

HTML Folien, mslide

Motivation

Im Rahmen der Linux User Gruppe halte ich immer wieder Vorträge. Graphische Werkzeuge wie LibreOffice Impress (Unter Windooof gibt es etwas analoge, ich glaube Power Point) sind nicht unbedingt meine Lieblinge zumal eine passende Software zum Abspielen der Präsentation vorhanden sein muss.

Eine Präsentation sollte auf alle möglichen Geräte angeboten werden können. Es bedeutet, dass es soweit wie nur möglich die Eigentümlichkeiten der Abspielgeräte Rücksicht nehmen.

Vorträge sollten auch verteilt werden, nur Folien auszuhändigen ist nicht immer ausreichend, es wäre Gut eine passende Dokumentation mit zu liefern. Dies würde bedeuten, dass 2 Dokumenten gepflegt werden, oder dass die Vortragsunterlagen bereits dir Erklärungen zu den einzelne Folien beinhalten, in der Folien jedoch nicht sichtbar sind.

Aus der vorbereite Unterlagen sollten die Folien sowie ein geeigneten Dokument, HTML Seite und oder PDF Datei.

1) Warum HTML

- HTML Code besteht aus der eigentliche Informationen und
- Aus der Layout Vorgaben (CSS).
- HTML benötigt nur ein Browser, diesen sind auf praktisch jeden Gerät verfügbar.
- HTML Seiten können dynamisch sein, über Tastatur oder Maus bzw. Berührung des Bildschirmes, bedient werden.
- Aus eine HTML Datei kann ein ausdrucksbaren Dokument erzeugt werden.

2) HTML und Ausgabe Medien

Beamer, Computer Bildschirme, Mobile Geräte, Drucker sind unterschiedlich.

2.1) Beamer

- Je nach Alter sind Format und Auflösung unterschiedlich:
- 800 x 600 Bildpunkten, Seitenverhältnis 4:3
- 1024 x 768 Bildpunkten, Seitenverhältnis 4:3
- 1920 x 1080 Bildpunkten. Seitenverhältnis 16:9

Beim Anschluss am PC/Notebook, wird das Format richtig erkannt.

2.2) PC Bildschirm

Hier ist die Welt vielfältiger!

- Übliche Seitenverhältnisse:
 - 16:9
 - 3:2
 - Sonder Formate wie 21:9
 - ...
- Breite in Pixel ?
 - 1368
 - 1920

- 3840
- ...

2.3) Fernseher

Sie können auch als Ausgabegerät dienen.

- Format oft 16:9 Größe wird beim Anschluss am PC erkannt.

Dies geht beispielsweise mit "Smart View". Es ist eine Funktion, die zu erreichen ist, wenn die Schnelleinstellungsleiste vollständig geöffnet ist. es sollte sich auf eine der Blätter befinden.

2.4) Drucker

- In Europa können die meisten Drucker DIN A4 Seite bedrucken.
- Es können Farb- oder Schwarz/Weiß Drucker sein.

Eine korrekte Ausgabe Formatierung, Kopf und Fußzeile sowie Ränder bedingte Seitenumbrüche (Schuster-Jungen Huren Kinder usw. müssen beachtet werden.

Alle diese Anforderungen können berücksichtigt werden, diese CSS-Regeln werden leider nicht von den Browser verarbeitet.

Mit Weasyprint, ein in Python geschriebenen Programm können jedoch, ausgehend von ein HTML Seite, PDF-Dokumente erstellt werden.

3) Impress PowerPoint und co.

Ob es es eine gute Idee ist?

Folien mit PowerPoint und co. werden nur skaliert, dies kann dazuführen, dass Texte unlesbar werden.

4) Aufbau einer HTML Datei

```
HTML
Kopfbereich
  Metadaten
  Style (CSS)
  Javascript
Körper
  Html Code
```

Diese stark vereinfachte Darstellung zeigt, dass die gesamte Seite aus Blöcke besteht, es ist leicht, ausgehend von Bruchstücke eine vollständige Seite aufzubauen, es ist das Verfahren, der von mslide verwendet wird.

Ein einfache Script schreibt die einzelnen Bruchstücke nacheinander in der Zielfeile.

Der Vorgang wird durch ein Makefile unterstützt, der erschienene Parameter zu deklarieren und die Dateien erzeugt, wenn die Quelle, eine Markdowndatei sich verändert hat.

5) Eigene Anpassungen.

- zusätzliche CSS Dateien können eingebunden werden
- Auch zusätzliche Javascript code kann eingebaut werden.

5.1) Eigene Anpassungen, CSS

Eintragung in der Markdown Datei:

```
::: {.col2 .h45}
! [Bild 1] (DSC00476.JPG) {.foto}

! [Bild 2] (DSC00487.JPG) {.foto}
:::
```



Bild 1



Bild 2

Hier werden 2 Bilder nebeneinander platziert. und als Papierbilder mit weißen Ränder dargestellt. Die leere Zeile zwischen beide Bilder Deklarationen ist wichtig!

5.1.1) CSS Code

```
1 | .imgCol2 p {
2 |     display: block;
3 |     column-count: 2;
4 |     column-gap: 1em;
5 |     margin-top: 0;
6 | }
7 | .imgCol2 {
8 |     background-color: transparent;
9 | }
10 | .foto {
11 |     border: .5em solid #ffffff;
12 | }
```

.imgCol2 und .foto sind sogenannten CSS Klassen.

Im Markdown Text bedeutet **:::** dass ein Element vom Typ DIV eingefügt wird, das zweites Auftreten schießt dieser Block. **{.foto}** sorgt dafür, dass er Mitglied der Klasse .foto wird.

Das nachfolgend eingefügten Element ist ein Paragraf- Diese wurden als nicht sichtbar deklariert es muss aufgehoben werden. Das Paragraph, hier im Beispiel wird mit 2 Spalten versehen, ein Abstand zwischen den Bildern wurde auch vorgesehen.

.imgCol2 ist normalerweise (bei normale Folien mit eine Farbe versehen, die nicht, diejenige des normaler Hintergrund entspricht, die Farbe wird auf transparent gestellt.

Schließlich werden mit .foto die Rahmen der Bilder deklariert.

Schlecht, dabei, ist jedoch, dass solch ein Konstrukt nicht sauber von Pandoc umgesetzt wird (Bilduntertitel fehlt).

5.1.2) Einbinden der dedizierten CSS Dateien

Im Makefile kann die Datei als Argument gegeben werden, dies dürfte nicht so schwierig sein.

```
1  # Globale Definitionen
2  D=../TEMPLATES/mslide
3  TITEL='HTML-Vortrag'
4  F=HTML-slides
5  LANG=de
6  AUTOR='Max Mustermann'
7
8  all: $F.pdf $F.html slide.html
9
10 $F.pdf: $F.html
11     weasyprint $F.html $F.pdf
12
13 $F.html: $F.md
14     $D/build-html.sh -c $D/docmain.css,doc2pictures.css \
15     -i $F.md -o $F.html -I -t ${TITEL} -l ${LANG} -a ${AUTOR}
16
17
18 slide.html: $F.md
19     $D/build-slide.sh -c $D/main.css,$D/color.css,$D/nop.css,$D/anim.css,2pictures.css \
20     -j $D/mslide.js -i $F.md -o slide.html -I -t ${TITEL} -n -l ${LANG} -a ${AUTOR}
21
22 clean:
23     rm $F.html slide.html $F.pdf
```

Zeilen 15 bzw. 20 wurden mit unseren “private” CSS Dateien versehen:

doc2pictures.css (Dokumente) bzw. **2pictures.css** (Vortrag).

6) Präsentation am Fernseher vom Smartphone aus

- Auf der Smartphone muss ein webserver installiert und gestartet werden:
 - <https://play.google.com/store/apps/details?id=com.phlox.simpleserver>
- Relevante Dateien in ein passenden Ordner auf das Smartphone kopiert.
- Verbindung zum Fernseher, beispielsweise mit “Smart View” (Samsung)

Der Webserver ist wichtig, Bildmaterial wird im Browser des Smartphone sonst nicht geladen. Der Webserver kann beim Bedarf gestartet werden.

Da die Quelle unseren Smartphone ist, muss es möglich sein das Bildschirm des Gerätes am Fernseher darzustellen, muss das Wiedergabegerät über einer WIFI Verbindung verfügen und “Screen Mirroring” unterstützen. Im entsprechenden Stores sollten geeignete Applikationen zu finden sein.

7) Präsentation auf fremden PC

- Präsentation auf USB-Stick
- Präsentation auf den Smartphone
 - Als Stick einbinden
 - Über Webserver Folien zur Verfügung stellen
- Präsentation auf ein eigenen öffentlichen Webserver

Voraussetzung sind immer ausreichende Rechte, Einbinden von externe Medien, Zugriff auf externe Webpräsenzen, WIFI-Verbindung wählen.

In manche Umgebungen sind Rechner stark geschützt, ein USB-Stick könnte Schadsoftware beinhalten, daher ist das Anschließen einen externe Medium unzulässig.

Manchmal wird auch, aus Sicherheitsgründen der Zugriff auf externe Webdienste verwehrt, Zugriffe auf interne Dienste ist allerdings vorhanden. Im letzteren Fall könnte das Smartphone als Webserver dienen, es setzt aber voraus, dass ein Zugriff auf das interne WLAN möglich ist, dies könnte auch eine Sicherheitslücke Bedeuten.

Ein Smartphone kann als Hot-Spot eingerichtet werden, der fremden Rechner kann, wenn es von den Administratoren nicht mit Absicht unterbunden, sich mit unsere Smartphone verbinden.

Irgend eine der Voraussetzung dürfte meistens vorhanden sein, wenn nicht Pech gehabt, eigener Notebook beim nächsten Mal schleppen!

8) Linux. ich verwende doch Windows!

Selbstschuld!

- wls installieren
 - debian vom Windows Store installieren.
 - sudo apt update
 - sudo apt upgrade
 - sudo apt install make
 - sudo apt install pandoc
 - sudo apt install weasyprint
- Wie gewohnt arbeiten

8.1) Windows: Texte Bearbeiten

- Ein echter Text Editor sollte installiert sein
 - Notepad++
 - Geany

Notepad++ ist unter Windows weit verbreitet. Alternativ kann auch Geany auf der Windows Rechner installiert werden. Beide Editoren kennen sich mit Markdowndateien aus, es kann die Arbeit erleichtern.

Von Geany aus können die Dateien erzeugt werden, dies kann von Vorteil sein, editieren und erzeugen werden einfacher.

8.2) Windows Erzeugung der Dateien

wsl make

Wsl muss auf Version 1 eingestellt werden, damit kann auf die Verzeichnisse und Dateien des Windows Systems zurückgegriffen werden.

die Datei beispielsweise **make.bat** in der reicht aus um die Erzeugung der Dateien zu starten.

Wsl startet eine bash Fenster und ruft anschließend **make** auf.

8.2.1) Von Geany aus erzeugen

Kommandos zum Erstellen konfigurieren

#	Label	Kommando	Arbeitsverzeichnis	Zurücksetzen
Kommandos für Markdown				
1.	HTML + PDF erzeugen	ws! make		
2.	HTML + PDF löschen	ws! make clean		
3.				
Regulärer Ausdruck für Fehlermeldungen:				
Dateitypunabhängige Befehle				
1.	Make	make		
2.	Make (eigenes Target)...	make		
3.	Make Objekt-Datei	make %e.o		
4.				
Regulärer Ausdruck für Fehlermeldungen:				
<i>Notiz: Element 2 öffnet ein Dialog und fügt das Ergebnis am Ende des Kommandos an</i>				
Befehle zum Ausführen				
1.	Ausführen	./%e		
2.				
<i>%d, %e, %f, %p, %l werden innerhalb der Kommando- und Verzeichnisfelder ersetzt - Details gibt es in der Dokumentation.</i>				
				<div>AbbrechenOK</div>

Erstellen → Kommandos zum Erstellen konfigurieren

Im Übrigen, ist es unter Linux nicht anders wirklich, die Kommandos sind bis auf das Zauberwort `ws!` identisch.