

# Das kiviletter-Paket v5.3

Marei Peischl  
marei@peitex.de

2023-08-02

## Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
<b>2. Grundlegende Eigenschaften des Pakets</b>	<b>2</b>
<b>3. Paketoptionen</b>	<b>2</b>
<b>4. Aufbau der Briefdokumente</b>	<b>2</b>
<b>5. Tabellen</b>	<b>2</b>
5.1. SimpleTabular . . . . .	3
5.1.1. Kopfzeile . . . . .	3
5.1.2. Spaltenkonfiguration (fortgeschrittene Nutzer*innen) . . . . .	3
<b>6. PricingTabular</b>	<b>4</b>
6.1. Spaltenbreiten . . . . .	4
6.2. Kopfzeileinträge . . . . .	5
6.3. Tabellenzellenausrichtung . . . . .	5
6.4. Farbige Tabellen . . . . .	5
6.5. Trennlinien zwischen den Einträgen . . . . .	6
6.6. Reihenfolge/Anzahl der Spalten ändern . . . . .	6
<b>A. Aufbau des kivitendo Vorlagensatzes</b>	<b>7</b>
A.1. Quickstart – Wo kann was angepasst werden? . . . . .	7
A.2. Aufbau . . . . .	7
A.3. Mandant*innen / Firma . . . . .	7
A.4. Identitäten . . . . .	7
A.5. Währungen/Konten . . . . .	8
A.6. Briefbogen/Logos . . . . .	8
A.7. Fußzeile . . . . .	8
A.8. Seitenstil/Basislayout . . . . .	8
A.9. Absenderergänzung . . . . .	9
A.10. Allgemeine T <sub>E</sub> Xnische Hinweise . . . . .	10
A.10.1. Änderung der Basisschriftart . . . . .	10
A.10.2. Unterscheidungen durch String-Vergleich . . . . .	10

## 1. Einleitung

Das kiviletter-Paket liefert eine Erweiterung der kivitendo Vorlagendateien. Es bietet eine deutlich höhere Flexibilität für die Gestaltung und Anpassung von Vorlagen.

`\usepackage[ <Optionen>]{kiviletter}`

`\SetupKiviletter` Eine spätere Modifikation ist ab Version 5.2 auch über `{<Optionen>}` nach dem Laden des Pakets möglich.

## 2. Grundlegende Eigenschaften des Pakets

Das Paket basiert auf dem KOMA-Script-Paket `scrletter`. Alle Mechanismen sind kompatibel zu diesem Paket konstruiert, sodass die Layoutanpassungen, die in der KOMA-Script Dokumentation [1] beschrieben sind auch für `kiviletter` gelten.

## 3. Paketoptionen

Um den Anforderungen von `kivitando` besser gerecht zu werden setzt `kiviletter` ein paar andere Voreinstellungen als `scrletter` dies tut. Die Grundkonfiguration lautet:

```
fromlogo=true, % Zeigt das Logo mit an, sofern gesetzt
fromalign=right, % Absenderadresse oben Rechts im Kopf
firstfoot=false, % Keine Unterscheidung zwischen dem ersten Seitenfuß
→ und übrigen Seiten
refline=nodate, % Keine Datumsangabe in der Geschäftszeile
```

Diese Optionen können durch selbst-gesetzte Paketoptionen überschrieben werden. Daneben erweitert `kiviletter` die Funktionalität auch um Eigene:

`infobox` (Schalter) (default: `true`)

Dieser Schalter schaltet zwischen einer Geschäftszeile und der Platzierung der Daten in einer Infobox rechts neben dem Adressfeld um. Grundlegend entspricht ein solcher Block der aktuellen Empfehlung der DIN5008:2020. Der Infoblock wird über die KOMA-Brief-Variable `location` umgesetzt. Die Inhalte können jederzeit vor der Erzeugung des Briefkopfes überschrieben werden.

`footer` (Schalter) (default: `true`)

Da die Fußzeile bei `kivitando` für alle Seiten einheitlich behandelt werden soll, wird hier nicht der Mechanismus aus `scrletter` verwendet. Diese Option erlaubt es den Fuß abzuschalten, z. B. falls dieser bereits auf dem Briefpapier vorgedruckt ist. Intern ist die KOMA-Option `firstfoot` dafür auf `false` gesetzt.

Alle unbekanntenen Optionen werden an das zugrundeliegende Paket `scrletter` weitergereicht.

## 4. Aufbau der Briefdokumente

Der Aufbau der Briefdokumente ist identisch zu `scrletter`. Da eine gute deutschsprachige Dokumentation hierfür existiert wird in diesem Dokument nicht weiter darauf eingegangen [1]. Die Dokumentation führt die Platzierung der Briefelemente unter dem Begriff der Pseudolängen ein. Darüber ist eine Anpassung derselben möglich.

## 5. Tabellen

Für eine vereinfachte Konfiguration der Tabellen wurden zwei Umgebungen implementiert, die beide unterschiedlich viele Automatismen nutzen. Grundsätzlich sollen beide Varianten Seitenumbrüche in den Tabellen erlauben und sich der Textbreite anpassen.

## 5.1. SimpleTabular

Der einfache Tabellentyp ist die Umgebung `SimpleTabular`. Diese basieren auf dem `xltabular`-Paket, die die sich der Textbreite anpasst. Sie wird im kivitendo Vorlae-gensatz in den Dateien `zahlungserinnerung_invoice.tex`, `zahlungserinnerung.tex` und `statement.tex` verwendet.

Sie verfügt über ein optionales Argument, um die Spaltenkonfiguration und die Kopfzeile anzupassen. Die Voreinstellung (also ohne optionales Argument) entspricht der, der folgenden Angabe:

```
\begin{SimpleTabular}[colspec=rrX,headline=  
→ {\usekomafont{tablehead}\position & \usekomafont{tablehead}\menge  
→ & \usekomafont{tablehead}\bezeichnung},  
  <Tabelleninhalt>  
\end{SimpleTabular}
```

`\SetupSimpleTabular` Um die Optionen global zu justieren existiert der Befehl `{<Optionen>}`  
Die zusätzlichen Optionen für alle nachfolgenden `SimpleTabular`-Umgebungen.

### 5.1.1. Kopfzeile

Die Kopfzeile wird über den Optionsschlüssel `headline` angepasst. Entsprechend dem  $\text{\LaTeX}$ -Standard werden Tabellen Spalten mit `&` getrennt. `\bfseries` setzt den Tabellenkopf zusätzlich in Fettschrift. Analog sind natürlich auch andere Schriftarteneinstellungsschalter möglich.

### 5.1.2. Spaltenkonfiguration (fortgeschrittene Nutzer\*innen)

Die voreingestellte Spaltenkonfiguration entspricht `rrX`, also zwei rechtsbündigen Spalten und einer Blocksatzspalte, die die restliche Breite bis die Textbreite erreicht ist auffüllt. Soll von dieser Spaltenkonfiguration abgewichen werden, steht der Optionsschlüssel `colspec` zur Verfügung. Das folgende Beispiel tauscht die beiden rechtsbündigen Spalten in linksbündige:

```
\begin{SimpleTabular}[colspec=llX]
```

Als Spaltentypen sind Konfigurationen aus den folgenden Einträgen am sinnvollsten:

- `l`, `r`, `c`: Linksbündig, rechtsbündig, zentriert. Spaltenbreite passt sich dem Inhalt an. Zeilenumbrüche sind in diese Zellen nicht möglich
- `p{<Breite>}`
- `X`: Blocksatz, Spaltenbreite füllt den übrigen Platz auf. Bei mehreren `X`-Spalten wird gleichmäßig aufgeteilt

Zusätzlich ist es möglich die Währung automatisch in der Spalte zu ergänzen. Der Mechanismus ist so konstruiert, dass diese nicht in der Kopfzeile sondern lediglich in den Inhaltszeilen eingefügt wird. In diesem Fall wird die Spaltenspezifikation durch `<{\tabcurrency}>` ergänzt. Eine rechtsbündige Spalte mit Währungsangabe wird somit durch `r<{\tabcurrency}>` erzeugt. Falls eine solche Spalte zusätzlich über eine feste Breite verfügen soll, definiert `kiviletter` zusätzlich den Spaltentyp `K{<Breite>}`.

## 6. PricingTabular

PricingTabular wurde entwickelt um Tabellen für Rechnungen vereinfacht erstellen zu können. Die Voreinstellung verfügt über die Spaltenpos, id, desc, amount, price, pricetotal. Alle Spalten, außer der Spaltesdesc haben eine feste Breite. Falls die voreingestellte Breite nicht passt, kann diese nachträglich modifiziert werden (siehe Abschnitt 6.1).

PricingTabular (env.) [ <Optionen>]

Zusätzliche Einstellungen können entweder als optionales Argument zur Umgebung vorgenommen werden oder über das Makro\SetupPricingTabular.

\SetupPricingTabular {<Optionen>}

Bei Nutzung von \SetupPricingTabular gelten die zusätzlichen Optionen für alle nachfolgenden PricingTabular.

Die PricingTabular selbst basiert auf dem longtable-Paket und wird entsprechend strukturiert und umgebrochen. Da die Umbrüche allerdings besser konfigurierbar sein sollen wurde zusätzlich noch die Variante PricingTabular\* entwickelt:

```
\begin{PricingTabular*}[ <<Optionen>>]
  % eigentliche Tabelle wird bei kivitendo über die Positionen in
  ↪ einer Schleife erzeugt
  \FakeTable{Position & Artikelnummer &%
    Kurzbeschreibung
    \ExtraDescription{Lange Beschreibung}
    &%
    Anzahll.&%
    Einzelpreis &%
    Gesamtpreis \tabularnewline
  }
  \begin{PricingTotal}%Tabellenblock für die Summen
    \schlussbetrag & Gesamtbetrag\\
  \end{PricingTotal}
\end{PricingTabular*}
```

Wie der Name der internen \FakeTable schon sagt, handelt es sich hier technisch nicht mehr um eine Tabelle. Die PricingTabular\* interiert über die einzelnen Positionszeilen und platziert die Teile einzeln. Dies ermöglicht mehr Flexibilität bei der Modifikation.

### 6.1. Spaltenbreiten

Die Spaltenbreiten werden angepasst, indem der Spaltenname als Optionsschlüssel verwendet wird. Um die Positionsspalte zu ändern ist somit die Option pos=<Breite> notwendig. Hier können alle Längenangaben verwendet werden, die L<sup>A</sup>T<sub>E</sub>X versteht. (cm, mm, em, ex, ...)

Die Spaltenbreite der Spaltesdesc für die Artikelbeschreibung nimmt dabei jeweils den übrigen Platz ein.

```
pos=5ex,
id=4em,
amount=5em,
price=7em,
pricetotal=7em,
```

## 6.2. Kopfzeileneinträge

Die Kopfzeileneinträge werden über die Option

```
<Spaltenname>/header=<Neue Beschriftung>
```

angepasst. Vorbelegt ist die Konfiguration:

```
pos / header=\position,  
id / header=\artikelnummer,  
desc / header=\bezeichnung,  
amount / header=\menge,  
price / header=\einzelpreis,  
pricetotal / header=\gesamtpreis,
```

## 6.3. Tabellenzellenausrichtung

Wie die übrigen Eigenschaften einzelner Spalten kann auch der Spaltentyp und damit die Ausrichtung konfiguriert werden.

```
<Spaltenname>/colspec=<Spaltentyp>
```

Dies gilt für alle Spalten außer der Beschriftungsspalte `desc`, da sich diese in der Konstruktion stark von den übrigen unterscheidet. Wichtig ist, dass die Spaltentypen hier ein Argument für die Breite akzeptieren müssen.

In der Voreinstellung sind alle Spalten vom Typ K beziehungsweise die Preis-Spalten vom Typ P. Diese beiden Spaltentypen sind definiert als:

```
\newcolumnntype{K}[1]{>{\raggedleft\arraybackslash}p{#1}}  
\newcolumnntype{P}[1]{K{#1}<{\_kivi_tab_column_currency:}}
```

Sie sind somit beide Spalten mit fester breite, die in der Box oben ausgerichtet sind (p) allerdings rechtsbündig ausgerichtet. Die P-Spalten tragen zusätzlich (Außerhalb der Kopfzeile) die Währung ein.

## 6.4. Farbige Tabellen

Versionen ab Juli 2021 enthalten die Möglichkeit farbige Tabellen zu nutzen. Die Optionen für die `PricingTabular` Umgebung können wie folgt konfiguriert werden:

<code>color-rows</code>	(Schalter)	(default: <code>false</code> )
	Schaltet Farbige Tabellen ein.	
<code>rowcolor-odd</code>	(Farbe)	(default: <code>black!10</code> )
	Farbe für Tabellenzeilen mit ungerader Zählung. Eine leere Farbangabe erzeugt keine Farbbox.	
<code>rowcolor-even</code>	(Farbe)	(default: <code>&lt;leer&gt;</code> )
	Wie <code>rowcolor-odd</code> bei gerade Zählung.	
<code>rowcolor-header</code>	(Farbe)	(default: <code>black!35</code> )
	Farbe der Kopfzeile	
<code>rowcolor-total</code>	(Farbe)	(default: <code>black!35</code> )
	Farbe für den Summenblock	
	Die Angabe hinter dem Kommentarzeichen entspricht der Voreinstellung.	

## Beispiel für die Farbigen Tabellen

Pos.	Art.- Nr.	Bezeichnung	Menge	E-Preis	G-Preis
1	ABC-1	Kurzbeschreibung Lange Beschreibung	1 Stk.	500 €	500 €
2	ABC-2	Kurzbeschreibung Lange Beschreibung	3 h	130 €	390 €
3	ABC-3	Kurzbeschreibung Lange Beschreibung	5 Stk.	12 €	60 €
<b>Gesamtbetrag</b>					<b>950 €</b>

### 6.5. Trennlinien zwischen den Einträgen

Die Umgebung `PricingTabular` hat die Möglichkeit horizontale Linien zwischen den Einträgen der `\FakeTable` einzuziehen. Die einfachste Möglichkeit hierfür ist die Option `hrule`, sie setzt automatisch eine Linie der Dicke `\lightrulewidth`. Da diese Linie formal nicht innerhalb der Tabelle platziert wird, können Linienmakros für Tabellen hier nicht verwendet werden. Falls dennoch eine manuelle Anpassung der Maße notwendig ist, kann direkt der Code zur Erzeugung der Linie übergeben werden. Die Option `hrule` entspricht dabei einer Tabellenlinie die optisch der `\midrule` des `booktabs` Pakets entspricht. In Kombination mit Farbigen Tabellen ist hier Vorsicht geboten, da der Abstand nicht mit zur farbigen Box gerechnet wird. Allerdings ist in diesem Fall eine Trennlinie ohnehin nicht notwendig, da durch die Farbige Unterscheidung bereits eine zusätzlich Abtrennung der Tabellenzeilen gesetzt wird.

### 6.6. Reihenfolge/Anzahl der Spalten ändern

Die Reihenfolge wurde über die Option `columns` festgelegt. Soll daher eine Tabelle mit nur drei Spalten und lediglich bestehend aus Produktnummer, Beschreibung und Menge genutzt werden, ist dies mit der Option `columns={id,desc,amount}` möglich.

Einzelne Spalten können auch über `<Spaltenname>=false` abgeschaltet werden. Dies ist z. B. dann hilfreich, wenn die Angabe einer Produktnummer aus Platzgründen nicht sinnvoll ist (`id=false`).

## A. Aufbau des kivitendo Vorlagensatzes

### A.1. Quickstart – Wo kann was angepasst werden?

In keinem Fall sollten Dateien mit der Endung `*.cls` oder `*.sty` geändert werden. Durch Änderungen an diesen Dateien verhindert man Updates auf neuer Versionen. Zwar kopiert kivitendo die Datei und bearbeitet diese beim Update nicht. Allerdings sind sämtliche Änderungen über die Konfigurationsdateien möglich und erhöhen die Wartbarkeit.

- `insettings.tex` :
  - Pfad zu Angaben über Mandant\*innen (default: firma)
  - Logo/Briefpapier, falls für alle Mandant\*innen in gleicher Struktur. Sonst in der `ident.tex`.
  - Layout der Kopf/Fußzeile
  - innerhalb dieser Datei werden auch die folgenden Dateien geladen:
    - `firma/ident.tex` Mandant\*innenspezifische Konfiguration, Adressdaten
    - `firma/<währungskürzel>_account.tex`
- Sprache/Übersetzungen.
  - Es muss mindestens eine Sprache angelegt werden!
  - deutsch.tex** Textschnipsel für Deutsch
    - Dafür eine Sprache mit Vorlagenkürzel DE anlegen
  - english.tex** Textschnipsel für Englisch
    - Dafür eine Sprache mit Vorlagenkürzel EN anlegen

Alle dokumententypspezifischen Einstellungen müssen in der jeweiligen Template-Datei modifiziert werden.

### A.2. Aufbau

Die Grundstruktur besteht je Dokumententyp aus einer Basisdatei und verschiedenen Setup-Dateien.

Die Basis wurde so überarbeitet, dass Dokumente nun generell auf der Dokumentenklasse `scrartcl.cls` basieren und das Paket `kiviletter.sty` benutzen.

Mandant\*innenspezifische Konfiguration findet sich in der Datei `insettings.tex` und dem Ordner eines spezifischen Mandant\*innen (default=`*firma/*`).

### A.3. Mandant\*innen / Firma

Um gleiche Vorlagen für verschiedene Firmen verwenden zu können, wird je nach dem Wert der Kivitendo-Variablen `<%kivicompany%>` ein Firmenverzeichnis ausgewählt (siehe `insettings.tex`), in dem Briefkopf, Identitäten und Währungs-/Kontoeinstellungen hinterlegt sind. `<%kivicompany%>` enthält den Namen des verwendeten Mandant\*innendaten. Ist kein Firmenname eingetragen, so wird das generische Unterverzeichnis `*firma*` verwendet.

### A.4. Identitäten

In jedem Firmen-Unterverzeichnis soll eine Datei `ident.tex` vorhanden sein, die mit `\newcommand` Werte für `\telefon`, `\fax`, `\firma`, `\strasse`, `\ort`, `\ustid`, `\email` und `\homepage` definiert.

## A.5. Währungen/Konten

Für jede Währung (siehe `insettings.tex`) soll eine Datei vorhanden sein, die das Währungssymbol (`\currency`) und folgende Angaben für ein Konto in dieser Währung enthält `\kontonummer`, `\bank`, `\bankleitzahl`, `\bic` und `\iban`. So kann in den Dokumenten je nach Währung ein anderes Konto angegeben werden. Nach demselben Schema können auch weitere, alternative Bankverbindungen angelegt werden, die dann in `insettings.tex` als Variable in der Fußzeile eingefügt werden.

Als Fallback (falls kivitendo keine Währung an das Druckvorlagen-System übergibt) ist Euro eingestellt. Dies lässt sich in der `insettings.tex` über das optionale Argument von `\setupCurrencyConfig` anpassen, z.B.

```
\setupCurrencyConfig[chf]{\identpath}{\lxcurrency}
```

für Schweizer Franken als Standardwährung.

## A.6. Briefbogen/Logos

Eine Hintergrundgrafik oder ein Logo kann in Abhängigkeit vom Medium (z.B. nur beim Verschicken mit E-Mail) eingebunden werden.

Desweiteren sind (auskommentierte) Beispiele enthalten für eine Grafik als Briefkopf, nur ein Logo, oder ein komplettes A4-PDF als Briefpapier.

Absolute Positionierung innerhalb des Brief-Layouts ist über die entsprechende Dokumentation des `sclayer`-Paketes möglich. Da die Voreinstellungen bereits einige Sonderfälle automatisch berücksichtigen ist mit den Anpassungen Vorsicht geboten. Sämtliche Einstellungen sollten jedoch außerhalb der `*.sty`-Dateien vorgenommen werden. Anpassungen der `insettings.tex` betreffen hierbei alle Mandant\*innen. Spezifischere Einstellungen sind über die zugehörige Konfigurationsdatei (`ident.tex`) möglich. In diesem Fall kann zum Ende der `insettings` eine weitere Konfigurationsdatei über die Verwendung von `\identpath` geladen werden. Ein Beispiel ist in der `insettings.tex` enthalten.

## A.7. Fußzeile

Die Tabelle im Fuß verwendet die Angaben aus `firma/ident.tex` und `*firma/_account.tex`. Ihre Struktur wird in der `insettings.tex` definiert. Sie kann anschließend auch Mandant\*innenspezifisch überschrieben werden.

## A.8. Seitenstil/Basislayout

Das Seitenlayout wird über `sclayer-scrpage` bestimmt. Die ausführliche Dokumentation findet sich in [1]. Es existieren in der Datei `insettings.tex` einige Hinweise zu den Anpassungen. Die Basiskonfiguration ist ebenfalls dort eingetragen.

Neben den in Abschnitt 3 beschriebenen Optionen zum Abschalten der Fußzeile kann der Inhalt der Fußzeile über die `sclayer-scrpage` Makros, wie

```
\cfoot[<Inhalt auf der ersten Briefseite>]{<Inhalt auf folgenden  
↪ Briefseiten>}
```

geändert werden.

Die Kopfzeile unterscheidet sich von Dokumententyp zu Dokumententyp leicht, da diese über Datenbankvariablen befüllt wird. Hierfür wird das Makro `\ourhead` in der `insettings.tex` definiert.

```
\ourhead  {\i{<Bezeichner>}}{\i{<Eintrag>}}{\i{<Titel>}}{\i{<Nummer>}}{\i{<Datum>}}
```

Diese Definition kann ebenfalls über die `insettings.tex` angepasst oder auch nachträglich überschrieben werden:

```

\newcommand{\ourhead}[5] {%
  \thead{%
    \makebox[\textwidth]{%
      \Ifstr{#1}{}{}{#1: #2 \hspace{0.7cm}}%
      #3%
      \Ifstr{#4}{}{}{~\nr: #4}%
      \Ifstr{#5}{}{}{\vom ~ #5}%
      \hspace{0.7cm} - \seite ~ \thepage/\letterlastpage  --%
    }%
  }%
}

```

In der Standard-Einstellung sieht eine Kopfzeile mit obigen Aufruf dann folgendermaßen aus:

```
arg1: arg2    arg3 Nr.: arg4vom arg5    - Seite 9/50 -
```

Erzeugt mit dem Aufruf

```
\ourhead{arg1}{arg2}{arg3}{arg4}{arg5}
```

## A.9. Absenderergänzung

Die Absenderergänzung wird über die Variable `location` in der `kiviletter.sty` folgendermaßen belegt:

```

\setkomavar{location}{
  \Ifkomavareempty{transaction}{}{
    \usekomafont{transaction}
    \usekomavar{transaction}
  }
}
\par
\medskip
\parbox{\useplength{locwidth}}{
  \locationentry{date}
  \locationentry{myref}
  \locationentry{customer}
  \locationentry{yourref}
  \locationentry{delivery}
  \locationentry{quote}
  \locationentry{orderID}
  \locationentry{projectID}
  \locationentry{taxpoint}
  \locationentry[\ansprechpartner]{fromname}
  \locationentry{fromphone}
  \locationentry*{fromemail}
}
}

```

Um die Reihenfolge der Variablenausgabe zu verändern, kann diese Definition als Basis in die `insettings.tex` oder `ident.tex` (Falls nur für eine Firma) kopiert und dort entsprechend modifiziert werden.

Das Vorgehen geht für alle vorbelegten Variablen analog.

## A.10. Allgemeine T<sub>E</sub>Xnische Hinweise

### A.10.1. Änderung der Basisschriftart

L<sup>A</sup>T<sub>E</sub>X kann grundsätzlich beliebige Schriftarten verwenden. Hierfür sollte allerdings immer darauf geachtet werden, dass die Lizenz der Schriftart das Einbetten von Glyphen erlaubt. Dies liegt in der Verantwortung der Anwender\*innen. Darüber hinaus ist wichtig, welches Kompilierungsprogramm verwendet werden muss. Um TrueType oder OpenType Schriftarten zu nutzen sollte `lualatex` verwendet werden. Bei Typ1 Schriftarten, die speziell für L<sup>A</sup>T<sub>E</sub>X installiert wurden, ist pdfLaTeX möglich. Da man heutzutage nur noch moderne Schriftformate verwenden sollte, wird hier lediglich die Variante für `lualatex` aufgelistet.

Die Konfiguration läuft hierbei über das `fontspec` Paket (Doku siehe [2]). Dann hängt es davon ab, ob die Basisschriftart eine Serifenschriftart ist oder nicht. In jedem Fall wird die Änderung entweder in der `insettings.tex`, sofern sie für alle Mandant\*innen gelten soll oder in der Mandant\*innenspezifischen Konfigurationsdatei gsesetzt.

#### Änderung, falls es ein Schriftpaket gibt

Wenn möglich sollte die Schriftart über ein entsprechendes Konfigurationspaket gesetzt werden. Ob ein solches existiert kann man sehr leicht über eine Suche nach dem Namen unter [ctan.org](http://ctan.org) herausfinden.

#### Änderung der Basisschriftart auf eine Schriftart mit Serifen

```
\setmainfont{<Name der Schriftart, z.B. SourceSerifPro>}
```

#### Änderung der Basisschriftart auf eine Schriftart ohne Serifen

```
\setsansfont{<Name der Schriftart, z.B. SourceSansPro>}
```

```
\renewcommand*{\familydefault}{\sfdefault}
```

### A.10.2. Unterscheidungen durch String-Vergleich

```
\ifstr{\lxmedia}{printer}{Falls gedruckt werden soll} {sonst}
```

## Literatur

- [1] Markus Kohm und Jens-Uwe Morawski. *KOMA-Script. Die Anleitung*. 12. Okt. 2022. URL: <http://mirrors.ctan.org/macros/latex/contrib/koma-script/doc/scrguide.pdf> (besucht am 01.01.2023).
- [2] Will Robertson. *The fontspec package. Font selection for X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and Lua<sub>l</sub>L<sup>A</sup>T<sub>E</sub>X*. Mit Erläut. von Khaled Hosny, Philipp Gesang, Joseph Wright u. a. Version v2.6g. 9. Nov. 2017. eprint: <http://wspr.io/fontspec/>. URL: <http://mirrors.ctan.org/macros/latex/contrib/fontspec/fontspec.pdf>.