

Orderrücken gestalten mit `ticket.sty`

Uwe Ziegenhagen

In diesem Artikel möchte ich kurz vorstellen, wie man mit `ticket.sty` einfach Orderrücken und andere Etiketten gestalten kann.

Das `ticket` Paket

Das Paket `ticket.sty` ist mein persönlicher Favorit, wenn es um die Erzeugung von Etiketten aller Art mit L^AT_EX geht. Das Paket wurde von Thomas Emmel entwickelt, aktuell ist die Version 0.40d aus dem Jahr 2016.

Zur Definition der einzelnen Etiketten trägt man die notwendigen Angaben in eine TDF-Datei ein, TDF steht dabei für »Ticket Definition File«. In diese Datei kommen Angaben zum Layout des Bogens (vertikaler und horizontaler Offset, gegebenenfalls druckerspezifisch) und die Angaben zu den einzelnen Etiketten: wieviele Spalten und Zeilen auf dem Bogen, Dimensionen eines einzelnen Tickets. In diesem Artikel soll die Nutzung des Pakets anhand von Orderrücken gezeigt werden, das Beispiel lässt sich aber auf nahezu jede Art von Etiketten und Labels übertragen.

Orderrücken gestalten

Betrachten wir am den folgenden Quellcode, der drei Etiketten für breite Ordner auf einem DIN A4 Blatt erzeugt.

```
\documentclass[landscape]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage{graphicx}
\usepackage[sfdefault]{plex-sans}

\begin{filecontents}[overwrite]{Ordner.tdf}
\unitlength=1mm
\hoffset=10mm
\voffset=-15mm
\ticketNumbers{1}{3}
\ticketSize{190}{58} % Breite und Hoehe der Labels in mm
\ticketDistance{0}{0} % Abstand der Labels
\end{filecontents}

\usepackage[Ordner]{ticket}
```

```

\renewcommand{\ticketdefault}{}%
\makeatletter
\@boxedtrue % Rahmen um Ticket
\@emptycrossmarkfalse % Falzmarken
\@cutmarktrue % Schnittmarken
\makeatother

\newcommand{\myticket}[2]{
\ticket{%
\put(15,15){\scalebox{#1}{\bfseries #2}}
}}

\begin{document}
\myticket{7}{Ausbildung}
\myticket{7}{Steuern}
\myticket{12}{ADAC}
\end{document}

```

Als Dokumentenklasse wähle ich `scrartcl`, da hier A4 als Dokumentengröße bereits vorgegeben ist und das Querformat einfach per Klassenoption gesetzt werden kann. Das Rumhantieren mit den Optionen vom `geometry` Paket kann daher entfallen. Aus dem `graphicx` Paket benötigen wir den `\scalebox` Befehl, um die Schriftgröße innerhalb eines Labels beliebig skalieren zu können, als Schriftart kommt die IBM Plex Sans zum Einsatz.

Um nur mit einer Datei arbeiten zu müssen und nicht mehreren für Ticket-Definition und dem eigentlichen \TeX -Dokument, lege ich die Ticketdefinition in eine `filecontents`-Umgebung. Die »`overwrite`« Option sorgt dabei dafür, dass die Datei beim Kompilieren stets überschrieben wird, selbst wenn sie bereits existiert.

Wir legen die einzelnen Tickets mit der Größe 190 x 58 Millimeter an, zwischen den einzelnen Tickets soll es weder horizontalen noch vertikalen Abstand geben. Es passen drei Tickets in einer Spalte auf eine Seite, die vertikalen und horizontalen Offsets spielen bei unserem Beispiel keine Rolle, da wir nicht auf fertige Etikettenbögen drucken und werden auf nicht völlig unpassende Werte gesetzt.

Im nächsten Schritt laden wir das `ticket`-Paket, als Option übergeben wir den Namen unserer vorher geschriebenen TDF-Datei. Mit dem `\renewcommand` Befehl setzen wir das Design des Ticket-Hintergrunds zurück, an dieser Stelle könnte man auch beispielsweise Bilder einbetten, die dann auf jedem Etikett gesetzt werden.

Über die Befehle innerhalb von `makeatletter` und `makeatother` definieren wir, dass unsere Etiketten a) von einem Rahmen umgeben sind, b) Schnittmarken gesetzt und c) Falzmarken nicht gesetzt werden.

Als nächsten definieren wir uns mit dem `\myticket` Befehl ein Kommando für die Erzeugung der individuellen Etiketten. Wir benötigen dabei zwei Parameter, zum einen den Skalierungsfaktor für die Schriftgröße, zum anderen den Text, der auf das jeweilige Etikett gesetzt werden soll.

In das eigentliche Dokument kommen dann nur noch die `\myticket` Befehle mit den jeweiligen Parametern für Textskalierung und Inhalt.

Gesetzt sieht das Ganze dann so aus wie in der folgenden Abbildung dargestellt.



Fazit

Mit `ticket.sty` lassen sich in kurzer Zeit Etiketten und Labels jeder Art gestalten. Eine TDF-Datei ist schnell geschrieben, die Einbettung der Definition in den \TeX -Code sorgt dafür, dass man stets nur eine Datei bearbeiten muss.