

Kurzanleitung zum Workshop »Twine«: Textformatierung & Variablen

Teile der nachfolgend beschriebenen Befehle funktionieren so nur, falls man zuvor das Geschichtenformat auf »SugarCube« umgestellt hat ...

TYPISCHE MARKUP-BEFEHLE:

"Fett" - Achtung: keine Anführungszeichen, sondern zweimal Apostroph

//Kursiv//

__Unterstrichen__

*Liste mit Punkt - Zeile muss linksbündig stehen, sonst erscheint keine Liste

!Level 1 Überschrift (entspricht h1 in HTML) - geht bis Level 6 - wirkt pro Zeile

markdownguide.org/cheat-sheet

TYPISCHE HTML-BEFEHLE:

Text oder <h1>Text</h1>

code-crowd.de/blog/die-10-wichtigsten-html-tags

Benötigt man eine Tabelle, findet man hier Hilfe: cssportal.com/html-table-generator

TYPISCHE CSS-BEFEHLE (inline):

<div style="font-size:30px; color:red">DIV-Befehl mit CSS</div>

tutorium.drdanielappel.de/Uebersicht_CSS.html

CSS-FILESHEET:

Die nachfolgenden Befehle gehört ins Stylesheet der Twine-Datei

ganze Seite in rot, es gehen auch Farbnamen wie *red* statt #ff5555

.rot {background-color: #ff5555;}

Schriftfarbe definieren:

.schriftschwarz {color: black;}

Der CSS-Befehl muss anschließend in der Passage oben mit +Tag unter seinem Namen eingefügt werden. HTML-Farben: cssportal.com/html-colors

MEDIEN EINBINDEN: relativer Pfad & speichern

Bild:

Imagemap erstellen: <https://www.image-map.net>

Audio: <audio src="audio_ordner/audio_name.mp3" autoplay />

Video: <video src="PFAD/video_name.mp4" controls>Ersatztext</video>

Video: <iframe width="480" height="320" src="video_name.mp4" autoplay;></iframe>

WICHTIG BEI DER VERWENDUNG VON MEDIEN: Es muss exportiert werden, damit die eingebundenen Bilder, Töne und Videos angezeigt/abgespielt werden. Ausnahme: Man verwendet absolute Pfade zu einer Internetseite (zum Beispiel zu einem Bild auf pixabay.com)

TEXTFELD ZUM AUFKLAPPEN:

```
<details style="background-color: #eeeeee; border: 1px solid grey;">
  <summary style="padding: 4px; border: 4px;">&#9656; Titelangaben:</summary>
  Hier steht der Inhalt, der wiederum mit HTML & CSS formatiert werden kann ...
</details>
```

VARIABLEN VERWENDEN:

Variablen sind case-sensitiv, es wird also zwischen Groß- und Kleinschreibung unterschieden.

```
<<set $anzahl = 0>> (Achtung: Diese Variable wird bei jedem Seitenaufruf wieder auf 0 gesetzt – vgl. StoryInit)
  $anzahl
  <<set $anzahl = $anzahl + 1>>
  $anzahl

  <<if $anzahl == 1>>
    Bei Wert 1 wird dieser Text gezeigt.
  <<elseif $anzahl == 2>>
    Bei Wert 2 wird dieser Text gezeigt.
  <<else>>
    Bei allen anderen Werten wird dieser Text gezeigt.
  <</if>> -- abschließendes /if nicht vergessen ..
```

STORYINIT, um Variablen zu initialisieren

Will man Variablen über mehrere Passagen behalten, muss man die Variable auf einer Passage namens **StoryInit** mit <<set \$anzahl = 0>> definieren.

lokale Variablen: variablenNamen

Variablen, die mit einem Unterstrich beginnen, sind nur lokal verfügbar, also nur in der Passage, in der sie definiert wurden.