

Listen einfacher setzen mit dem Paket `easylist`

Adelheid Grob

Einleitung

\LaTeX stellt mit den Umgebungen `itemize` und `enumerate` zwei Möglichkeiten zur Verfügung, um Listen zu erstellen. Allerdings erlauben diese nur eine Gliederungstiefe von maximal vier – mal ganz davon abgesehen, dass verschachtelte `itemize`-Umgebungen schnell unübersichtlich werden. Desweiteren ist es zwar möglich, die Eigenschaften der Zähler oder der Liste als solches zu ändern, dies erfordert jedoch einiges an Aufwand. Abhilfe kann hier das sehr umfangreiche Paket von Paul Isambert namens `easylist` schaffen, das beliebig tiefe Gliederung erlaubt und mit dem auf sehr einfache Art und Weise das Aussehen einer Liste angepasst werden kann.

Einbinden und Optionen

Das Paket wird in der Präambel mittels `\usepackage[OPT]{easylist}` eingebunden. Als Optionen stehen hierbei zur Verfügung

- eine Nummer, die die Gliederungstiefe angibt (standardmässig 10)
- die Auswahl des Zeichens, mit dem die verschiedenen Items gekennzeichnet werden sollen (Standard ist §, möglich sind §, @, ¶ und #).

Befehlsübersicht

Das Paket stellt die Umgebung `easylist` zur Verfügung, die zum Setzen der Liste verwendet wird. Außerdem werden die Kommandos `\NewList` und `\ListProperties(key=value)` definiert. Mit ersterem ist es möglich, die Listeneigenschaften komfortabel zu ändern, das letztere wird benötigt um alle Änderungen wieder zurückzusetzen, was hilfreich sein kann, wenn die für eine Liste getroffenen Definitionen an späterer Stelle im Text nicht mehr benötigt werden. Es ist auch möglich, dem Kommando `NewList(key=value)` direkt die gewünschten Eigenschaften für eine Liste mitzugeben. Darüber hinaus existieren die beiden Kommandos `\Activate` und `Deactivate`, die benötigt werden, um auch innerhalb anderer Umgebungen `easylist` nutzen zu können.

Das Kommando `ListProperties(key=value)`

`\ListProperties(key=value)` bietet die Möglichkeit, mit Hilfe verschiedener `keys` das Aussehen einer Liste zu verändern – diese sind im Wesentlichen so strukturiert, dass sich eine Änderung entweder auf alle Gliederungsebenen auswirkt oder nur auf eine bestimmte (N-te) Ebene (`key/keyN`). `\ListProperties()` wird in der Regel der *easylist*-Umgebung vorangestellt. Es kann aber auch innerhalb der Umgebung aufgerufen werden, bezieht sich dann aber nur auf die nächstfolgende Zeile.

In nachfolgender Tabelle sind alle möglichen `keys` aufgelistet.

<i>key</i>	<i>Bedeutung</i>
StartN/StartN*	Angabe der Startnummer der N-ten Ebene, auch möglich ist hierbei die Angabe T _E X-interner Zähler wie beispielweise <code>\thesection</code> (in diesem Fall muss die Sternvariante verwendet werden). Soll einem Zähler einer höheren als der ersten Gliederungsebene ein anderer Wert mitgegeben werden, muss der Aufruf von <code>\ListProperties()</code> innerhalb der Liste erfolgen. (siehe Bsp. 1)
Mark/MarkN	Angabe der Punctuation innerhalb des Zählers (standardmässig wird ein normaler Punkt gesetzt). (siehe Bsp. 2)
FinalMark/ FinalMarkN	Angabe des abschließenden Zeichens für den letzten Gliederungspunkt einer Ebene. (siehe Bsp. 2)
Hide/HideN	Verbergen aller bzw. des N-ten Zählers. Dies ist erforderlich, wenn z.B. über <code>CtrCom/CtrComN</code> ein anderes Aufzählungssymbol definiert wird. <code>Hide3=2</code> verbirgt so bei der dritten Gliederungstiefe die ersten beiden Zählerpunkte. (siehe Bsp. 3)
Style/StyleN	Änderung des Stils der gesamten Liste (bzw. der N-ten Gliederungstiefe), beispielsweise Farbe, Aufzählungssymbole (siehe Bsp. 3)
Style*/StyleN*	wie <code>Style/StyleN</code> , allerdings betreffen die Angaben nur die Zähler (siehe Bsp. 3)

Style**/StyleN**	wie Style/StyleN, allerdings betreffen die Angaben nur den Text (siehe Bsp. 3)
CtrCom/ CtrComN	Ausführen eines L ^A T _E X-Kommandos für alle Ebenen oder die N-te. (siehe Bsp. 4)
Numbers/ NumbersN	Angabe der Art der Nummerierung für die gesamte Liste oder bestimmter Teilebenen, dabei stehen grosse und kleine römische Zahlen zur Verfügung (R,r), grosse und kleinen Buchstaben (L,l), die Zeichen von Zapf Dingbats (z) oder arabische Zahlen (a, Standard). (siehe Bsp. 5)
Margin/ MarginN	Gibt den Abstand vom linken Textrand an. Margin setzt dabei einen allgemeinen Rand, MarginN den Rand für den N-ten Unterpunkt. (siehe Bsp. 6)
Progressive/ Progressive*	Ermöglicht ein progressives Einrücken der Aufzählungspunkte. Die Sternversion startet die Einrückung erst ab der zweiten Gliederungstiefe. (siehe Bsp. 7)
Hang/HangN	Legt fest, ob der Text aller oder eines Punktes nach einem Zeilenumbruch eingerückt wird (true/false, s. Bsp. 8).
Space/SpaceN	Verändern des Abstandes zweier aufeinander folgender Punkte unterschiedlicher Gliederungstiefen (für alle oder für die N-te Ebene, siehe Bsp. 9)
Space*/SpaceN*	wie Space/SpaceN, allerdings bezogen auf Punkte mit der gleichen Gliederungstiefe (siehe Bsp. 9).
FinalSpace/ FinalSpaceN	Festlegung des vertikalen Abstands zwischen dem letzten Zähler einer Ebene und dem Text (siehe Bsp. 9)
Indent/IndentN	Einrückung aller oder des N-ten Zählers (siehe Bsp. 10)

Align/AlignN Festlegung der Ausrichtung des Textes. Bei Verwendung von `Hang / HangN = true` kann es zu unschönem Flattersatz kommen. Mit `Align` kann man einen Text entweder am längsten (bzw. am N-ten) Zähler oder an einem festen Wert, z.B. 2cm, ausrichten. Es gibt die Möglichkeit, den Text am Abschlusszeichen des Zählers auszurichten (`Align=move`) – hierbei muss berücksichtigt werden, dass mit `Margin` dafür gesorgt wird, dass der globale Textrand eingehalten wird. Mit `Align=fixed` wird der Text aller Punkte am Textbeginn des ersten Punktes ausgerichtet – hierbei muss evtl. mit `FinalSpace` garantiert werden, dass bei größeren Zählern der Text nicht mit dem Zähler überlappt. (s. Bsp. 11)

Vordefinierte Listen

Es existieren im Moment sechs vordefinierte Stile, diese werden der `easylist`-Umgebung beim Aufruf als Option mitgegeben: `\begin{easylist}[STIL].tractatus` definiert die Punctuation nach dem ersten Zählers als einfachen Punkt (Mark1=.), alle anderen Zähler enden ohne abschließendes Zeichen – dieser Stil kann so also nur bis zur zweiten Gliederungsebene verwendet werden. Der Stil `checklist` erstellt eine Liste mit Checkboxen als Zähler. `booktoc` und `articletoc` erstellen Listen, die auf der ersten Ebene die Auszeichnungsdefinitionen der `section` (bzw. des `chapter`) übernehmen - auf der zweiten Ebene entsprechend der `subsection` und so weiter. Zu guter letzt gibt es noch die beiden Stile `enumerate` und `itemize`, die den beiden Standardumgebungen nachempfunden sind, wobei Punkte ab der fünften Gliederungsebene ignoriert werden und nach der Liste als Fließtext gesetzt werden.

Diesen vordefinierten Stilen kann man mit `\ListProperties()` wiederum zusätzliche oder andere Werte übergeben.

Alle Stile sind am Schluß dieses Artikels abgebildet.

Probleme

Die Verwendung von UTF8-Encoding ist schwierig – zum Einen ist es nicht möglich, das Standardgliederungssymbol § zu verwenden. Zum Anderen werden Umlaute inner halb einer *easylist*-Umgebung als fehlerhaft erkannt.

Ein weiteres Problem besteht, wenn man z.B. um eine Liste eine Box zeichnen möchte. Dies liegt daran, dass innerhalb einer `\fbox` die für *easylist* benötigten Werte §, @, ¶ oder ‡ usw. nicht mehr „aktiv“ sind. Abhilfe schaffen die beiden Kommandos `\Activate` und `\Deactivate`, die das Zeichen, das für die Listengenerierung verwendet wird, auch innerhalb des entsprechenden Befehls aktivieren. Allerdings funktioniert dies nicht, wenn man mehr als eine Zeile in der *easylist*-Umgebung hat! Für das Bsp. 12 wurde deswegen folgendes Kommando definiert, mit dem eine Umrahmung auch mehrerer Zeilen möglich ist.

```
\newcommand{\MitBox}[2]{%
\fbbox{%
\begin{minipage}{#1\textwidth}
#2
\end{minipage}}%
}
```

Beispiele

Beispiel 1: StartN*/StartN

5. Punkt 5.1. Unterpunkt 5.1.1. Unterunterpunkt 5. noch ein Punkt 5.15. Unterpunkt x.15	<ol style="list-style-type: none"> 1 <code>\ListProperties(Start1*=\thepage)</code> 2 <code>\begin{easylist}</code> 3 § Punkt 4 §§ Unterpunkt 5 §§§ Unterunterpunkt 6 § noch ein Punkt 7 <code>\ListProperties(Start2=15)</code> 8 §§ Unterpunkt x.15 9 <code>\end{easylist}</code>
---	--

Beispiel 2: Mark/MarkN und FinalMark/FinalMarkN

1· Punkt
1.1· Unterpunkt
1.1.1) Unterunterpunkt
2· noch ein Punkt
2.1· Unterpunkt

```

1 \ListProperties(Mark=$\cdot$, %
2 Mark3={})
3 \begin{easylist}
4 § Punkt
5 §§ Unterpunkt
6 §§§ Unterunterpunkt
7 § noch ein Punkt
8 §§ Unterpunkt
9 \end{easylist}

```

1) Punkt
1.1) Unterpunkt
1.1.1: Unterunterpunkt
2) noch ein Punkt
2.1) Unterpunkt

```

1 \ListProperties(FinalMark={}), %
2 FinalMark3=:)
3 \begin{easylist}
4 § Punkt
5 §§ Unterpunkt
6 §§§ Unterunterpunkt
7 § noch ein Punkt
8 §§ Unterpunkt
9 \end{easylist}

```

Beispiel 3: Style/StyleN und Sternversionen

1. Punkt
1.1. <i>Unterpunkt</i>
1.1.1. <i>Unterunterpunkt</i>
2. noch ein Punkt
2.1. <i>Unterpunkt</i>

```

1 \ListProperties(Style=\color{blue}, %
2 Style2=\itshape)
3 \begin{easylist}
4 § Punkt
5 §§ Unterpunkt
6 §§§ Unterunterpunkt
7 § noch ein Punkt
8 §§ Unterpunkt
9 \end{easylist}

```

1. Punkt
→ Unterpunkt
1.1.1. Unterunterpunkt
2. noch ein Punkt
→ Unterpunkt

```

1 \ListProperties(Style*=\itshape,%
2 Style2**=\color{blue},
3 Style2*=\rightarrow$,
4 Hide2=2)
5 \begin{easylist}
6 § Punkt
7 §§ Unterpunkt
8 §§§ Unterunterpunkt
9 § noch ein Punkt
10 §§ Unterpunkt
11 \end{easylist}

```

Beispiel 4: CtrCom/CtrComN

1. Punkt
1.1. Unterpunkt
1.1.1. Unterunterpunkt
2. noch ein Punkt
2.1. Unterpunkt

```

1 \ListProperties(CtrCom=\fbox,%
2 CtrCom2=\colorbox{orange!30})
3 \begin{easylist}
4 § Punkt
5 §§ Unterpunkt
6 §§§ Unterunterpunkt
7 § noch ein Punkt
8 §§ Unterpunkt
9 \end{easylist}

```

Beispiel 5: Numbers/NumbersN

I. Punkt
I.I. Unterpunkt
I.I.a. Unterunterpunkt
II. noch ein Punkt
II.I. Unterpunkt

```

1 \ListProperties(Numbers=R,%
2 Numbers3=1)
3 \begin{easylist}
4 § Punkt
5 §§ Unterpunkt
6 §§§ Unterunterpunkt
7 § noch ein Punkt
8 §§ Unterpunkt
9 \end{easylist}

```

Beispiel 6: Margin/MarginN

1. Punkt
1.1. Unterpunkt
1.1.1. Unterunterpunkt
2. noch ein Punkt
2.1. Unterpunkt

```

1 \ListProperties(Margin=5ex)
2 \begin{easylist}
3 § Punkt
4 §§ Unterpunkt
5 §§§ Unterunterpunkt
6 § noch ein Punkt
7 §§ Unterpunkt
8 \end{easylist}

```

1. Punkt
1.1. Unterpunkt
1.1.1. Unterunterpunkt
2. noch ein Punkt
2.1. Unterpunkt

```

1 \ListProperties(Margin1=2cm,
2 Margin2=0.5cm, Margin3=1cm)
3 \begin{easylist}
4 § Punkt
5 §§ Unterpunkt
6 §§§ Unterunterpunkt
7 § noch ein Punkt
8 §§ Unterpunkt
9 \end{easylist}

```

Beispiel 7: Progressive/Progressive*

1. Punkt
1.1. Unterpunkt
1.1.1. Unterunterpunkt
2. noch ein Punkt
2.1. Unterpunkt

```

1 \ListProperties(Progressive=3ex)
2 \begin{easylist}
3 § Punkt
4 §§ Unterpunkt
5 §§§ Unterunterpunkt
6 § noch ein Punkt
7 §§ Unterpunkt
8 \end{easylist}

```

1. Punkt
1.1. Unterpunkt
1.1.1. Unterunterpunkt
2. noch ein Punkt
2.1. Unterpunkt

```

1 \ListProperties(Progressive*=3ex)
2 \begin{easylist}
3 § Punkt
4 §§ Unterpunkt
5 §§§ Unterunterpunkt
6 § noch ein Punkt
7 §§ Unterpunkt
8 \end{easylist}

```


Beispiel 8: `Hang/HangN`

1. langer Punkt um zu zeigen,
dass bei einem Zeilenumbruch
der Text eingerückt wird.
1.1. hier ist `Hang=false`, der Text
nach dem Zeilenumbruch wird
nicht eingerückt.
1.1.1. Der Punkt der dritten Ebene
wird wieder eingerückt.

```

1 \begin{easylist}
2 \ListProperties(Hang=true,%
3 Hang2=false)
4 § langer Punkt um zu zeigen, dass
5 bei einem Zeilenumbruch der Text
6 eingerückt wird.
7 §§ hier ist Hang=false, der Text
8 nach dem Zeilenumbruch wird nicht
9 eingerückt.
10 §§§ Der Punkt der dritten Ebene
11 wird wieder eingerückt.
12 \end{easylist}

```

Beispiel 9: `Space/SpaceN`, `Space*/SpaceN*` und `FinalSpace/FinalSpaceN`

1. Punkt

1.1. Unterpunkt

1.2. weiterer Unterpunkt

1.2.1. Unterunterpunkt

```

1 \begin{easylist}
2 \ListProperties(Space=0.8cm,%
3 Space2*=.3cm)
4 § Punkt
5 §§ Unterpunkt
6 §§§ weiterer Unterpunkt
7 §§§§ Unterunterpunkt
8 \end{easylist}

```

1. Punkt
1.1. Unterpunkt 1
1.2. Unterpunkt 2
1.3. Unterpunkt 3
1.3.1. Unterunterpunkt
2. noch ein Punkt

```

1 \begin{easylist}
2 \ListProperties(FinalSpace=0.5cm,%
3 FinalSpace3=1.5cm)
4 § Punkt
5 §§ Unterpunkt 1
6 §§§ Unterpunkt 2
7 §§§§ Unterpunkt 3
8 §§§§§ Unterunterpunkt
9 § noch ein Punkt
10 \end{easylist}

```

Beispiel 10: `Indent/IndentN`

<p>1. ein erster Punkt, dem ein paar Unterpunkte folgen</p> <p> 1.1. ein Unterpunkt mit Umbruch, um Indent zu demonstrieren</p> <p> 1.1.1. ein Unterunterpunkt zum Unterpunkt, der auch umbricht</p>
--

```

1 \begin{easylist}
2 \ListProperties(Indent=1cm,%
3 Indent2=2cm)
4 § ein erster Punkt, dem
5 ein paar Unterpunkte
6 folgen
7 §§ ein Unterpunkt mit
8 Umbruch, um Indent zu
9 demonstrieren
10 §§§ ein Unterunterpunkt
11 zum Unterpunkt, der
12 auch umbricht
13 \end{easylist}

```

Beispiel 11: Align/AlignN

<p>1. ein erster Punkt</p> <p>1000. ein weiterer Punkt mit einem sehr hohen Zähler</p>
--

```

1 \begin{easylist}
2 \ListProperties(Hang=true,%
3 Align=fixed,FinalSpace=2em)
4 § ein erster Punkt
5 \ListProperties(Start1=1000)
6 § ein weiterer Punkt mit einem
7 sehr hohen Zähler
8 \end{easylist}

```

<p>1. ein erster Punkt</p> <p>1000. ein weiterer Punkt mit einem sehr hohen Zähler</p>
--

```

1 \begin{easylist}
2 \ListProperties(Hang=true,%
3 Align=move,Margin=2em)
4 § ein erster Punkt
5 \ListProperties(Start1=1000)
6 § ein weiterer Punkt mit einem
7 sehr hohen Zähler
8 \end{easylist}

```

1.	ein erster Punkt
1000.	ein weiterer Punkt mit einem sehr hohen Zähler

```

1 \begin{easylist}
2 \ListProperties(Hang=true,%
3 Align=2cm)
4 § ein erster Punkt
5 \ListProperties(Start1=1000)
6 § ein weiterer Punkt mit einem
7 sehr hohen Zähler
8 \end{easylist}

```

Beispiel 12: Activate/Deactivate

1. Punkt
1.1. Unterpunkt
1.1.1. Unterunterpunkt
2. noch ein Punkt
2.1. Unterpunkt

```

1 \Activate\MitBox{0.9}{
2 \begin{easylist}
3 § Punkt
4 §§ Unterpunkt
5 §§§ Unterunterpunkt
6 § noch ein Punkt
7 §§ Unterpunkt
8 \end{easylist}}
9 \Deactivate

```

Vordefinierte Stile

tractatus:

1. Punkt
1.1 Unterpunkt
1.11 Unterunterpunkt
2. noch ein Punkt
2.1 Unterpunkt

checklist:

<input type="checkbox"/> Punkt
<input type="checkbox"/> Unterpunkt
<input type="checkbox"/> Unterunterpunkt
<input type="checkbox"/> noch ein Punkt
<input type="checkbox"/> Unterpunkt

booktoc:

I chapter
1 section
1.1 subsection
1.1.1 subsubsection

articletoc:

1 section
1.1 subsection
1.1.1 subsubsection
2 section
2.0.1 subsection

enumerate:

- 1. erster Punkt
 - (a) zweiter Punkt
 - i. dritter Punkt
 - A. vierter Punkt
- 1.a)i.A.1. fünfter Punkt

itemize:

- erster Punkt
 - zweiter Punkt
 - * dritter Punkt
 - vierter Punkt
- fünfter Punkt