssen Sie so anpassen, damit auf dem Deckblatt, der Ehrenwörtlichen Erklärung etc. die korrekten Daten aufgeführt sind. Überprüfen Sie diese Stellen unbedingt in der generierten Ausgabe, bevor Sie Ihre Arbeit abgeben!
4. Testen Sie die Funktionsfähigkeit Ihrer Hauptdatei indem Sie auf der Kommandozeile folgenden Befehl ausführen:

```
quarto render Thesis2024UR.qmd --to pdf
```
5. Die erzeugte PDF-Datei finden Sie im Verzeichnis `_output`
6. Beginnen Sie damit, Ihre Arbeit zu verfassen

3.4 Test der Silbentrennung

Mit Stand vom 2024-05-15 gibt es offenbar ein Problem mit der Silbentrennung, wenn das Quarto-interne `tinyTeX` verwendet wird. Durch setzen der Option `latex-tinytex: false` im Frontmatter-Bereich der Hauptdatei, lässt sich dies offenbar vermeiden. Voraussetzung ist allerdings eine funktionierende, von Quarto unabhängige `TeX`-Installation.

[illegible]

4 Zusammenfassung

Dieses Dokument und die damit verbundene Vorlage sollen eine Hilfe sein, um die Formalien für eine schriftliche Ausarbeitung an der FHDW bei der Verwendung von Quarto zu erfüllen. Es wurde dabei versucht, möglichst viele Automatismen von Quarto und \LaTeX zu nutzen um das Deckblatt, verschiedene Erklärungen und Verzeichnisse einfach in die Arbeit zu integrieren. Diese weitreichenden Funktionalitäten entbinden den Nutzer allerdings nicht von seiner Sorgfaltspflicht. Eine Absprache mit dem betreuenden Dozenten ist in jedem Fall ratsam.

A Erster Anhang

Dieser Code-Block kann verwendet werden, um die aktuell aktive Python-Installation zu bestimmen:

```
1 import sys
2 print(sys.executable)
3 print(sys.version)
```

```
/Users/uli/.conda/envs/quartofhdw/bin/python
3.10.13 (main, Sep 11 2023, 08:16:02) [Clang 14.0.6 ]
```

B Zweiter Anhang

Der zweite Anhang enthält eine Tabelle:

Tabelle A1: Alle Zahlen von 1..9

Col1	Col2	Col3
1	2	3
4	5	6
7	8	9

C Struktur der Vorlage

Folgende Dateien werden von dieser Vorlage verwendet.

- Hauptdatei `fhdwVorlage.qmd` (bzw. `MeineArbeit.qmd` bzw. eine entsprechende Kopie)
 - `Einleitung.qmd` (bzw. `MeineArbeit01.qmd` bzw. eine entsprechende Kopie)
 - `Hauptteil.qmd` (bzw. `MeineArbeit02.qmd` bzw. eine entsprechende Kopie)
 - `_quarto.yml`
 - * `before-body.tex`
 - * `_titlepage.tex`
 - * `toc.tex`
 - * `after-body.tex`
 - * `in-header.tex`
 - `references.csl`

Hier ein Verweis auf Abbildung 1 mit der Unterschrift von Max Mustermann. Die Abbildung befindet sich in einer anderen Datei.

Quellen

- Füller, M., Nüßer, W. & Rustemeyer, T. (2012). Context Driven Process Selection and Integration of Mobile and Pervasive Systems. *Pervasive and Mobile Computing*, 8 (3), 467–482. Elsevier B.V.
- Hocking, C. G., Furnell, S. M., Clarke, N. L. & Reynolds, P. L. (2010). A Distributed and Cooperative User Authentication Framework. *Proceedings of International Conference on Information Assurance and Security (IAS)* (S. 304–310).
- Maslennikov, D. (2011). ZeusS-in-the-Mobile – Facts and Theories. Zugriff am 20.12.2012. Verfügbar unter: <https://securelist.com/zeus-in-the-mobile-facts-and-theories/36424/>
- Matthes, E. (2023). *Python Crash Course, 3rd Edition: A Hands-on, Project-Based Introduction to Programming* (3rd edition.). San Francisco: No Starch Press.
- Meier, R. (2010). *Professional Android 2 Application Development* (S. 576). Indianapolis, IN, USA: Wiley.
- Miniconda — Miniconda Documentation. (o. J.). Zugriff am 16.11.2023. Verfügbar unter: <https://docs.conda.io/projects/miniconda/en/latest/>

Ehrenwörtliche Erklärung

Hiermit erkläre ich, Friedrich Fleißigerstudent, dass ich die vorliegende Arbeit selbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Paderborn, 2024-05-67



Friedrich Fleißigerstudent