Formatautomat

Automatische Textumwandlung mit pandoc

Wer Anleitungen, Dokumentationen oder Briefe in einer neutralen Auszeichnungssprache verfasst, kann daraus mit wenig Aufwand ein passendes Format für verschiedene Anwendungen machen. Das Programm pandoc übernimmt den Formatwandel unter Windows, Linux und macOS.

Von Jan Mahn

Schon seit Jahren hat es sich in der Webentwicklung herumgesprochen, dass es keine gute Idee ist, Inhalte und Layout in einem Dokument zu vermischen. Stattdessen trennt man Stylesheet und Inhalte und spart sich Veränderungen am fertigen Dokument, wenn sich das Logo oder die Farbe der Überschrift ändert. Im Büro ist genau diese Vermischung aber weiterhin Alltag: Briefe, Dokumente und Beschilderungen werden traditionell in Word auf einer Briefkopf-Vorlage zusammengebaut und als DOCX gespeichert. Ändert sich nach einigen Jahren der Firmenname oder die Faxnummer im Briefkopf, werden die Änderungen umständlich per Hand in jede Vorlage geschrieben.

Wer stattdessen auch bei Texten darauf achtet, diese vom Layout zu trennen, muss anfangs sich oder die Kollegen umgewöhnen, exportiert später aber problemlos Dokumente für Web, Druck oder E-Book-Reader.

Ausgezeichnet

Dokumente bestehen für gewöhnlich aus einer Handvoll Elemente: Texte, Grafiken, Tabellen, Überschriften und Listen. Um diese benutzerfreundlich und flexibel zu kennzeichnen, gab es viele Ansätze: Das von Tim Berners Lee erfundene HTML stellte sich mit den Tags in ‹ › für Texte aber als zu sperrig heraus und fand nie Verbreitung außerhalb von Entwicklerkreisen. Wesentlich nutzerfreundlicher ist die Auszeichnungssprache Markdown, die erst 2004 erfunden wurde. Eine Überschrift der obersten Ebene leiten Sie beispielsweise mit # am Anfang einer Zeile ein, eine Unterüberschrift mit ##. Die häufigsten Elemente finden Sie über ct.de/y1nh.

Markdown können Sie in jedem Texteditor schreiben. Einen optischen Eindruck vom Ergebnis liefern Markdown-Editoren oder Plug-ins für Programmierumgebungen (siehe ct.de/y1nh). Auf die Dateiendung kommt es nicht weiter an, .txt oder .md erfüllen ihren Zweck.

Die Verwandlung

Die Umwandlung der Markdown-Texte in ein leserfreundliches Endformat übernimmt das kostenlose Kommandozeilenwerkzeug pandoc. Windows-Anwender müssen nur den Installer herunterladen und ausführen (Download via ct.de/y1nh). Um PDF-Dateien zu erzeugen, braucht pandoc einen LaTeX-Konverter wie MiKTeX. Da Windows keine eigene Paketverwaltung hat, müssen Sie diesen selbst installieren (siehe ct.de/y1nh).

In den Paketquellen der großen Linux-Distributionen ist pandoc meist in etwas älteren Versionen enthalten. Unter Ubuntu installieren Sie es zusammen mit einem LaTeX-Konverter zum Beispiel mit

```
sudo apt install pandoc
sudo apt install texlive
```

Wer ein aktuelles Paket für Linux herunterladen oder selbst kompilieren möchte, findet die Anleitung auf der Seite der Ent-

wickler. Unter macOS muss die Paketverwaltung homebrew installiert sein. Dann landet das Programm mit dem Aufruf

```
brew install pandoc
```

zusammen mit allen Abhängigkeiten auf dem Mac.

Welche Version installiert ist, zeigt der Kommandozeilenbefehl `pandoc --version`. Für die erste Verwandlung benötigen Sie eine kurze Textdatei test.md mit einem Beispiel-Inhalt wie

```
# Ein Test
* eine Aufzählung
* weiteres Element
```

Navigieren Sie auf der Kommandozeile in den Ordner, in dem sich die Datei befindet und lassen Sie pandoc eine HTML-Datei erzeugen:

```
pandoc test.md -o test.html
```

Der Parameter `-o` sorgt dafür, dass das Ergebnis in die Datei `test.html` geschrieben wird – pandoc erkennt an der Dateiendung, welches Exportformat sinnvoll ist. Das Ergebnis ist ein HTML-Schnipsel:

```
<h1 id="ein-test">Ein Test</h1>
<ul>
<li>eine Aufzählung</li>
<li>weiteres Element</li>
</ul>
```

HTML-Kommentare, eingeleitet mit `<!--` und beendet durch `-->` verwandelt pandoc auch im fertigen Dokument in unsichtbare Kommentare. Möchten Sie sie ganz entfernen, fügen Sie den Parameter `--strip-comments` in den Programmaufruf ein. Statt der Markdown-Datei können Sie auch weitere Eingabeformate testen: Pandoc leistet auch beim Einlesen von Word-Dokumenten gute Arbeit (sofern die Layout-Möglichkeiten von Word nicht ausgereizt wurden). Auch Webseiten liest das Programm ein und extrahiert die Inhalte. Fügen Sie beim Programmaufruf einfach eine URL mit `https://` statt eines Dateinamens ein.

Bisher spuckt pandoc nur Fragmente einer Webseite aus. Um ein vollständiges HTML-Gerüst mit `<head>` und `<body>` zu generieren, gibt es den Parameter `-s` (für „standalone“). Bei einem Blick in den Quelltext einer so erzeugten Seite werden Sie Elemente bemerken, die Sie nicht selbst definiert haben. Sie stammen aus dem Standard-Template, das pandoc für HTML mitbringt. Ein eigenes Template für HTML ist schnell gebaut. Legen Sie die Datei `html_temp.html` im gleichen Verzeichnis wie die Markdown-Datei an:

```
<html>
<head>
<title>$title$</title>
</head>
<body>
<h1>$title$</h1>
$body$
<footer>$copy$</footer>
</body>
</html>
```

Um das Template zu nutzen, hängen Sie den Parameter `--template=html_temp.html` an den pandoc-Befehl an. Die Variable `$body$` ersetzt pandoc automatisch durch den umgewandelten Inhalt der Quelldatei. `$title$` und `$copy$` versucht es aus einem Meta-Block auszulesen, den Sie am Anfang der Datei im YAML-Format zwischen -- und … platzieren:

```
--
title: Testdokument
author:
 - Autor 1
 - Autor 2
copy:Kein Copyright
...
```

Innerhalb dieser Meta-Informationen können Sie sich frei entfalten und komplexere Templates mit mehreren Variablen bauen. Soll ein Element im späteren Dokument nur angezeigt werden, wenn die Variable im YAML-Block gesetzt wurde, können Sie in der Template-Datei If-Abfragen verwenden:

```
$if(author)$
<ul>
$for(author)$
  <li>$author$</li>
$endfor$
</ul>
$endif$
```

Ist die Variable `author` definiert, enthält das generierte Dokument eine ungeordnete Liste mit einem Eintrag für jeden Autorennamen. Bei der Arbeit mit langen Dokumenten, die aus Kapiteln bestehen (wissenschaftlichen Arbeiten oder Dokumentationen), ist ein Inhaltsverzeichnis sinnvoll. Pandoc kann das automatisch aus den Überschriften auslesen und die Hierarchie der Überschriftenebenen darstellen. Im Template legen Sie die Position des Inhaltsverzeichnisses mit folgenden Zeilen fest:

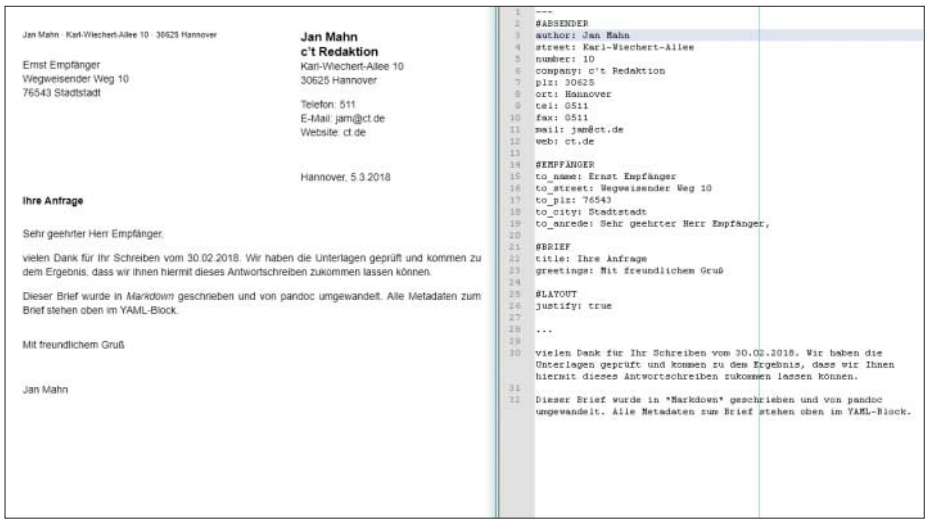
```
$if(toc)$
$toc$
$endif$
```

Beim Aufruf von pandoc führt der Parameter `--toc` dazu, dass das Inhaltsverzeichnis produziert wird. Mit der zusätz-

```
$watcher = New-Object System.IO.FileSystemWatcher
$watcher.Path = ".\incoming\"
$watcher.Filter = "*.md"
$watcher.IncludeSubdirectories = $true
$watcher.EnableRaisingEvents = $true
$action = { $path = $Event.SourceEventArgs.FullPath
 $changeType = $Event.SourceEventArgs.ChangeType
 $date = Get-Date -Format d.M.yyyy
 $dateiname = (Get-Item $path).BaseName

 $output = ".\output\"+$dateiname+".html"
  pandoc $path -o $output --template=./templates/↵
  ↳template_brief.htm --metadata date=$date
}
Register-ObjectEvent $watcher "Created" -Action $action
Register-ObjectEvent $watcher "Changed" -Action $action
Register-ObjectEvent $watcher "Renamed" -Action $action
while ($true) {sleep 1}
PAUSE
```

Der FileSystemWatcher überwacht den Ordner „incoming“ und startet pandoc, wenn er eine neue Datei findet.



Den fertigen Brief hat pandoc aus der Markdown-Datei und einer Vorlage erstellt.

lichen Angabe `--toc-depth=` geben Sie an, bis zu welcher Ebene die Überschriften berücksichtigt werden (ohne Angabe bis zur dritten Ebene, also `###`). Möchten Sie die einzelnen Kapitel in kleineren Einzeldateien schreiben, können Sie beim Export einfach mehrere Markdown-Dateien verbinden. Setzen Sie die Dateinamen in der richtigen Reihenfolge in den Programmaufruf ein, pandoc erledigt die Vereinigung:

```
pandoc -s kapitel1.md kapitel2.md ↵
↳ -o paper.htm
```

Brief an den Vater

Bei einem Geschäftsbrief, der in einem Fensterumschlag verschickt werden soll, kommt es auf die genaue Positionierung des Adressfeldes an. Lässt man den Briefschreiber den Brief in Markdown schreiben, kann pandoc die Platzierung erledigen. Mit etwas CSS entsteht eine leicht anpassbare Vorlage für Geschäftsbriefe nach DIN 5008 als HTML. Unter ct.de/y1nh finden Sie eine fertige Template-Datei, die per CSS alle Elemente in die richtige Position bringt und die gängigen Druckränder automatisch berücksichtigt, außerdem eine Muster-Markdown-Datei für einen Geschäftsbrief. Die Verwandlung starten Sie über die Kommandozeile:

```
pandoc quelldatei.md  -o brief.html ↵
↳--template=template_brief.htm ↵
↳--metadata date='%date%'
```

Um den Brief nicht per Hand datieren zu müssen, nutzen Sie die interne Datumsfunktion des Betriebssystems und übergeben das Datum an pandoc: Unter Windows geht das per ‚%date%', unter Linux und macOS liefert `date +'%d.%m.%Y'` das Datum im europäischen Format.

Der Prozess

Bisher musste die Verwandlung noch manuell angestoßen werden. Noch praktischer wäre es jedoch, einen Ordner „incoming" zu schaffen, in den Sie nur die Rohdatei für den Brief werfen müssen, um ein fertig formatiertes Schreiben zu erhalten. Je nach Betriebssystem müssen Sie anders vorgehen, um pandoc mit den richtigen Parametern auszuführen.

Unter Windows helfen die PowerShell und die Klasse `System.IO.FileSystemWatcher`. Im Kasten auf Seite 169 finden Sie das passende Skript (zum Download unter ct.de/y1nh).

Das Skript geht davon aus, dass es in einem Ordner liegt, in dem die drei Unterordner „templates", „incoming" und „output" liegen. Es reagiert nur auf Dateien mit der Endung „.md". Damit das Programm ständig läuft, können Sie es als geplante Task in Windows einrichten, die bei jedem Start des Rechners anläuft.

Unter Linux benötigen Sie das Programm incron, das Ordnerzugriffe überwacht und mit dem Start eines Programms reagiert. Es ähnelt im Aufbau dem Linux-Klassiker cron, das für zeitgesteuerte Aufrufe zuständig ist. Installieren Sie incron mit

```
sudo apt install incron
```

Die Regeln bearbeiten Sie mit `incrontab -e`. Um alle in /home/joe/post/incoming geworfenen Briefe umzuwandeln, fügen Sie die folgende Zeile ein:

```
/home/joe/post/incoming IN_MOVED_TO ↵
↳pandoc -o /home/joe/post/out/$#.html
```

Mit `$#` gibt incron den Dateinamen der Quelldatei an pandoc weiter. Speichern Sie die Incrontab-Datei und legen Sie einen Brief in den Ordner „incoming". Wenige Sekunden später liegt die verwandelte Datei im Ordner „out".

Unter macOS gibt es die Möglichkeit, Ordneraktionen einzurichten und mit Apples Skriptsprache AppleScript dafür zu sorgen, dass pandoc startet. Diese Sprache ist so angelegt, dass sie für Menschen besonders leicht lesbar sein soll – dadurch wird selbst eine einfache Aufgabe sehr lang. Das Skript haben wir unter ct.de/y1nh veröffentlicht. Laden Sie es herunter und legen es im Ordner „/Library/Scripts/Folder Action Scripts/" ab, damit macOS darauf zugreifen kann. Klicken Sie rechts auf den zu überwachenden Ordner und wählen Sie im Menü „Ordneraktionen konfigurieren". Im Dialog können Sie mit einem Klick auf das „+" die Aktion hinzufügen. Aktivieren Sie außerdem oben links die Ordneraktion.

Das Urteil

Die Arbeit mit pandoc lohnt vor allem, wenn man regelmäßig wiederkehrende Aufgaben erledigen muss. Administratoren können beispielsweise eine firmenweite Umgebung bereitstellen, um Briefe zu produzieren. Der Benutzer legt das Markdown-Textdokument in ein Netzlaufwerk, das Ergebnis wird automatisch gedruckt und in der Poststelle verpackt. Gleichzeitig wandert eine Kopie des Textes ins Archiv.

Auch für wissenschaftliche Arbeiten ist das Programm – in Verbindung mit Markdown – eine reizvolle Alternative zum direkten Schreiben von LaTeX-Dokumenten mit ihrer wenig nutzerfreundlichen Syntax. Zahlreiche Vorlagen gibt es im Internet, auch an die Unterstützung gängiger Bibliographie-Formate und die Ausgabe mathematischer Formeln haben die Entwickler gedacht. Wer E-Books für verschiedene Reader produziert, findet ebenfalls umfangreiche Anleitungen für pandoc. *(jam@ct.de)* c't

Pandoc, Vorlagen, Dokumentation:
ct.de/y1nh