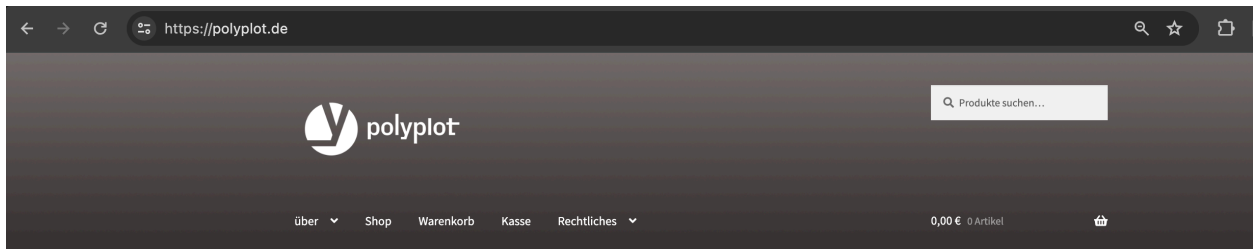


polyplot Bedienungsanleitung

Version 0.2.0¹



Willkommen bei polyplot.

Wir machen digitale Literatur und erforschen, was diese eigentlich können kann. Mit der polyplot Software kann man interaktive Literatur schreiben/programmieren. [Gefährliche Menschen](#) ist der erste polyplot-Roman von Frank Rieger und Christiane Hütter.

[Zur kostenfreien Leseprobe](#)



Willkommen bei polyplot. polyplot ermöglicht es Autor*innen interaktive Literatur in Kollaboration oder solo zu entwickeln und zu veröffentlichen.

Auf polyplot.de ist "Gefährliche Menschen", der erste polyplot-Roman von Frank Rieger und Christiane Hütter zu finden. polyplot wird produziert von Christiane Hütter.

¹ polyplot ist derzeit in Entwicklung und es wird in Zukunft auch neue Versionen des Starter-Pack geben. Diese sind anhand der Versionsnummer identifizierbar.

polyplot unterstützt drei Szenarien:

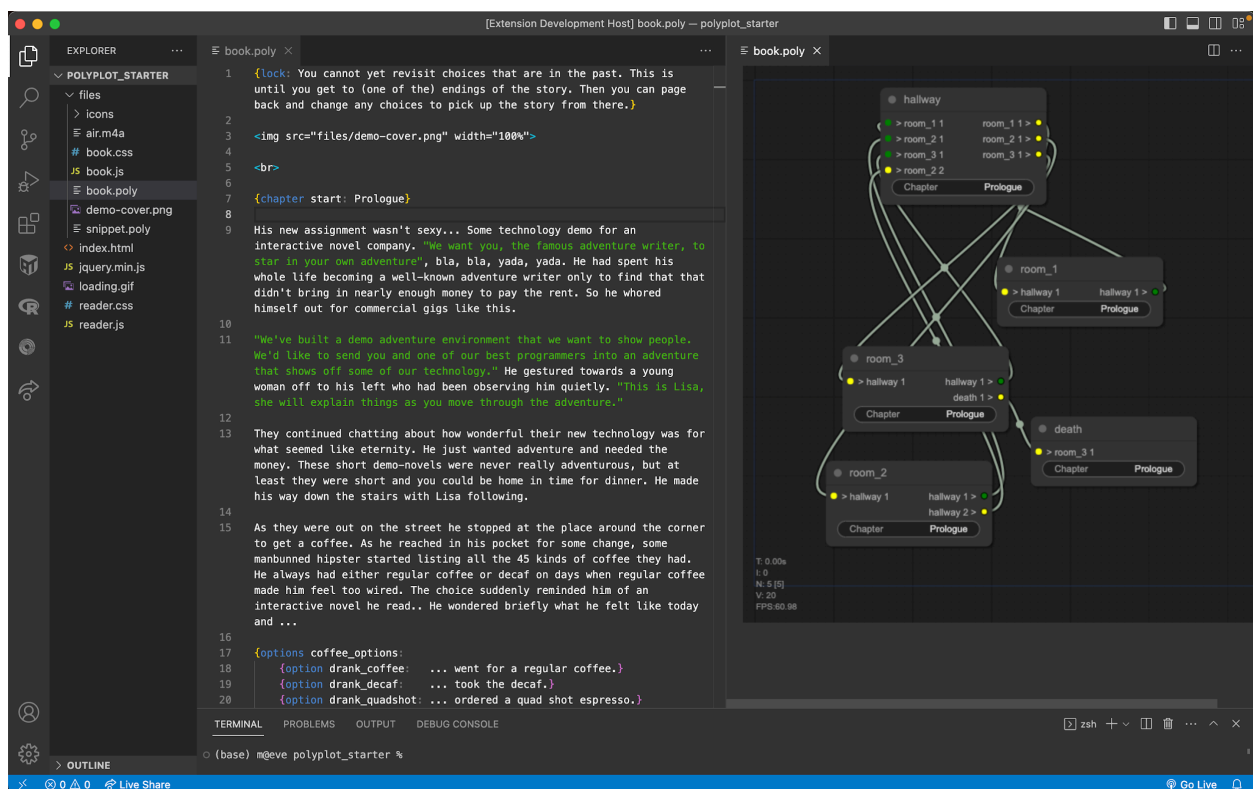
Szenario A: "Alleine schreiben": eine Person arbeitet lokal an einem polyplot. Dies funktioniert auch ohne Internetverbindung.

Szenario B: "Gemeinsam schreiben": mehrere Autor*innen arbeiten gemeinsam an einem polyplot und synchronisieren ihre jeweiligen Änderungen.

Szenario C: "Live Share Session": es ist auch möglich andere Autor*innen einzuladen und zur selben Zeit live gemeinsam an einem polyplot zu schreiben.

Während der Entstehung einer Geschichte können diese drei Szenarien miteinander kombiniert werden.

1. Einrichten der polyplot Umgebung



Die polyplot Umgebung ist eine Erweiterung für den VSCode Editor und läuft auf aktuellen Mac, Windows und Linux Rechnern.

Die Einrichtung erfordert folgende Schritte:

1. Den VSCode Editor installieren: <https://code.visualstudio.com/>
2. Die LiveServer VSCode Erweiterung installieren:
<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>

Zur Vorschau der Geschichte wird der Chrome-Browser empfohlen.

Die folgenden zusätzlichen Schritte sind notwendig für die Szenarien "**Gemeinsam schreiben**" und "**Live Share Session**". Für das Szenario "**Alleine schreiben**" können diese übersprungen werden.

3. git installieren

Windows: <https://git-scm.com/download/win>

Mac: Ein Terminalfenster öffnen und den Befehl "git --version" eingeben. Daraufhin kommt ein Pop-up mit Installationsanweisungen; ggf. muss das Passwort eingegeben werden.

Linux: git ist bereits installiert²

4. Einen Github Account anlegen: <https://github.com/>

5. Die Live Share VSCode Erweiterung installieren:
<https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare>

² Die git Version muss mindestens 2.28 betragen. Ältere Versionen müssen mit dem entsprechenden Linux-Paketmanager aktualisiert werden.

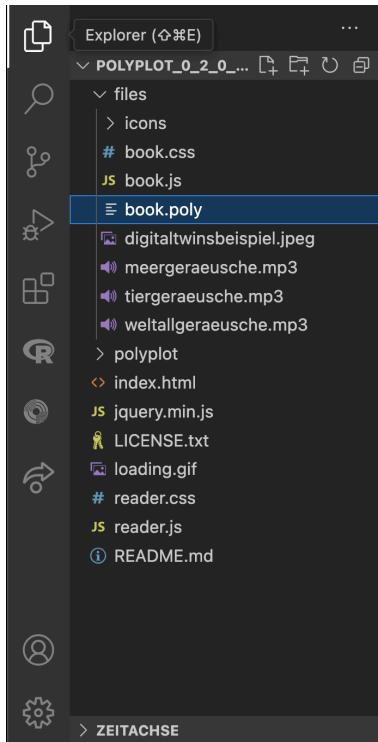
2. Erste Schritte mit dem polyplot Starter-Pack

▼	polyplot_0_2_0_starter_pack	8. Apr 2024 at 08:30
>	files	7. Apr 2024 at 00:56
	index.html	7. Apr 2024 at 00:03
	jquery.min.js	13. Jul 2023 at 11:30
	LICENSE.txt	6. Apr 2024 at 23:24
	loading.gif	13. Jul 2023 at 11:30
▼	polyplot	8. Apr 2024 at 21:29
	Polyplot Bedienungsanleitung.pdf	6. Apr 2024 at 23:36
	polyplot-0.2.0.vsix	8. Apr 2024 at 21:29
	reader.css	13. Jul 2023 at 11:30
	reader.js	1. Aug 2023 at 15:50
	README.md	8. Apr 2024 at 17:03

Das Polyplot Starter-Pack wird als Zip-Datei ausgeliefert. Sie enthält eine Beispielgeschichte (Starter-Story), die polyplot VSCode Erweiterung (.vsix) und diese Anleitung. Nach dem Entpacken der Datei wird zunächst die Erweiterung installiert: In VSCode **Erweiterungen > ... > Aus VSIX installieren** wählen, zur polyplot Erweiterung mit der Dateiendung ".vsix" navigieren und die anschließende Installation bestätigen.

Danach wird das Farbschema für die Schreibumgebung ausgewählt, unter **Code > Einstellungen > Design > Farbdesign** entweder "Polyplot light" mit hellem oder "Polyplot dark" mit dunklem Hintergrund auswählen.

Das Story-Verzeichnis und book.poly



Durch `Datei > Ordner öffnen...` wird die Beispielgeschichte geladen. Wichtig: immer den kompletten Story-Ordner laden, so dass die Datei "index.html" in der Verzeichnisanzeige (`Explorer`) sichtbar ist. Zum Bearbeiten des polyplot wird die Datei "book.poly" im Unterordner "files" geöffnet.



Während der Arbeit am polyplot sollte "book.poly" regelmäßig gespeichert werden. Das Symbol neben dem Dateinamen zeigt an wenn der Arbeitszustand der Datei gespeichert (Abbildung links) oder nicht gespeichert (Abbildung rechts) ist.

Die Live-Vorschau



Prolog

Herzlich willkommen.

Wie schön, dass Sie Vorsorge für den Fall Ihres Ablebens treffen möchten.

Dass Sie an jene denken, die SIE vermissen werden.

Wir bieten Ihnen ein umfassendes Tracking.

Alle Ihre Geräte- Sie brauchen keine neuen!- recorden Ihre Performance. Und ich, ICH HIER, höre Ihnen zu. Ich werde Ihr ganz persönliches SIESELBST.

Ich würde nun gerne mit Ihnen darüber sprechen, wie Sie sich Ihr SIESELBST nach Ihrem Ableben vorstellen. Sind Sie bereit?

Ja.

Nein.

Weiß nicht.

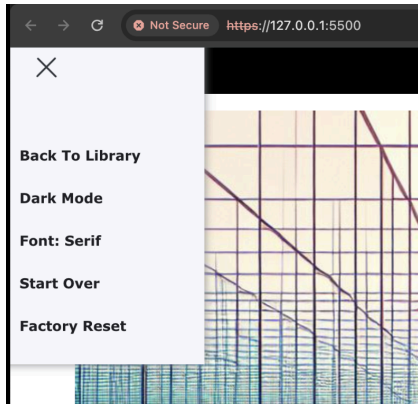
Das ist gut.

Möchten Sie, dass ich Sie mit entspannenden Geräuschen unterstütze?

Ja gerne.

Mit einem Klick auf `Go Live` in der Fußleiste öffnet sich ein Browserfenster mit der Live-Vorschau.

Während der polyplot editiert wird, wird diese automatisch aktualisiert. Jedoch muss das Browserfenster gelegentlich neu geladen werden, um alle Änderungen korrekt anzuzeigen.



Zum Zurücksetzen der Story in der Live-Vorschau auf den Startzustand gibt es den "Factory reset" im Vorschau-Menü oben links.

Die Knotenansicht



Die Knotenansicht wird durch einen Klick auf **Show Nodes** (rechts oben) aktiviert. Sie stellt die Struktur der Story in Form eines Graphen dar. Jeder Knoten entspricht einem **{anchor}** und seine Eingänge links und Ausgänge rechts bilden die Sprungziele (**{jump}**) ab. Die Farben haben folgende Bedeutung:

- grün: es ist ein Sprungziel vorhanden
- gelb: es ist ein bedingtes Sprungziel vorhanden (z.B. aus einer **{option}**)

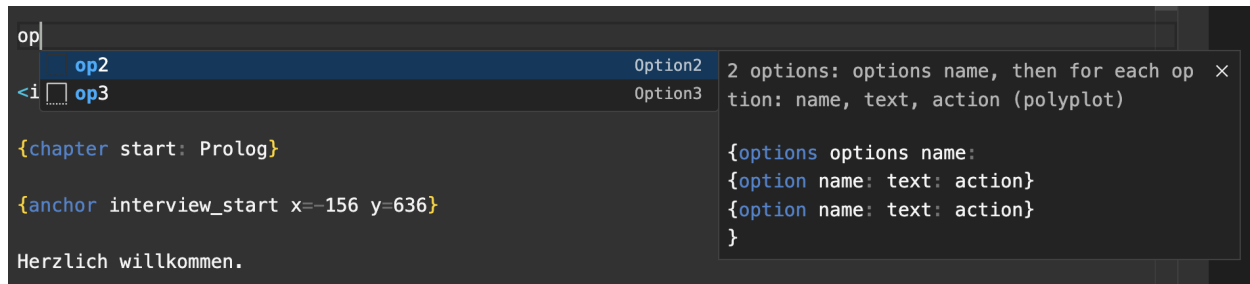
- rot: hier fehlt ein Sprungziel - Hinweis auf einen Fehler

Die Knoten können zur besseren Übersicht beliebig angeordnet werden³.

Ein Doppelklick auf einen Knoten springt zur entsprechenden Stelle des `{anchor}` im Text.

Wie auch die Vorschau aktualisiert sich die Knotenansicht automatisch, jedoch kann es auch hier vorkommen, dass Text- und Knoten außer Takt geraten. In diesem Fall bitte den Text speichern, die Knotenansicht schließen und neu öffnen.

Die Snippets-Funktion



Mit der Snippets-Funktion können polyplot Konstrukte durch Eingabe eines Kürzel wie z.B. "op3" für "`{option}` mit 3 Varianten" eingefügt werden. Dies erspart das Eintippen des kompletten Befehls und hilft, Fehler zu vermeiden. Die Auswahl eines Kürzels ist auch ctrl-Leertaste (Windows)/^Leertaste(Mac) möglich. Während der Eingabe kann man mit der Tabulatortaste von Platzhalter zu Platzhalter. Eine Auflistung der Konstrukte und Kürzel steht im Abschnitt "9. Liste der Polyplot Konstrukte".

3. polyplot Grammatik - die wichtigsten Konstrukte

Ein polyplot besteht aus Text, Befehlen und verknüpften Medien. Text und Befehle sind farblich voneinander abgehoben. Die Struktur eines

³ Hinweis: Beim Verschieben werden die Koordinaten des Knotens in "book.poly" gespeichert. Damit diese Änderungen nicht verloren gehen, muss die Datei gespeichert werden.

polyplot wird aus Kapiteln `{chapter}` und Abschnitten `{anchor}` gebildet. Es muss mindestens ein Kapitel geben.

Mit `{option}` werden Auswahlfelder erzeugt, die die Leser*in anklicken kann, um in der Geschichte zu navigieren. `{jump}` springt zu Kapiteln und Abschnitten.

Variablen enthalten Werte, die sich im Laufe der Geschichte verändern. Dies können Zahlen `{$zaehler=1}`, Texte `{$heute="Montag"}` oder die logischen Werte true/false `{$diamanten_gefunden=false}` sein. Mit Vergleichen und `{if}` `{then}` Konstrukten lassen sich Verzweigungen und bedingte Entscheidungen darstellen. Der aktuelle Wert einer Variablen wird mit `{print}` ausgegeben.

`{tt}` hebt Namen hervor.

Audiodateien werden mit `{audio}` und Bilder mit dem HTML-Befehl `` eingefügt⁴.

`{lock}` verhindert ein "Vor-und Zurückspulen" in der Story und `{unlock}` hebt diese Sperre auf.

Die Starter-Story gibt weitere Beispiele für die Verwendung der Befehle. Eine Auflistung von Befehlen und Kürzeln steht im Abschnitt "9. Liste der Polyplot Konstrukte".

4. Szenario A: Alleine schreiben

Wie in den letzten beiden Abschnitten beschrieben, lässt sich ausgehend vom Starter-Pack ein polyplot lokal auf dem eigenen Rechner schreiben/programmieren und mit der Vorschau im Browser durchspielen. Dies funktioniert auch ohne Internetverbindung. Die "book.poly" Datei enthält die komplette Story. Zusätzlich können Bild- und Audiodateien im Unterverzeichnis "files" abgelegt und mit `{audio}` und `` in die Geschichte eingebunden werden.

⁴ Der `` Befehl enthält spitze Klammern, im Gegensatz zu den meisten anderen Konstrukten, die in geschweifte Klammern eingerahmt sind.

Es empfiehlt sich, von der "book.poly" Datei bzw. dem polyplot Verzeichnis regelmäßig Sicherungskopien zu erstellen.

5. Szenario B: Gemeinsam schreiben

[Screenshot]

Mit polyplot kann interaktive Literatur auch von mehreren Autor*innen kollaborativ entwickelt werden. Dabei dient ein Online-Speicherort, das "Repo", zur Synchronisierung.

Zur Vorbereitung sind hier zusätzlich die Schritte 3, 4 und 5 aus Kapitel 1 "Einrichten der polyplot Umgebung" nötig. Außerdem muss das Projekt in einem Repo angelegt worden sein.⁵

Die beteiligten Autor*innen laden zunächst eine Starter-Story aus dem Repo herunter. Danach können sie unabhängig voneinander an ihrer Story arbeiten, was jeweils 3 Schritte erfordert:

1. Vor Beginn der Arbeit an der Story den aktuellen Stand aus dem Repo herunterladen.
2. Den polyplot editieren wie gewohnt.
3. Den neuen Stand in das Repo hochladen (oder verwerfen).

Die entsprechenden Aktionen erfolgen derzeit über die in VSCode integrierte Versionskontrolle:

[Screenshot]

Das Projekt aus dem Repo initialisieren (einmalig): `[]`
Den aktuellen Stand aus dem Repo laden: `[]`
Den neuen Stand sichern und in das Repo hochladen: `[]`
Den neuen Stand verwerfen: `[]`

⁵ Dies erfolgt derzeit noch manuell nach Absprache durch polyplot; später wird es möglich sein selbst polyplots anzulegen. Ein Repo kann je nach Wunsch öffentlich oder privat sein.

Beim Hochladen werden neue Audio- und Bilddateien, die im Unterverzeichnis "files" angelegt wurden, ebenfalls übertragen. Dabei ist darauf zu achten, dass diese Dateien nicht zu groß sind: einige Megabyte (MB) sind noch ok, eine hundert MB große Audiodatei ist problematisch und würde auch später beim Anzeigen des polyplot zu Problemen führen. In diesem Fall müssen die Mediendateien vor dem Hochladen entsprechend bearbeitet werden.

Die im Repo befindliche Online-Version kann - ähnlich wie in der Vorschau - im Browser durchgespielt werden. Diese Funktion muss im Repo aktiviert werden; anschliessend kann die Story über eine Webadresse erreicht werden. In einem privaten Repo ist der Zugriff auf die an der Produktion Beteiligten beschränkt. In Zukunft werden verschiedene Publishing-Modelle, z.B. die Veröffentlichung im Shop von polyplot.de möglich sein.

Die beschriebene Prozedur funktioniert reibungslos, wenn die Beteiligten ihre Änderungen nicht gleichzeitig vornehmen. Falls gleichzeitige Änderungen stattfinden, müssen diese in einem derzeit noch komplizierten Verfahren miteinander abgeglichen werden. Deswegen empfehlen wir lieber Szenario C anzuwenden, wenn gleichzeitig an einer Story gearbeitet werden soll.

6. Szenario C: Live Share Session

[Screenshot]

Live Share ermöglicht es, andere Autor*innen einzuladen, um gemeinsam live in einer Schreib-Session an einem polyplot zu arbeiten. Zur Vorbereitung sind auch hier die Schritte 3, 4 und 5 aus Kapitel 1 "Einrichten der polyplot Umgebung" nötig.

Die Session wird durch Anklicken von **Live Share** in der Statusleiste gestartet. Anschließend kann ein Link geteilt werden, mit dem andere in die Session einsteigen können. Während der Session werden die Bildschirm-Cursor aller Beteiligten angezeigt und es gibt auch ein Chat-Fenster. Zum Beenden der Session klickt die einladende Person auf **" "**. Die anderen können die Session jederzeit mit **" "** verlassen.

[Screenshot]

In einer Live Share Session wird nur der lokale polyplot der einladenden Person verändert. Falls das betreffende Projekt nach **Szenario B: Gemeinsam schreiben** angelegt ist, muss die einladende Person nach der Session den neuen Stand sichern und in das Repo hochladen, wie im letzten Abschnitt beschrieben.

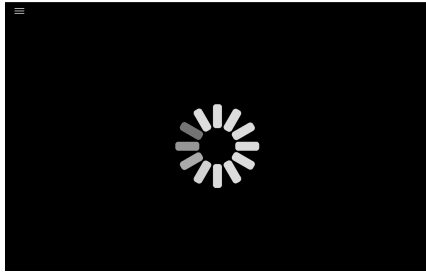
7. Anpassen von VSCode

Der VSCode Editor kann über zahlreiche Einstellungen und Erweiterungen an die eigenen Bedürfnisse angepasst werden. Hier einige Beispiele:

- Helles oder dunkle Farbschema auswählen: "Polyplot light" / "Polyplot dark" in `Code > Einstellungen > Design > Farbdesign`
- Schriftart auswählen: `Code > Einstellungen > Einstellungen`, dann in der Suchzeile "schriftfamilie" eingeben.
- Schriftgröße anpassen: `Code > Einstellungen > Einstellungen`, dann in der Suchzeile "schriftgröße" eingeben.
- Snippets an/oder ausschalten: `Code > Einstellungen > Einstellungen`, dann in der Suchzeile "snippet vorschläge" eingeben.
- Einstellungen für Barrierefreiheit
<https://code.visualstudio.com/docs/editor/accessibility>
- Deutsche Rechtschreibprüfung:
<https://marketplace.visualstudio.com/items?itemName=streetsidesoftware.code-spell-checker-german>

8. Falls es einmal hakt - Tipps zur Fehlerbehebung

Wie in "2. Erste Schritte mit dem polyplot Starter-Pack" erwähnt, gibt es im Vorschau-Menü oben links den Eintrag "Factory reset". Dieser setzt die Geschichte in der Live Vorschau auf den Startzustand zurück.



Falls statt der Live Vorschau ein Ladezeichen angezeigt wird, dann gibt es einen Fehler in der Syntax.

Ein paar generelle Tipps zur Fehlersuche:

- Falls beim Öffnen der "book.poly" Datei der gesamte Text einfarbig ist, muss die polyplot Erweiterung installiert werden. Wie steht im Abschnitt "Erste Schritte mit dem polyplot Starter-Pack".
- Befehle fangen immer direkt nach der geschweiften Klammer an ohne Leerzeichen dazwischen "`{chapter ...}`", nicht: "`{ chapter ...}`"
- Eine rote geschweifte Klammer zeigt an, dass die zugehörige öffnende / schließende Klammer fehlt.
- Konstrukte wie chapter, anchor, optionen und Variablen müssen eindeutige Bezeichner haben. Variablen beginnen mit "\$".
- Bezeichner nur in Kleinbuchstaben, Unterstrich (nicht am Anfang) und Zahlen (jedoch nicht am Anfang) schreiben. Keine Umlaute, Sonderzeichen, Leerzeichen.
- Darauf achten, dass nur gerade Anführungszeichen wie diese hier: `"` verwendet werden.
- Vergleiche in `{if}`-Konstrukten verwenden `"=="`, nicht `"="`.
- Pfade zu Bildern und Musikdateien beginnen immer mit `"files/"`, z.B. `"/files/meinbild.jpg"`. Tipp: Snippets enthalten diese Pfade bereits. Auch in Dateinamen keine Umlaute, Sonderzeichen, Leerzeichen verwenden.
- Ein roter Punkt in Knotenansicht zeigen an dass hier ein Sprungziel (`{jump}`) fehlt.
- Hinweis für Expert*innen: in der JavaScript Konsole im Browser sind nähere Informationen zum aktuellen Fehler zu finden

9. Liste der Polyplot Konstrukte

Snippets/Kürzel stehen **fett** vor dem zugehörigen Konstrukt.

anc {anchor wohnzimmer}

aud {audio /files/audio.mp3}⁶
{audio.volume = 0.3}
{audio stop}

cha {chapter einleitung: Der erste Tag}

com {comment Kommentar, wird nicht angezeigt}

if {if \$stage > 3: "Es sind mehr als drei Tage vergangen" sagte sie}
{if \$stage == 3: "Es genau drei Tage vergangen" sagte sie}

ifel {if \$raum_1 || \$raum_2 || \$raum_3⁷: Wir sind in einem der drei Räume.}
{else: Wir sind außerhalb der Räume.}

inp {inputfield tresorkombination}⁸

jum {jump wohnzimmer}⁹

{lock: Du hast an dieser Stelle schon eine Entscheidung getroffen.
Diese kannst du nicht mehr ändern. Wenn du ein Ende der Geschichte
erreicht hast, kannst du zurückblättern und dich anders entscheiden.}

op2 {options verschiedene_wege:¹⁰
 {option tuer_1: Gehe in den ersten Raum: {jump raum_1}}
 {option tuer_2: Gehe in den zweiten Raum: {jump raum_2}}
}

⁶ Die Audiodatei spielt einmal oder bis {audio stop} ausgeführt wird.

⁷ raum_1, raum_2 und raum_3 müssen vorher als {anchor} gesetzt sein.

⁸ Setzt die Variable tresorkombination auf den Wert der Eingabe

⁹ "wohnzimmer" muss als {chapter} oder {anchor} vorhanden sein.

¹⁰ Springt je nach Auswahl zum {anchor} "raum_1", "raum_2" oder "raum_3".

```

op3 {options kaffee_optionen:11 12
    {option normaler_kaffee: ... trank einen Kaffee.}
    {option ohne_koffein: ... nahm den entkoffeinierten Kaffee.}
    {option espresso: ... bestellte einen doppelten Espresso.}
}

pri {print $hundename} wacht auf und schüttelt sich den Schlaf aus dem
Fell.}

{tt}
{tt NAME}

{unlock} -> siehe {lock}

var {$hundename = "Fairy"} {$stage = 0}
{$stage++}

```

HTML Befehle

```

<blockquote>Zitat</blockquote>
img 

```

¹¹ setzt je nach Auswahl die Variable "normaler_kaffee", "ohne_koffein" oder "espresso" auf den Wert "true".

¹² Texte in {options} dürfen keine Anführungszeichen enthalten. Verzweigungen in Dialogen müssen daher mit {if}/{else} oder {anchor}/{jump} realisiert werden.