

SCHULE

LaTeX-Klassen und Pakete für den Einsatz im Bereich der Schule

0.10.0 2024-04-01

Paketdokumentation

Johannes PIEPER Johannes KUHAUPT Ludger HUMBERT
André HILBIG Adrian SALAMON Daniel SPITTANK

<https://gitlab.com/gi-fg-ibnw/schule>

schulepaket@johpie.de

Diese Zusammenstellung wird entwickelt, um Pakete und damit Befehle bereit zu stellen, die für den Textsatz von Dokumenten zur Unterrichtsvorbereitung für den (Informatik)Unterricht nützlich sind. Zur Zeit liegt der Schwerpunkt auf dem Informatikunterricht, eine Ergänzung für den Physikunterricht wird nach und nach eingearbeitet. Weitere Ergänzungen für andere Fächer werden gerne entgegen genommen.

Diese Sammlung umfasst Pakete und Klassen zum Setzen von speziellen Dokumenten für Klausuren, Lernzielkontrollen, Unterrichtsbesuche, Arbeits-, Informations- und Lösungsblättern, sowie speziellen Elementen, wie Struktogramme, Syntax-, Sequenz-, Objekt- und Klassendiagramme.

Ein besonderer Dank geht an Martin Weise für seine Hilfe bei der Übersetzung der Readme-Dateien und Zusammenfassung auf CTAN* ins Englische.

Inhaltsverzeichnis

I. Allgemeines	5	1.4. Arten der Nutzung	8
1. Allgemeines zum Paket	5	1.4.1. Nutzung für Dokumente	8
1.1. Wichtiger Hinweis zur neuen Version	5	1.4.2. Eingebettete Nutzung	8
1.2. Manuelle Installation	5	1.4.3. Nutzung über die Dokumentenklassen .	8
1.2.1. Voraussetzungen	6	1.5. Kompilieren der Dokumente .	8
1.3. Begriffsklärungen	7		

II. Module	10	6.2. Kopf- Fußzeilen	23
2. Nutzung der Module	10	6.2.1. Paketooptionen	23
2.1. Standardmodule	10	6.2.2. Befehle	23
2.2. Laden weiterer Module	10	6.3. Seitenzahlen	23
3. Aufgaben	10	6.3.1. Paketooptionen	23
3.1. Aufgaben	10	6.3.2. Befehle	24
3.1.1. Befehle	10	6.4. Strukturelemente	24
3.1.2. Umgebungen	11	6.4.1. Umgebungen	24
3.1.3. Aufgabentemplates	11	6.5. Wörtliche Rede, Zitate und	
3.2. Teilaufgaben	12	Anführungszeichen	25
3.2.1. Befehle	12	6.5.1. Paketooptionen	25
3.2.2. Umgebung	12	6.5.2. Befehle	25
3.3. Lösungen	13	7. Formulare	26
3.3.1. Paketooptionen	13	7.1. Befehle	26
3.3.2. Umgebungen	13	8. Kuerzel	26
3.4. Lückentexte	14	8.1. Paketooptionen	27
3.4.1. Befehle	14	8.2. Befehle	27
3.5. Multiple-Choice	14	9. Lizenzen	28
3.5.1. Befehle	14	9.1. Paketooptionen	28
3.6. Umgebungen	15	9.2. Befehle	28
3.7. Bearbeitungshinweise	15	10. Metadaten	29
3.7.1. Umgebungen	15	10.1. Paketooptionen	29
3.7.2. Befehle	15	10.2. Befehle	29
4. Aufgabenpool	16	11. Papiertypen	29
4.1. Befehle	16	11.1. Befehle	30
4.2. Umgebungen	17	12. Symbole	31
4.3. Optionen an Aufgaben	17	12.1. Befehle	31
4.3.1. Befehle	17	13. Texte	34
4.4. Anhänge an Aufgaben	17	13.1. Befehle	34
4.4.1. Umgebungen	18	13.2. Umgebungen	34
4.4.2. Befehle	18	III. Fächer	37
5. Bewertung	18	14. Nutzung der Fachmodule	37
5.1. Paketooptionen	18	15. Informatik	37
5.1.1. Umgebungen	20	15.1. Objektdiagramme	37
5.1.2. Befehle	20		
6. Format	20		
6.1. Formatierungen	20		
6.1.1. Paketooptionen	20		
6.1.2. Befehle	21		

15.2. Sequenzdiagramme	39	28. Das alte Schule-Paket	53
15.3. Struktogramme	40		
15.4. Syntaxdiagramme	41	VI. Häufig gestellte Fragen	54
15.5. Flussdiagramme	41		
16. Physik	42	29. Formatierung	54
16.1. Konstanten	43	29.1. Kann ich ein anderes Pa- pierformat als A4 verwenden?	54
16.2. Schaltpläne	43	29.2. Kann ich die Seitenränder festlegen?	54
17. Geschichte	44	29.3. Ist es möglich, in den er- stellten Materialien Schreib- schriften zu verwenden?	54
17.1. Befehle	45		
IV. Dokumenttypen	46		
18. Arbeitsblatt	46	VII. Beispiele	56
19. Klausur	46	30. Minimalbeispiele	56
19.1. Paketoptionen	46	30.1. Nutzung von Modulen	58
20. Leitprogramm	47	31. Praxisbeispiele	59
20.1. Paketoptionen	47	31.1. Geschichte	59
20.2. Befehle	47	31.2. Informatik	60
20.3. Umgebungen	48	31.3. Physik	60
21. Lernzielkontrolle	48		
22. Übungsblatt	48	VIII. Entwicklungsdokumentation	61
23. Unterrichtsbesuch	48	32. Lizenzen	61
23.1. Befehle für Angaben zum Unterrichtsbesuch	48	33. Richtlinien	61
24. Folie	49	34. Modularität	62
25. Beurteilung	49	34.1. Erläuterungen zum Modul- system	62
25.1. Paketoptionen	50	34.2. Aufbau eines Moduls	62
25.2. Befehle	50	34.2.1. Beispiel	62
25.3. Umgebungen	50	34.3. Aufbau eines Fachmoduls . .	64
		34.4. Aufbau eines Dokumenttyps .	64
V. Zusatzpakete	52	35. Funktionen für Entwickler	64
26. Nutzung der Zusatzpakete	52	35.1. Fehlerbehandlung und De- bugging	64
27. Schaltungen mit Relais	52	35.1.1. Paketoptionen	64
		35.1.2. Befehle	65

35.2. Interne Makros	65	37. ToDo	67
35.2.1. Befehle	65	37.1. Must-have	67
		37.2. Nice-to-have	67
36. Changelog	66	38. Index	69

Teil I.

Allgemeines

1. Allgemeines zum Paket

1.1. Wichtiger Hinweis zur neuen Version

Das Schule-Paket wurde vollständig überarbeitet. Diese Version enthält grundlegende, strukturelle Veränderungen. So wird unter anderem die Vielzahl an Dokumentenklassen stark reduziert und die Konfiguration erfolgt nun über Paketoptionen.

Dies führt zu großen Veränderungen der Schnittstelle. Die neue Version ist damit **nicht kompatibel** zu allen vorhergehenden Versionen. Es besteht allerdings ein **Kompatibilitätsmodus**, der automatisch für alle alten Dokumentenklassen aktiv ist. Alte Dokumente lassen sich somit weiterhin setzen, die Schnittstelle wird aber nicht weiterentwickelt. Bestehende Fehler werden in der alten Version nicht behoben.

Diese Änderungen ermöglichen die Lösung einiger bestehender Probleme (u. a. Quelltexte in Aufgaben und Lösungen). Zusätzlich wurde die Nutzung des Pakets vereinheitlicht und die Nutzung in anderen Dokumentenklassen ermöglicht, sodass etwa die Aufgabenumgebungen auch in Beamer-Präsentationen übernommen werden können. Klausuren unterstützen nun die automatische Erzeugung von Erwartungshorizonten.

Eine weitere große Veränderung ist die Ausgliederung der ausbildungsrelevanten Teile (Unterrichtsbesuche, Stundenverläufe etc.) des Pakets. In der Vergangenheit hat sich gezeigt, dass sich die Anforderungen der verschiedenen, an der Lehrerbildung beteiligten Stellen stark voneinander unterscheiden. Daher werden die entsprechenden Funktionen des Pakets ausgegliedert, sodass sie einfach in eigenen Dokumenten genutzt werden können. Die bestehenden Vorlagen werden als eigenständige Klassen mitgeliefert.

1.2. Manuelle Installation

Um die Pakete und Klassen nutzen zu können, gibt es drei Varianten. In der folgenden Beschreibung dieser Möglichkeiten wird von einer standardisierten \LaTeX -Installation ausgegangen – weitere Hinweise können der Dokumentation der jeweiligen \TeX -Distribution entnommen werden:

Global Für die globale/systemweite Installation der Pakete und Klassen müssen diese in das globale \LaTeX -Verzeichnis der \TeX -Installation kopiert werden: unter Linux in der Regel `/usr/share/texmf/tex/latex/`. In diesem kann ein weiteres Verzeichnis wie z. B. `schule` angelegt werden, in das alle Dateien des Schulepakets kopiert werden.

Damit die Quellen anschließend dem System bekannt sind, muss der Cache von \LaTeX neu aufgebaut werden. Bei den meisten Linux-Installationen geschieht dieses durch den Aufruf von `texhash`.

1. Allgemeines zum Paket

Benutzer Damit ein Nutzer auf die Quellen zugreifen kann, müssen diese im Benutzerverzeichnis (Home directory) abgelegt werden. Dies geschieht durch das Kopieren der Pakete und Klassen in das Verzeichnis `texmf/tex/latex/` im Benutzerverzeichnis, das ggf. erst angelegt werden muss. Auch hier sollte – wie bei der globalen Installation – ein eigenes Unterverzeichnis angelegt werden.

Lokal Um die Klassen und Pakete ohne weitere Installation nutzen zu können, ist es darüber hinaus möglich, die benötigten Dateien in das Verzeichnis zu kopieren, in dem die Datei liegt, die übersetzt werden soll. Dies ist jedoch aufgrund des Umfangs des Schulepakets weniger empfehlenswert.

1.2.1. Voraussetzungen

Ein Grund für die Nutzung des Schule-Pakets und der damit verbundenen speziellen Klassen und Pakete liegt darin, viele der häufig benötigten Pakete zusammen zu fassen. Daher müssen diese für die Benutzung vorhanden sein. Die meisten sind Standardpakete, die mit jeder normalen Installation mitgeliefert sind. Es folgt eine Aufstellung der Voraussetzungen für das Paket `schule` und die vorhandenen Module. Mit einem Stern (*) markierte Pakete sind im Paket `schule` bereits enthalten:

- `amsmath`
- `babel`
- `environ`
- `fontenc`
- `forarray`
- `graphicx`
- `hyperref`
- `ifthen`
- `inputenc`
- `pgfopts`
- `schulealt` *
- `tikz`
- `xcolor`
- `xparse`
- `xstring`
- `zref-totpages`

Folgende Pakete werden zusätzlich für das Fach »Informatik« benötigt:

- `listings`
- `pgf-umlcd`
- `pgf-umlsd`
- `relaycircuit` *
- `struktex`
- `syntaxdi`

Folgende TikZ-Bibliotheken werden für das Fach »Informatik« benötigt:

- `er`

Folgende Pakete werden zusätzlich für das Fach »Physik« benötigt:

- `circuittikz`
- `units`
- `mhchem`

Folgende Pakete werden zusätzlich für das Fach »Geschichte« benötigt:

- `uni-wtal-ger`
- `marginnote`

Folgende Pakete werden zusätzlich für das Modul »Aufgaben« benötigt:

1. Allgemeines zum Paket

- xsim
- utfsym *

Folgende Pakete werden zusätzlich für das Modul »Format« benötigt:

- amssymb
- array
- colortbl
- csquotes
- ctable
- enumitem
- eurosym
- graphicx
- longtable
- multicol
- multirow
- setspace
- tikz
- ulem
- utfsym *
- xspace

Folgende Pakete werden zusätzlich für das Modul »Symbole« benötigt:

- utfsym *

Folgende Pakete werden zusätzlich für das Modul »Texte« benötigt:

- lineno
- multicol

Folgende Pakete werden zusätzlich für das Modul »Texte« benötigt:

- standalone

Folgende TikZ-Bibliotheken werden für das Zusatzpaket `relaxcircuit` benötigt:

- arrows
- scopes
- shadows
- shapes.misc

1.3. Begriffsklärungen

Zusatzpaket Das Paket `schule` liefert einige \LaTeX -Pakete mit, die für das Paket entwickelt wurden, aber von diesem unabhängig nutzbar sind.

Diese Pakete werden im Folgenden als Zusatzpaket bezeichnet.

Modul Im Gegensatz zu einem Zusatzpaket ist ein Modul enger mit dem Hauptpaket verzahnt. Es lässt sich nicht unabhängig von diesem nutzen.

Module bestehen aus einer oder mehreren \LaTeX -Quelldateien, die in das Paket eingebunden werden.

Siehe auch die Beschreibung in der Entwicklungsdokumentation im Abschnitt 34.2, S. 62.

Fachmodul Ein Fachmodul ist ähnlich aufgebaut wie ein normales Modul für das Schulpaket, wird allerdings für fachspezifische Erweiterungen genutzt und erfüllt somit einen anderen Zweck.

Siehe auch die Beschreibung in der Entwicklungsdokumentation im Abschnitt 34.3, S. 64.

Dokumenttyp Ein Dokumenttyp ist ähnlich aufgebaut wie ein normales Modul für das Schulpaket, wird allerdings für typspezifische Erweiterungen genutzt und erfüllt somit einen anderen Zweck.

Siehe auch die Beschreibung in der Entwicklungsdokumentation im Abschnitt 34.4, S. 64.

1.4. Arten der Nutzung

1.4.1. Nutzung für Dokumente

Wenn zumindest ein `typ` in den Paketoptionen angegeben wird, werden viele Module und mit diesen auch viele externe Pakete geladen und konfiguriert, von denen einige auch die grundlegende Struktur der zu setzenden Dokumente verändern.

Außerdem werden Entscheidungen für das Aussehen der Dokumente getroffen. Man hat hier noch viele Freiheiten, ist jedoch auf die grundlegenden Vorgaben des Schule-Pakets festgelegt.

Dies kann auch zu Inkompatibilitäten mit bestimmten Dokumentenklassen oder externen Paketen führen, z. B. könnten Option-Clashes auftreten.

1.4.2. Eingebettete Nutzung

Es trat immer wieder der Wunsch auf, dass Funktionen aus dem Schulpaket auch in anderen Dokumenten oder gar in Dokumentenklassen oder anderen Paketen nutzen zu können.

Das war aus den oben genannten Gründen schwierig. Inzwischen ist dies möglich, in dem man beim Laden des Pakets die Option `typ = {ohne}` angibt.

Damit wird das Paket in einen »minimalinvasiven« Modus geschaltet, der nur die nötigsten Module lädt und so wenig Vorgaben macht wie möglich.

Weitere Module können dann natürlich geladen werden.

1.4.3. Nutzung über die Dokumentenklassen

Die Nutzungsvariante mit den wenigsten Freiheiten ist die über eine der Dokumentenklassen. Anpassungen sind hier nur sehr eingeschränkt möglich und es werden sehr viele Vorgaben gemacht. Sie ist allerdings gleichzeitig die Variante, bei der man am wenigsten konfigurieren und eigene Einstellungen vornehmen muss. Siehe auch 3.3.1.

1.5. Kompilieren der Dokumente

Das Schulpaket ist für die Nutzung von `pdflatex` optimiert und wurde nur damit getestet.

Aufgrund des komplexen Aufbaus kann es besonders bei der Nutzung des Moduls `Aufgaben` notwendig sein, mindestens zwei Läufe von `pdflatex` durchzuführen. Dies liegt daran, dass eine Menge Zwischendateien mit Punkten und anderen Metadaten zu den Aufgaben erstellt werden müssen und dann damit Berechnungen durchgeführt werden.

1. Allgemeines zum Paket

Es kann deshalb passieren, dass nach der Änderung der Anzahl der Aufgaben der erste Durchlauf mit sehr vielen Fehlern fehlschlägt.

Teil II.

Module

2. Nutzung der Module

2.1. Standardmodule

Standardmäßig wird davon ausgegangen, dass ein Dokument mit Schulkontext gesetzt werden soll (Arbeitsblatt, Klausur, etc). Dann lädt das Schule-Paket die Module **Metadaten**, **Format** und **Aufgaben**.

Wird das Paket eingebettet verwendet, also mit der Paketoption `typ = {ohne}` geladen, lädt das Schule-Paket nur die Module **Metadaten** und **Format**.

Wird ein nicht definierter Typ angegeben, wird ein Arbeitsblatt gesetzt und der angegebene Typ wird als Bezeichner verwendet.

2.2. Laden weiterer Module

`module = {\Modul1,Modul2,...}` (zunächst leer)

Weitere Module können geladen werden, indem sie der Paketoption `module` als komma-separierte Liste übergeben werden.

3. Aufgaben

Das Modul **Aufgaben** ist das umfangreichste Modul des Schule-Pakets. Es umfasst alles, was zum Setzen von verschiedenen Arbeitsblättern, Klausuren, Klassenarbeiten, Lernzielkontrollen usw. notwendig ist.

Im Kern baut das Modul auf dem Paket `xsim` auf, sodass alle Funktionen dieses Pakets nutzbar sind.

Die vom Schulepaket gemachten Ergänzungen sind voll kompatibel zu `xsim`, so werden die Hinweise etwa in den Eigenschaften der Aufgaben gespeichert.

3.1. Aufgaben

3.1.1. Befehle

`\setzeSymbol{\Symbol}`

kann nur innerhalb der Aufgabenumgebung genutzt werden und stellt der jeweiligen Aufgabe ein Symbol voran. Dies kann etwa genutzt werden, um die Arbeitsform oder bestimmte Aufgabentypen zu kennzeichnen.

Eine Kombination mit dem Modul **Symbole** bietet sich an. So könnte etwa zur Kennzeichnung von Höraufgaben `\setzeSymbol{\symOhr}` genutzt werden.

Alternativ lässt sich das Symbol auch als Eigenschaft der Aufgabe direkt setzen. Dieses erfolgt z. B. durch `\begin{aufgabe}[symbol=\symOhr]`.

3. Aufgaben

`\punkteAufgabe`

liefert die Punkte der aktuellen Aufgabe inkl. der Bezeichnung.

`\punkteTotal`

liefert die Gesamtpunktzahl aller Aufgaben inkl. der Bezeichnung.

`\punktuebersicht*` [*⟨Darstellungsart⟩*]

Voreinstellung: **kurz**

setzt eine Übersichtstabelle über die in allen Aufgaben erreichbaren Punkte und Zusatzpunkte, sowie einer Leerzeile für die erreichten Punkte. Als optionalen Parameter kann zwischen verschiedenen Darstellungen gewählt werden. Alternativ zur Standardoptionen ist **default**. Ist die Übersicht als letztes auf den Aufgabenseiten, so kann der optionale Stern gesetzt werden. Dann wird diese Seite als letzte mit Inhalt angenommen.

3.1.2. Umgebungen

`\begin{aufgabe*}` [*⟨Optionen⟩*]

setzt eine Aufgabe. Alle Aufgaben werden automatisch durchnummeriert. Wird der optionale Stern angegeben, wird die Aufgabe als Zusatzaufgabe gesetzt.

Bei den optionales Argument können alle von `xsim` bereitgestellten Optionen angegeben werden. Dazu gehören unter anderem folgende:

`points = ⟨Punkte⟩`

legt die Punkte der Aufgabe fest.

`bonus-points = ⟨Zusatzaufgabe⟩`

legt die Punkte der Aufgabe fest.

`subtitle = {⟨Titel⟩}`

setzt den Title der Aufgabe.

3.1.3. Aufgabentemplates

Die Darstellung der Aufgaben erfolgt auf der Grundlage verschiedener Templates. Das Paket `schule` liefert dabei folgende Templates mit, die in darunter dargestellt sind.

- `schule-binnen`
- `schule-default`
- `schule-keinenummer`
- `schule-keinepunkte`
- `schule-keintitel`
- `schule-randpunkte`
- `schule-tcolorbox`

3. Aufgaben

1.) Diese Aufgabe ist im Stil: Binnen
(4 Punkte)

Aufgabe 2 *Default* (4 (+2) Punkte)
Diese Aufgabe ist im Stil: Default

Aufgabe *Keine Nummer* (4 (+2) Punkte)
Diese Aufgabe ist im Stil: Keine Nummer

Aufgabe 4 *Keine Punkte*
Diese Aufgabe ist im Stil: Keine Punkte

Diese Aufgabe ist im Stil: Kein Titel —/ 4 (+2) P.

Aufgabe 6 *Randpunkte*
Diese Aufgabe ist im Stil: Randpunkte —/ 4 (+2) P.

Aufgabe 7 *TColorBox*4 (+2) Punkte

Diese Aufgabe ist im Stil: TColorbox

`\setzeAufgabentemplate{<Templatename>}`

setzt das Template mit dem die folgenden Aufgaben dargestellt werden.

`\setzeLoesungentemplate{<Templatename>}`

setzt das Template mit dem die folgenden Lösungen dargestellt werden. Diese sind identisch mit denen der Aufgaben.

3.2. Teilaufgaben

3.2.1. Befehle

`\teilaufgabe[<Punkte>]`

leitet innerhalb einer `teilaufgaben`-Umgebung eine Teilaufgabe ein. Teilaufgaben werden mit den Kleinbuchstaben von *a* bis *z* gekennzeichnet.

Über den optionalen Parameter kann eine Punktzahl angegeben werden.

`\teilaufgabeOhneLoesung`

Dient als Platzhalter bei Teilaufgaben, bei den keine Lösung angegeben wird. Die entsprechende Nummer wird bei den Lösungen nicht aufgeführt und die folgende Teilaufgaben bekommt den nächsten Buchstaben, so dass es übereinstimmt mit der Aufgabenstellung.

3.2.2. Umgebung

`\begin{teilaufgaben}`

bietet die Möglichkeit, eine Aufgabe in verschiedene Teilaufgaben zu unterteilen.

```
1 \begin{aufgabe}
2   Inhalt...
3   \begin{teilaufgaben}
```

3. Aufgaben

```
4     \teilaufgabe Erstens.
5     \teilaufgabe[5] Zweitens.
6     \end{teilaufgaben}
7 \end{aufgabe}
```

☞ **Hinweis:** Teilaufgaben können auch in einer `loesung-` und `bearbeitungshinweis-` Umgebung verwendet werden!

3.3. Lösungen

3.3.1. Paketoptionen

`loesungen` = `folgend|keine|seite` Voreinstellung: `keine`
legt fest, ob die Lösungen direkt hinter die Aufgaben, als eigenständige Lösungsseite oder gar nicht gesetzt werden.

⚠ **Achtung:** Die Option `loesungen = {seite}` ist nur für eigenständige Dokumente, z. B. mit der Dokumentenklasse `scartcl` gedacht. Sie greift tief in den Übersetzungsprozess ein und ist geeignet Fehler im Zusammenspiel mit anderen Paketen zu provozieren.

`\printsolutions`

Wenn keine der Standardoptionen genutzt wird, kann der Befehl zur Ausgabe der Lösungen aus dem `xsim`-Paket genutzt werden.

3.3.2. Umgebungen

`\begin{loesung*}`

wird innerhalb oder direkt hinter einer `aufgabe` verwendet, um eine Lösung dazu anzugeben. Die Inhalte dieser Umgebung werden standardmäßig nicht gesetzt, sondern durch die entsprechende Konfiguration von `loesungen` an der entsprechenden Stelle gesetzt. Wichtig ist, dass bei Zusatzaufgaben auch bei der Lösung der Stern gesetzt werden muss.

```
1 \begin{aufgabe}
2   Inhalt...
3   \begin{teilaufgaben}
4     \teilaufgabe Erstens.
5     \teilaufgabe[5] Zweitens.
6   \end{teilaufgaben}
7   \begin{loesung}
8     \begin{teilaufgaben}
9       \teilaufgabe Erste Lösung.
10      \teilaufgabe Zweite Lösung.
11    \end{teilaufgaben}
12  \end{loesung}
13 \end{aufgabe}
```

3.4. Lückentexte

3.4.1. Befehle

`\luecke`[*(Optionen für blank)*]{*(Länge)*}

Setzt eine Lücke mit der angegebenen Länge. Der Befehl nutzt dazu den `\blank`-Befehl aus `xsim`. Mit dem optionalen Parameter können zusätzliche Optionen an diesen weitergereicht werden, z. B. kann mit `style = line|wave|dline|dotted|dashed` der Stil der Unterstreichung festgelegt werden.

`\textluecke`[*(Optionen für blank)*]{*(Text)*}

Setzt eine Lücke für den angegebenen Text, die Länge wird durch den angegebenen Text vorgegeben. Standardmäßig wird als Korrekturfaktor für das handschriftliche Ausfüllen 2 genutzt.

Der Befehl nutzt dazu den `\blank`-Befehl aus `xsim`. Mit dem optionalen Parameter können zusätzliche Optionen an diesen weitergereicht werden, z. B. kann mit `style=\dashuline{#1}` eine unterstrichelte Linie gesetzt werden. Mit `scale = 3` ließe sich der Korrekturfaktor auf 3 anpassen. Wird als Option `nichts` angegeben, so wird die Lücke ohne Inhalt und Weite eingesetzt.

Innerhalb von Lösungsumgebungen wird der Text in die Lücke eingesetzt.

3.5. Multiple-Choice

Zwar ist es über das `Format`-Modul möglich, einzelne Kästchen zum Ankreuzen zu setzen. In der Regel sollten allerdings echte Multiple-Choice-Aufgaben vorgezogen werden, da diese besser formatiert werden können und sich auch direkt Lösungen angeben lassen.

⚠ Achtung: Die direkte Nutzung der Lösung funktioniert nur dann, wenn innerhalb einer Aufgabe nur eine einzige Multiple-Choice-Umgebung genutzt wird.

3.5.1. Befehle

`\choice`[*(richtig)*]

Innerhalb einer `mcumgebung` können mit `\choice` die einzelnen Wahlmöglichkeiten angegeben werden.

Falls im optionalen Parameter `\mcrichtig` steht, wird die Wahlmöglichkeit als richtig markiert und in Lösungsumgebungen entsprechend gesetzt.

`\mcrichtig`

markiert innerhalb einer `mcumgebung` eine Wahlmöglichkeit als richtig.

`\mcloesung`

kann in einer Lösung angegeben werden, damit in dieser die Multiple-Choice-Aufgabe mit den korrekten Lösungen aufgeführt wird.

3.6. Umgebungen

`\begin{mcumgebung}(\langle Spaltenzahl \rangle)`

ermöglicht es Multiple-Choice-Aufgaben zu setzen. Wird die Spaltenzahl nicht explizit angegeben, so ist 3 als Default-Wert gesetzt.

```

1 \begin{aufgabe}
2   \begin{mcumgebung}(4)
3     \choice[\mcrichtig] Erstens
4     \choice Zweitens
5     \choice[\mcrichtig] Drittens
6   \end{mcumgebung}
7 \end{aufgabe}
8 \begin{loesung}
9   \mcloesung
10 \end{loesung}

```

3.7. Bearbeitungshinweise

Die Bearbeitungshinweise sind dazu gedacht, dass man den Lernenden Tipps zu den Aufgaben mitgibt. Dieses ist z. B. bei der Bearbeitung von Leitprogrammen (siehe 20) der Fall. Dabei ist es angedacht, diese nicht direkt bei den Aufgaben stehen zu haben, sondern an einer anderen Stelle, damit sie nur bei Bedarf genutzt werden.

3.7.1. Umgebungen

`\begin{bearbeitungshinweis}`

erlaubt es, zu einzelnen Aufgaben Hinweise anzugeben. Der Hinweis kann dabei fast beliebigen \LaTeX -Code enthalten. Verbatim-Elemente, wie z. B. die Verwendung von Quellcode machen an Probleme. Es kann aber `\lstinputlisting` genutzt werden.

3.7.2. Befehle

`\bearbeitungshinweisZuAufgabe[\langle Aufgabentyp \rangle]{\langle AufgabenId \rangle}` Voreinstellung: `aufgabe`

Setzt die Bearbeitungshinweise für die angegebene Aufgabe. Die ID ist dabei fortlaufend über alle Aufgabentypen. Der optionale Parameter erlaubt es auch für andere Aufgabentypen wie der Zusatzaufgabe mit `aufgabe*` den Hinweis direkt auszugeben. Wird als AufgabenId nichts angegeben, so wird die aktuelle Aufgabe genommen.

`\bearbeitungshinweisliste`

Setzt die Bearbeitungshinweise zu allen Aufgaben als Liste.

4. Aufgabenpool

Eingeführt in
Version 0.8.4

Die Idee für das Modul `Aufgabenpool` bestand darin, dass Aufgaben einfach bei verschiedenen Klassenarbeiten oder Klausuren wieder genutzt werden können. Diese Aufgaben dürfen auch in verschiedenen Verzeichnissen liegen und sich einzeln setzen lassen. Besondere Berücksichtigung benötigen dabei eingebundene Dateien, wie z. B. Bilder oder Programmdateien, die relativ zur Aufgabendatei, an einem anderen Ort liegen als die Quelldatei für die Klausur. Für diesen Fall wird auf ein *basedir* gesetzt, dass für jede eingebundene Aufgabe entsprechend gesetzt wird. Mit der Nutzung über `\getBasedir` lässt sich jede Datei relativ zur Aufgabendatei einbinden. Dieses gilt auch innerhalb der *loesung*-Umgebung.

Zur Verwendung einer Aufgabe aus dem Aufgabenpool lassen sich auch Hinweise in der Aufgabe ablegen. Diese werden dann nicht mit ausgegeben, wenn die Aufgabe in einer Klassenarbeit bzw. Klausur mit eingebunden wird.

4.1. Befehle

`\aufgabeninput` [*optionen*] {*Verzeichnis*} {*Datei*}

Bindet die Datei mit der Aufgabe ein, die im angegebenen Verzeichnis steht. Die Angabe des Verzeichnis wird als Grundlage für das *basedir* für diese Aufgabe genutzt. Als optionalen Parameter lassen sich durch Komma getrennte Optionen angeben. Dadurch lässt sich in der eigentlichen Aufgabe steuern, ob bestimmte Elemente gesetzt werden sollen oder nicht. Näheres unter 4.3.

`\getBasedir`

Liefert das aktuelle *basedir* zurück, um es in einer Aufgabe als Pfadergänzung bei der Einbindung von Dateien zu nutzen.

`\setBasedir`

Eröffnet die Möglichkeit, dass *basedir* als Eigenschaft bei anderen Aufgabentypen, die auf Grundlage des *xsim*-Pakets definiert sind, zu setzen. Bei Aufgabe ist dieses bereits gesetzt durch:

```

1 \xsimsetup{
2   aufgabe/begin-hook = \setBasedir,
3 }
```

`\inputOnce` {*Datei*}

Eingeführt in
Version 0.9.0

Die angegebene Datei wird eingebunden. Dieses geschieht innerhalb eines Dokumentes aber genau einmal, auch wenn dieser Aufruf in mehreren Aufgaben angegeben ist. So kann z. B. auf einen gemeinsamen Text zurückgegriffen werden. Das *Basedir* wird im Pfad zur Datei mit berücksichtigt, so dass `\getBasedir` nicht in der Pfadangabe mit genutzt werden darf.

4.2. Umgebungen

`\begin{aufgabenpoolHinweis}`

Eingeführt in
Version 0.9.0

Innerhalb dieser Umgebung können Hinweise zur Aufgabe gemacht werden, die an den Steller der Arbeit bzw. Klausur gerichtet sind. Diese werden dann in der einzelnd stehenden Aufgabe auch angezeigt. Wird die Aufgabe in einer Klassenarbeit bzw. Klausur eingebunden, werden diese Hinweise nicht mit gesetzt.

4.3. Optionen an Aufgaben

Eingeführt in
Version 0.9.0

Zu jeder Aufgabe lassen sich einstellbare Optionen angeben. Mit diesen kann man regeln, ob bestimmte Teile im fertigen Dokument mit eingebunden werden sollen oder nicht. So lassen sich zum Beispiel bestimmte Teilaufgaben ausklammern und nicht bei jeder Nutzung der Aufgabe darstellen. Es gibt auch die Möglichkeit die Einbindung weiterer Dokumente so zu regeln, dass sie einmal in einer Klassenarbeit/Klausur gesetzt werden, auch wenn sie in mehreren Aufgaben aus dem Aufgabenpool vorkommen.

4.3.1. Befehle

`\ifAufgabenpoolOptionTF{<Optionname>}{<Wahr>}{<Falsch>}`

Dieses bietet die Möglichkeit anhand der gewählten Option bestimmte Teile anzeigen zu lassen oder nicht. Der Wahr-Teil wird ausgeführt, wenn die angegeben Option gewählt ist. Gleiches gilt für den Falsch-Teil, wenn die Option nicht gewählt ist. Zur Vereinfachung gibt es diesen Befehl auch nur mit dem Wahr- oder Falsch-Teil.

`\ifAufgabenpoolOptionT{<Optionname>}{<Wahr>}`

Siehe `\ifAufgabenpoolOptionTF`.

`\ifAufgabenpoolOptionF{<Optionname>}{<Falsch>}`

Siehe `\ifAufgabenpoolOptionTF`

`\setAufgabenpoolOptionen{<Optionennamen>}`

Dieses kann im Kopf der Aufgabendatei genutzt werden, um die Optionen zu setzen. So lassen sich die verschiedenen Ansichten bereits im Aufgabendokument anzeigen.

4.4. Anhänge an Aufgaben

Eingeführt in
Version 0.10.0

Bei der Nutzung des Aufgabenpools kann es vorkommen, dass Aufgaben auch Elemente haben, die im Anhang angegeben werden. Teilweise können auch mehrere Aufgaben den gleichen Anhang nutzen. Daher werden zwei verschiedene Möglichkeiten zum Einbinden von Anhängen bereitgestellt. Die Ausgabe des Anhangs wird dann unterdrückt, wenn der Aufruf des Anhangs sich innerhalb einer eingebundenen Aufgabendatei befindet. Für das Gesamtdokument gibt es mit `\showIncludeAppendix` einen extra Befehl, der dann für die Ausgabe sorgt.

5. Bewertung

4.4.1. Umgebungen

`\begin{onceAppendix}{\langle Name \rangle}`

Innerhalb der Umgebung kann Inhalt für den Anhang angegeben werden. Durch die Angabe des Namens wird dafür gesorgt, dass Anhang mit gleichem Namen nur einmal ausgegeben wird.

4.4.2. Befehle

`\inputOnceAppendix{\langle Datei \rangle}`

Eine Datei kann, mit `\getBasedir` für den Dateipfad, angegeben werden. Diese wird nur einmal als Anhang eingebunden in das Dokument.

`\showIncludeAppendix`

Der Befehl wird an einer Stelle verwendet, an der die Anhänge eingebunden werden sollen.

5. Bewertung

Das Modul `Bewertung` ergänzt das Modul `Aufgaben` um die Möglichkeiten eines Erwartungshorizonts und der Berechnung der Notenverteilung. Die Punkteangaben beim Erwartungshorizont werden auch als Punkte für die Aufgaben herangezogen und müssen so nicht doppelt angegeben werden.

⚠ Achtung: Soll in einem Dokument ein Erwartungshorizont gesetzt werden, müssen alle Aufgaben Erwartungen enthalten!

5.1. Paketoptionen

`erwartungshorizontAnzeigen`

hängt den Erwartungshorizont im gewählten Stil automatisch an das Dokument an, setzt vorher die Seitennummerierung und die Dokumentbezeichnung in der Kopfzeile zurück.

Unter dem Erwartungshorizont wird automatisch eine Notenverteilung gesetzt.

⚠ Achtung: Diese Option ist nur für eigenständige Dokumente, z. B. mit der Dokumentenklasse `scartcl` gedacht. Sie greift tief in den Übersetzungsprozess ein und ist geeignet Fehler im Zusammenspiel mit anderen Paketen zu provozieren.

`erwartungshorizontKeineSeiten`

entfernt die Seitennummerierung im Erwartungshorizont.

`erwartungshorizontStil = einzeltabellen|simpl|standard` Voreinstellung: `standard` legt den Stil des Erwartungshorizonts fest. Bisher gibt es drei verschiedene Stile:

`erwartungshorizontStil = einzeltabellen` setzt für jede Aufgabe eine eigene Überschrift und darunter eine Tabelle mit den einzelnen Erwartungen. Unter die Erwartungen aller Aufgaben wird mit `\punktuebersicht` eine Übersicht über die erreichten Punkte gesetzt.

5. Bewertung

Erwartungshorizont		max. Punkte	erreicht
Aufgabe 1			
Die Schülerin/der Schüler...			
a) hat den Text gelesen.		5	
b) nimmt begründet Stellung zum Text und berücksichtigt, dass es sich um einen Blindtext handelt.		10	
Gesamt		15	
Zusatzaufgabe 1			
Die Schülerin/der Schüler...			
a) gibt eine richtige Lösung zu Teilaufgabe a) an.		+5	
b) gibt eine richtige Lösung zu Teilaufgabe b) an.		+5	
Gesamt		+10	
Gesamt			
Aufgabe	1	1	Σ
Punkte	15	0	15
erreicht			

`erwartungshorizontStil = simpel` setzt einen Bewertungsbogen ohne Punkte mit drei Smiley-Feldern zum Ankreuzen. Die Notenverteilung wird hier ebenfalls nicht gesetzt.

Erwartungshorizont		max. Punkte	erreicht
Aufgabe 1			
a) Du hast den Text gelesen.			☺
b) Du hast alle Namen unterschlagen!			☹
Zusatzaufgabe 1			
a) Deine Lösung zu Teilaufgabe a) ist richtig.			☺
b) Deine Lösung zu Teilaufgabe b) ist richtig.			☹

`erwartungshorizontStil = standard` setzt einen klassischen Erwartungshorizont in einer zusammenhängenden Tabelle. Die Umgebung ist `longtable`, die Tabelle bricht also bei längeren Erwartungshorizonten auf die nächste Seite um.

Erwartungshorizont		max. Punkte	erreicht
Aufgabe 1			
Die Schülerin/der Schüler...			
a) hat den Text gelesen.		5	
b) nimmt begründet Stellung zum Text und berücksichtigt, dass es sich um einen Blindtext handelt.		10	
Gesamt		15	
Zusatzaufgabe 1			
Die Schülerin/der Schüler...			
a) gibt eine richtige Lösung zu Teilaufgabe a) an.		+5	
b) gibt eine richtige Lösung zu Teilaufgabe b) an.		+5	
Gesamt		15 +10	
Notenverteilung			
Note	≥ 95	Note	≥ 90
erreicht	15	erreicht	10
max. Punkte	15	erreicht	10
erreicht	15	erreicht	10

kmkPunkte

schaltet alle benotungsrelevanten Funktionen vom normalen Notensystem (*ungenügend* bis *sehr gut*) auf KMK-Notenpunkte (0 bis 15) um.

notenOhneTendenz

Gibt in der Notenübersicht nur die sechs Notenstufen ohne Tendenzen an. Diese Option ist nicht mit der Option `kmkPunkte` vereinbar.

Eingeführt in
Version 0.8.3

notenschema = {⟨15=.95,...⟩}

gibt ein Notenschema für die Berechnung der Notenverteilung an. Es muss eine Liste mit der Zuordnung von Notenpunkten zu Prozentwerten übergeben werden. Die Prozentwerte geben dabei jeweils die untere Grenze für die jeweilige Note an.

Das Standardnotenschema ist $15 = .95$, $14 = .9$, $13 = .85$, $12 = .8$, $11 = .75$, $10 = .7$, $9 = .65$, $8 = .6$, $7 = .55$, $6 = .5$, $5 = .45$, $4 = .39$, $3 = .33$, $2 = .27$, $1 = .2$

6. Format

5.1.1. Umgebungen

`\begin{erwartungen}`

Diese Umgebung wird nicht mehr für Erwartungen benötigt und gibt eine Warnung aus.

Veraltet
seit 0.10.0

5.1.2. Befehle

`\erwartung{⟨Erwartung⟩}{⟨Punkte⟩}[⟨Zusatzpunkte⟩]`

definiert eine einzelne Erwartung, die innerhalb einer `aufgabe` oder einer `loesung` angegeben werden muss. Der Parameter kann beliebigen L^AT_EX-Code enthalten bis auf Verbatim-Elemente. Des weiteren werden die Punkte für diese Erwartung als Parameter erwartet. Als optionalen Parameter können Zusatzpunkte angegeben werden. Wird die Erwartung innerhalb einer `teilaufgaben`-Umgebung eingefügt, so bekommt die Erwartung die Nummer der Teilaufgabe mit.

`\erwartungshorizont`

setzt den Erwartungshorizont im gewählten Stil, falls die automatische Erzeugung über die Paketoption `erwartungshorizontAnzeigen` nicht genutzt wird.

`\notenverteilung`

setzt die Notenverteilung, falls die automatische Erzeugung über den Erwartungshorizont nicht genutzt wird. Die Verteilung wird über die Gesamtpunkte aller Aufgaben unter Berücksichtigung des gewählten Notenschemas ermittelt.

6. Format

Dieses Modul definiert einige grundlegende Paketoptionen für die Formatierung von Dokumenten und stellt passende Makros bereit. Außerdem bindet es das Paket `ulem` für verschiedene Textformatierungen ein.

6.1. Formatierungen

Über verschiedene Paketoptionen kann das Aussehen der vom Schule-Paket erstellten Dokumente beeinflusst werden. Es sind zudem einige Makros vorhanden, die häufig verwendete Formatierungen und Sonderzeichen bereitstellen.

6.1.1. Paketoptionen

`farbig`

aktiviert die farbige Darstellung.

`sprache`

fügt eine Liste von CSV Sprachen dem Babelpaket hinzu. `ngerman` ist immer geladen (als Hauptsprache)

6. Format

6.1.2. Befehle

`\achtung{⟨Text⟩}`

Der Befehl `\achtung` stellt den angegebenen Text mit einem vorangestellte Warnsymbol und einem fettgedruckten »Achtung:« dar.

```
1 \achtung{Dies ist ein Beispiel.}     ⚠ Achtung: Dies ist ein Beispiel.
```

`\chb*`

setzt eine ankreuzbares Kästchen, der optionale Stern markiert dieses.

```
1 \chb \chb*                              
```

`\dashuline{⟨Text⟩}`

Der Befehl `\dashuline` stellt den angegebenen Text unterstrichelt dar.

```
1 \dashuline{Dies ist ein Beispiel.}     Dies ist ein Beispiel.
```

`\dotuline{⟨Text⟩}`

Der Befehl `\dotuline` stellt den angegebenen Text unterpunktet dar.

```
1 \dotuline{Dies ist ein Beispiel.}     ..Dies ist ein Beispiel.
```

`\hinweis{⟨Text⟩}`

Der Befehl `\hinweis` stellt den angegebenen Text mit einem vorangestellte Warnsymbol und einem fettgedruckten »Hinweis:« dar.

```
1 \hinweis{Dies ist ein Beispiel.}     ⚠ Hinweis: Dies ist ein Beispiel.
```

`\person{⟨Name⟩}`

Der Name einer Person wird mit dem Befehl `\person{⟨Name⟩}` hervorgehoben.

6. Format

```
1 \person{Einstein}
```

EINSTEIN

`\so{<Text>}`

Der Befehl `\so` stellt den angegebenen Text durchgestrichen dar und ermöglicht es so in Wertetabellen bzw. Schreibtischtests einzelne Werte durchzustreichen.

```
1 \so{Dies ist ein Beispiel.}
```

~~Dies ist ein Beispiel.~~

`\uline{<Text>}`

Der Befehl `\uline` stellt den angegebenen Text unterstrichen dar.

```
1 \uline{Dies ist ein Beispiel.}
```

Dies ist ein Beispiel.

`\uuline{<Text>}`

Der Befehl `\uuline` stellt den angegebenen Text doppelt unterstrichen dar.

```
1 \uuline{Dies ist ein Beispiel.}
```

Dies ist ein Beispiel.

`\uwave{<Text>}`

Der Befehl `\uwave` stellt den angegebenen Text unterschlängelt dar.

```
1 \uwave{Dies ist ein Beispiel.}
```

~Dies ist ein Beispiel.

`\xout{<Text>}`

Der Befehl `\xout` stellt den angegebenen Text durchgekreuzt dar.

```
1 \xout{Dies ist ein Beispiel.}
```

~~Dies ist ein Beispiel.~~

6.2. Kopf- Fußzeilen

Das Modul stellt einige Standardformatierungen für Kopf- und Fußzeilen bereit, die in den Dokumenttypen verwendet werden.

6.2.1. Paketoptionen

`datumAnzeigen`

aktiviert die Darstellung des Datums in der Kopfzeile.

`namensfeldAnzeigen`

aktiviert die Darstellung eines Namensfelds in der Kopfzeile.

6.2.2. Befehle

`\schule@kopfUmbruch`

setzt einen Umbruch, wenn die Kopfzeile durch eine gesetzte Option mehrzeilig wird. Kann in Kopfzeilen verwendet werden, um sie gleichmäßig auszurichten.

`\schule@kopfInnen`

setzt eine Kopfzeile mit der vollständigen Lerngruppenbezeichnung (Fach, Lerngruppe) und je nach Paketoption einem Namensfeld in der zweiten Zeile. Ein etwaiger Umbruch der anderen Kopfzeilenfelder wird berücksichtigt.

`\schule@kopfMitte`

setzt eine Kopfzeile mit dem Titel des Dokuments. Ein etwaiger Umbruch der anderen Kopfzeilenfelder wird berücksichtigt.

`\schule@kopfAussen`

setzt eine Kopfzeile mit dem gegebenen Parameter, üblicherweise dem Dokumenttypbezeichner und je nach Paketoption einem Datumsfeld in der zweiten Zeile. Ein etwaiger Umbruch der anderen Kopfzeilenfelder wird berücksichtigt.

6.3. Seitenzahlen

Die Darstellung von Seitenzahlen in Dokumenten kann ebenfalls beeinflusst werden.

6.3.1. Paketoptionen

`seitenzahlen = auto|autoGesamt|immer|immerGesamt|kein` Voreinstellung: `autoGesamt` legt die Art der Darstellung von Seitenzahlen fest. Die verschiedenen Varianten sind davon abhängig, ob es sich um ein einseitiges oder mehrseitiges Dokument handelt:

Variante	einseitig	mehrseitig
auto		1
autoGesamt		1 von n
immer	1	1
immerGesamt	1 von 1	1 von n
keine		

6.3.2. Befehle

`\Seitenzahlen`

setzt die Seitenzahlen gemäß der über die Paketoption `seitenzahlen` gewählten Variante. Dieser Befehl kann in Kopf- oder Fußzeilen verwendet werden.

6.4. Strukturelemente

Verschiedene, häufig verwendete Strukturelemente gehören ebenfalls zum Umfang des Pakets. Darunter sind verschiedene Listen und Platzhalter.

6.4.1. Umgebungen

`\begin{smalldescription}`

Die Listenumgebung `smalldescription` ist identisch zur `description`-Standardumgebungen von \LaTeX , bis auf die Tatsache, dass zwischen den einzelnen Punkten der Abstand verkleinert wurde.

`\begin{smallenumerate}`

Die Listenumgebung `smallenumerate` ist identisch zur `enumerate`-Standardumgebungen von \LaTeX , bis auf die Tatsache, dass zwischen den einzelnen Punkten der Abstand verkleinert wurde.

`\begin{smallitemize}`

Die Listenumgebung `smallitemize` ist identisch zur `itemize`-Standardumgebungen von \LaTeX , bis auf die Tatsache, dass zwischen den einzelnen Punkten der Abstand verkleinert wurde.

Der Unterschied wird besonders dann deutlich, wenn man die Umgebungen nebeneinander sieht:

```

1 \begin{minipage}[t]{.4\textwidth}
2   \texttt{itemize}-Umgebung:
3   \begin{itemize}
4     \item Punkt
5     \item Punkt
6     \item Punkt
7   \end{itemize}

```

```

8 \end{minipage}
9 \begin{minipage}[t]{.4\textwidth}
10   \texttt{smallitemize}-Umgebung:
11   \begin{smallitemize}
12     \item Punkt
13     \item Punkt
14     \item Punkt
15   \end{smallitemize}
16 \end{minipage}

```

itemize-Umgebung:

- Punkt
- Punkt
- Punkt

smallitemize-Umgebung:

- Punkt
- Punkt
- Punkt

6.5. Wörtliche Rede, Zitate und Anführungszeichen

6.5.1. Paketoptionen

`zitate = guillemets|quotes|swiss` Voreinstellung: `guillemets`
 Standardmäßig werden die deutschen »Möwchen« geladen. Über `quotes` können doppelte „Hochkommata“ (99-66) geladen werden. Die Darstellung von doppelten Hochkommata im “Modus 66-99” kann mittels `swiss` erreicht werden.

6.5.2. Befehle

`\enquote{<Text>}`

Setzen von Passagen in typographische Anführungszeichen.

```

1 \enquote{Beispiel}           »Beispiel«

```

`\diastring{<Zeichenkette>}`

Darstellung von Zeichenketten (strings) in Diagrammen usw.

```

1 \diastring{Beispiel}       "Beispiel"

```

Hinweis: Teilweise kann es zu Fehlern kommen, wenn das Paket `csquotes` mit eigenen Optionen geladen wird.

7. Formulare

Das Modul `Formulare` befindet sich noch in der Entwicklung. Es soll die Möglichkeit bieten, Formulare aus dem `hyperref` in PDF-Dokumenten komfortabel einzubinden. Dabei sollen auch die Möglichkeit erhalten bleiben, dass die gesetzten Dokumente in ausgedruckter Form für den Unterricht nutzbar sind.

Durch Einbindung des Moduls wird automatisch eine `Form`-Umgebung um das komplette Dokument erstellt. Die Übergabe von Parametern an diese Umgebung muss er noch implementiert werden.

7.1. Befehle

`\feldLinFormular` [*Abstand*] {*Anzahl*} [*Textfeldargumente*]

erstellt ein Textfeld für das Formular, das mit der gegebenen Anzahl an Linien für den Ausdruck versehen ist. Der Abstand der Linien zueinander kann angegeben werden, er beträgt standardmäßig 0,7cm. Dem Befehl können auch Argumente mitgegeben werden, die an das intern genutzte `TextField` übergeben werden.

Bei dem Textfeld ist die Schriftgröße 18pt gesetzt, die ungefähr passend zu den Linienabstand von 0,7cm ist. Außerdem wurde die Hintergrundfarbe auf leer gesetzt, damit die Linien zu erkennen sind.

```
1 \feldLinFormular[0.5cm]{3}
```

8. Kuerzel

Das Modul `Kuerzel` stellt einige Makros bereit, die Kurzschreibweisen für häufig verwendete Schreibweisen bereitstellen. Sofern relevant, wird dabei die Schreibweise an die gewählte Variante des Genderings, im Sinne einer geschlechtergerechten Sprache angepasst.

8.1. Paketoptionen

`gendering` = `binneni` | `fem` | `gap` | `mas` | `split` | `star` Voreinstellung: `split`
 Standardmäßig wird die amtlich geforderte Schreibweise des Splittings (etwa Schülerinnen und Schüler) verwendet.

- Splitting: `gendering = {split}`

Außerdem werden folgende Varianten unterstützt:

- Gender-Gap: `gendering = {gap}`
- Gender-Star: `gendering = {star}`
- Binnen-I: `gendering = {binneni}`

Für spezielle Fälle, kann auch die ausschließliche Nutzung einer Geschlechtsform erzwungen werden:

- Generisches Femininum: `gendering = {fem}`
- Generisches Maskulinum: `gendering = {mas}`

8.2. Befehle

`\Lkr`
Lehrkraft

`\GK`
Grundkurs

`\Lkre`
Lehrkräfte

`\GKe`
Grundkurse

`\Lpr`
Lehrperson

`\GKen`
Grundkursen

`\Lprn`
Lehrpersonen

`\LK`
Leistungskurs

`\EuE`
Eltern und Erziehungsberechtigte

`\LKe`
Leistungskurse

`\EuEn`
Eltern und Erziehungsberechtigten

`\LKen`
Leistungskursen

`\EK`
Erweiterungskurs

`\SuS`

`\EKe`
Erweiterungskurse

Gendering	Ergebnis
<code>binneni</code>	SchülerInnen
<code>fem</code>	Schülerinnen
<code>gap</code>	Schüler_innen
<code>mas</code>	Schüler
<code>split</code>	Schülerinnen und Schüler
<code>star</code>	Schüler*innen

`\EKen`
Erweiterungskursen

\SuSn

Gendering	Ergebnis
binnen	SchülerInnen
fem	Schülerinnen
gap	Schüler_innen
mas	SchülerIn
split	Schülerinnen und Schülern
star	Schüler*innen

\LuLn

Gendering	Ergebnis
binnen	LehrerInnen
fem	Lehrerinnen
gap	Lehrer_innen
mas	Lehrern
split	Lehrerinnen und Lehrern
star	Lehrer*innen

\LuL

Gendering	Ergebnis
binnen	LehrerInnen
fem	Lehrerinnen
gap	Lehrer_innen
mas	Lehrer
split	Lehrerinnen und Lehrer
star	Lehrer*innen

\KuK

Gendering	Ergebnis
binnen	KollegInnen
fem	Kolleginnen
gap	Kolleg_innen
mas	Kollegen
split	Kolleginnen und Kollegen
star	Kolleg*en*innen

9. Lizenzen

Dieses Modul definiert die Paketooptionen zum Festlegen der Lizenz des Dokuments und bietet Makros zum Setzen des Lizenznamens und der Lizenzsymbole an. Es setzt dabei nun auf das Paket `doclicense`, das auch entsprechende Werte in die PDF-Datei einbindet. Aus Kompatibilitätsgründen wird das alte Verhalten beibehalten.

Geändert in
Version 0.8.3

9.1. Paketooptionen

`lizenz = {\Lizenzcode}`

Voreinstellung: `cc-by-nc-sa-4`

legt die Lizenz für das Dokument fest. Aktuell werden folgende Codes unterstützt:

- `cc-by-4`
- `cc-by-sa-4`
- `cc-by-nc-sa-4`

`nohyperxmp`

deaktiviert die Einbindung der Lizenz über `hyperxmp` in das Dokument.

Eingeführt in
Version 0.8.4

9.2. Befehle

`\lizenzName`

gibt den vollständigen Namen der Lizenz des Dokuments zurück.

`\lizenzNameKurz`

gibt den gekürzten Namen der Lizenz des Dokuments zurück.

`\lizenzSymbol`

setzt das Symbol der Lizenz des Dokuments.

10. Metadaten

Dieses Modul definiert die Paketoptionen zum Setzen bestimmter Metadaten und bietet Makros zum Zugriff darauf an.

Im Gegensatz zu älteren Versionen des Schule-Paketes werden für Metadaten immer die Standardmakros von L^AT_EX eingesetzt, soweit dies möglich ist. Dies gilt etwa für Autor (`\author{\langle Autor \rangle}`), Datum (`\date{\langle Datum \rangle}`) und Titel (`\title{\langle Titel \rangle}`) des Dokumentes. Es werden allerdings auch für diese Metadaten Makros zum einfachen Zugriff aus dem Dokument heraus definiert.

10.1. Paketoptionen

`fach = {\langle Fach \rangle}`

legt das Fach für das Dokument fest, siehe Abschnitt III, S. 37.

`lerngruppe = {\langle Lerngruppe \rangle}`

legt die Lerngruppe für das Dokument fest.

`nummer = {\langle Dokumentnummer \rangle}`

legt die Dokumentnummer fest.

10.2. Befehle

`\Autor`

gibt den Autor des Dokuments zurück.

`\Datum`

gibt das Datum des Dokuments zurück.

`\Fach`

gibt das Fach des Dokuments zurück.

`\Lerngruppe`

gibt die Lerngruppe des Dokuments zurück. Als Alternative ist auch `\Kurs` definiert.

`\Nummer`

gibt die Nummer des Dokuments zurück.

`\Titel`

gibt den Titel des Dokuments zurück.

11. Papiertypen

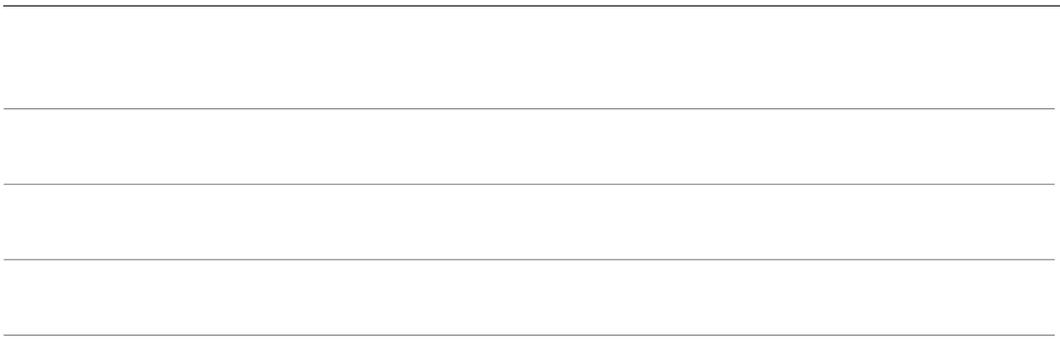
Das Modul `Papiertypen` stellt einige Makros bereit, die es erlauben, Freiräume zum Bearbeiten von Aufgaben zu setzen. Hierzu stehen verschiedene Muster zur Auswahl. Die entsprechenden Felder werden dabei in der Breite jeweils auf `\linewidth` skaliert, allerdings so, dass ein vollständiges Muster entsteht. Die zur Verfügung stehende Breite wird also optimal genutzt.

11.1. Befehle

`\feldLin[⟨Abstand⟩]{⟨Anzahl⟩}`

setzt die angegebene Anzahl Linien mit dem angegebenen Abstand zueinander. Der Standardabstand beträgt 1cm .

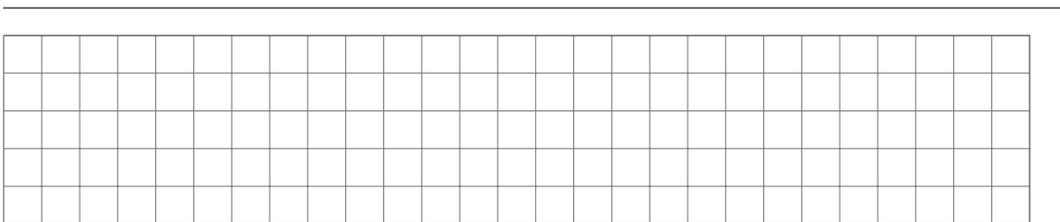
```
1 \feldLin[1cm]{4}
```



`\feldKar[⟨Seitenlänge⟩]{⟨Anzahl⟩}`

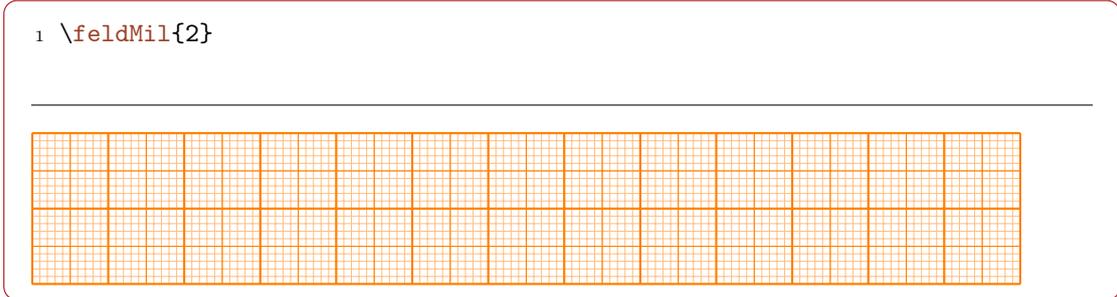
setzt die angegebene Anzahl von Karo-Kästchen mit einer gegebenen Seitenlänge. Der Standard für die Seitenlänge beträgt $0,5\text{cm}$.

```
1 \feldKar[0.5cm]{5}
```



`\feldMil{⟨Anzahl⟩}`

setzt die angegebene Anzahl von Kästchen im Millimeterpapiermuster untereinander. Die Farbe wird von der Paketooption `farbig` beeinflusst. Ohne das Setzen dieser Option wird das Millimeterpapiermuster nur in schwarz/weiß dargestellt.



12. Symbole

Dieses Modul stellt einige für den Schulkontext relevante Unicode-Symbole aus den vom Paket `utfsym` unterstützten Blöcken als benannte Makros zur Verfügung.

12.1. Befehle

Körperteile	<code>\symZeigefinger</code>	Material	<code>\symPin</code>
<code>\symNase</code>  (1F443)	 (1F446)	<code>\symBleistift</code>  (1F589)	 (1F4CC)
<code>\symAuge</code>  (1F441)	<code>\symApplaus</code>  (1F44F)	<code>\symFueller</code>  (1F58B)	<code>\symNadel</code>  (1F4CD)
<code>\symAugen</code>  (1F440)	Kommunikation	<code>\symKuli</code>  (1F58A)	<code>\symPinself</code>  (1F58C)
<code>\symMund</code>  (1F444)	<code>\symSprechblase</code>  (1F5E9)	<code>\symBuntstift</code>  (1F58D)	<code>\symBuch</code>  (1F56E)
<code>\symZunge</code>  (1F445)	<code>\symZweiSprechblasen</code>  (1F5EA)	<code>\symLineal</code>  (1F4CF)	<code>\symBild</code>  (1F5BC)
<code>\symOhr</code>  (1F442)	<code>\symDreiSprechblasen</code>  (1F5EB)	<code>\symGeodreieck</code>  (1F4D0)	<code>\symMikroskop</code>  (1F52C)
<code>\symDaumenHoch</code>  (1F44D)	<code>\symDenkblase</code>  (1F5ED)	<code>\symBueroklammer</code>  (1F4CE)	<code>\symHeft</code>  (1F4D3)
<code>\symDaumenRunter</code>  (1F44E)	Kunst	<code>\symBueroklammern</code>  (1F587)	<code>\symBuecher</code>  (1F4DA)
	<code>\symPalette</code>  (1F3A8)		

12. Symbole

`\symKlembrett`
 (1F4CB)

`\symCD`
 (1F4BF)

`\symZeitung`
 (1F4F0)

`\symThermometer`
 (1F321)

`\symSchere`
 (2700)

`\symSchloss`
 (1F512)

`\symSchlossOffen`
 (1F513)

`\symSchluessel`
 (1F511)

`\symGlocke`
 (1F514)

`\symKeineGlocke`
 (1F515)

`\symLupe`
 (1F50D)

Musik

`\symNote`
 (1F39C)

`\symNoten`
 (1F3B6)

Smileys

`\symSmileyLachend`
 (1F642)

`\symSmileyNeutral`
 (1F610)

`\symSmileyTraurig`
 (1F641)

`\symSmileyGrinsend`
 (1F600)

`\symSmileySchlafend`
 (1F614)

`\symSmileyZwinkernd`
 (1F609)

Sonstiges

`\symKlee`
 (1F340)

`\symSonne`
 (1F323)

`\symMond`
 (1F319)

`\symStern`
 (1F31F)

`\symUhr`
 (1F551)

`\symHaken`
 (1F5F8)

Spielkarten

`\symSpielkarte`
 (1F0A0)

`\symPik`
 (2660)

`\symHerz`
 (2665)

`\symKaro`
 (2666)

`\symKreuz`
 (2663)

`\symPikAss`
 (1F0A1)

`\symPikZwei`
 (1F0A2)

`\symPikDrei`
 (1F0A3)

`\symPikVier`
 (1F0A4)

`\symPikFuenf`
 (1F0A5)

`\symPikSechs`
 (1F0A6)

`\symPikSieben`
 (1F0A7)

`\symPikAcht`
 (1F0A8)

`\symPikNeun`
 (1F0A9)

`\symPikZehn`
 (1F0AA)

`\symPikBube`
 (1F0AB)

`\symPikDame`
 (1F0AD)

`\symPikKoenig`
 (1F0AE)

`\symHerzAss`
 (1F0B1)

`\symHerzZwei`
 (1F0B2)

`\symHerzDrei`
 (1F0B3)

`\symHerzVier`
 (1F0B4)

`\symHerzFuenf`
 (1F0B5)

`\symHerzSechs`
 (1F0B6)

`\symHerzSieben`
 (1F0B7)

`\symHerzAcht`
 (1F0B8)

`\symHerzNeun`
 (1F0B9)

12. Symbole

<code>\symHerzZehn</code>  (1F0BA)	<code>\symKaroBube</code>  (1F0CB)	<code>\symKreuzDame</code>  (1F0DD)	<code>\symTennis</code>  (1F3BE)
<code>\symHerzBube</code>  (1F0BB)	<code>\symKaroDame</code>  (1F0CD)	<code>\symKreuzKoenig</code>  (1F0DE)	<code>\symTischtennis</code>  (1F3D3)
<code>\symHerzDame</code>  (1F0BD)	<code>\symKaroKoenig</code>  (1F0CE)	Sport	
<code>\symHerzKoenig</code>  (1F0BE)	<code>\symKreuzAss</code>  (1F0D1)	<code>\symBaseball</code>  (26BE)	<code>\symPokal</code>  (1F3C6)
<code>\symKaroAss</code>  (1F0C1)	<code>\symKreuzZwei</code>  (1F0D2)	<code>\symBasketball</code>  (1F3C0)	<code>\symMedaille</code>  (1F3C5)
<code>\symKaroZwei</code>  (1F0C2)	<code>\symKreuzDrei</code>  (1F0D3)	<code>\symFussball</code>  (26BD)	<code>\symZielflagge</code>  (1F3C1)
<code>\symKaroDrei</code>  (1F0C3)	<code>\symKreuzVier</code>  (1F0D4)	<code>\symVolleyball</code>  (1F3D0)	
<code>\symKaroVier</code>  (1F0C4)	<code>\symKreuzFuenf</code>  (1F0D5)	<code>\symHockey</code>  (1F3D1)	Technik
<code>\symKaroFuenf</code>  (1F0C5)	<code>\symKreuzSechs</code>  (1F0D6)	<code>\symLaufen</code>  (1F3C3)	
<code>\symKaroSechs</code>  (1F0C6)	<code>\symKreuzSieben</code>  (1F0D7)	<code>\symReiten</code>  (1F3C7)	<code>\symHandy</code>  (1F4F1)
<code>\symKaroSieben</code>  (1F0C7)	<code>\symKreuzAcht</code>  (1F0D8)	<code>\symSchwimmen</code>  (1F3CA)	<code>\symKeinHandy</code>  (1F4F5)
<code>\symKaroAcht</code>  (1F0C8)	<code>\symKreuzNeun</code>  (1F0D9)	<code>\symSki</code>  (26F7)	
<code>\symKaroNeun</code>  (1F0C9)	<code>\symKreuzZehn</code>  (1F0DA)	<code>\symSnowboard</code>  (1F3C2)	Theater
<code>\symKaroZehn</code>  (1F0CA)	<code>\symKreuzBube</code>  (1F0DB)	<code>\symSurfen</code>  (1F3C4)	<code>\symTheater</code>  (1F0DD)

Verkehrsmittel	<code>\symSchwebebahn</code>  (1F69F)	<code>\symFussgaenger</code>  (1F6B8)	<code>\symWuerfelDrei</code>  (2682)
<code>\symAuto</code>  (1F698)	<code>\symSeilbahn</code>  (1F6A1)	<code>\symRollstuhl</code>  (267F)	<code>\symWuerfelVier</code>  (2683)
<code>\symBus</code>  (1F68C)	<code>\symSchiff</code>  (1F6A2)	Würfel	<code>\symWuerfelFuenf</code>  (2684)
<code>\symBahn</code>  (1F682)	<code>\symBoot</code>  (1F6A3)	<code>\symWuerfelEins</code>  (2680)	
<code>\symStrassenbahn</code>  (1F68B)	<code>\symFahrrad</code>  (1F6B2)	<code>\symWuerfelZwei</code>  (2681)	<code>\symWuerfelSechs</code>  (2685)

13. Texte

Dieses Modul definiert einige Umgebungen, die für die Formatierung längerer Texte hilfreich sind.

⚠ **Achtung:** Da die Umgebungen mit Zeilennummern nicht ohne große Klammern in umrahmte Boxen gesetzt werden können, sehen die Beispiele hier ein wenig anders aus.

13.1. Befehle

`\resetZeilenNr`

Standardmäßig werden die Zeilennummern über die Umgebungsgrenzen hinweg vergeben. Möchte man in jeder neuen Umgebung mit 1 beginnen, so muss man die Zeilennummer mit diesem Befehl zunächst zurücksetzen.

13.2. Umgebungen

`\begin{mehrsparzig}[\langle Anzahl \rangle]`

Setzt einen gegebenen Text mehrspaltig, wobei die Anzahl der Spalten angegeben werden kann. Die Standardanzahl ist 2.

```

1 \begin{mehrsparzig}
2   \blindtext
3 \end{mehrsparzig}

```

13. Texte

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch

die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

`\begin{zeilenNr}[\langle Modulo \rangle]`

Setzt einen gegebenen Text mit Zeilennummern, wobei ein Modulo für den Abstand der Zeilennummern angegeben werden kann.

Der Standardmodulo beträgt 5.

```
1 \begin{zeilenNr}[1]
2   \blindtext
3 \end{zeilenNr}
```

- 1 Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist
- 2 selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so?
- 3 Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“?
- 4 Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich
- 5 die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander
- 6 stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele
- 7 verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss
- 8 keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“
- 9 dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

`\begin{zeilenNrMehrspaltig}[\langle Modulo \rangle]{\langle Anzahl \rangle}`

Setzt einen gegebenen Text mehrspaltig mit Zeilennummern, wobei die Anzahl der Spalten angegeben werden muss und zusätzlich ein Modulo für den Abstand der Zeilennummern angegeben werden kann.

Die Zeilennummern stehen jeweils links neben der jeweiligen Spalte. Der Standardmodulo beträgt 5.

```
1 \begin{zeilenNrMehrspaltig}[5]{3}
2   \blindtext
```

```
3 \end{zeilenNrMehrspaltig}
```

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. 15 Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein 20 Blindtext“ oder „Huardest gefburn“? Kjift – mitnich-

ten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit 25 oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buch-

staben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, 30 da sie eine falsche Anmutung vermitteln.

```
\begin{zeilenNrZweispaltig}[\langle Modulo \rangle]
```

Setzt einen gegebenen Text zweispaltig mit Zeilennummern, wobei ein Modulo für den Abstand der Zeilennummern angegeben werden kann.

Die Zeilennummern stehen links neben der linken Spalte und rechts neben der rechten Spalte. Der Standardmodulo beträgt 5.

```
1 \begin{zeilenNrZweispaltig}[3]
2   \blindtext
3 \end{zeilenNrZweispaltig}
```

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Teil III.

Fächer

14. Nutzung der Fachmodule

Die fachspezifischen Funktionen und Vorgaben sind in sogenannte Fachmodule aufgeteilt, die über Paketoptionen flexibel geladen werden können.

`fach = <Fach>`

Mit der Paketoption `fach` kann das Fach für das Dokument festgelegt werden. Es werden dann alle fachspezifischen Funktionen und Vorgaben für das Fach geladen.

Mit der Angabe von `fach = ohne` kann auf die Angabe eines Faches verzichtet werden, etwa für die Einbindung in Dokumentationen etc.

Wird ein Fach angegeben, zu dem kein Fachmodul existiert, so wird dieses nur als Bezeichner verwendet.

`weitereFaecher = {\langle Fach 1, Fach 2, \dots \rangle}`

Für fächerübergreifenden Unterricht können weitere Fachmodule geladen werden, indem eine kommaseparierte Liste von Fachmodulen angegeben wird.

Hierbei wird auf das Laden möglicher »Standalone-Abschnitte« der Fachmodule verzichtet, vgl. Abschnitt 34, S. 62fachmodul:

15. Informatik

Das Fachmodul `Informatik` bindet Pakete ein, um Klassen- sowie Objektdiagramme (`pgf-umlcd`), Syntaxdiagramme (`syntaxdi`), Struktogramme (`struktex`) und Sequenzdiagramme (`pgf-umlsd`) setzen zu können. Die entsprechenden Dokumentationen sind bei den jeweiligen Paketen zu finden. Hier sind lediglich Abweichungen und Erweiterungen vom Standardumfang der Pakete dokumentiert.

15.1. Objektdiagramme

`\anchormark[\langle Horizontale Verschiebung \rangle] {\langle Nodename \rangle} [\langle Skalierung \rangle]`

Durch den Befehl `\anchormark` können Objektdiagramme mit Beziehungsattributen ausgestattet werden, die an der korrekten Stelle hinter dem Attributbezeichner beginnen.

⚠ Achtung: Dieser Befehl ist nicht skalierungssicher!

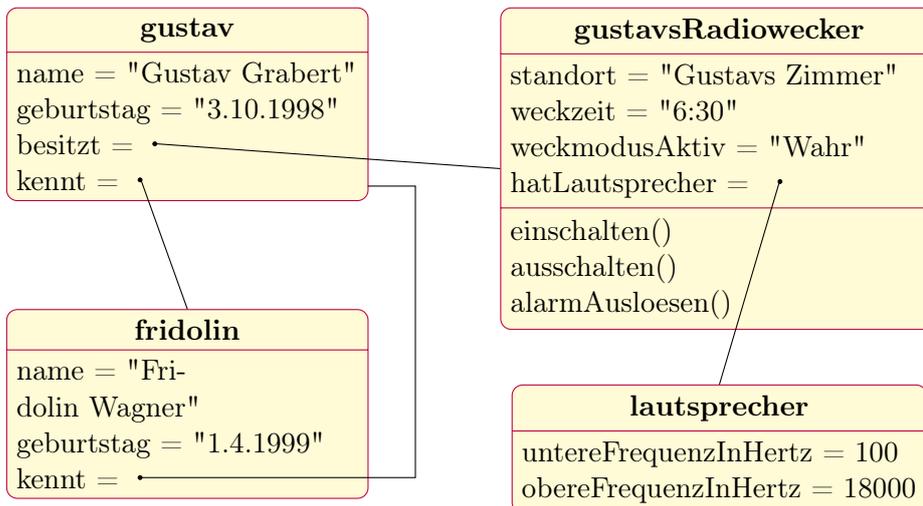
```
1 \begin{tikzpicture}[remember picture]
2   \begin{object}[text width=5.5cm]{gustavsRadiowecker}{-3,0}
3     \attribute{standort = \diastring{Gustavs Zimmer}}
```

15. Informatik

```

4      \attribute{weckzeit = \diastring{6:30}}
5      \attribute{weckmodusAktiv = \diastring{Wahr}}
6      \attribute{hatLautsprecher = \anchormark{hatLautsprecher}[0.025]}
7      \operation{einschalten()}
8      \operation{ausschalten()}
9      \operation{alarmAusloesen()}
10     \end{object}
11     \begin{object}[text width=4.5cm]{gustav}{-10,0}
12         \attribute{name = \diastring{Gustav Grabert}}
13         \attribute{geburtstag = \diastring{3.10.1998}}
14         \attribute{besitzt = \anchormark{besitzt}[0.025]}
15         \attribute{kennt = \anchormark{gKennt}[0.025]}
16     \end{object}
17     \begin{object}[text width=4.5cm]{fridolin}{-10,-4}
18         \attribute{name = \diastring{Fridolin Wagner}}
19         \attribute{geburtstag = \diastring{1.4.1999}}
20         \attribute{kennt = \anchormark{fKennt}[0.025]}
21     \end{object}
22     \begin{object}[text width=5.2cm]{lautsprecher}{-3,-5}
23         \attribute{untereFrequenzInHertz = 100}
24         \attribute{obereFrequenzInHertz = 18000}
25     \end{object}
26
27     \draw (hatLautsprecher) -- (lautsprecher.north);
28     \draw (gKennt.south east) -- (fridolin.north);
29     \draw (besitzt.east) -- (gustavsRadiowecker.west);
30     \draw (fKennt.east) -- ($(fKennt.east)+(3.5,0)$
31         -| ($(gustav.south)+(3,0.2)$) -- ($(gustav.south east)
32             +(-0.01,0.2)$);
33 \end{tikzpicture}

```



15.2. Sequenzdiagramme

`\skalieresequenzdiagramm{<Faktor>}`

⚠ **Achtung:** Sollte nicht mehr verwendet werden: Besser `resizebox` oder `scalebox`

Da es vorkommen kann, dass Sequenzdiagramme zu breit für eine Seite sind, kann mit dem Befehl `\skalieresequenzdiagramm{<Faktor>}` die Größe des Sequenzdiagramms angepasst werden, wenn er innerhalb der Umgebung `sequencediagram` ausgeführt wird.

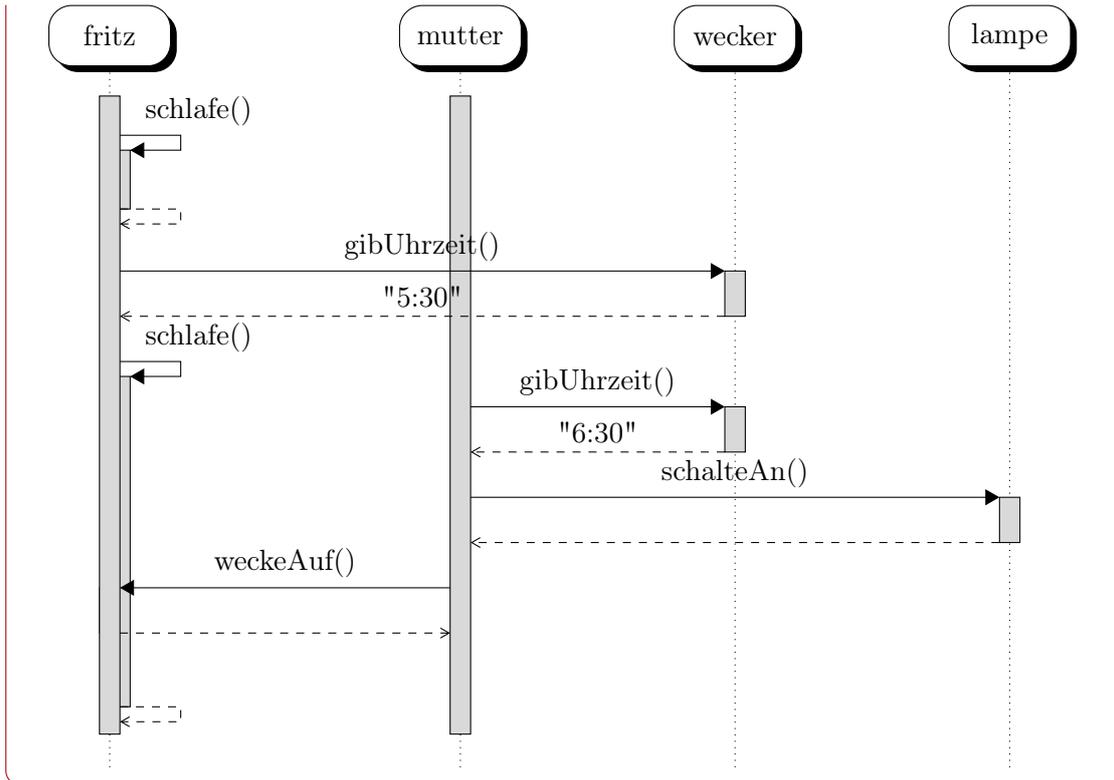
`\newthreadtwo[<Farbe>]{<Bezeichnung>}{<Name>}{<Abstand>}`

Threads haben im Gegensatz zu Instanzen im Paket `pgf-umlsd` immer einen festen Abstand zu den Nachbarn. Durch den neuen Befehl `\newthreadtwo` ist es über den dritten Parameter möglich, diesen Abstand zu verändern. Dabei verhält sich der neue Parameter für den Abstand genauso wie der zugehörige optionale Parameter bei Instanzen.

```

1 \begin{sequencediagram}
2   \newthread{fritz}{fritz}
3   \newthreadtwo{mutter}{mutter}{3cm}
4   \newinst[2]{wecker}{wecker}
5   \newinst[2]{lampe}{lampe}
6
7   \begin{callself}[2]{fritz}{schlafe()}{}
8   \end{callself}
9   \begin{call}{fritz}{gibUhrzeit()}{wecker}{\diastring{5:30}}
10  \end{call}
11  \begin{callself}[2]{fritz}{schlafe()}{}
12    \begin{call}{mutter}{gibUhrzeit()}{wecker}{\diastring{6:30}}
13    \end{call}
14  \begin{call}{mutter}{schalteAn()}{lampe}{}
15  \end{call}
16  \begin{call}{mutter}{weckeAuf()}{fritz}{}
17  \end{call}
18  \end{callself}
19 \end{sequencediagram}

```



`\nextlevel`

Im Paket für Sequenzdiagramme ist vorgesehen, dass man mit `\prevlevel` wieder einen Schritt nach oben gehen kann. Zusätzlich wird ein Befehl `\nextlevel` bereitgestellt, mit dem man auch einen zusätzlichen Schritt nach unten gehen kann, um ggf. etwas mehr Platz und Abstand zu schaffen.

15.3. Struktogramme

Mit dem Paket `struktex` lassen sich sehr einfach Struktogramme setzen:

```

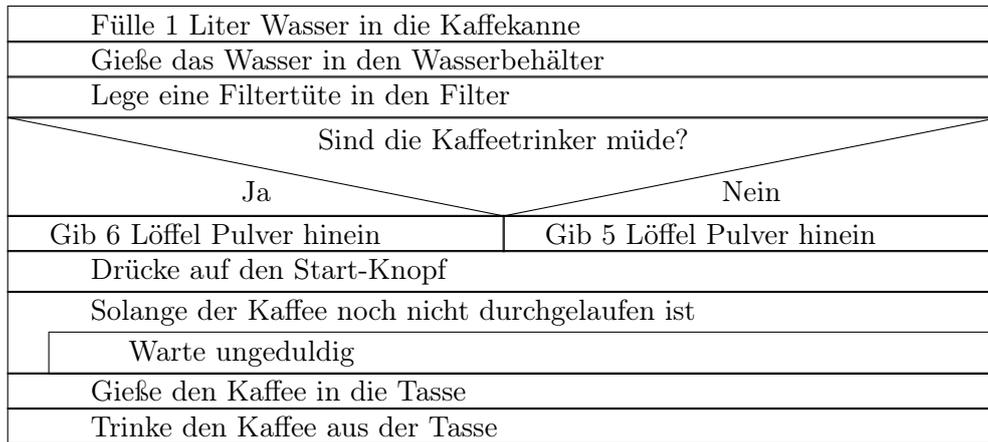
1 \begin{struktogramm}(130,60)[koche Kaffee]
2 \assign{F"ulle 1 Liter Wasser in die Kaffekanne}
3 \assign{Gie"se das Wasser in den Wasserbeh"alter}
4 \assign{Lege eine Filtert"ute in den Filter}
5 \ifthenelse{5}{5}{Sind die Kaffeetrinker m"ude?}{Ja}{Nein}
6 \assign{Gib 6 L"offel Pulver hinein}
7 \change
8 \assign{Gib 5 L"offel Pulver hinein}
9 \ifend
10 \assign{Dr"ucke auf den Start-Knopf}
  
```

```

11 \while{Solange der Kaffee noch nicht durchgelaufen ist}
12   \assign{Warte ungeduldig}
13 \whileend
14 \assign{Gie"se den Kaffee in die Tasse}
15 \assign{Trinke den Kaffee aus der Tasse}
16 \end{struktogramm}

```

koche Kaffee



15.4. Syntaxdiagramme

Beispiele sind dem Paket `syntaxdi` zu entnehmen.

15.5. Flussdiagramme

Für Flussdiagramme, bzw. Programmablaufpläne steht der Style `pap` bereit, der in `tikzpicture` genutzt werden kann. Damit werden `\node` ein entsprechendes Aussehen gegeben. Es stehen zur Verfügung:

startstop Für den Beginn bzw. das Ende eines Ablaufs, als Rechteck mit runden Ecken.

verzweigung Für Abfragen und Wiederholungen als Diamant.

aktion Einfache Aktionen in einem Rechteck.

einausgabe Ein Rhomboid wird für Ein- oder Ausgaben genutzt.

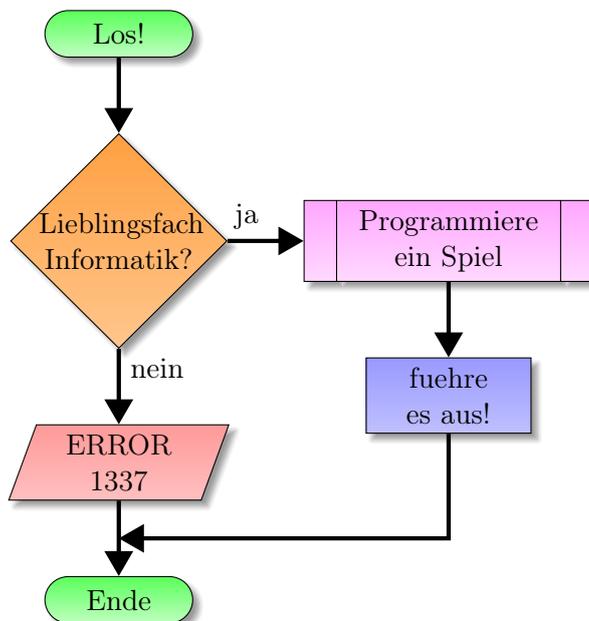
unterprogramm Ein Rechteck ergänzt um freie Flächen auf der linken und rechten Seite stellt den Aufruf eines Unterprogramms dar.

Weiterhin können Linien mit dem Style `linie` versehen werden, um diese deutlicher darzustellen.

```

1 \begin{tikzpicture}[pap]
2   \node[startstop] (s1){Los!};
3   \node[verzweigung, below = of s1] (v1) {Lieblingsfach Informatik?};
4   \node[unterprogramm, right = of v1] (up1) {\nodepart[text width=7em]{
two} Pro\gram\mie\re ein Spiel};
5   \node[aktion, below = of up1] (a1) {fuehre es aus!};
6   \node[einausgabe, below = of v1] (ea1) {ERROR 1337};
7   \node[startstop, below = of ea1] (e1){Ende};
8
9   \draw[linie] (s1)--(v1);
10  \draw[linie] (v1)--(up1) node[near start, above] {ja};
11  \draw[linie] (v1)--(ea1) node[near start, right] {nein};
12  \draw[linie] (up1)--(a1);
13
14  \draw[linie] (a1) |- ($(e1.north) + (0,0.5)$);
15  \draw[linie] (ea1)--(e1);
16 \end{tikzpicture}

```



16. Physik

Das Paket `Physik` bindet die Pakete `units`, `circuitikz` und `mhchem`. ein, so dass Formeln und Schaltpläne gesetzt werden können. Außerdem werden in diesem Modul Standardkonstanten für Formeln definiert.

Durch das Einbinden des Pakets `ziffer` wird dafür gesorgt, dass das im deutschen gebräuchliche Komma bei Zahlen richtig gesetzt wird.

16.1. Konstanten

Sehr regelmäßig müssen in Formeln physikalische Konstanten eingesetzt werden. Einige werden im Paket definiert, um die Eingabe zu erleichtern.

Eingeführt in
Version 0.8.3

\elementarladung

Die Elementarladung mit $1,602 \cdot 10^{-19}$ C.

\planckscheEV

Das plancksche Wirkungsquantum mit $4,1357 \cdot 10^{-15}$ eVs.

\planckscheJ

Das plancksche Wirkungsquantum mit $6,626 \cdot 10^{-34}$ Js.

\elektronenmasse

Die Masse eines Elektrons mit $9,109 \cdot 10^{-31}$ kg.

\protonenmasse

Die Masse eines Protons mit $1,673 \cdot 10^{-27}$ kg.

\lichtgeschwindigkeit

Die Lichtgeschwindigkeit mit $2,9979 \cdot 10^8$ m/s.

\rydbergfrequenz

Die Rydberg-Frequenz mit $3,28984 \cdot 10^{15}$ Hz.

```
1 \[ v = \sqrt{\frac{2 \cdot U \cdot Q}{m}} \]
2 \[ v = \sqrt{\frac{2 \cdot \unit[15000]{V} \cdot \elementarladung \cdot \protonenmasse}} = \unitfrac[1,70 \cdot 10^6]{m}{s} \]
```

$$v = \sqrt{\frac{2 \cdot U \cdot Q}{m}}$$

$$v = \sqrt{\frac{2 \cdot 15000 \text{ V} \cdot 1,602 \cdot 10^{-19} \text{ C}}{1,673 \cdot 10^{-27} \text{ kg}}} = 1,70 \cdot 10^6 \text{ m/s}$$

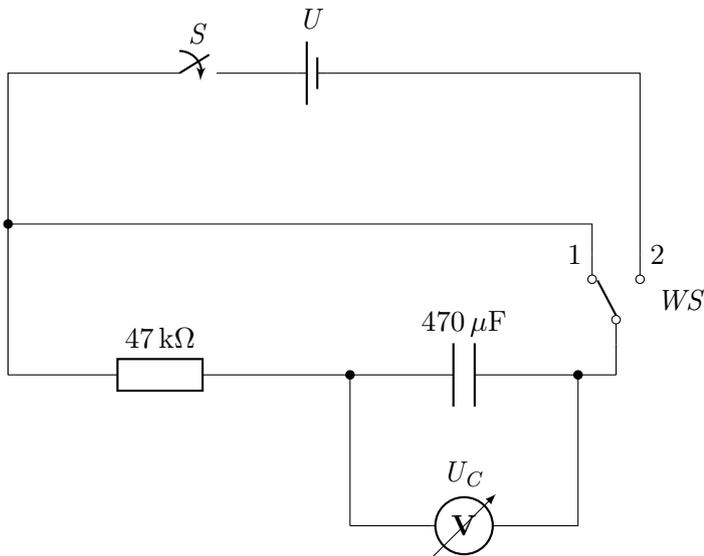
16.2. Schaltpläne

An dieser Stelle ein Beispiel für einen Schaltplan mit circuitkz:

```

1 \begin{circuitikz}
2   \draw
3     (0,0)--(1,0) to[european resistor,l=47k\Omega] (3,0)--(5,0)
4     to[C,l=470\muF] (7,0) -- (8,0)
5     (4.5,0) to[short,-*] (4.5,0) -- (4.5,-2)
6     (4.5,-2) -- (5,-2) to[voltmeter,l=U_C] (7,-2) -- (7.5,-2)
7     (7.5,-2) to[short,-*] (7.5,0)
8     (8,1) node[spdt,rotate=90] (Ums) {}
9     (Ums) node[right=0.4cm] {WS}
10    (Ums.out 1) node[left] {1}
11    (Ums.out 2) node[right] {2}
12    (0,0) |- (2,4) to[closing switch,l=SS] (3,4) to[battery1,
13    l=US] (5,4) -| (Ums.out 2)
14    (Ums.in) -- (8,0)
15    (Ums.out 1) |- (0,2) to[short,-*] (0,2)
16  ;
17 \end{circuitikz}

```



17. Geschichte

Das Fachmodul [Geschichte](#) bindet das Paket `biblatex` mit den Einstellungen für die humanwissenschaftliche Zitierweise ein. Des weiteren werden Befehle für durchnummerierte Quellen, Materialien und Verfassertexte zur Verfügung gestellt.

17.1. Befehle

`\material[Ebene]{Titel}` Voreinstellung: `\subsection`

Erzeugt eine Überschrift, mit der Material markiert werden kann. Dazu wird am rechten Rand ein M mit einer fortlaufenden Nummer gesetzt. Standardmäßig ist als Überschriftsebene `\subsection` gesetzt, dass über den optionalen Parameter geändert werden kann.

`\quelle[Ebene]{Titel}` Voreinstellung: `\subsection`

Erzeugt eine Überschrift, mit der eine Quelle markiert werden kann. Dazu wird am rechten Rand ein Q mit einer fortlaufenden Nummer gesetzt. Standardmäßig ist als Überschriftsebene `\subsection` gesetzt, dass über den optionalen Parameter geändert werden kann.

`\vt[Ebene]{Titel}` Voreinstellung: `\subsection`

Erzeugt eine Überschrift, mit der ein Verfassertext markiert werden kann. Dazu wird am rechten Rand ein VT mit einer fortlaufenden Nummer gesetzt. Standardmäßig ist als Überschriftsebene `\subsection` gesetzt, dass über den optionalen Parameter geändert werden kann.

Die so erstellten Textabschnitte können mit `\nameref{sec:$REFERENZ_ART$NUMMER}` referenziert werden, z. B. per `\nameref{sec:vt1}`. Für weitere Hinweise siehe Bsp. 9, S. 59.

Teil IV.

Dokumenttypen

Dokumenttypen dienen dazu, Vorgaben für spezielle Arten von Dokumenten zu machen und entsprechende Makros bereitzustellen. Üblicherweise wird eine bestimmte Dokumentklasse für die Verwendung empfohlen, grundsätzlich sind die Dokumenttypen aber unabhängig von der verwendeten Klasse.

Die Typen werden über die Paketoption `typ` geladen. Ist der Typ unbekannt, wird ein Arbeitsblatt gesetzt und der die Bezeichnung des Dokuments wird auf den angegebenen Typ eingestellt.

Wird kein Typ angegeben, ist der Kompatibilitätsmodus zum alten Schule-Paket aktiv. Es wird folglich kein Dokumenttyp geladen. Ebenso wenig wird ein Dokumenttyp geladen, wenn der Typ auf `typ = {ohne}` gesetzt wird, da sich das Paket dann im eingebetteten Modus befindet, also kein eigenständiges Dokument gesetzt werden soll.

Somit lässt sich `schule` auch in Präsentationen mit `beamer` setzen, siehe Bsp. 6, S. 58.

18. Arbeitsblatt

Dieser Dokumenttyp ist der Standard des Schulepakets. Wird ein unbekannter Dokumenttyp verwendet, wird stattdessen ein Arbeitsblatt gesetzt. Um gezielt das Arbeitsblatt zu verwenden ist als Typ `ab` anzugeben.

Die empfohlene Dokumentklasse ist `scartcl`.

Die Vorgaben des Dokumenttyps definieren Kopf- und Fußzeilen in der üblichen Darstellung des Schule-Pakets. Darüber hinaus werden keine weiteren Vorgaben gemacht.

19. Klausur

Dieser Dokumenttyp wird für Klausuren und Klassenarbeiten verwendet. Für die Verwendung der Klausur oder Klassenarbeit ist als Typ `k1` anzugeben.

Die empfohlene Dokumentklasse ist `scartcl`.

Die Vorgaben des Dokumenttyps definieren Kopf- und Fußzeilen in der üblichen Darstellung des Schule-Pakets. Außerdem werden Namens- und Datumfelder erzwungen.

19.1. Paketoptionen

`klausurtyp = klausur|klasse|kurs` Voreinstellung: `klausur`
legt fest, ob die Klausur als Klausur (Standard), Kursarbeit (`klausurtyp = {kurs}`) oder Klassenarbeit (`klausurtyp = {klasse}`) bezeichnet wird.

20. Leitprogramm

Als Leitprogramm wird eine Grundlage für den Unterricht bezeichnet, mit dem Schülerinnen und Schüler sich ein größeres Thema erarbeiten können. Ein Leitprogramm enthält dafür erklärende Texte sowie Aufgaben mit Hinweisen und Lösungen. Diese werden von Elementen werden von den Lernenden selbstständig gelesen und bearbeitet. Zum Abschluss eines Kapitels gehört in der Regel ein Kapiteltest. Dieses holen sich die Schülerinnen und Schüler bei der Lehrkraft ab um ihn zu bearbeiten und ihn anschließend direkt von der Lehrkraft kontrollieren zu lassen. Dieser Test wird dabei nur auf Grundlage des Erlernten und ohne direktes Hinzunehmen des Leitprogramms absolviert.

Der Dokumententyp Leitprogramm, als Typ ist `leit` anzugeben, stellt die layouttechnischen Grundlagen bereit und sorgt für die Verknüpfungen zwischen den Aufgaben und den dazugehörigen Hinweisen und Lösungen. Der Dokumententyp lässt sich aber auch für ein Skript nutzen, dass aus verschiedenen Kapiteln besteht. Die empfohlene Dokumentklasse ist `scrreprt`. Ein Beispiel ist unter Bsp. 5, S. 57 aufgeführt.

20.1. Paketoptionen

Beim Leitprogramm werden standardmäßig von der Aufgabe Links zu möglichen vorhandenen Lösungen oder Bearbeitungshinweisen gesetzt. Da diese Schaltflächen auch angezeigt werden, wenn die Lösungen bzw. Hinweise nicht eingebunden wurden, kann die Anzeige über Paketoptionen ausgeschaltet werden.

`hinweisLinkVerbergen`

verbirgt Links bei der Aufgabe zu möglichen Bearbeitungshinweisen.

`loesungLinkVerbergen`

verbirgt Links bei der Aufgabe zu möglichen Lösungen.

20.2. Befehle

`\TextField{<Höhe>}`

Erstellt ein Formularfeld mit der angegebenen Höhe und der aktuellen Spaltenbreite. Mit passenden Anzeigeprogrammen kann dann an dieser Stelle im PDF-Dokument Text eingegeben werden.

`\monatWort{<Monatszahl>}`

Übersetzt den als *Zahl* angegeben Monat in den deutschen Namen. Sollte die Zahl nicht erkannt werden, wird »unbekannter Monat« ausgegeben.

`\uebungBild`

Erstellt ein Symbol für eine Übung, dass allen Aufgaben innerhalb eines Leitprogramms vorangestellt wird.

`\hinweisBild`

Erstellt ein Symbol für ein Hinweis.

20.3. Umgebungen

`\begin{hinweisBox}`

Erzeugt eine optisch hervorgehobene Box, die mit dem Symbol für einen Hinweis gekennzeichnet ist.

21. Lernzielkontrolle

Dieser Dokumenttyp wird für Lernzielkontrollen verwendet. Für die Verwendung der Lernzielkontrolle ist als Typ `lzk` anzugeben.

Die empfohlene Dokumentklasse ist `scartcl`.

Die Vorgaben des Dokumenttyps definieren Kopf- und Fußzeilen in der üblichen Darstellung des Schule-Pakets. Außerdem werden Namens- und Datumsfelder erzwungen.

22. Übungsblatt

Dieser Dokumenttyp wird für Übungsblätter verwendet. Der Hauptunterschied zum Typ »Arbeitsblatt« liegt darin, dass Aufgaben als »Übungen« bezeichnet werden. Für die Verwendung des Übungsblatts ist als Typ `ueb` anzugeben.

Die empfohlene Dokumentklasse ist `scartcl`.

Die Vorgaben des Dokumenttyps definieren Kopf- und Fußzeilen in der üblichen Darstellung des Schule-Pakets. Darüber hinaus werden keine weiteren Vorgaben gemacht.

23. Unterrichtsbesuch

Dieser Dokumenttyp dient als Grobvorlage für Unterrichtsbesuche. Eine komplette Vorlage wird nicht angeboten, da die Studienseminare unterschiedliche Anforderungen stellen und es auch in den einzelnen Seminaren sehr häufig Änderungen an den layouttechnischen Aspekten gibt. Die Hauptanwendungen dieses Dokumenttyps sind daher Unterrichtsbesuche, bei denen es keine festen Vorgaben gibt, wie z. B. bei Revisionen oder der Materialsammlung für Informatik, die vollständig dieses \LaTeX -Paket nutzt. Für die Verwendung dieses Dokumententyps ist `ub` anzugeben.

Als Dokumentklasse wird für diesen Typ `scartcl` empfohlen. Darin wird durch den Dokumenttyp Kopf- und Fußzeile gesetzt, sowie eine Titelseite erzeugt, die mit Angaben gefüllt wird, die für einen Unterrichtsbesuch typisch sind und entsprechend angegeben werden müssen.

23.1. Befehle für Angaben zum Unterrichtsbesuch

Mit den folgenden Befehlen werden Angaben gesetzt, die auf der Titelseite des Unterrichtsbesuch angezeigt werden.

`\besuchtitel{<Titel>}`

setzt den Eintrag, um was für eine Art es sich bei dem Unterrichtsbesuch handelt. Dieses kann z. B. sein: „2. Unterrichtsbesuch im Fach Informatik“.

`\lehrer{<Lehrername>}`

setzt den Namen des Lehrers, der neben der Titelseite auch im Seitenkopf angezeigt wird.

`\schulform{<Schulform>}`

setzt den Eintrag für die Schulform wie z. B. Gesamtschule.

`\lerngruppe[<Kurzform der Lerngruppe>] {<Name der Lerngruppe>} {<Anzahl weiblich>} {<Anzahl männlich>}`

sorgt dafür, dass die Angaben zur Lerngruppe gesetzt werden. Der Name wird auf dem Titelblatt und im Seitenkopf angegeben, außer die optionale Möglichkeit der Kurzform wurde genutzt. In diesem Fall wird die Kurzform im Seitenkopf angegeben. Aus der Anzahl der weiblichen und männlichen Schülerinnen und Schüler wird automatisch die Gesamtzahl bestimmt, daher sind für diese Angaben nur Zahlen erlaubt.

`\zeit{<Startzeit>} {<Endzeit>} {<Stunde>}`

bietet die Möglichkeit, die Zeiten der Besuchsstunde anzugeben. Neben der Uhrzeit des Beginns und des Endes muss angegeben werden, um welche Stunde es sich an dem entsprechenden Tag handelt.

`\schule{<Name der Schule>}`

hierüber lässt sich der Name der Schule angeben, der auf der Titelseite angezeigt wird.

`\raum{<Raumbezeichnung>}`

bietet die Möglichkeit die Bezeichnung des Raumes anzugeben, in dem die Besuchsstunde stattfinden soll.

24. Folie

Bei der Nutzung des Dokumenttyps Folie wird eine Seite mit wenig Rand zur Verfügung gestellt, der die Fuß und Kopfzeile fehlt. So kann möglichst viel auf eine Folie gedruckt werden, wenn diese im Unterricht zum Einsatz kommen soll. Um den Dokumenttyp verwenden zu können muss `folie` als Typ angegeben werden.

25. Beurteilung

Bei der Nutzung des Dokumenttyps Beurteilung werden einige Makros geladen, so dass Beurteilungen von Lehramtsanwärterinnen und Lehramtsanwärtern bzw. Lehrkräften in Ausbildung einfach gesetzt werden können. U. a. wird dazu die Titelseite entsprechend gestaltet und Umgebungen für die Handlungsfelder vorformatiert.

25.1. Paketoptionen

Die folgenden Optionen sollten in der Präambel dem Schule-Paket mitgegeben werden.

beurteilung

Gibt den Namen des Beurteilenden an, z. B. `beurteilung={Marine Musterfrau}`.

ref

Gibt den Namen des Beurteilten an, z. B. `ref={Sabine Musterref}`.

zeitraum

Gibt den Zeitraum des Gutachtens an (nicht nur die einzelnen gesehenen Unterrichte), z. B. `zeitraum={02.02.19-03.05.19}`.

schulname

Gibt den Namen der Schule an.

schullogo

Lädt das Schullogo auf die Titelseite rechts oben. Das Schullogo sollte L^AT_EX als `logo.pdf` vorliegen.

schulanschrift

Wenn ein Schullogo gesetzt wird, kann zusätzlich der Name der Schule,

schulstr

die Straße der Schule und

schulort

der Ort der Schule ausführlich unter das Logo geschrieben werden.

lehramt

Gibt an, um welches Lehramt es sich handelt, z. B. GyGe.

25.2. Befehle

`\setzeGrundlagen{Termine}`

Über den Mechanismus können auf der Titelseite Einträge in die Tabelle zum hospitierten bzw. gegebenen Unterricht eingetragen werden. Dabei werden drei Spalten erwartet. Z. B.:

`\setzeGrundlagen{02.04.-01.05.2019 & 9b & Themen}`

25.3. Umgebungen

`\begin{handlungsfeld1}`

`\begin{handlungsfeld2}`

25. Beurteilung

```
\begin{handlungsfeld3}
```

```
\begin{handlungsfeld4}
```

```
\begin{handlungsfeld5}
```

```
\begin{handlungsfeld6}
```

In den Umgebungen `handlungsfeldX` können die Beurteilungen entsprechend zu den Handlungsfeldern geschrieben werden. Jedes Handlungsfeld wird dazu in einer `tcolorbox` gesetzt. Die zugehörigen Kompetenzen und Titel werden passend zugeordnet. Optional können der Umgebung weitere Optionen für die `tcolorbox` mitgegeben werden:

```
\begin{handlungsfeld1}[colback=red].
```

Teil V.

Zusatzpakete

26. Nutzung der Zusatzpakete

Die Zusatzpakete sind normale L^AT_EX-Pakete und können auch so eingesetzt werden. Allerdings erfolgt die Nutzung innerhalb des Schule-Pakets in der Regel über die Einbindung in Modulen oder Fachmodulen. Dennoch ist die direkte Einbindung möglich, auch unabhängig vom Schule-Paket.

27. Schaltungen mit Relais

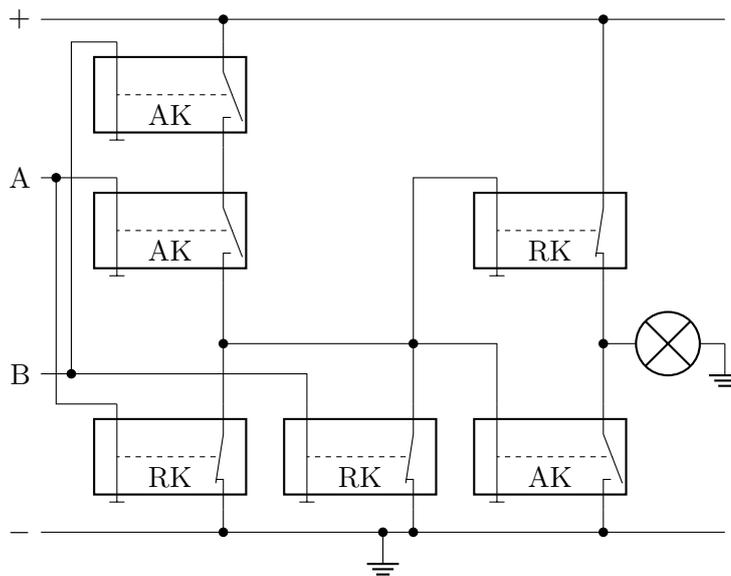
Durch das Paket `relaycircuit` ist es möglich Schaltungen mit Relais zu zeichnen. Dazu wird die neue Knotenform `relais` deklariert, die sich in *arbeits relais* (Bezeichnung: AK) und *ruhe relais* (Bezeichnung: RK) aufteilen. So kann der Schaltplan eines logischen NAND mittels Relais wie folgt gesetzt werden:

```
1 \begin{tikzpicture}
2   \draw (0,6.8) node [left] {\(+\)} -- (9,6.8);
3   \draw (0,0) node [left] {\(-\)} -- (9,0);
4   \draw (4.5,0) to[short, *-] (4.5,0) node [ground] {};
5
6   \draw (7.4,2.5) to[short,*-] (7.5,2.5) to[lamp] (9,2.5)
7     node[ground] {};
8
9   \draw (2.5,5.8) node[arbeits relais] (a1) {};
10  \draw (2.5,4) node[arbeits relais] (a2) {};
11  \draw (2.4,6.8) to[short,*-] (a1.anschluss);
12  \draw (a1.ausgabe) -- (a2.anschluss);
13
14  \draw (2.5,1) node[ruhe relais] (r1) {};
15  \draw (a2.ausgabe) -- (r1.anschluss);
16  \draw (r1.ausgabe) to[short,-*] (2.4,0);
17  \draw (5,1) node[ruhe relais] (r2) {};
18  \draw (r2.ausgabe) to[short,-*] (4.9,0);
19
20  \draw (7.5,1) node[arbeits relais] (a3) {};
21  \draw (7.5,4) node[ruhe relais] (r3) {};
22  \draw (a3.anschluss) -- (r3.ausgabe);
23  \draw (a3.ausgabe) to[short,-*] (7.4,0);
24  \draw (r3.anschluss) to[short,-*] (7.4,6.8);
25
```

```

26 \draw (2.4,2.5) to[short,*-] (4.9,2.5) -| (a3.eingabe);
27 \draw (r2.anschluss) |- (r3.eingabe);
28
29 \draw (0,4.7) node [left] {A} to[short,*-] (0.2,4.7) --
30 (a2.eingabe);
31 \draw (0.2,4.7) |- (r1.eingabe);
32
33 \draw (0,2.1) node [left] {B} to[short,*-] (0.4,2.1) -|
34 (r2.eingabe);
35 \draw (0.4,2.1) |- (a1.eingabe);
36 \end{tikzpicture}

```



28. Das alte Schule-Paket

Bei der mit Version 0.7 verbundenen Neuentwicklung wurde das alte Schule-Paket in `schulealt` umbenannt und die Dokumentklassen entsprechend angepasst.

Um zu vermeiden, dass alle alten Dokumente nicht mehr gesetzt werden können, wird das alte Paket mitgeliefert. Zudem wurde ein Kompatibilitätsmodus implementiert, der dafür sorgt, dass das Einbinden des Schule-Pakets ohne die Paketoption `typ` dazu führt, dass das Laden des neuen Pakets frühzeitig unterbrochen und stattdessen das alte Paket geladen wird. Somit sollten alle alten Dokumente unverändert gesetzt werden können.

Für die Befehle des alten Schule-Pakets kann in der Doku nachgesehen werden, die dem Zusatzpaket `schulealt` beiliegt.

⚠ Achtung: Wahrscheinlich wird das alte Paket in der Zukunft entfernt werden. Es sollte also keinesfalls für neue Dokumente verwendet werden.

Teil VI.

Häufig gestellte Fragen

29. Formatierung

29.1. Kann ich ein anderes Papierformat als A4 verwenden?

Ja! Denn dies wird von der Dokumentenklasse und nicht vom Schule-Paket festgelegt, z. B. ein A5-Blatt im Querformat:

```
1 \documentclass[a5paper,landscape]{scrartcl}
```

29.2. Kann ich die Seitenränder festlegen?

Ja! Dazu kann einfach das übliche Paket `geometry` im Dokument genutzt werden, z. B.:

```
1 \usepackage[
2   left=3cm,
3   right=2cm,
4   top=2cm,
5   bottom=2cm,
6   footskip=1cm
7 ]{geometry}
```

29.3. Ist es möglich, in den erstellten Materialien Schreibschriften zu verwenden?

Ja! Das ist kein Problem, allerdings ist das in \LaTeX bereits so einfach, dass das Schule-Paket hier keine abweichenden Funktionen implementiert.

Zu empfehlen ist hier das Paket `schulschriften`. Dieses bringt Fonts für die folgenden Schreibschriften mit:

wesu Sütterlinschrift (1911)

wedn Deutsche Normalschrift (1941)

wela lateinische Ausgangsschrift (1953)

wesa Schulausgangsschrift (1968, ehem. DDR)

29. Formatierung

weva Vereinfachte Ausgangsschrift (1972)

Diese können einfach verwendet werden, indem das jeweilige Paket, z. B. **weva**, eingebunden wird. Danach kann die zugehörige Schrift verwendet werden

`1 \weva Vereinfachte Ausgangsschrift`

Vereinfachte Ausgangsschrift

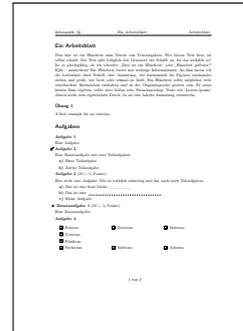
Teil VII.

Beispiele

30. Minimalbeispiele

Beispiel 1 »Arbeitsblatt« [T_EX] [PDF]

```
1 \documentclass[a4paper]{scrartcl}
2 \usepackage[
3     typ=ab,
4     fach=Informatik,
5     lerngruppe=9g,
6     loesungen=seite,
7     module={Symbole,Bewertung},
8 ]{schule}
9
10 % Dieses Dokument gehört zu den Beispiel des LaTeX
    Paketes Schule und ist von den Autoren
```



Beispiel 2 »Informationsblatt« [T_EX] [PDF]

```
1 \documentclass[a4paper]{scrartcl}
2 \usepackage[
3     typ=ib,
4     fach=Informatik,
5     lerngruppe=9g,
6 ]{schule}
7
8 % Dieses Dokument gehört zu den Beispiel des LaTeX
    Paketes Schule und ist von den Autoren
9 % des Pakets erstellt worden.
10 %
```



Beispiel 3 »Klausur« [T_EX] [PDF]

```

1 \documentclass[a4paper]{scrartcl}
2 \usepackage[
3   typ=kl,
4   klausurtyp=klausur,
5   fach=Informatik,
6   lerngruppe=9g,
7   loesungen=seite,
8   erwartungshorizontAnzeigen,
9 ]{schule}

```

Beispiel 4 »Klassenarbeit« [T_EX] [PDF]

```

1 \documentclass[a4paper]{scrartcl}
2 \usepackage[
3   typ=kl,
4   klausurtyp=klasse,
5   fach=Deutsch,
6   lerngruppe=1g,
7   loesungen=seite,
8   erwartungshorizontAnzeigen,
9   erwartungshorizontStil=simpel,
10 ]{schule}

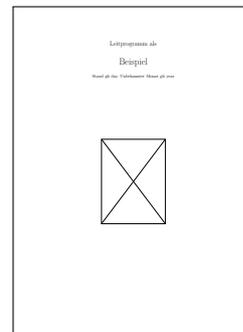
```

Beispiel 5 »Leitprogramm« [T_EX] [PDF]

```

1 \documentclass[12pt,a4paper,parskip=half,
2   chapterprefix,bibliography=totoc,footheight=57pt,
3   numbers=noendperiod]{scrreprt}
4 \usepackage[
5   typ=leit,
6   fach=Informatik,
7   farbig,
8   module={Lizenzen},
9   lizenz={cc-by-nc-sa-4},
10 ]{schule}
11 % Dieses Dokument gehört zu den Beispiel des LaTeX
12 % Paketes Schule und ist von den Autoren

```



Beispiel 6 »Beamerpräsentation« [T_EX] [PDF]

```

1 \documentclass{beamer}
2 \usepackage[
3   fach=Informatik,
4   typ={ohne},
5   farbig,
6   loesungen=folgend,
7   module={
8     Aufgaben,
9   },
10 ]{schule}

```



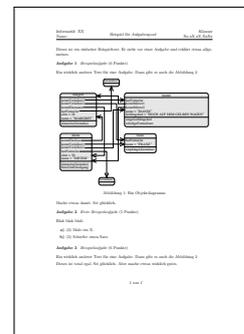
30.1. Nutzung von Modulen

Beispiel 7 »Klausur mit Nutzung des Aufgabenpools« [T_EX] [PDF]

```

1 \documentclass[a4paper,parskip=half]{scrartcl}
2 \usepackage[utf8]{inputenc}
3 \usepackage[
4   typ=kl,
5   klausurtyp=klausur,
6   fach=Informatik,
7   lerngruppe=XX,
8   erwartungshorizontAnzeigen,
9   kmkPunkte,
10  loesungen=seite,

```

Beispiel 8 »Klausur mit Erwartungshorizont mit Einzeltabellen« [T_EX] [PDF]

```

1 \documentclass[a4paper]{scrartcl}
2 \usepackage[
3   typ=kl,
4   klausurtyp=klausur,
5   fach=Informatik,
6   lerngruppe=9g,
7   loesungen=seite,
8   erwartungshorizontAnzeigen,
9   erwartungshorizontStil=einzeltabellen,
10 ]{schule}

```



31. Praxisbeispiele

31.1. Geschichte

Beispiel 9 »Beispieldokument Geschichte« [TeX] [PDF]

```

1 \documentclass[]{\scrartcl}
2 \usepackage[
3   typ=ab,
4   fach=Geschichte,
5   lerngruppe=9g,
6   loesungen=folgend,
7   module={Symbole,Texte},
8   sprache={english,french},
9 ]{schule}
10 \usepackage{blindtext}

```



Beispiel 10 »Informationsblatt Hieroglyphen« [TeX] [PDF]

```

1 \documentclass[a4paper]{scrartcl}
2 \usepackage[
3   fach=Geschichte,
4   lerngruppe={6a},
5   typ=ib,
6   seitenzahlen=keine,
7   loesungen=keine,
8 ]{schule}
9
10 % Dieses Dokument gehört zu den Beispiel des LaTeX
    Paketes Schule und ist von den Autoren

```



31.2. Informatik

Beispiel 11 »Arbeitsblatt Fahrkartenauskunft« [TEX] [PDF]

```

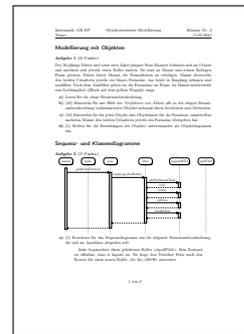
1 \documentclass[12pt,a4paper]{scrartcl}
2 \usepackage[
3     typ=ab,
4     fach=Informatik,
5     lerngruppe=EF,
6 ]{schule}
7
8 % Dieses Dokument ist in der Zusammenarbeit von
9 % verschiedenen
10 % Informatikreferendaren und Informatiklehrern
    entstanden.
11 % Der Herausgeber dieses Dokuments ist die
    Fachgruppe Informatische
    
```



Beispiel 12 »Klausur Objektorientierte Modellierung« [TEX] [PDF]

```

1 \documentclass[a4paper]{scrartcl}
2 \usepackage[
3     fach=Informatik,
4     lerngruppe={GK EF},
5     typ=kl,
6     klausurtyp=klausur,
7     nummer=2,
8     farbig,
9     datumAnzeigen,
10    seitenzahlen=autoGesamt,
    ]{schule}
    
```

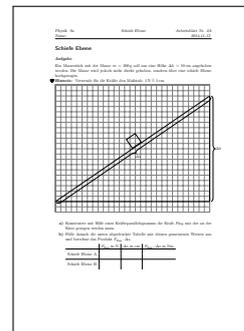


31.3. Physik

Beispiel 13 »Arbeitsblatt Schiefe Ebene« [TEX] [PDF]

```

1 \documentclass[a4paper]{scrartcl}
2 \usepackage[
3     typ=ab,
4     fach=Physik,
5     lerngruppe=9a,
6     nummer=2A,
7     datumAnzeigen,
8     namensfeldAnzeigen,
9 ]{schule}
    
```



Teil VIII.

Entwicklungsdokumentation

32. Lizenzen

Es ist erlaubt, diese Software unter den Bedingungen der L^AT_EX Project Public License (LPPL), Version 1.3c oder später, zu kopieren und zu verteilen (<http://www.latex-project.org/lppl.txt>). Sie hat den Status “maintained.”

Einzelne Code-Beispiele in dieser Dokumentation stammen aus der Materialsammlung Informatik² und unterliegen damit der CC-BY-NC-SA (Creative Commons Attribution-NonCommercial-ShareAlike License).

33. Richtlinien

Um die Quelltexte des Pakets über einen längeren Zeitraum halbwegs konsistent und dokumentiert zu halten und somit die leichtere Einarbeitung neuer Betreuer zu ermöglichen, wurden folgende Richtlinien vereinbart:

- Bei sämtliche Klassen- und Paketquellen sollten lange Zeilen vermieden werden.
- Bei der Dokumentation (`dokumentation.tex`) soll kein automatischer Umbruch im Dokument stattfinden, um Änderungen im Text einfacher nachhalten zu können.
- Auf Einrückungen sollte geachtet werden. Zeilenumbrüche sind ggf. entsprechend auszukommentieren.
- Einrückungen erfolgen mit Leerzeichen in einer Weite von 4.
- Alle internen Makros und Variablen sollen im Namensraum `schule@` stehen, z. B. `schule@ergebnishorizontAnzeigen`.
- Sprechende Bezeichner sollten verwendet werden.
- Für neue Funktionen sollte ein neues Modul oder ein neuer Dokumenttyp angelegt werden, sofern es sich um keine klare Ergänzung handelt.
- Neue Funktionen und Änderungen sind in der Dokumentation mit den Mitteln von `cnltx-doc` zu kennzeichnen, z. B. mit `\changedversion{<version>}` für Änderungen oder `\sinceversion{<version>}` für neue Funktionen. Außerdem sind sie im Changelog zu vermerken.

2. <http://ddi.uni-wuppertal.de/material/materialsammlung/index.html>

34. Modularität

34.1. Erläuterungen zum Modulsystem

Ein zentrales Problem des alten Schule-Pakets bis Version 0.6 war, dass es sehr monolithisch aufgebaut und alles integrierte, was im Schulalltag und in der Lehrerausbildung nützlich sein könnte. So wurden die Weiterentwicklung und Übergabe an neue Maintainer schwierig, da stets eine Einarbeitung in alle Bereiche erforderlich war.

Die grundlegende Entscheidung für das neue Schule-Paket ist es also, eine Modularisierung zu etablieren, die, zusätzlich zu einem stabilen Kern, verschiedene Funktionen und Fachspezifika entsprechend kapselt und von diesem Kern trennt.

Ein eindeutiger Aufbau dieser Module soll dafür sorgen, dass es leichter wird, neue Funktionen zu ergänzen, ohne den gesamten Quelltext des Pakets kennen und verstehen zu müssen.

Grundsätzlich gibt es drei verschiedene Arten von Modulen im Schule-Paket: **Module**, **Fachmodule** und **Zusatzpakete**, vgl. Abschnitt 1.3, S. 7.

Erweiterungen ohne direkten Bezug (also alles, das auch ohne das Schule-Paket sinnvoll genutzt werden kann) zum Paket sollten in Form unabhängiger Zusatzmodule implementiert werden.

34.2. Aufbau eines Moduls

Ein Modul für das Schule-Paket besteht aus mehreren Dateien, deren erster Teil des Namens `schule.mod.Modulname` identisch ist. Je nach Funktion werden dort drei Möglichkeiten angehängt, die als Abschnitte bezeichnet werden. Sobald eine der Dateien mit diesem Schema vorhanden ist, kann das Modul über seinen Namen eingebunden werden, vgl. Abschnitt II, S. 10. Alle Dateien vorgegebenen Dateien liegen dazu im Verzeichnis »latex«.

Der Modulmechanismus sorgt dafür, dass die entsprechenden Abschnitte des Moduls an den richtigen Stellen des Quelltextes eingebunden werden. Es sind die folgenden drei Abschnitte definiert:

`optionen.tex` → Paketoptionen des Moduls, vgl. Paket `pgfopts`

`pakete.tex` → Paketabhängigkeiten des Moduls

`code.tex` → Implementierung des Moduls

All diese Abschnitte sind optional und werden geladen, wenn sie vorhanden sind. Ein Modul kann also beispielsweise nur aus einer `code.tex`-Datei bestehen, wenn es nur einige Makros definiert.

34.2.1. Beispiel

schule.mod.HalloWelt.optionen.tex

```

1 % *****
2 % * Paketoptionen *
3 % *****
4
5 % Boolesche Optionen
6 % *****
7 \newboolean{schule@nutzeGoodbye}
8
9 % Standardwerte
10 % *****
11 \newcommand{\schule@weltname}{Welt}
12
13 % Definition der Paketoptionen
14 % *****
15 \pgfkeys{
16   /schule/.cd,
17   weltname/.store in=\schule@weltname,
18   nutzeGoodbye/.value forbidden,
19   nutzeGoodbye/.code=\setboolean{schule@datumAnzeigen}{true},
20 }

```

schule.mod.HalloWelt.pakete.tex

```

1 % *****
2 % * Paketabhängigkeiten *
3 % *****
4
5 \RequirePackage{ifthenelse}

```

schule.mod.HalloWelt.code.tex

```

1 % *****
2 % * Hallo Welt! *
3 % *****
4
5 \newcommand{\halloWelt}{
6   \ifthenelse{\boolean{schule@nutzeGoodbye}}{
7     Goodbye \schule@weltname!
8   }{
9     Hallo \schule@weltname!

```

```
10 }  
11 }
```

34.3. Aufbau eines Fachmoduls

Ein Fachmodul für das Schule-Paket wird analog zu einem normalen Modul erstellt, vgl. Abschnitt 34.2, S. 62. Hier steht nur im Dateinamen nach dem zweiten Punkt `fach` anstatt `mod`.

Die Dateien für das Fach Geschichte sind dann beispielsweise folgende:

- `schule.fach.Geschichte.optionen.tex`
- `schule.fach.Geschichte.pakete.tex`
- `schule.fach.Geschichte.code.tex`

34.4. Aufbau eines Dokumenttyps

Auch ein Dokumenttyp für das Schule-Paket hat den gleichen Aufbau wie ein normales Modul, vgl. Abschnitt 34.2, S. 62. Hier steht nur im Dateinamen nach dem zweiten Punkt `typ` anstatt `mod`.

Die Dateien für das Arbeitsblatt (Typ `ab`) sind dann beispielsweise folgende:

- `schule.typ.ab.optionen.tex`
- `schule.typ.ab.pakete.tex`
- `schule.typ.ab.code.tex`

35. Funktionen für Entwickler

35.1. Fehlerbehandlung und Debugging

35.1.1. Paketoptionen

`debug`

schaltet das Paket in den Debugmodus. Verhindert zudem die Unterdrückung ungefährlicher Warnungen, die standardmäßig aktiv ist.

Diese Paketoption setzt zudem die boolesche Variable `schule@debug` auf `true`, sodass jederzeit geprüft werden kann, ob der Debugmodus aktiv ist oder nicht.

35.1.2. Befehle

`\sinfo{⟨Text⟩}`

schreibt den angegebenen Text in die Logdatei.

`\swarnung{⟨Text⟩}`

schreibt den angegebenen Text als Warnung in die Logdatei.

`\sfehler{⟨Text⟩}`

schreibt den angegebenen Text als Fehler in die Logdatei und beendet die Kompilierung.

`\sdinfo{⟨Text⟩}`

schreibt den angegebenen Text in die Logdatei, falls der Debugmodus aktiv ist.

`\sdwarnung{⟨Text⟩}`

schreibt den angegebenen Text als Warnung in die Logdatei, falls der Debugmodus aktiv ist.

35.2. Interne Makros

35.2.1. Befehle

`\schule@kopfUmbruch`

liefert einen Zeilenumbruch, wenn die Kopfzeile an einer Stelle zweizeilig ist, z. B. durch die Darstellung eines Namensfelds oder die Anzeige des Datums.

`\schule@modulDateiLaden {⟨Kategorie⟩}{⟨Modulname⟩}{⟨Abschnitt⟩}`

Lädt einen Abschnitt eines Moduls. Interne Hilfsfunktion für das Laden von Modulen.

⚠ Achtung: Sollte nicht manuell verwendet werden, sondern nur an den entsprechenden Stellen der `schule.sty`.

⚠ Achtung: Kann mit viel Sorgfalt zur Erfüllung von Abhängigkeiten zwischen Modulen genutzt werden. Es ist darauf zu achten, dass dann zu Beginn der Abschnitte des ladenden Moduls die entsprechenden Abschnitte des zu ladenden Moduls eingebunden werden.

Dies führt dazu, dass die Ladereihenfolge der Module nicht mehr sichergestellt werden kann.

`\schule@modulNachladen{⟨Modulname⟩}`

Lädt ein Modul mit allen Abschnitten. Interne Hilfsfunktion für die Erfüllung von Abhängigkeiten in Modulen.

⚠ Achtung: Die Verwendung zu anderen Zwecken wird nicht empfohlen, da die Reihenfolge der zu ladenden Module hier nicht beachtet werden kann. Ebenfalls kann so nicht sichergestellt werden, dass der geladene Code an der richtigen Stelle im Quelltext landet, vielmehr wird er genau an der Stelle des Befehls geladen.

36. Changelog

Im Laufe der Jahre wurde das Paket immer wieder erweitert. Nicht nur die Anpassung an veränderte Anforderungen, etwa bei den Unterrichtsbesuchen, sondern auch neue Funktionalitäten fließen in das Paket ein. Die folgende Liste bietet eine Übersicht über die letzten Änderungen.

- **0.10.0** – 2024-04-01
 - Das Modul Aufgabenpool um Möglichkeiten für Anhänge erweitert
 - Einbinden von Erwartungen für den Erwartungshorizont flexibler gestaltet
 - Bugfix
- **0.9.0** – 2023-10-07
 - Modul Aufgabenpool ausgebaut, damit es variable einsetzbar ist
 - Aufgabentemplate auch für Lösungen setzbar gemacht
 - Reihenfolge beim Laden von Lizenzen geändert um Problemen mit hyperxmp besser aus dem Weg zu gehen
- **0.8.4** – 2023-03-19
 - Modul Aufgabenpool hinzugefügt
 - Option: Einbindung von hyperxmp kann bei Lizenzen unterbunden werden
 - Konstanten für das Fach Physik ergänzt
 - Abhängigkeit von mdframe entfernt
- **0.8.3** – 2021-08-09
 - Konstanten für das Fach Physik eingebaut
 - Einbindung von ziffer im Modul Physik
 - Notenübersicht ohne Tendenz
 - Umstellung der Linzenzen auf doclicense
- **0.8.2** – 2020-10-16
 - Anpassungen an neue Version von xsim
 - Modul Formulare begonnen
 - Punktübersicht als Ankerpunkt für die letzte Seite mit Inhalt ermöglicht
 - Überarbeitung der mcumgebung bzgl. Lösungen.
 - Auslagerung von syntaxdi und utfsym in extra Pakete
- **0.8.1.1** – 2018-09-06
 - Fix für die Benutzung mit beamer

37. ToDo

- **0.8.1** – 2018-08-22
 - Umbau auf flache Verzeichnistiefe für die Anforderungen von TeXLive
- **0.8.0** – 2018-08-12
 - Vollständiger Umbau von `exsheets` auf `xsim`
 - Modul »Bewertung« hinzugefügt
 - Dokumententyp »Leitprogramm« hinzugefügt
 - Dokumententyp »Folie« hinzugefügt
- **0.7.2** – 2017-01-29
 - Modul »Lizenzen« hinzugefügt
 - Fix: Bearbeitungshinweise konnten keine Makros enthalten
- **0.7.1** – 2017-01-08
 - Dokumentklassen verworfen
 - Dokumenttypen als Module implementiert
 - Bearbeitungshinweise zu Aufgaben hinzugefügt
 - Deklaration von `\aufgabeMC` und `\aufgabeLueckentext` für Hinweise angepasst **Inkompatibel** mit der bisherigen Schnittstelle
 - Ungenauigkeiten in der Doku zu Erwartungen behoben
- **0.7** – 2016-09-01
 - Vollständige Überarbeitung des Pakets

37. ToDo

Die folgende Liste soll die nächsten geplanten Funktionen bzw. Entwicklungsschritte angeben.

37.1. Must-have

- ...

37.2. Nice-to-have

- Weitere für die Schule nützliche Dokumenttypen integrieren, z. B. Lerntagebücher.
- Praxisbeispiel Klausur: Objekt- und Klassendiagramm mit TikZ setzen.
- TikZ-Stile nicht als Paketooptionen formatieren.
- Dokumentieren des Generierens von Klausuren aus Aufgabendatenbanken wie es mit `exsheets` möglich war.

37. *ToDo*

- Bessere Trennung von Inhalt und Formatierung (Modul Format sollte alle notwendigen Makros enthalten, Modul Aufgaben sollte sich nicht mehr um die Formatierung kümmern)

38. Index

A	
<code>\achtung</code>	21
<code>amsmath</code> (Paket)	6
<code>amssymb</code> (Paket)	7
<code>\anchormark</code>	37
<code>array</code> (Paket)	7
<code>aufgabe</code> (Umgebung)	11, 13, 20
<code>aufgabe*</code> (Umgebung)	11
<code>\aufgabeLueckentext</code>	67
<code>\aufgabeMC</code>	67
<code>Aufgaben</code>	8, 10, 18
<code>\aufgabeninput</code>	16
<code>Aufgabenpool</code>	16
<code>aufgabenpoolHinweis</code> (Umgebung)	17
<code>\author</code>	29
<code>\Autor</code>	29
B	
<code>babel</code> (Paket)	6
<code>beamer</code> (Klasse)	46
<code>beamer</code> (Paket)	66
<code>bearbeitungshinweis</code> (Umgebung)	13, 15
<code>\bearbeitungshinweisliste</code>	15
<code>\bearbeitungshinweisZuAufgabe</code>	15
<code>\begin</code>	50 f.
<code>\besuchtitel</code>	49
<code>beurteilung</code>	50
<code>Bewertung</code>	18
<code>biblatex</code> (Paket)	44
<code>\blank</code>	14
<code>bonus-points</code>	11
C	
<code>\changedversion</code>	61
<code>\chb</code>	21
<code>\choice</code>	14
<code>circuitikz</code> (Paket)	42 f.
<code>circuittikz</code> (Paket)	6
<code>cnltx-doc</code> (Paket)	61
<code>colortbl</code> (Paket)	7
<code>csquotes</code> (Paket)	7, 26
<code>ctable</code> (Paket)	7
<code>CTAN</code>	1
D	
<code>\dashuline</code>	21
<code>\date</code>	29
<code>\Datum</code>	29
<code>datumAnzeigen</code>	23
<code>debug</code>	64
<code>\diastring</code>	25
<code>doclicense</code> (Paket)	28, 66
<code>\dotuline</code>	21
E	
<code>\EK</code>	27
<code>\EKe</code>	27
<code>\EKen</code>	27
<code>\elektronenmasse</code>	43
<code>\elementarladung</code>	43
<code>\enquote</code>	25
<code>enumitem</code> (Paket)	7
<code>environ</code> (Paket)	6
<code>\erwartung</code>	20
<code>erwartungen</code> (Umgebung)	20
<code>\erwartungshorizont</code>	20
<code>erwartungshorizontAnzeigen</code>	18, 20
<code>erwartungshorizontKeineSeiten</code>	18
<code>erwartungshorizontStil</code>	18 f.
<code>\EuE</code>	27
<code>\EuEn</code>	27
<code>eurosym</code> (Paket)	7
<code>exsheets</code> (Paket)	67
F	
<code>\Fach</code>	29
<code>fach</code>	29, 37
<code>farbig</code>	20, 30
<code>\feldKar</code>	30
<code>\feldLin</code>	30
<code>\feldLinFormular</code>	26
<code>\feldMil</code>	30
<code>fontenc</code> (Paket)	6
<code>forarray</code> (Paket)	6
<code>Format</code>	10, 14
<code>Formulare</code>	26
G	
<code>gendering</code>	27
<code>geometry</code> (Paket)	54
<code>Geschichte</code>	44
<code>\getBasedir</code>	16, 18
<code>\GK</code>	27
<code>\GKe</code>	27
<code>\GKen</code>	27
<code>graphicx</code> (Paket)	6 f.
H	
<code>\hinweis</code>	21
<code>\hinweisBild</code>	47
<code>hinweisBox</code> (Umgebung)	48
<code>hinweisLinkVerbergen</code>	47
<code>hyperref</code> (Paket)	6, 26

INDEX

I	
<code>\ifAufgabenpoolOptionF</code>	17
<code>\ifAufgabenpoolOptionT</code>	17
<code>\ifAufgabenpoolOptionTF</code>	17
<code>ifthen</code> (Paket).....	6
<code>Informatik</code>	37
<code>inputenc</code> (Paket).....	6
<code>\inputOnce</code>	16
<code>\inputOnceAppendix</code>	18
K	
<code>klausurtyp</code>	46
<code>kmkPunkte</code>	19
<code>Kuerzel</code>	26
<code>\KuK</code>	28
<code>\Kurs</code>	29
L	
<code>lehramt</code>	50
<code>\lehrer</code>	49
<code>\Lerngruppe</code>	29
<code>\lerngruppe</code>	49
<code>lerngruppe</code>	29
<code>\lichtgeschwindigkeit</code>	43
<code>lineno</code> (Paket).....	7
<code>listings</code> (Paket).....	6
<code>lizenz</code>	28
<code>\lizenzName</code>	28
<code>\lizenzNameKurz</code>	28
<code>\lizenzSymbol</code>	28
<code>\LK</code>	27
<code>\Lke</code>	27
<code>\Lken</code>	27
<code>\Lkr</code>	27
<code>\Lkre</code>	27
<code>loesung</code> (Umgebung).....	13, 16, 20
<code>loesung*</code> (Umgebung).....	13
<code>loesungen</code>	13
<code>loesungLinkVerbergen</code>	47
<code>longtable</code> (Umgebung).....	19
<code>longtable</code> (Paket).....	7
<code>LPPL</code>	61
<code>\Lpr</code>	27
<code>\Lprn</code>	27
<code>\lstinputlisting</code>	15
<code>\luecke</code>	14
<code>\LuL</code>	28
<code>\LuLn</code>	28
M	
<code>marginnote</code> (Paket).....	6
<code>\material</code>	45
<code>\mcloesung</code>	14
<code>\mcrichtig</code>	14
<code>mcumgebung</code> (Umgebung).....	66
<code>mcumgebung</code> (Umgebung).....	14f.
<code>mehrspaltig</code> (Umgebung).....	34
<code>Metadaten</code>	10
<code>mhchem</code> (Paket).....	6, 42
<code>module</code>	10
<code>\monatWort</code>	47
<code>multicol</code> (Paket).....	7
<code>multirow</code> (Paket).....	7
N	
<code>namensfeldAnzeigen</code>	23
<code>\newthreadtwo</code>	39
<code>\nextlevel</code>	40
<code>\node</code>	41
<code>nohyperxmp</code>	28
<code>notenOhneTendenz</code>	19
<code>notenschema</code>	19
<code>\notenverteilung</code>	20
<code>\Nummer</code>	29
<code>nummer</code>	29
O	
<code>onceAppendix</code> (Umgebung).....	18
P	
<code>Papiertypen</code>	29
<code>\person</code>	21
<code>pgf-umlcd</code> (Paket).....	6
<code>pgf-umlsd</code> (Paket).....	6, 39
<code>pgfopts</code> (Paket).....	6, 62
<code>Physik</code>	42
<code>\planckscheEV</code>	43
<code>\planckscheJ</code>	43
<code>points</code>	11
<code>\prevlevel</code>	40
<code>\printsolutions</code>	13
<code>\protonenmasse</code>	43
<code>\punkteAufgabe</code>	11
<code>\punkteTotal</code>	11
<code>\punktuebersicht</code>	18
<code>\punktuebersicht*</code>	11
Q	
<code>\quelle</code>	45
R	
<code>\raum</code>	49
<code>ref</code>	50
<code>relaycircuit</code> (Paket).....	6f.
<code>\resetZeilenNr</code>	34
<code>\rydbergfrequenz</code>	43
S	
<code>schulanschrift</code>	50

INDEX

<code>\schule</code>	49	<code>\symBueroklammer</code>	31
<code>schule</code> (Paket)	6 f., 11, 46	<code>\symBueroklammern</code>	31
<code>\schule@kopfUmbruch</code>	65	<code>\symBuntstift</code>	31
<code>\schule@modulDateiLaden</code>	65	<code>\symBus</code>	34
<code>schulealt</code> (Paket)	6, 53	<code>\symCD</code>	32
<code>\schulform</code>	49	<code>\symDaumenHoch</code>	31
<code>schullogo</code>	50	<code>\symDaumenRunter</code>	31
<code>schulname</code>	50	<code>\symDenkblase</code>	31
<code>schulort</code>	50	<code>\symDreiSprechblasen</code>	31
<code>schulschriften</code> (Paket)	54	<code>\symFahrrad</code>	34
<code>schulstr</code>	50	<code>\symFueller</code>	31
<code>scrartcl</code> (Klasse)	13, 18, 46, 48	<code>\symFussball</code>	33
<code>scrreprt</code> (Klasse)	47	<code>\symFussgaenger</code>	34
<code>\sdinfo</code>	65	<code>\symGeodreieck</code>	31
<code>\sdwarnung</code>	65	<code>\symGlocke</code>	32
<code>\Seitenzahlen</code>	24	<code>\symHaken</code>	32
<code>seitenzahlen</code>	23 f.	<code>\symHandy</code>	33
<code>sequencediagram</code> (Umgebung)	39	<code>\symHeft</code>	31
<code>\setAufgabenpoolOptionen</code>	17	<code>\symHerz</code>	32
<code>\setBasedir</code>	16	<code>\symHerzAcht</code>	32
<code>setspace</code> (Paket)	7	<code>\symHerzAss</code>	32
<code>\setzeAufgabentemplate</code>	12	<code>\symHerzBube</code>	33
<code>\setzeGrundlagen</code>	50	<code>\symHerzDame</code>	33
<code>\setzeLoesungentemplate</code>	12	<code>\symHerzDrei</code>	32
<code>\setzeSymbol</code>	10	<code>\symHerzFuenf</code>	32
<code>\sfehler</code>	65	<code>\symHerzKoenig</code>	33
<code>\showIncludeAppendix</code>	17 f.	<code>\symHerzNeun</code>	32
<code>\sinceversion</code>	61	<code>\symHerzSechs</code>	32
<code>\sinfo</code>	65	<code>\symHerzSieben</code>	32
<code>\skaliereSequenzdiagramm</code>	39	<code>\symHerzVier</code>	32
<code>smalldescription</code> (Umgebung)	24	<code>\symHerzZehn</code>	33
<code>smallenumerate</code> (Umgebung)	24	<code>\symHerzZwei</code>	32
<code>smallitemize</code> (Umgebung)	24	<code>\symHockey</code>	33
<code>\so</code>	22	<code>\symKaro</code>	32
<code>sprache</code>	20	<code>\symKaroAcht</code>	33
<code>standalone</code> (Paket)	7	<code>\symKaroAss</code>	33
<code>struktex</code> (Paket)	6, 37, 40	<code>\symKaroBube</code>	33
<code>subtitle</code>	11	<code>\symKaroDame</code>	33
<code>\SuS</code>	27	<code>\symKaroDrei</code>	33
<code>\SuSn</code>	28	<code>\symKaroFuenf</code>	33
<code>\swarnung</code>	65	<code>\symKaroKoenig</code>	33
<code>\symApplaus</code>	31	<code>\symKaroNeun</code>	33
<code>\symAuge</code>	31	<code>\symKaroSechs</code>	33
<code>\symAugen</code>	31	<code>\symKaroSieben</code>	33
<code>\symAuto</code>	34	<code>\symKaroVier</code>	33
<code>\symBahn</code>	34	<code>\symKaroZehn</code>	33
<code>\symBaseball</code>	33	<code>\symKaroZwei</code>	33
<code>\symBasketball</code>	33	<code>\symKeineGlocke</code>	32
<code>\symBild</code>	31	<code>\symKeinHandy</code>	33
<code>\symBleistift</code>	31	<code>\symKlee</code>	32
<code>Symbole</code>	10	<code>\symKlemmbrett</code>	32
<code>\symBoot</code>	34	<code>\symKreuz</code>	32
<code>\symBuch</code>	31	<code>\symKreuzAcht</code>	33
<code>\symBuecher</code>	31	<code>\symKreuzAss</code>	33

INDEX

<code>\symKreuzBube</code>	33	<code>\symSmileyLachend</code>	32
<code>\symKreuzDame</code>	33	<code>\symSmileyNeutral</code>	32
<code>\symKreuzDrei</code>	33	<code>\symSmileySchlafend</code>	32
<code>\symKreuzFuenf</code>	33	<code>\symSmileyTraurig</code>	32
<code>\symKreuzKoenig</code>	33	<code>\symSmileyZwinkernd</code>	32
<code>\symKreuzNeun</code>	33	<code>\symSnowboard</code>	33
<code>\symKreuzSechs</code>	33	<code>\symSonne</code>	32
<code>\symKreuzSieben</code>	33	<code>\symSpielkarte</code>	32
<code>\symKreuzVier</code>	33	<code>\symSprechblase</code>	31
<code>\symKreuzZehn</code>	33	<code>\symStern</code>	32
<code>\symKreuzZwei</code>	33	<code>\symStrassenbahn</code>	34
<code>\symKuli</code>	31	<code>\symSurfen</code>	33
<code>\symLaufen</code>	33	<code>\symTennis</code>	33
<code>\symLineal</code>	31	<code>\symTheater</code>	33
<code>\symLupe</code>	32	<code>\symThermometer</code>	32
<code>\symMedaille</code>	33	<code>\symTischtennis</code>	33
<code>\symMikroskop</code>	31	<code>\symUhr</code>	32
<code>\symMond</code>	32	<code>\symVolleyball</code>	33
<code>\symMund</code>	31	<code>\symWuerfelDrei</code>	34
<code>\symNadel</code>	31	<code>\symWuerfelEins</code>	34
<code>\symNase</code>	31	<code>\symWuerfelFuenf</code>	34
<code>\symNote</code>	32	<code>\symWuerfelSechs</code>	34
<code>\symNoten</code>	32	<code>\symWuerfelVier</code>	34
<code>\symOhr</code>	31	<code>\symWuerfelZwei</code>	34
<code>\symPalette</code>	31	<code>\symZeigefinger</code>	31
<code>\symPik</code>	32	<code>\symZeitung</code>	32
<code>\symPikAcht</code>	32	<code>\symZielflagge</code>	33
<code>\symPikAss</code>	32	<code>\symZunge</code>	31
<code>\symPikBube</code>	32	<code>\symZweiSprechblasen</code>	31
<code>\symPikDame</code>	32	<code>syntaxdi (Paket)</code>	6, 37, 41, 66
<code>\symPikDrei</code>	32		
<code>\symPikFuenf</code>	32	T	
<code>\symPikKoenig</code>	32	<code>\teilaufgabe</code>	12
<code>\symPikNeun</code>	32	<code>teilaufgaben (Umgebung)</code>	12, 20
<code>\symPikSechs</code>	32	<code>\teilaufgabeOhneLoesung</code>	12
<code>\symPikSieben</code>	32	<code>\TextFeld</code>	47
<code>\symPikVier</code>	32	<code>\textluecke</code>	14
<code>\symPikZehn</code>	32	<code>tikz (Paket)</code>	6 f.
<code>\symPikZwei</code>	32	<code>tikzpicture (Umgebung)</code>	41
<code>\symPin</code>	31	<code>\Titel</code>	29
<code>\symPinsel</code>	31	<code>\titel</code>	29
<code>\symPokal</code>	33	<code>typ</code>	8, 10, 46, 53
<code>\symReiten</code>	33	U	
<code>\symRollstuhl</code>	34	<code>\uebungBild</code>	47
<code>\symSchere</code>	32	<code>ulem (Paket)</code>	7, 20
<code>\symSchiff</code>	34	<code>\uline</code>	22
<code>\symSchloss</code>	32	<code>uni-wtal-ger (Paket)</code>	6
<code>\symSchlossOffen</code>	32	<code>units (Paket)</code>	6, 42
<code>\symSchluessel</code>	32	<code>utfsym (Paket)</code>	7, 31, 66
<code>\symSchwebebahn</code>	34	<code>\uuline</code>	22
<code>\symSchwimmen</code>	33	<code>\uwave</code>	22
<code>\symSeilbahn</code>	34		
<code>\symSki</code>	33	V	
<code>\symSmileyGrinsend</code>	32	<code>\vt</code>	45

INDEX

W

<code>weitereFaecher</code>	37
<code>weva</code> (Paket)	55

X

<code>xcolor</code> (Paket)	6
<code>\xout</code>	22
<code>xparse</code> (Paket)	6
<code>xsim</code> (Paket)	7, 10 f., 13 f., 16, 66 f.
<code>xspace</code> (Paket)	7
<code>xstring</code> (Paket)	6

Z

<code>zeilenNr</code> (Umgebung)	35
<code>zeilenNrMehrspaltig</code> (Umgebung)	35
<code>zeilenNrZweispaltig</code> (Umgebung)	36
<code>\zeit</code>	49
<code>zeitraum</code>	50
<code>ziffer</code> (Paket)	42, 66
<code>zitate</code>	25
<code>zref-totpages</code> (Paket)	6