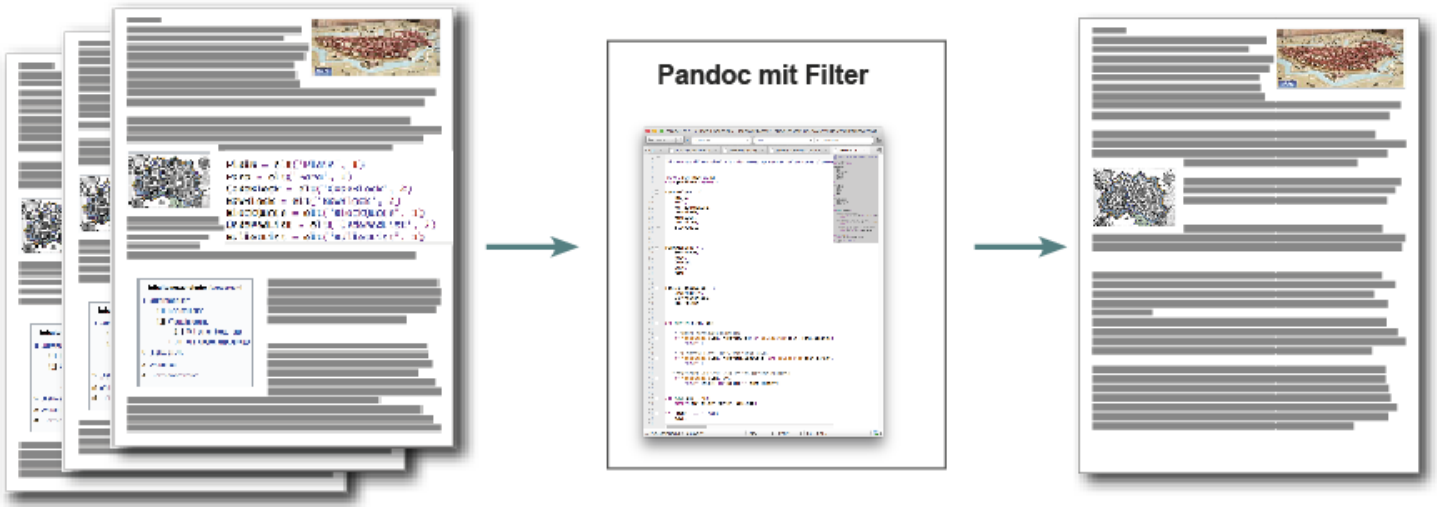
Normalisierung von Dokumentformaten für einfache Beschreibungstexte

Frieda Josi

12. Juni 2017

Aufgabenstellung: Normalisierung mit Pandoc

- Filter für Kommandozeilentool Pandoc entwickeln:
  - Aus Dokumentformate (Word, OpenOffice, HTML, LaTeX, Markdown,..) werden nur Fotos, Abbildungen, und deren Beschreibunsgtexte in das Ausgabeformat übernommen.
  - Der Filter wird mit Python umgesetzt



Import von Word-, HTML-, ... Dateien

Export einer Word-, HTML-, ... Datei

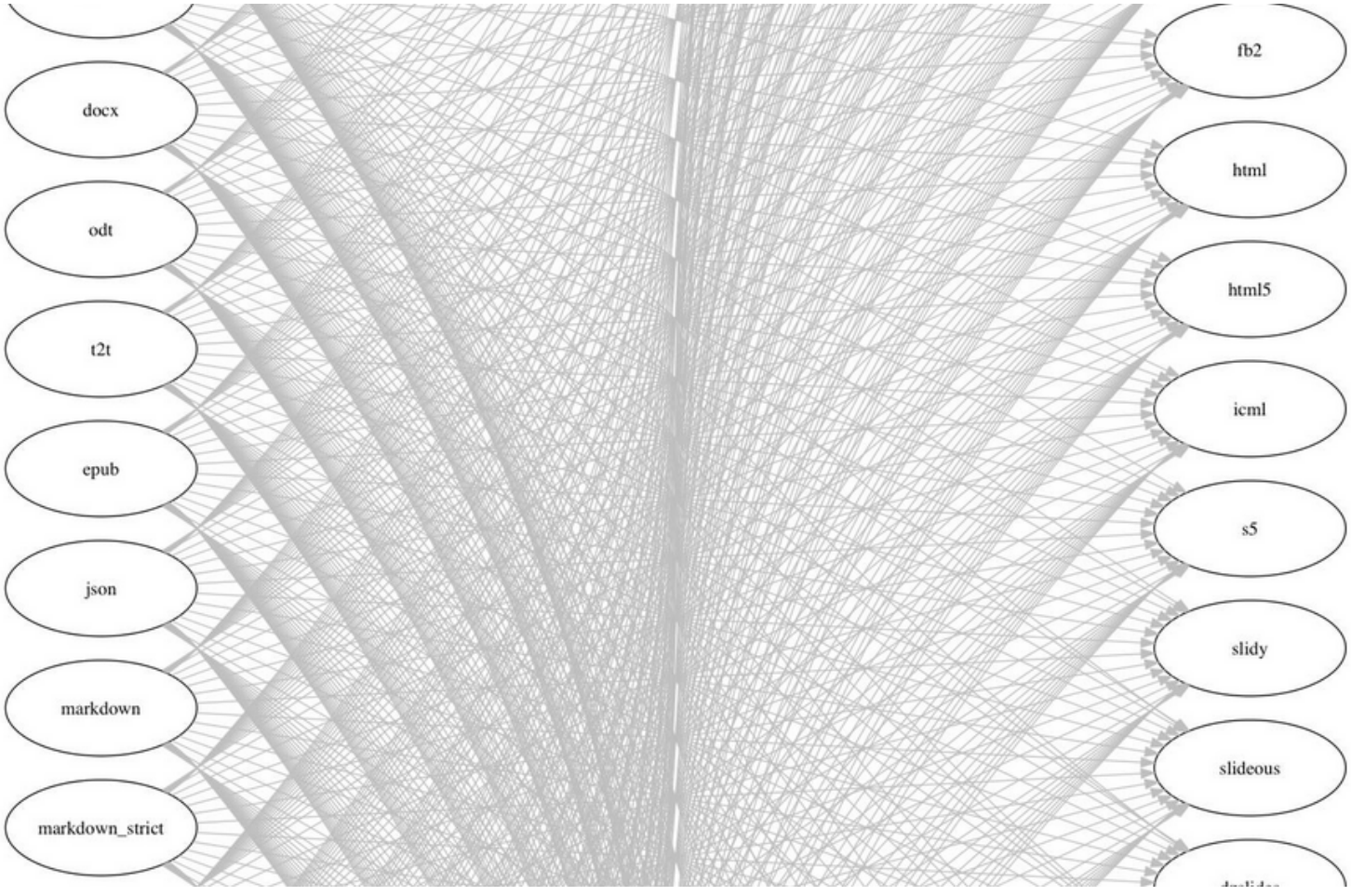
Funktionsweise von Pandoc: Mögliche Dokumentformate

Pandoc ist ein Programm, das viele verschiedene Dokumentformate ineinander umwandeln kann.

Lesen/Importieren:	**Schreiben/Exportieren:**
Markdown, CommonMark, reStructuredText, textile, MediaWiki, HTML, DocBook, LaTeX, Microsoft Word DOCX (OOXML), LibreOffice/OpenOffice ODT, EPUB, TWiki, Org-mode	HTML-Formate, Textverarbeitungsformate (Microsoft Word, LibreOffice/OpenOffice ODT,...), Ebooks, Dokumentationsformate (DocBook,...), Seitenlayoutformate (z.B. InDesign), TeX-Formate (z.B. LaTeX), PDF via LaTeX, Vereinfachte Auszeichnungssprachen (z.B. CommonMark, Markdown, reStructuredText, ...)

Quellen: `https://de.wikipedia.org/wiki/Pandoc` **und** `http://pandoc.org/`

Funktionsweise von Pandoc: Ausschnitt Dokumentformate



Quelle: `http://pandoc.org/`

Funktionsweise von Pandoc: Markdown

- Pandoc basiert auf einer Erweiterung der Auszeichnungssprache Markdown.
- Bei der Umwandlung sollen Strukturelemente eines Dokuments erhalten bleiben, nicht ihre Formatierung. Problematisch sind aber komplexe Tabellenstrukturen.

```
# Überschrift in Ebene 1
#### Überschrift in Ebene 4

> Dies ist ein Zitat mit zwei Absätzen. Lorem ipsum dolor sit amet,
> consectetuer adipiscing elit.
>
> Donec sit amet nisl. Aliquam semper ipsum sit amet velit.

![nur ein Beispiel](https://commons.wikimedia.org/wiki/File:Example_de.jpg "Beispielbild")

1.  Ein Punkt in einer geordneten Liste
1.  Hund
    1. Hund im Hund 1
    2. Die Katze im Hund 2
    3. Die Katze im Hund 3

Jetzt eine horizontale Linie
---------------------------
Ein normaler Absatz, mit wenig Text. Normaler Text wird so dargestellt wie eingegeben.

Eine Leerzeile erzeugt einen Absatz.

    Ein Code-Block
    durch Einrückung mit vier Leerzeichen
```

Quellen: https://de.wikipedia.org/wiki/Markdown und https://github.com/jgm/pandoc

Funktionsweise von Pandoc: Umwandlung

- Pandoc besteht aus einer Haskellbibliothek und einem eigenständigen Kommandozeilenprogramm.
- Diese Bibliothek besteht aus Modulen für jedes Eingabe- und Ausgabeformat. Es können neue Dokumentformate über neue Module hinzugefügt werden.
- Es gibt verschiedene Kommandozeilen Lese- und Schreiboptionen (hier -s für das Lesen der Importdatei (smart) und -o für die Exportdatei und auch das Ausgabeformat (output).

Pandoc verwenden über Terminaleingabe:
pandoc -s ausstellung.md -o ausstellung.html

Quellen: `http://pandoc.org`

Pandoc Filter Funktionsweise

Pandoc bietet Schnittstelle für Einfluss auf Syntaxbaum der Zwischendarstellung. Dafür werden Filter (z.B. in Python, Perl, Haskell, . . . geschrieben) eingesetzt.

```
INPUT --reader--> AST --filter--> AST --writer--> OUTPUT
```

- Das Eingabedokument wird von Pandoc in eine Zwischendarstellung (AST) gelesen,
- der Filter nimmt Einfluss darauf (z.B. Alle Überschriften werden mit Grossbuchstaben geschrieben),
- dann wird modifizierter Syntaxbaum erstellt,
- danach erfolgt das Schreiben des Zieldokumentformates.

Terminaleingabe: pandoc -s ausstellung.md –filter remove.py -o ausgabe.html

Quelle: `http://pandoc.org/scripting.html`

Pandoc-Elemente

Erstellte Übersicht als Vorbereitung für den Filter.

Elemente, die gelöscht werden:	**Elemente, die bleiben:**	**Elemente, die teilweise bleiben:**
CodeBlock, RawBlock, BlockQuote, HorizontalRule, Table, Null, Strikeout, Quoted, Code, Math, RawInline, Note	Plain, Para, LineBlock, Header, Div, Str, Emph, Strong, Superscript, Subscrip, SmallCaps, Cite, Space, SoftBreak, LineBreak, Link, Image, Span	OrderedList, DefinitionList, BulletList

Quelle: `http://hackage.haskell.org/packagepandoc-types-1.17.0.5/docs/Text-Pandoc-Definition.htm`

Eigener Filter Teil 1

Der erstellte Python-Filter entfernt alle nicht gewollten Elemente.
Datei: remove.py

```
import panflute as pf
from panflute import *

removeBlock = (
    Table,
    Null,
    HorizontalRule,
    CodeBlock,
    RawBlock,
    RawInline,
    BlockQuote )

removeInline = (
    Strikeout,
    Note,
    Quoted,
    Code,
    Math )

removeNestedLists = (
    OrderedList,
    DefinitionList,
    BulletList )
```

Eigener Filter Teil 2

```
def action(elem, doc):
    # remove forbidden elements
    if isinstance(elem, removeBlock) or isinstance(elem, removeInline):
        return []

    # no nested lists, only the first level
    if isinstance(elem, removeNestedLists) and isinstance(elem.parent,
                                                          pf.ListItem):
        return []

    # transform Div into list of its content elements
    if isinstance(elem, Div):
        return [block for block in elem.content]

def main(doc=None):
    return run_filter(action, doc=doc)

if __name__ == '__main__':
```

Durchführung

Terminaleingabe:

pandoc -s ausstellung.md –filter remove.py -o ausstellung.html

Import Markdowndatei -> Filter Einsatz -> Export HTML-Datei

Import Markdown

ausstellung.md | ausgabe.html | Definition.hs

```
---
title: Hinweistafeln die in der Ausstellung  **Mi
...
Weit hinten, hinter den Wortbergen, fern der Länd

Weit hinten, hinter den Wortbergen

Weit hinten, hinter den Wortbergen
Weit hinten, hinter den Wortbergen

| was  | wo  |  wie | warum  | wieso  |
|---|---|---|---|---|
|  Nichts |  Noch weniger | Gar nichts  | Gar nic
|  Nichts |  Noch weniger | Gar nichts  | Gar nic

> Dies ist ein Zitat mit zwei Absätzen. Lorem ips
> consectetuer adipiscing elit.
>
> Donec sit amet nisl. Aliquam semper ipsum sit a
> Suspendisse id sem consectetuer libero luctus a

1.  Hund
    1. Hund im Hund
    2. Katze in Katze
    3. Katze in Katze
1.  Katze
```

Export HTML

ausstellung.md | ausgabe.html | Defini

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XH
<html xmlns="http://www.w3.org/1999/x
<head>
  <meta http-equiv="Content-Type" con
  <meta http-equiv="Content-Style-Typ
  <meta name="generator" content="pan
  <title>Hinweistafeln die in der Aus
  <style type="text/css">code{white-s
</head>
<body>
<div id="header">
<h1 class="title">Hinweistafeln die i
</div>
<p>Weit hinten, hinter den Wortbergen
<p>Weit hinten, hinter den Wortbergen
<p>Weit hinten, hinter den Wortbergen
<ol style="list-style-type: decimal">
<li>Hund</li>
<li>Katze</li>
<li>Maus</li>
</ol>
<p>Noch mehr einfacher Text. Noch meh
<ul>
<li>Lorem ipsum dolor sit amet, conse
<li>Donec sit amet nisl. Aliquam semp
</ul>
```

Quellen

Pandoc: http://pandoc.org/
Pandoc Github: https://github.com/jgm/pandoc
Pandoc Filters:
https://github.com/jgm/pandoc/wiki/Pandoc-Filters
Pandoc Panflute: https://github.com/sergiocorreia/panflute
Pandoc Optionen: http://pandoc.org/MANUAL.html#options
Pandoc Elemente:
http://hackage.haskell.org/packagepandoc-types-1.17.0.5/docs/Text-Pandoc-Definition.html
Pandoc online testen: http://pandoc.org/try/
Pandoc Filter schreiben: http://pandoc.org/scripting.html

Ergänzung

Umwandeln einer PDF-Datei zu Word, HTML,... geht mit Pandoc nicht.

Alternative: Acrobat Pro DC

- PDF -> Word, Excel (Sehr guter Export)
- PDF -> HTML (Sehr guter Export)
- PDF -> RTF (Formatierung kaum übernommen)
- PDF -> Text (Sehr guter Export, aber ohne Formatierung)
- PDF -> Einzelne Bilder in jeweils eine separate Bilddatei exportieren

Quelle: `https://helpx.adobe.com/de/acrobat/using/file-format-options-pdf-export.html#file_format_options_for_pdf_export`, **27. März 2017**