



Duale Hochschule Baden-Württemberg Mannheim

<Bachelor>/<Projekt>arbeit

<Titel Ihrer Arbeit>

**Studiengang Wirtschaftsinformatik**

**Studienrichtung <Ihre Studienrichtung>**

Verfasser(in):	<Ihr Vor- und Nachname>
Matrikelnummer:	<Ihre Matrikelnummer>
Firma:	<Ihre Firma>
Abteilung:	<Ihre Abteilung>
Kurs:	<Ihr Kurs>
Studiengangsleiter:	<Ihr Studiengangsleiter>
Wissenschaftliche(r) Betreuer(in):	<Ihr(e) wissenschaftliche(r) Betreuer(in)>
Firmenbetreuer(in):	<Ihr(e) Firmenbetreuer(in)>
Bearbeitungszeitraum:	dd.mm.yyyy – DD.MM.YYYY

# Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Titel “<Titel Ihrer Arbeit>” selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

<Ihr Vor- und Nachname>

# Sperrvermerk

**Ein Sperrvermerk sollte nur bei berechtigtem Bedarf gesetzt werden!**

**Beachten Sie, dass mit Sperrvermerk versehene Arbeiten nicht für weitere wissenschaftliche Zwecke außerhalb des Firmenkontextes oder zur Publikation verwendet werden dürfen.**

**Wir empfehlen, wenn möglich, auf den Sperrvermerk zu verzichten.**

**Besprechen Sie diese Problematik mit Ihrer Firma!**

(Mustertext) Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung der Ausbildungsstätte vorliegt.

# Danksagung

Hier können Sie eine Danksagung schreiben.

# Contents

List of Figures	v
List of Tables	vi
Quelltextverzeichnis	vii
Algorithmenverzeichnis	viii
Abkürzungsverzeichnis	ix
Kurzfassung (Abstract)	x
<b>1 Einleitung</b>	<b>1</b>
1.1 Beispiel Abschnitt: $\text{\LaTeX}$ -Installation . . . . .	1
1.1.1 Beispiel Unterabschnitt: Aufbau eines $\text{\LaTeX}$ -Dokuments . . . . .	1
1.1.2 Beispiel Unterabschnitt auf zweiter Ebene: Präambel . . . . .	1
1.2 Noch ein Beispiel-Abschnitt . . . . .	2
<b>2 Vorbereitung der Daten</b>	<b>3</b>
<b>3 Säubern der Daten</b>	<b>6</b>
<b>4 Zusammenfassung</b>	<b>10</b>
4.1 Fazit . . . . .	10
4.2 Ausblick . . . . .	10
<b>Anhang</b>	
<b>A Beispiel-Anhang: Testanhang</b>	<b>11</b>
A.1 Abschnitt im Anhang . . . . .	11
A.2 Noch ein Abschnitt im Anhang . . . . .	11
<b>B Beispiel-Anhang: Noch ein Testanhang</b>	<b>13</b>
<b>Bibliography</b>	<b>14</b>

# List of Figures

2.1	Webseite von Intel . . . . .	4
-----	------------------------------	---

# List of Tables

# Quelltextverzeichnis

2.1	<code>clean.sh</code>	4
3.1	<code>database.R</code>	6



# Algorithmenverzeichnis

# Abkürzungsverzeichnis

<b>AD</b>	Archiv für Diplomatie, Schriftgeschichte, Siegel- und Wappenkunde
<b>BMBF</b>	Bundesministerium für Bildung und Forschung
<b>DHBW</b>	Duale Hochschule Baden-Württemberg
<b>ECU</b>	European Currency Unit
<b>EU</b>	Europäische Union
<b>RDBMS</b>	Relational Database Management System

# Kurzfassung (Abstract)

Hier können Sie die Kurzfassung (engl. Abstract) der Arbeit schreiben. Beachten Sie dabei die Hinweise zum Verfassen der Kurzfassung.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

# 1 Einleitung

Dieses Kapitel enthält die Einleitung mit ihren verschiedenen Abschnitten/Sections und Unterabschnitten.

## 1.1 Beispiel Abschnitt: $\text{\LaTeX}$ -Installation

Zur Verwendung von  $\text{\LaTeX}$ -Installation einer Distribution z. B. TeXLive, MikTeX etc. sowie eines Editors z. B. TeXStudio, TeXnicCenter etc. notwendig.

Installieren Sie zunächst die Distribution und anschließend den Editor. Beim ersten Start des Editors öffnet sich ein Konfigurationsassistent, der zunächst nach dem Pfad der installierten Distribution fragt.

Nach der Installation können Einstellungen z. B. für einen PostScript-Viewer gemacht werden. Dieser Schritt kann ohne Weiteres übersprungen werden. Entscheidend sind die Einstellungen für den pdf-Viewer.

Jetzt kann  $\text{\LaTeX}$  verwendet werden. Um die Ausgabe eines Dokumentes zu erzeugen, muss das Dokument kompiliert werden (Ausgabe > Aktives Dokument > Erstellen und betrachten).

### 1.1.1 Beispiel Unterabschnitt: Aufbau eines $\text{\LaTeX}$ -Dokuments

Ein  $\text{\LaTeX}$ -Dokument besteht in der Regel aus folgenden Komponenten:

- Präambel
- Titelseite
- Textteil

### 1.1.2 Beispiel Unterabschnitt auf zweiter Ebene: Präambel

In der Präambel werden global die Einstellungen für das gesamte Dokument definiert. Hierbei können z. B. die Seitenränder, der Zeilenabstand oder auch die Sprache für die Silbentrennung festgelegt werden. In der ersten Zeile eines jeden Dokumentes wird dabei immer die zu verwendende Klasse festgelegt. Standardmäßig kann hier die Artikel-Klasse gewählt werden:

```
\documentclass[12pt,titlepage]{article}
```

In den eckigen Klammern wird dabei u.a. die Standardschriftgröße für das gesamte Dokument festgelegt.

Außerdem werden in der Präambel die für das Dokument benötigten Pakete festgelegt. Gebräuchlich sind vor allem folgende Pakete:

- `\usepackage[ngerman]{babel}`
- `\usepackage[latin1]{inputenc}`
- `\usepackage{color}`
- `\usepackage[a4paper]{geometry}`
- `\usepackage{amssymb}`
- `\usepackage{amsthm}`
- `\usepackage{graphicx}`

Im vorliegenden Fall werden die Pakete in der Konfigurationsdatei `config.tex` festgelegt, deren Inhalt durch `\input{config}` in das Hauptdokument `master.tex` inkludiert wird.

### Beispiel Unterabschnitt auf zweiter Ebene: Titelseite

Nachdem die Dokumenten-Klasse und die zu verwendenden Pakete festgelegt worden sind, folgt die Titelseite. Da die Titelseite bereits Teil des eigentlichen Dokuments ist, muss ihr unbedingt der Befehl `\begin{document}` vorausgehen. Am Ende des Dokuments sollte der Befehl `\end{document}` gesetzt werden. Alles was nach diesem Befehl steht, wird vom Compiler nicht mehr beachtet.

## 1.2 Noch ein Beispiel-Abschnitt

Der Textteil beinhaltet nun den eigentlichen Text des Dokuments.

## 2 Vorbereitung der Daten

Intel bietet auf ihrer Webseite die Möglichkeit unterschiedliche Prozessoren miteinander zu vergleichen. Es werden alle Desktop-Prozessoren der Intel Core i7-Reihe an Prozessoren ausgewählt, damit es eine große Anzahl an Datensätzen gibt und auch die Spannweite zwischen gewissen Werten groß genug ist, dass Unterschiede leichter zu erkennen sind. Z.B. ist die Kernanzahl pro Prozessor seit der 4. Generation stark gestiegen, wohingegen die Anzahl an Prozessorkernen bei Intel Core i3-Modellen nicht so stark angestiegen sind. Die Daten werden direkt von Intels Webseite bezogen. Da Intel keine API anbietet, um deren CPU-Spezifikationen zu vergleichen und zu exportieren, muss die CSV-Datei selber heruntergeladen werden. Bevor die Daten aus der CSV ausgelesen werden können, müssen diese noch etwas aufbereitet werden. Wie in Abbildung 2.1 zu sehen ist, sind hier leider die Prozessormodelle die Spalten anstatt der Zeilen.

Bevor die CSV bereinigen können, muss also die Matrix der CSV noch einmal transponiert werden. Sonst wären die einzelnen Prozessoren die Spalten in der Datenbank, wo die bereinigte Daten der CSV abgespeichert werden. Weiterhin sind nicht alle Einträge in der CSV interessant für die Auswertung der Daten. Viele der Einträge in der CSV listen nur Features oder Prozessorerweiterungen auf, die nicht einfach zu vergleichen sind oder sich nicht visualisieren lassen. Weiterhin sind manche Einträge bei älteren Prozessoren komplett leer, weswegen die Datenmenge in diesen Spalten stark absinkt. Daher werden nur ausgewählte Spalten ausgewählt, die interessant für die Fragestellung sind und auch genügend Datenpunkte haben.

56 Intel® Produkte vergleichen

Vergleich exportieren

Intel® Core™ i7-8700K Processor

Intel® Core™ i7-8700 Processor

Intel® Core™ i7-8700T Processor

Intel® Core™ i7-12700 Processor

Intel® Core™ i7-12700F Processor

Intel® Core™ i7-12700K Processor

**Hauptdaten**

Produktsortiment	Intel® Core™ i7 Prozessoren der 8. Generation	Intel® Core™ i7 Prozessoren der 8. Generation	Intel® Core™ i7 Prozessoren der 8. Generation	Intel® Core™ i7 Prozessoren der 12. Generation	Intel® Core™ i7 Prozessoren der 12. Generation	Intel® Core™ i7 Prozessoren der 12. Generation
Vertikales Segment	Desktop	Desktop	Desktop	Desktop	Desktop	Desktop
Prozessornummer	i7-8700K	i7-8700	i7-8700T	i7-12700	i7-12700F	i7-12700K
Lithographie	14 nm	14 nm	14 nm	Intel 7	Intel 7	Intel 7
Empfohlener Kundenpreis	\$359.00	\$303.00	\$303.00	\$373.00-\$383.00	\$345.00-\$355.00	\$450.00

**CPU-Spezifikationen**

Anzahl der Kerne	6	6	6	12	12	12
------------------	---	---	---	----	----	----

Figure 2.1: Webseite zum Produktvergleich von Intel-Prozessoren  
Quelle: [2]

Aufgrund der mächtigen Standardwerkzeugen eines POSIX-kompatiblen Betriebssystems wie Linux, wurde 'awk' verwendet, um die Spalten in Zeilen umzubauen und umgekehrt. 'awk' arbeitet nicht nur mit Zeilen, wie z.B. 'grep' oder 'sed', sondern kann auch in einzelnen Spalten filtern. Normalerweise sind die Standardtrennzeichen von Spalten für 'awk' Leerzeichen und Tabulatoren. Es ist aber möglich mit dem '-F' Flag anzugeben, dass ein anderes Trennzeichen verwendet wird. In 'awk' wird normalerweise mit regulären Ausdrücken gefiltert, aber aufgrund einer statischen CSV-Datei kann direkt nach Spaltennamen sortiert werden. Das entsprechende Skript sieht folgendermaßen aus:

```

1 #!/bin/bash
2
3 awk -F ',' '
4 NR==5 { | |
5 NR==7 { | |
6 NR==8 { | |
7 NR==12 { | |
8 NR==13 { | |
9 NR==14 { | |
10 NR==15 { | |
11 NR==16 { | |
12 NR==17 { | |
13 NR==18 { | |
14 NR==19 { | |
15 NR==22 { | |
16 NR==23 { | |

```

```

17 NR==24    ||
18 NR==25    ||
19 NR==26    ||
20 NR==27    ||
21 NR==28    ||
22 NR==29    ||
23 NR==30    ||
24 NR==31    ||
25 NR==36    ||
26 NR==42    ||
27 NR==56    ||
28 NR==57    ||
29 NR==58    ||
30 NR==59    ||
31 NR==60    ||
32 NR==61    ||
33 NR==62    ||
34 NR==82    ||
35 NR==90
36 {print $0}' "Intel_UPE_ComparisonChart_2024_11_04_i7.csv" |
37
38 awk '
39 {
40     for (i=1; i<=NF; i++)
41         a[i] = a[i] ? a[i] "," $i : $i
42 }
43 END {
44     for (i=1; i in a; i++)
45         print a[i]
46 }
47 ' FS=, OFS=, > clean.csv

```

Quelltext 2.1: clean.sh

Das Skript besteht aus zwei Teilen: Im ersten Teil werden die entsprechenden Spalten mit 'awk' nach Spaltennummer ausgefiltert, welche vorher als interessant deklariert wurden, und in den zweiten Teil gepiped. Im zweiten Teil werden die jeweiligen Spalten in einer for-Schleife in neue Zeilen geladen. Nachdem die gesamte Datei eingelesen wurde, kann diese wieder transponiert ausgegeben werden. Nun kann mit dem eigentlichen Säubern der Daten begonnen werden.



# 3 Säubern der Daten

Nun ist die CSV vorbereitet für die eigentliche Säuberung. Die Daten liegen in den entsprechenden Spalten vor und alle uninteressanten Daten wurden entfernt. Bevor die Daten jedoch in einer Datenbank in Tabellenform gespeichert werden können, sollten noch entsprechende Zahlenwerte als Zahl erkannt werden. Weiterhin beinhalten viele Spaltennamen immer noch Sonderzeichen und Leerzeichen, weswegen sie schwierig zu bearbeiten sind.

Um diese Probleme zu adressieren, wurde R zur Bereinigung der Daten ausgewählt. R ist eine Skriptsprache für statistische Berechnungen. Es bietet mächtige Werkzeuge zur Auswertung und Säuberung von Daten und ist damit eine exzellente Wahl um dieses Problem zu lösen. Es bietet Möglichkeiten die Spaltennamen automatisch umzubenennen, indem es problematische Zeichen, wie z.B. Leerzeichen und Sonderzeichen, erkennt und durch unproblematische Zeichen ersetzt. Weiterhin werden auch alle Großbuchstaben durch Kleinbuchstaben ersetzt. Darüber hinaus ist R eine dynamisch typisierte Skriptsprache. Datentypen werden je nach Kontext automatisiert erkannt. Wenn die entsprechenden Spalten von allen Zeichen, die keine Zahlen sind, gesäubert werden, lassen sich diese problemlos in Zahlen umwandeln. Die entsprechenden Einheiten, welche vorher direkt in den Spalten gespeichert wurden, werden stattdessen noch in den Spaltennamen hinzugefügt.

Wenn alle Spalten und Spaltennamen gesäubert bzw. angepasst wurden, können die Daten in einer Datenbank gespeichert werden.

Mit den vorher genannten Überlegungen im Hinterkopf ist folgendes Skript entstanden:

```
1 #!/usr/bin/Rscript
2
3 library(RPostgres)
4
5 intel <- read.csv("clean.csv")           #Reading
6 intel <- janitor::clean_names(intel)      #Cleaning the column
      names
7
8 #cleaning and mutating columns to numbers
9 intel$max_turbo_taktfrequenz <- as.numeric(gsub(" GHz", "",
      intel$max_turbo_taktfrequenz))
10 intel$lithographie <- gsub("Intel 7", "10 nm",
      intel$lithographie)
11 intel$lithographie <- as.numeric(gsub(" nm", "",
      intel$lithographie))
```

```

12 intel$intel_turbo_boost_technik_2_0_taktfrequenz <- as.
    numeric(gsub(" GHz", "",
        intel$intel_turbo_boost_technik_2_0_taktfrequenz))
13 intel$grundtaktfrequenz_des_prozessors <- gsub(" \\| ", ".",
    intel$grundtaktfrequenz_des_prozessors)
14 intel$grundtaktfrequenz_des_prozessors <- as.numeric(gsub("
    GHz", "", intel$grundtaktfrequenz_des_prozessors))
15 intel$cache <- gsub(" MB Intel Smart Cache", "",
    intel$cache)
16 intel$cache <- as.numeric(gsub(" MB", "", intel$cache))
17 intel$bus_taktfrequenz <- as.numeric(gsub(" GT/s", "",
    intel$bus_taktfrequenz))
18 intel$verlustleistung_tdp <- as.numeric(gsub(" W", "",
    intel$verlustleistung_tdp))
19 intel$intel_turbo_boost_max_technology_3_0_frequency <- gsub
    (" \\| ", ".",
        intel$intel_turbo_boost_max_technology_3_0_frequency)
20 intel$intel_turbo_boost_max_technology_3_0_frequency <- as.
    numeric(gsub(" GHz", "",
        intel$intel_turbo_boost_max_technology_3_0_frequency))
21 intel$single_p_core_turbo_frequency <- as.numeric(gsub(" GHz
    ", "", intel$single_p_core_turbo_frequency))
22 intel$single_e_core_turbo_frequency <- as.numeric(gsub(" GHz
    ", "", intel$single_e_core_turbo_frequency))
23 intel$e_core_base_frequency <- gsub(" GHz", "",
    intel$e_core_base_frequency)
24 intel$e_core_base_frequency <- as.numeric(gsub("900 MHz",
    "0.9", intel$e_core_base_frequency))
25 intel$total_l2_cache <- as.numeric(gsub(" MB", "",
    intel$total_l2_cache))
26 intel$processor_base_power <- as.numeric(gsub(" W", "",
    intel$processor_base_power))
27 intel$maximum_turbo_power <- as.numeric(gsub(" W", "",
    intel$maximum_turbo_power))
28 intel$grundtaktfrequenz_der_grafik <- as.numeric(gsub(" MHz",
    "", intel$grundtaktfrequenz_der_grafik))
29 intel$max_dynamische_grafikfrequenz <- as.numeric(gsub(" GHz
    ", "", intel$max_dynamische_grafikfrequenz))
30 intel$max_videospeicher_der_grafik <- as.numeric(gsub(" GB",

```

```

    "", intel$max_videospeicher_der_grafik))
31 intel$x4k_unterstutzung <- gsub("Hz", "",
    intel$x4k_unterstutzung)
32 intel$x4k_unterstutzung <- as.numeric(gsub("Yes \\| at ",
    "", intel$x4k_unterstutzung))
33
34
35 #Putting the units back into the column names
36 colnames(intel)[colnames(intel) == "max_turbo_taktfrequenz"]
    <- "max_turbo_taktfrequenz_GHz"
37 colnames(intel)[colnames(intel) == "lithographie"] <- "
    litographie_nm"
38 colnames(intel)[colnames(intel) == "
    intel_turbo_boost_technik_2_0_taktfrequenz"] <- "
    intel_turbo_boost_technik_2_0_taktfrequenz_GHz"
39 colnames(intel)[colnames(intel) == "
    grundtaktfrequenz_des_prozessors"] <- "
    grundtaktfrequenz_des_prozessors_GHz"
40 colnames(intel)[colnames(intel) == "cache"] <- "cache_MB"
41 colnames(intel)[colnames(intel) == "bus_taktfrequenz"] <- "
    bus_taktfrequenz_GT_per_s"
42 colnames(intel)[colnames(intel) == "verlustleistung_tdp"] <-
    "verlustleistung_tdp_W"
43 colnames(intel)[colnames(intel) == "
    intel_turbo_boost_max_technology_3_0_frequency"] <- "
    intel_turbo_boost_max_technology_3_0_frequency_GHz"
44 colnames(intel)[colnames(intel) == "
    single_p_core_turbo_frequency"] <- "
    single_p_core_turbo_frequency_GHz"
45 colnames(intel)[colnames(intel) == "
    single_e_core_turbo_frequency"] <- "
    single_e_core_turbo_frequency_GHz"
46 colnames(intel)[colnames(intel) == "e_core_base_frequency"]
    <- "e_core_base_frequency_GHz"
47 colnames(intel)[colnames(intel) == "total_l2_cache"] <- "
    total_l2_cache_MB"
48 colnames(intel)[colnames(intel) == "processor_base_power"] <-
    "processor_base_power_W"
49 colnames(intel)[colnames(intel) == "maximum_turbo_power"] <-

```

```

    "maximum_turbo_power_W"
50 colnames(intel)[colnames(intel) == "
    grundtaktfrequenz_der_grafik"] <- "
    grundtaktfrequenz_der_grafik_MHz"
51 colnames(intel)[colnames(intel) == "
    max_dynamische_grafikfrequenz"] <- "
    max_dynamische_grafikfrequenz_GHz"
52 colnames(intel)[colnames(intel) == "
    max_videospeicher_der_grafik"] <- "
    max_videospeicher_der_grafik_GB"
53 colnames(intel)[colnames(intel) == "x4k_unterstutzung"] <- "
    x4k_unterstutzung_at"
54
55
56 con <- dbConnect(
57   RPostgres::Postgres(),
58   dbname = "bda",
59   host = "localhost" ,
60   port = 5432,
61   user = "bda",
62   password = "bda",
63 )
64
65 dbWriteTable(con, "intel", intel, row.names = FALSE,
66               overwrite = TRUE)
67 dbDisconnect(con)

```

Quelltext 3.1: database.R

Als erstes wird die transponierte CSV eingelesen und die Spaltennamen mit Hilfe der 'janitor' Library gesäubert. Danach kommt ein großer Block an Funktionen, die die einzelnen Spalten von Einheiten und komischer Formatierung säubern. Danach kommt ein großer Block zur Umbenennung der Spaltennamen. Im letzten Schritt wird der Dataframe in einer existierenden Datenbank gespeichert. Ein großer Vorteil an der Funktion ist, dass die Tabelle innerhalb der Datenbank vorher nicht existieren muss. R erstellt von selber eine neue Tabelle mit entsprechenden Datentypen basierend auf den Datentypen innerhalb des Dataframes. Daher ist es wichtig die Daten innerhalb des Skripts schon zu Nummern zu konvertieren.

## 4 Speichern der Daten

Aus dem vorherigen Kapitel ist bereits hervorgegangen, dass die Daten in einer Datenbank gespeichert werden. In diesem Fall ist die Wahl auf PostgreSQL gefallen. PostgreSQL ist ein relationales Datenbankmanagementsystem. Es ist komplett ACID-kompatibel und speichert Daten in geordneter Form als Tabellen ab. Da die Datenquelle bereits als CSV geliefert wird, bietet es sich an diese in einer entsprechenden Datenbank zu speichern. CSV-Dateien speichern bereits Daten in Tabellenform, weswegen die Wahl direkt auf eine Datenbank gefallen ist.

Als Alternativen zu PostgreSQL gibt es viele, sind aber teils kostenpflichtig oder nicht so weit verbreitet wie PostgreSQL. Datenbankmanagementsysteme wie Oracle RDBMS oder Microsoft sind für große, Enterprise-Skala Deployments konzipiert und kosten entsprechend eine gute Menge Geld. MariaDB bzw. MySQL und SQLite wären Open-Source Alternativen zu PostgreSQL, welche auch gut geeignet sind. SQLite jedoch bietet keine Serverfunktionalität weswegen nur lokal gearbeitet werden kann. MariaDB/MySQL haben keinen nennenswerten Funktionalitätsunterschied, aber wurden nicht gewählt, weil mit PostgreSQL mehr Erfahrung im Team existiert.

# 5 Zusammenfassung

Dieses Kapitel enthält die Zusammenfassung der Arbeit mit Fazit und Ausblick.

## 5.1 Fazit

...

## 5.2 Ausblick

...

# A Beispiel-Anhang: Testanhang

Anhänge werden am Ende Ihrer Arbeit vor dem Literaturverzeichnis und dem Index eingefügt.

## A.1 Abschnitt im Anhang

Blabla

## A.2 Noch ein Abschnitt im Anhang

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.



# B Beispiel-Anhang: Noch ein Testanhang

nochmal: lipsum ...

# Bibliography

- [1] Min Chen et al. *Big Data: Related Technologies, Challenges and Future Prospects*. SpringerBriefs in Computer Science. Cham: Springer International Publishing, 2014. ISBN: 978-3-319-06244-0 978-3-319-06245-7. DOI: 10.1007/978-3-319-06245-7. URL: <https://link.springer.com/10.1007/978-3-319-06245-7> (visited on 11/13/2024).
- [2] I7. *Intel Core i7 Generation 4 bis 14*. i7. Nov. 4, 2024. URL: <https://www.intel.de/content/www/de/de/products/compare.html?productIds=212279,212280,212047,212048,212251,199325,199335,199314,199316,199318,191048,191792,193738,190885,186604,148263,140642,126684,97129,93339,88200,88195,87718,88040,80807,80808,80809,80814,80806,77656,76642,75121,75122,75123,75124,75125,236781,236794,236854,236783,236789,230492,230490,230491,230500,230489,134596,134591,134592,134594,134595,129948,126686,97122,97128,88196> (visited on 11/04/2024).
- [3] Mohsen Marjani et al. "Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges". In: *IEEE Access* 5 (2017). Conference Name: IEEE Access, pp. 5247–5261. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2689040. URL: <https://ieeexplore.ieee.org/abstract/document/7888916> (visited on 11/13/2024).