

# NGBSCHULE

LaTeX-Paket zur Erweiterung des schule-Pakets.

O.O.6 2018-12-15

Paketdokumentation

Jonas NEUGEBAUER

<https://ngb.schule>

[schule@neugebauer.cc](mailto:schule@neugebauer.cc)

Erweiterungen und Ergänzungen des schule-Pakets.

## Inhaltsverzeichnis

<b>I. Einleitung</b>	<b>2</b>	6.2. Checkup und Selbstlernbögen . . .	8
<b>1. Typografie und Farben</b>	<b>2</b>	6.3. Diagnosebögen . . . . .	8
1.1. Schriftarten . . . . .	2	<b>7. Klausuren und Klassenarbeiten</b>	<b>8</b>
1.1.1. Überschriften . . . . .	2	7.1. Klausuren . . . . .	8
1.2. Farben . . . . .	2	7.2. Erwartungshorizonte . . . . .	8
<b>II. Präambel</b>	<b>3</b>	<b>V. Fächer</b>	<b>9</b>
<b>2. Pakete einbinden</b>	<b>3</b>	<b>8. Fach Informatik</b>	<b>9</b>
<b>3. Dokumentinformationen</b>	<b>3</b>	8.1. Materialien Informatik . . . . .	9
<b>III. Allgemeine Erweiterungen</b>	<b>4</b>	8.1.1. UML-Diagramme . . . . .	9
<b>4. Aufgabensammlung</b>	<b>6</b>	8.1.2. Quelltexte . . . . .	10
<b>5. Materialsammlung</b>	<b>6</b>	8.1.3. Struktogramme . . . . .	10
<b>IV. Dokumenttypen und Subtypen</b>	<b>8</b>	<b>9. Fach Mathematik</b>	<b>10</b>
<b>6. Arbeitsblätter</b>	<b>8</b>	9.1. Materialien Informatik . . . . .	10
6.1. Allgemeine Arbeitsblätter . . . . .	8	<b>VI. Anhang</b>	<b>11</b>
		<b>10.Index</b>	<b>12</b>

# Teil I.

## Einleitung

### 1. Typografie und Farben

Die Gestaltung der Dokumente wird zum größten Teil in der Datei `theme.tex` festgelegt, die per `\input` eingebunden wird.

Derzeit wird für alle Dokumente dieselbe Theme-Datei eingebunden, allerdings ist eine Idee in Zukunft unterschiedliche Themes für Material der Sekundarstufe I und II einzubauen.

#### 1.1. Schriftarten

Als Schriftarten werden FiraSans für Textabsätze und tgschola für Überschriften genutzt. Als Monospace-Font wird courier geladen.

##### 1.1.1. Überschriften

Generell werden die Kommandos zum ändern von Schriftarten des KOMA-Paketes genutzt (`\setkomafont` und `\addtokomafont`), um die Schriftarten anzupassen.

Zusätzlich zu den dort verfügbaren gibt es eine neue Überschrift für den Reihentitel (`reihe`). Wie alle KOMA Schriften kann sie über `\usekomafont{reihe}` genutzt werden, was aber in der Regel von den entsprechenden Makros für Reihe und Titel automatisch übernommen wird.

#### 1.2. Farben

## **Teil II.**

# **Präambel**

## **2. Pakete einbinden**

## **3. Dokumentinformationen**

## Teil III.

# Allgemeine Erweiterungen

`\linie[⟨breite⟩]`

Voreinstellung: 6cm

Zeichnet eine Linie der angegebenen Breite.

```
1 \linie[4cm]
2
```



`\Namensfeld[⟨breite⟩]`

Voreinstellung: 5cm

Erzeugt ein Feld, um einen Namen einzutragen.

```
1 \Namensfeld
2
```

Name: \_\_\_\_\_

`\tab`

Kurz für `\hspace{1cm}`.

`\titlerule`

Erzeugt eine Linie mit Breite `\textwidth` und Dicke 1pt, die als Unterstreichung von Überschriften benutzt wird..

`\code{⟨text⟩}`

Wird zur Auszeichnung von Quelltextauszügen benutzt. Das Makro wird auch von anderen Paketen erzeugt, daher prüft **NGBSCHULE** zunächst, ob es schon existiert und erzeugt nur dann eine eigene Version.

`\operator{⟨operator⟩}`

Wird benutzt, um in Aufgabentexten Operatoren auszuzeichnen.

```
1 \operator{Implementiere} die Operation \texttt{push} aus der Klasse
\texttt{Stack}.
```

2

---

IMPLEMENTIERE die Operation `push` aus der Klasse `Stack`.

`\TITEL`

Erzeugt einen großen, zentrierten Titel.

1        `\TITEL`  
2

---

**Dokumentation des Pakets `ngbschule`**

`\ReiheTitel`

Erzeugt einen Dokumenttitel bestehend aus Reihe und Titel.

1        `\ReiheTitel`  
2

---

**Beispiel einer Reihe**  
**Dokumentation des Pakets `ngbschule`**

---

`\begin{rahmen} [⟨color⟩]`

Voreinstellung: `black`

Erzeugt eine `tcolorbox` mit einem schwarzen einfachen Rahmen.

1        `\begin{rahmen}`  
2        `\blindtext`  
3        `\end{rahmen}`  
4

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.

## 4. Aufgabensammlung

Um das Aufbauen einer Aufgabensammlung zu vereinfachen steht allen Dokumenttypen das Makro `\aufgabeLaden{<name>}` zur Verfügung. Mit diesem können Aufgabendateien (mit der Endung `.afg.tex`) aus einem Ordner in das aktuelle Dokument eingefügt werden. Der Pfad, in dem nach den Aufgabendateien gesucht wird kann durch die Option `aufgaben` für jedes Dokument angepasst werden.

`aufgaben = {<pfad>}`

Voreinstellung: `ngb/afg`

Setzt den Pfad zur Aufgabensammlung. Als Standard wird in einem Unterordner des Pakets selbst gesucht, von dem aus die Aufgaben jeder  $\TeX$ -Datei zur Verfügung stehen. Um die Aufgabensammlung an einem anderen Ort zu speichern muss (derzeit) der Pfad in jeder Datei über diese Option angegeben werden.

Der Pfad kann absolut sein (z.B. `/user/tex/Aufgaben`) oder eine relative Pfadangabe (z.B. `../../Aufgaben`).

`\aufgabeLaden{<name>}`

Lädt eine Aufgabe aus der Aufgabensammlung, wenn sie vorhanden ist. Die Datei muss den Namen `<name>.afg.tex` haben und im Ordner der Aufgabensammlung liegen.

Ohne Angabe eines neuen Pfades für die Aufgabensammlung ist der zusammengesetzte Pfad für eine Aufgabe dann also `ngb/afg/<name>.afg.tex`.

Es können auch Unterordner angegeben werden. (Z.B. `\aufgabeLaden{if/aufgabe-1}`. Die Aufgabendatei liegt in dem Fall also unter `ngb/afg/if/aufgabe-1.afg.tex`.)

## 5. Materialsammlung

Das Paket liefert eine Reihe von Materialien mit, die (hauptsächlich bezogen auf die Lehrpläne in NRW) häufig in Dokumenten auftauchen. Die einzelnen Fächer stellen Makros bereit, um fachspezifische Materialien nachzuladen. Dies ist vor allem für die Informatik relevant, für das vorgefertigte *UML-Diagramme*, *Struktogramme* und *Quelltexte* eingebunden werden können.

## 5. Materialsammlung

Details dazu sind bei der Beschreibung der Fächer unter Abschnitt 8.1, S. 9 und Abschnitt 9.1, S. 10 beschrieben.

Generell sollten die Materialdateien möglichst so gehalten sein, dass sie auch ohne das Paket **NGBSCHULE** verwendet werden können (Struktogramme sollten also zum Beispiel nicht die Makros `\Anweisung` etc. verwenden, sondern direkt die Makros aus dem stuktex-Paket).

## **Teil IV.**

# **Dokumenttypen und Subtypen**

## **6. Arbeitsblätter**

### **6.1. Allgemeine Arbeitsblätter**

### **6.2. Checkup und Selbstlernbögen**

### **6.3. Diagnosebögen**

## **7. Klausuren und Klassenarbeiten**

### **7.1. Klausuren**

### **7.2. Erwartungshorizonte**



# Teil V.

## Fächer

### 8. Fach Informatik

#### 8.1. Materialien Informatik

Sobald für mich relevant werden auch Sequenzdiagramme und PAPS ergänzt.

Für das Fach Informatik lassen sich auf einfache Weise eine Reihe vorgefertigte UML-Diagramme, Struktogramme und Quelltexte einbinden.

`\cd{<name>}`

Fügt das Klassendiagramm mit dem Namen *<name>* in das Dokument ein.

`\lst{<name>}`

Fügt die Quelltextdatei mit dem Namen *<name>* in das Dokument ein.

`\nss{<name>}`

Fügt das Struktogramm mit dem Namen *<name>* in das Dokument ein.

Die Namen der Materialien halten sich an die Punktnotation, um eine möglichst einfache Organisation zu gewährleisten. Die Namen lassen sich so auch mehrfach verwenden, um zum Beispiel verschiedene Darstellungen eines Algorithmus darzustellen.

Die Operation `pop` des `Stack` ist zum Beispiel unter dem Namen `stack.pop` gespeichert. Mit `\lst{stack.pop}` wird eine Quelltextdarstellung der Operation eingebunden (Standard: Java), während `\nss{stack.pop}` das entsprechende Struktogramm anzeigt. `\cd{stack}` bindet wiederum ein Klassendiagramm des `Stack` ein und `\lst{stack.pop.pseudo}` einen Pseudocode.

Alle Dateien für das Fach Informatik liegen im Unterordner `if`. Die Dateinamen setzen sich aus dem Namen des Materials und einer Endung pro Typ zusammen (abgeleitet aus dem Namen des Makros, z.B. `.cd.tex` für `\cd`).

##### 8.1.1. UML-Diagramme

Klassen- und Objektdiagramme liegen in Dateien mit der Endung `.cd.tex` im Materialordner. Zum Beispiel `ngb/inc/if/stack.cd.tex`. Dabei ist zu beachten, dass die Namen, die Operationen/Methoden entsprechen, in der Regel Objektdiagramme enthalten, die die Veränderungen der Zeiger beim Aufruf der Operation darstellen.

- `stack`
- `stack.pop`
- `stack.push`
- `queue`
- `queue.enqueue`
- `queue.dequeue`
- `list`
- `list.insert`
- `list.append`
- `list.delete`

### 8.1.2. Quelltexte

Quelltexte liegen in Dateien mit der Endung `.lst.tex` im Materialordner.

Zum Beispiel `ngb/inc/if/stack.pop.lst.tex`. Bei mehreren verfügbaren Sprachen wird der Name der Sprache angehängt, also zum Beispiel `stack.pop.java.lst.tex`. Das Makro sucht zunächst nach einer Datei mit dem exakten angegebenen Namen und als Fallback nach der Java-Version.

- |                                     |                                     |
|-------------------------------------|-------------------------------------|
| • <code>stack.pop.java</code>       | • <code>list.insert.pseudo</code>   |
| • <code>stack.push.java</code>      | • <code>list.append.pseudo</code>   |
| • <code>queue.enqueue.java</code>   | • <code>list.delete.pseudo</code>   |
| • <code>queue.dequeue.java</code>   | • <code>stack.pop.python</code>     |
| • <code>list.insert.java</code>     | • <code>stack.push.python</code>    |
| • <code>list.append.java</code>     | • <code>queue.enqueue.python</code> |
| • <code>list.delete.java</code>     | • <code>queue.dequeue.python</code> |
| • <code>stack.pop.pseudo</code>     | • <code>list.insert.python</code>   |
| • <code>stack.push.pseudo</code>    | • <code>list.append.python</code>   |
| • <code>queue.enqueue.pseudo</code> | • <code>list.delete.python</code>   |
| • <code>queue.dequeue.pseudo</code> |                                     |

### 8.1.3. Struktogramme

Struktogramme liegen in Dateien mit der Endung `.nss` im Materialordner.

Zum Beispiel `ngb/inc/if/stack.pop.nss`.

- |                              |                            |
|------------------------------|----------------------------|
| • <code>stack.pop</code>     | • <code>list.insert</code> |
| • <code>stack.push</code>    | • <code>list.append</code> |
| • <code>queue.enqueue</code> | • <code>list.delete</code> |
| • <code>queue.dequeue</code> |                            |

## 9. Fach Mathematik

### 9.1. Materialien Informatik

Für Mathematik sind vor allem geometrische Konstruktionen und Merksätze / Definitionen ange-dacht.

# **Teil VI.**

## **Anhang**

## 10. Index

<b>A</b>		<code>\nss</code> .....9
<code>\aufgabeLaden</code> .....6		
<code>aufgaben</code> .....6	<b>O</b>	
<b>C</b>	<code>\operator</code> .....4	
<code>\cd</code> .....9		
<code>\code</code> .....4	<b>R</b>	
courier (Paket).....2	rahmen (Umgebung).....5	
<b>F</b>	<code>\ReiheTitel</code> .....5	
FiraSans (Paket).....2		
<b>L</b>	<b>T</b>	
<code>\linie</code> .....4	<code>\tab</code> .....4	
<code>\lst</code> .....9	tcolorbox (Paket).....5	
<b>N</b>	tgschola (Paket).....2	
<code>\Namensfeld</code> .....4	<code>\TITEL</code> .....5	
	<code>\titlerule</code> .....4	