Dokumenten-KungFoo für Hacker mit Pandoc Filter

David-Elias Künstle

Einmal hin...

```
Fachschaftssitzung KW 22
========================

Top 0: Post
-----------

- Hackaton in Karlsruhe
- Wer hat die Kokosnuss?


Top 1: tübix
------------

**Helfer gesucht!**
Wir brauchen $\frac{2^9}{3}$
Würstchen!
```

Einmal hin...

```
pandoc protokoll.md
       --output=protokoll.pdf
```

Einmal hin. . .

Fachschaftssitzung KW 22

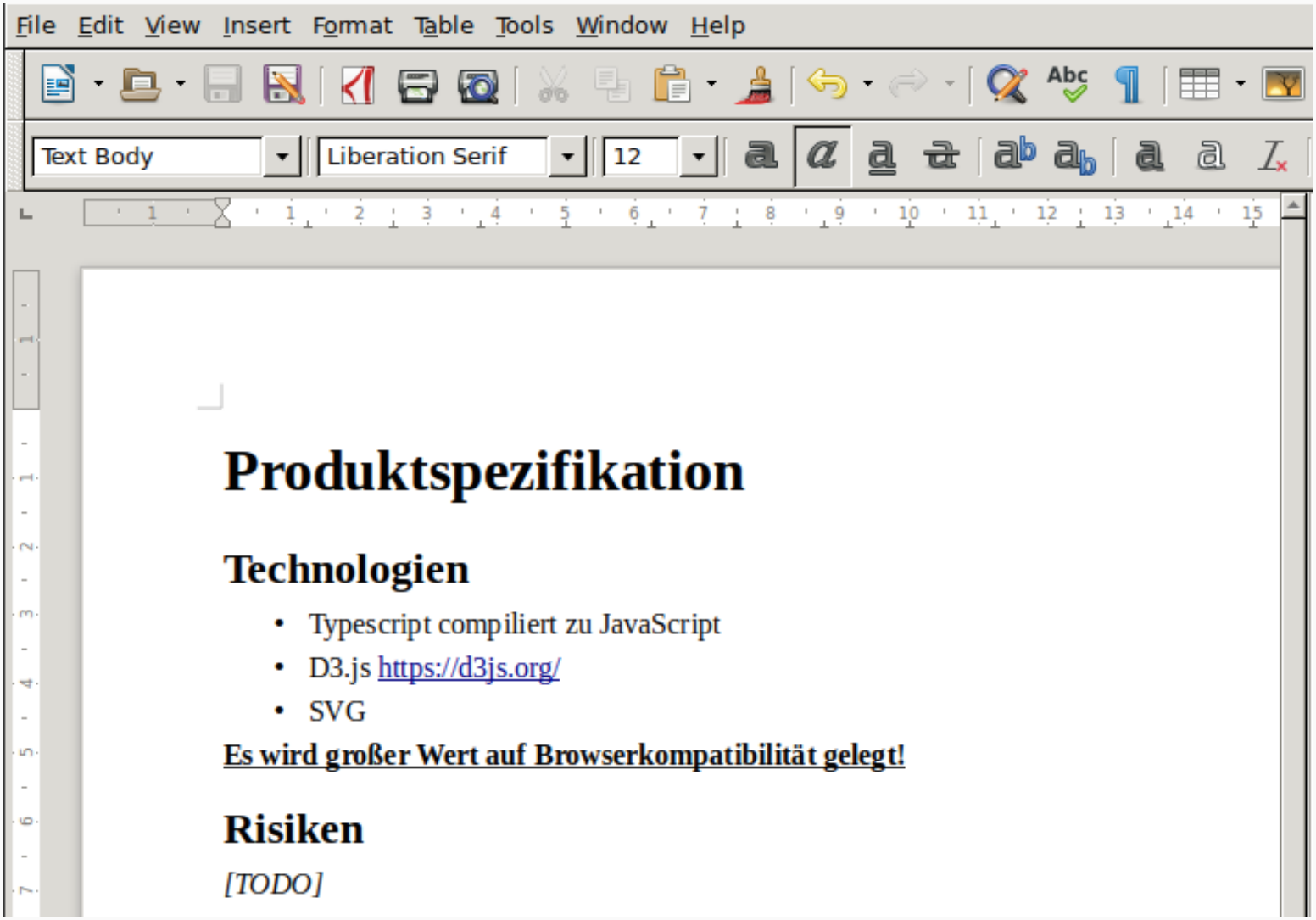
Top 0: Post

- Hackaton in Karlsruhe
- Vertreter für Kommission gesucht

Top 1: tübix

Helfer gesucht! Wir brauchen $\frac{2^6}{2}$ Würstchen!

. . . und wieder zurück!



. . . und wieder zurück!

```
pandoc spezifikation.docx
       --output=spezifikation.md
```

. . . und wieder zurück!

```
Produktspezifikation
====================

Technologien
------------

-   Typescript compiliert zu JavaScript
-   D3.js <https://d3js.org/>
-   SVG

***Es wird großer Wert auf Browserkompatibilität gelegt!***

Risiken
-------

*\[*TODO*\]*
```

Wie geht's?

$$\text{Reader} \rightarrow \boxed{\text{AST}} \rightarrow \text{Writer}$$

Rein in den Kaninchenbau!

```
## 2. Inspection of the response variable
``` `{r}
summary(dat$Response)
range(dat$Response)
mean(dat$Response)
``` `
```

What is the range of the response variable?

The response variable contains values between 0.68 and 2.98.

What is the mean?

2.05

```
``` `{r, fig.width=4, fig.height=4}
avg <- tapply(dat$Response,
...
```
```

Rein in den Kaninchenbau!

2. Inspection of the response variable

```
summary(dat$Response)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.680   1.355   2.300   2.049   2.680   2.980
```

```
range(dat$Response)
```

```
## [1] 0.68 2.98
```

```
mean(dat$Response)
```

```
## [1] 2.0487
```

What is the range of the response variable? The response variab

What is the mean? 2.05

```
avg <- tapply( dat$Response,
               list( dat$AgeGroup ),
               mean )
```

Rein in den Kaninchenbau!

Reader $\rightarrow$ $\boxed{\text{AST}}$ $\rightarrow$ JSON-Writer $\rightarrow$ Filter-Script $\rightarrow$
JSON-Reader $\rightarrow$ $\boxed{\text{AST}}$ $\rightarrow$ Writer

```
pandoc --to=JSON my_file.txt \
| magicfilter.py \
| pandoc --from=JSON -o my_file.html
```

Beispielfilter

```
from pandocfilters import toJSONFilter, Emph, Para

max_header_depth = 2

def flat_head(key, value, format, meta):
  if key == 'Header' and value[0] > max_header_depth:
    return Para([Emph(value[2])])

toJSONFilter(flat_head)
```

Libraries

- Haskell (nativ pandoc)
- Python: pandocfilters
- Python: panflute
- PHP: pandocfilters-php
- Node.js: pandoc-filter-node
- Perl: Pandoc::Elements
- Groovy: groovy-pandoc
- ...