E-LEARNING mit *decker*

…für nicht-Programmierer*innen

Zusammengestellt von Valerie Kiendl
(Stand: Dez 2020)

INSTALLATION

- **Decker** lässt sich auf der Homepage herunterladen und installieren. https://elearning.uni-wuerzburg.de/decker/

- Daneben brauchen Sie ein **Eingabeaufforderung** „terminal" (Mac) / „powershell" (Windows)
- Sowie ein **markdown-basiertes Textverarbeitungsprogramm**: dieses können Sie kostenfrei hier herunterladen: https://www.sublimetext.com/3 ***sublime Text*** (wenn Sie Erfahrungen mit LaTex etc haben, kennen Sie die Textverarbeitungssprache markdown)

Wenn diese drei Programme vorhanden sind, kann es losgehen.

a) Öffnen Sie Terminal/Powershell: Zum Start müssen Sie sich auf Ihrem Desktop befinden. Die Eingabereihenfolge im Terminal sieht dann wie folgt aus: cd steht für change document und navigiert Sie durch die Programme, es ist quasi das, was man sonst mit der Maus tun würde; ausgeführt wird immer mit [enter-Taste]:
b) **cd desktop** (Nötiger Schritt bei Mac)
c) **decker example** (>ein neuer Decker-Foliensatz wird erstellt, der Ordner „example" befindet sich auf dem Desktop)
d) **cd example** (>nun befinden Sie sich im Ordner „example")
e) **decker server** (>nun kann der Foliensatz im Browser angesehen werden)
f) **Live-Modus im Browser**: dazu gibt man in die Browserleiste http://localhost:8888/ ein.
> So funktioniert später auch der Live-Bearbeitungsmodus; in sublime geschriebene Foliensätze können jederzeit parallel angesehen und so kontrolliert werden: dazu einfach die Änderung in sublime mit strg+s speichern, der browser aktualisiert sich automatisch, ansonsten refresh drücken)

Tipp:
Man kann sich auch folgenden Tutorial-Ordner erstellen lassen, darin sieht man, was man alles mit welchen Codes machen kann.
Gehen Sie dazu wie folgt vor:
Öffnen Sie das Terminal/Powershell
cd desktop
decker tutorial (nun wird ein Beispiel-Ordner „tutorial" auf dem Desktop erstellt.)
Gleiches Spiel wie oben, um sich die Decks im Browser ansehen zu können:
cd tutorial
decker server

(Um keine Probleme zu bekommen, beenden Sie Ihr laufendes Eingabefenster und öffnen Sie immer ein neues terminal/powershell)

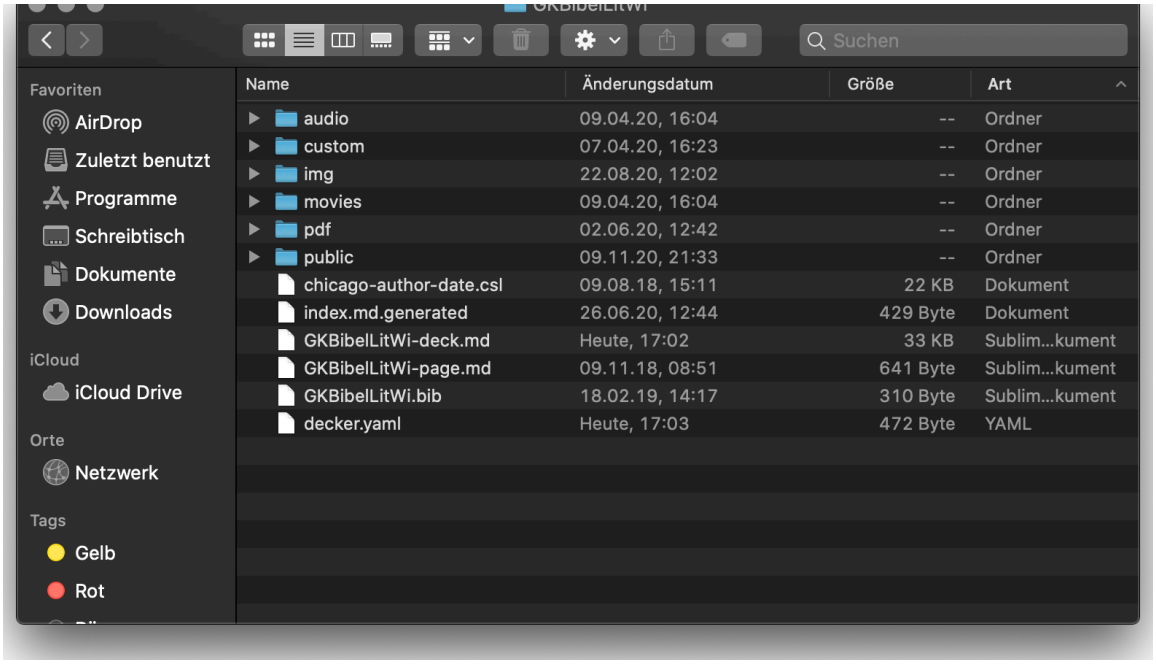
ARBEITEN MIT MARKDOWN

Der automatisch generierte exmaple-Ordner kann nun individuell überschrieben werden.

(Um-)Benennung des example-Ordners:
Der Ordner und die Dateien -deck.md und die -page.md und .bib müssen gleich benannt werden, um sich aufeinander beziehen zu können.

Ein Blick in den Ordner:

Im Ordner befinden sich nun mehrere Dateien.



1) Hauptdatei ist die -deck.md – Datei. Sie sollte so heißen, wie auch der Ordner an sich. Hierin wird der Content geschrieben.
2) Die .yaml-Datei legt das Grundlayout und die Basisprogrammierung fest.
3) Die Medien-Ordnerstruktur im Ordner für eingebaute Medien ist wichtig, denn damit werden später Pfade geschrieben, auf die das Deck dann zugreift. Auch innerhalb der Ordner ist es zu empfehlen, die Dateinamen konkret und übersichtlich zu benennen, um sie schneller wiederzufinden: Bspw.
   -img für Bilder
   -movies für Filme
   -audio für Tonaufnahmen und Musik
   -pdf (habe ich mir zusätzlich angelegt, sollten fixe PDF-Dateien integriert werden)
4) Der custom-Ordner mit den Dateien custom.css (darin kann individualisiert werden) und custom.md (darin befindet sich unsere Selbsttest-Ampel)

Der custom-Ordner für individualisiertes Layout
Textgrößen, Farben, etc.
Kann durch ein custom-style.css angelegt und definiert werden.
Dieses kann im yaml-Ordner hinterlegt werden. Dann greift das Deck darauf zu.

Textgrößen anlegen
Textgrößen können im custom.css individuell bestimmt werden

```
35
36  h1 {
37    font-size: 25pt !important;
38  }
39
40  h2 {
41    font-size: 20pt !important;
42  }
43
44  h3 {
45    font-size: 18pt !important;
46  }
47
48  p {
49    font-size: 15pt !important;
50  }
51
52  li {
53    font-size: 15pt !important;
54  }
55
```

- h1, h2, h3 steht für die Überschriften # ## ###
- p meint den Fließtext
- li steht für Text in Listen
- Weitere Textgrößen können hier bestimmt werden und müssen dann mit bei den Textelementen als Style angegeben werden.

Farben anlegen
Neben den vordefinierten Farboxen können weitere Farben angelegt werden.
Die Zuweisung funktioniert analog zu {.alert}, also {.individuelleFarbe}
Die Farbe lässt sich einfach unter
https://www.hexcolortool.com/?fbclid=IwAR35Y3dmEoxK9fJuBeP7ZdCcKsVLZTHCMtLOzYxB0DqL9SPwR-r8M5MbdC0#40baa4,0.5
mischen und der rgba (also rot grün blau Mischung samt Intensität) übernehmen.

```
55
56  .petrol {
57    background-color: rgba(46, 112, 122, 1);
58  }
59
60  .carneol {
61    background-color: rgba(153, 0, 0, 0.8);
62  }
63
64  .violet {
```

EINFACHE TEXTBEFEHLE

Aufbau eines Slides

```
# h1 : (Überschrift erster Ordnung), öffnet neues Slide
## h2 : (Überschrift zweiter Ordnung)
Texttexttexttext kann einfach so frei eingegeben werden

Text-Hervorhebungen

**bold**
__bold__
*italic*
_italic_
~~durchgestrichen~~
<u>unterstrichen</u>
H~2~O      (tiefgestellt)
CO^2^      (hochgestellt)

<br> steht für einen Zeilenbruch

Es gibt eine Reihe von Klassifizierungen, die ein Slide gestalten und
immer in { } hinter den jeweiligen Text gestellt werden.
Auch wenn sich Blöcke dann über den Slide verteilen, wird immer
untereinander geschrieben.
```

Vertikale Slides

```
# vertikales Slide {.sub}
das {.sub} am Ende bedeutet, dass sich die Folie in die Vertikale
aufschlägt.
```

Spalten / Felder / Raster

```
# Ich baue mir Spalten/ein Raster {.grid} (war vorher: {layout=„columns“})
## links oben {.top-left}
## mitte oben {.top}
## rechts oben {.top-right}
## linke Spalte {.left}
## Sandwich-Spalte {.center}
## rechte Spalte {.right}
## links unten {.bottom-left}
## mitte unten {.bottom}
## rechts unten {.bottom-right}

> es können aber auch nur die genommen werden, die man befüllen will.
```

Farbige Boxen

In der neueren Decker-Version wurden Markierungen minimalistischer; sie erscheinen nur noch als farbige Balken links neben der markierten Stelle. Wenn die farbigen Boxen wieder aktiviert werden sollen, dann muss dies im custom-Ordner geschehen.
(In den Vorlage-Ordnern habe ich beides zur Verfügung gestellt, custom-Boxen ist mit Boxen, custom ist ohne Boxen)

Boxen werden aktiviert, indem der jeweilige Abschnitt ## oder ### mit {.Farbname} versehen wird.

MEDIEN

1. Bild/Video/Audio

Ob Bild, Video oder Audio eingefügt werden soll,

```
![BUoderkeineBU](img/FotonameGenauWieImOrdner.jpg){width=70%}
```

Zur Erklärung:
! Hier wird ein Medium eingefügt
[hier kann eine Bildunterschrift eingefügt werden] sonst leer [] lassen
(hier wird der Pfad eingefügt)
img/ heißt im Ordner img=Bilder suchen
FotonameGenauWieImOrdner.Dateityp
Es können jpg oder png oder gif oder tif eingefügt werden, es muss nur übereinstimmen
{width=70%} gibt die proportionale Breite des Bildes an, dieses kann beliebig angepasst werden.

Genauso wird mit Videos verfahren:

```
![BUoderkeineBU](movies/filmdateinameGenauWieImOrdner.mp4){width=50%}
```

Bilder, die auf der Titel-Seite im Header verwendet werden, müssen per Größenkorrektur angeglichen werden; es empfiehlt sich: 1000px breit, 350px hoch; dann bleibt noch genug Platz für Daten zu Kurs und Autor*in.

2. Youtube-Link

```
![Titelodernicht](youtube:youtube-ID){width=70% start="178" end="210"}
```

Bei der Seite https://www.youtube.com/watch?v=VKpzgC_Bhik
Ist das nach v= die gebrauchte ID: VKpzgC_Bhik

Die genauere Bestimmung in {} gibt wieder folgendes an:
width Proportionale Größe wie bei den anderen Medien
start kann den genauen Startpunkt eines Videos angeben, muss in Sekunden angegeben werden
end kann das Video an beliebiger Stelle enden lassen, ebenfalls in Sekunden angegeben

QUIZZES

Die **Quizzes** haben eine neue einheitliche Syntax - [ ], außer die Match-Aufgabe; die Quiz-Typen werden durch Klassifizierung auf Ebene von H2 ## „eingestellt"

Bei ## wird die Quiz-Art definiert.
Bei ## kann gleich die Frage eingefügt werden, oder die Frage steht in der nächsten Textzeile, das macht keinen Unterschied

Multiple Choice Aufgaben

{.qmc} für Quiz Multiple Choice oder {.qmi .plain} für eine cleanere Version, letztere hab ich euch jetzt überall eingefügt.

Der code sieht dann so aus:
- [X] für richtige Antwort und - [ ] für falsche Antwort

Free Text Aufgaben
Es können jetzt mehrere richtige Antworten hinterlegt werden!

{.qft} für Quiz Free Text

Fragefragefrage frage frage

- [X] richtige Antwort
- [X] alternative richtige Antwort

Insert Choice Aufgaben
Gerade wenn die Füller für Lücken schon vorgegeben werden, empfiehlt sich auch die Insert Choice Variante.

{.qic} für Quiz Insert Choice

Text text textextext

- [X] richtig
- [ ] falsch
- [ ] falsch

Text geht weiter geht weiter text text. Und gleich kommt die nächste

- [ ] falsche Lücke
- [X] richtige Lücke
- [X] alternative richtige Lücke

Damit die Lücke gefüllt wird. Fertig.

Match / drag&drop Aufgaben

{.qmi} für Quiz Match Items

Item, zu dem gematcht werden soll
: Item, das zugeordnet werden soll