



# Twine-Tutorial

Frank Sindermann

Version 0.9 vom 22. Juni 2018

## Inhaltsverzeichnis

<b>1</b>	<b>Vorbemerkung</b>	<b>2</b>
<b>2</b>	<b>Erste Schritte</b>	<b>3</b>
2.1	Installation . . . . .	3
2.2	Der Hauptbildschirm . . . . .	3
<b>3</b>	<b>Grundlagen</b>	<b>5</b>
3.1	Abschnitte und Verzweigungen . . . . .	5
<b>4</b>	<b>Erweiterte Möglichkeiten</b>	<b>9</b>
4.1	Variablen . . . . .	9
4.2	Bedingungen . . . . .	11
4.3	Zufall . . . . .	12
4.4	Layoutänderungen . . . . .	13
<b>5</b>	<b>Schlussbemerkung</b>	<b>14</b>

Dieses Dokument ist lizenziert unter einer Creative Commons Namensnennung – Nicht kommerziell – Keine Bearbeitungen 4.0 International Lizenz.



# 1 Vorbemerkung

Dieses Dokument erklärt, wie man mit dem Programm Twine interaktive Geschichten im Stil klassischer Spielbücher erstellt. Dabei handelt es sich um Texte, die nicht linear von Anfang bis Ende durchgelesen werden, sondern abschnittsweise. Am Ende der meisten Abschnitte können die Leser\_innen durch Anklicken eines Links selbst entscheiden, wie die Geschichte weitergehen soll.

In dieser ersten Version der Anleitung wird kaum auf Fragen des Spieldesigns eingegangen. Es wird nicht vermittelt, wie man eine Twine-Geschichte von Grund auf konzipiert und schreibt, welche logischen oder stilistischen Probleme man vermeiden sollte etc., sondern allein die technischen Kenntnisse, die man dafür benötigt, um mit Twine zu arbeiten.

Einzutippender Text wird in `Schreibmaschinenschrift` wiedergegeben.  
Anzuklickender Text ist in `Kästchen` gesetzt.

Abgesetzte Codebeispiele stehen in einem Rahmen und sind in Schreibmaschinenschrift gesetzt.

## **Tipp**

Hinweise und Tipps stehen in gelben Kästen.

## **Achtung!**

Warnungen stehen in roten Kästen.

Anregungen und Kritik an dieser Anleitung sind jederzeit willkommen!

Und nun: Viel Spaß beim Entdecken von Twine!

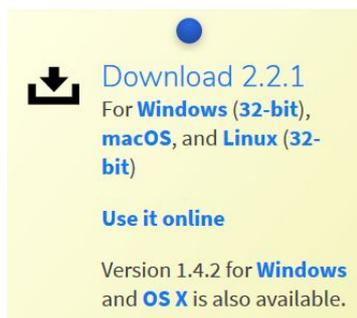
Leipzig, im Juni 2018

Frank Sindermann

## 2 Erste Schritte

### 2.1 Installation

Twine kann in der aktuellen Version 2.2.1 (Stand: 22. Juni 2018) über die Webseite [www.twinery.org](http://www.twinery.org) direkt online verwendet oder als Programm auf dem Computer installiert werden. Die entsprechenden Links finden sich rechts oben auf der Seite:



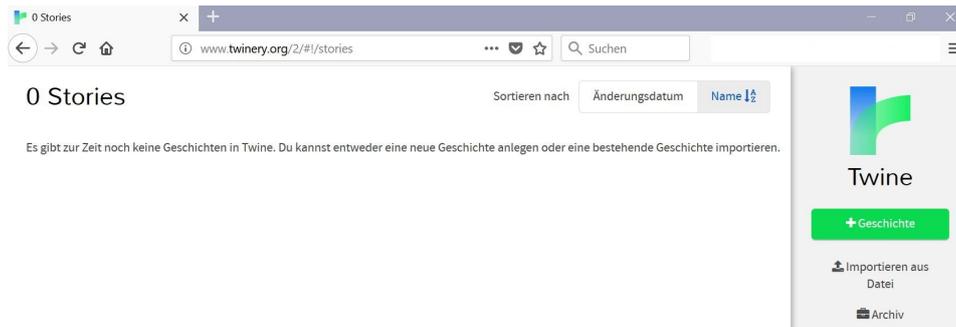
Auf die Installation wird hier nicht näher eingegangen, zumal die Online-Version sehr gut funktioniert und die Arbeitsdaten und fertigen Geschichten problemlos auf der Festplatte gespeichert werden können.

#### **Achtung!**

Die ebenfalls zum Download angebotene Version 1.4.2 unterscheidet sich in einigen Punkten deutlich von der neuen Version. Diese Anleitung bezieht sich ausschließlich auf Version 2.2.1.

### 2.2 Der Hauptbildschirm

Nach Schließen des Begrüßungsbildschirms mit Klick auf Überspringen gelangt man zum Hauptbildschirm von Twine, von dem aus man eine Geschichte zur Bearbeitung öffnen, eine neue beginnen und seine Arbeitsdaten speichern oder wiederherstellen kann.



**Geschichte bearbeiten** Durch Doppelklick auf eine der bisherigen Geschichten wird diese zur Bearbeitung geöffnet. Beim ersten Arbeiten mit Twine ist noch keine Geschichte vorhanden.

**neue Geschichte beginnen** Mit Klick auf  legt man eine neue Geschichte an. Im nachfolgenden Bildschirm gibt man einen (vorläufigen) Namen für die Geschichte ein und gelangt schließlich in den Twine-Arbeitsbereich.

**Arbeitsdaten speichern** Mit Klick auf  gelangt man in einen Speicherdialog und kann den aktuellen Bearbeitungsstand **aller** Geschichten sichern.

**Arbeitsdaten wiederherstellen** Mit Klick auf  kann man eine Archivdatei laden und damit alle oder ausgewählte Geschichten frühere Bearbeitungsstände wiederherstellen. Dies kann z. B. dann sinnvoll sein, wenn man Änderungen vorgenommen hat, die zu Fehlern führen oder einem nicht mehr gefallen.

### Achtung!

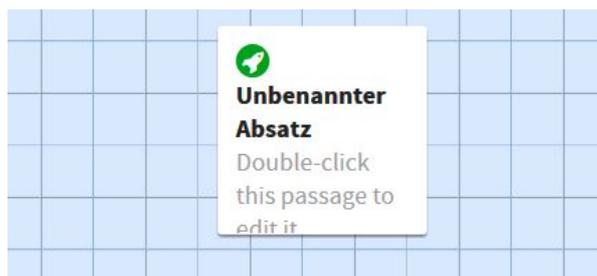
Die Arbeitsdaten sollten regelmäßig, am besten nach jeder Arbeitsphase, gesichert werden. Da Twine die Arbeitsdaten der Geschichten lokal im Browser speichert, können diese sehr leicht versehentlich gelöscht werden, z. B. durch das Leeren des Caches oder das Löschen des Browserverlaufs.

## 3 Grundlagen

### 3.1 Abschnitte und Verzweigungen

#### Eine neue Geschichte anlegen

Ein neues Spiel besteht zunächst aus einer leeren Textbox:



Durch Doppelklick auf die Box öffnet sich der Abschnitt zur Bearbeitung. Ganz oben wird der Abschnittsname eingetragen, in diesem Fall z. B. **Anfang**, im großen Feld dann der eigentliche Text. Das Feld zum Setzen von Tags wird in dieser Anleitung nicht verwendet.

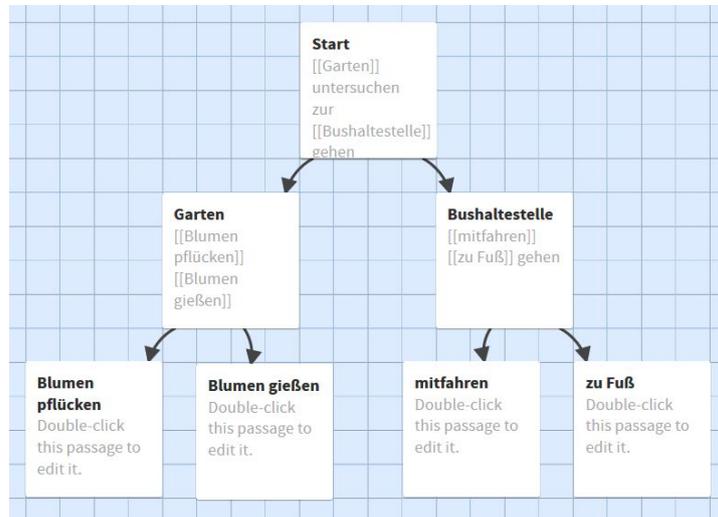


#### Verzweigungen einfügen

Damit die Geschichte interaktiv wird, fügt man Links hinzu, die zu weiteren Textabschnitten führen. Damit bietet man den Spieler\_innen Optionen an, wie die Handlung weitergehen soll. Einen Link setzt man, indem man ein Wort oder einen Satz in doppelte eckige Klammern setzt:

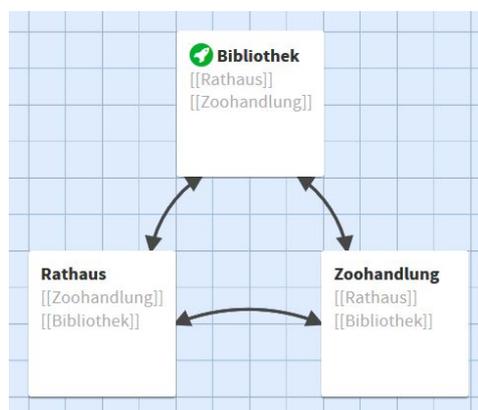
Möchtest du nach `[[links]]` oder nach `[[rechts]]` gehen oder dich noch `[[weiter umsehen]]`?

Sobald einem Abschnitt Links hinzugefügt werden, legt Twine beim Schließen der Textbox automatisch die Zielabschnitte an. Verbindungspfeile zeigen die Beziehungen der Textboxen an:



Möchte man auf einen schon bestehenden Abschnitt verweisen, gibt man einfach dessen Namen innerhalb der eckigen Klammern an. Dadurch ist es möglich, wieder zum vorherigen Abschnitt zurückzuspringen oder von mehreren Abschnitten zum selben Zielabschnitt zu verweisen.

Je nach Art der Geschichte können die Abschnitte unterschiedliche inhaltliche Funktionen erfüllen. In sehr handlungsorientierten Geschichten stehen Abschnitte oft für Ereignisse innerhalb der Handlung, während sie in Geschichten, bei denen es um Erkundung und Orientierung geht, häufig Orte darstellen. Werden Abschnitte in mehrere Richtungen miteinander versetzt, kann man sich wie auf einer Landkarte bewegen und immer wieder zwischen den Orten des Geschehens wechseln:



## Tipp

In vielen Fällen kann es sinnvoll sein, zwischen dem anklickbaren Link und dem Namen des verlinkten Abschnittes zu unterscheiden. Vielleicht sollen die Spieler\_innen an mehreren Stellen der Geschichte auf "weiter" klicken, aber jedesmal zu einem anderen Abschnitt gelangen. Dies erreicht man, indem man Linktext und Linkziel gemeinsam in die doppelten eckigen Klammern schreibt und durch einen Pfeil anzeigt, welcher der Texte wohin führen soll:

```
[[weiter->Wohnzimmer]]
```

## Texte mehrfach einfügen

Texte, die an mehreren Stellen der Geschichte erscheinen sollen, muss man nicht komplett in jeden einzelnen Abschnitt schreiben oder kopieren. Mit der Funktion (`display:`) kann man einen Abschnitt innerhalb eines anderen anzeigen lassen. Dazu legt man zunächst mittels Klick auf `+ Passage` (grünes Kästchen im unteren Menü) einen neuen Abschnitt an (grüner Kasten) und ruft diesen dann in den gewünschten Abschnitten auf:



Das Beispiel oben ergibt im Spiel folgende Ausgabe:

```
Du trittst ans Fenster und schaust in den Garten.  
Es regnet in Strömen.
```

Das Verwenden dieser Möglichkeit ist sehr sinnvoll, da man bei eventuellen Änderungen oder Korrekturen nur eine Stelle überarbeiten muss. Vielleicht möchte man die Geschichte etwas freundlicher gestalten und entscheidet sich für Sonnenschein statt Regen? Dann muss man nur im Wetterabschnitt die Beschreibung ändern und sie wird in allen darauf verweisenden Abschnitten aktualisiert.

## Ein Spiel testen

Mit einem Klick auf Spielen in der unteren Menüleiste kann man das Spiel testen.

### Tipp

Wenn die Geschichte länger wird, kann es hilfreich sein, beim Testen nicht immer von vorn beginnen zu müssen. Dazu kann man neu festlegen, wo das Spiel beginnen soll, indem man mit der Maus über den gewünschten Startabschnitt fährt, im erscheinenden Kontextmenü die drei Punkte anklickt und dann den Eintrag Start Story Here wählt. Welcher Abschnitt aktuell Startabschnitt ist, erkennt man an dem kleinen Raketen-symbol.

## Eine spielbare Datei erstellen

Mit den bisher gezeigten Möglichkeiten lässt sich bereits eine einfache interaktive Geschichte schreiben. Damit andere diese lesen bzw. spielen können, muss man sie aber zunächst als HTML-Datei abspeichern. Das Dreieck (neben dem Haus, im unteren Menü) öffnet ein Untermenü mit dem Eintrag Als Datei veröffentlichen (ganz unten). Wenn man darauf klickt, gelangt man in einen Speicherdialog und kann dort seine Geschichte als HTML-Datei abspeichern. Diese Datei lässt sich in jedem gängigen Browser – auch auf dem Smartphone – spielen, indem man entweder die Datei direkt weitergibt (Offline-Spiel) oder auf einen Server hochlädt (Online-Spiel). Zum Spielen ist es also nicht notwendig, Twine zu verwenden oder auch nur zu kennen!

### Tipp

Vor dem Veröffentlichen kann es nicht schaden, sich per Funktion Korrekturfassung anzeigen (im selben Menü) eine fortlaufende Textausgabe der gesamten Geschichte ausgeben zu lassen, die man auf Fehler prüfen lassen (kann).

## 4 Erweiterte Möglichkeiten

Die in diesem Kapitel gezeigten erweiterten Funktionen sind teilweise etwas komplexer in der Anwendung, steigern aber die Möglichkeiten enorm. Durch den Einsatz von Variablen, Bedingungen und Zufallselementen wird eine Twine-Geschichte noch interaktiver und lebendiger.

### 4.1 Variablen

Variablen bieten die Möglichkeit, festzuhalten, welche Entscheidungen die Spieler\_innen treffen, welche Fragen sie richtig oder falsch beantworten oder welche Gegenstände sie mitnehmen. Variablen werden mit der Funktion (`set:`) eingerichtet oder verändert. Der Name von Variablen beginnt mit einem `$`-Zeichen, gefolgt von einer frei wählbaren Kombination aus Buchstaben und Zahlen, z. B. `$geld`, `$hinweis12` oder `$name`. Der Name einer Variablen sollte so gewählt werden, dass man an ihm erkennt, welche Funktion sie in der Geschichte hat bzw. an welcher Stelle sie eingesetzt werden soll.

#### Typen von Variablen

Je nachdem, welche Informationen man in der Variablen speichern möchte, kann man sich für einen von drei Typen entscheiden:

- **Zahl:** Dieser Typ ist sinnvoll, wenn man einen Geldbetrag, eine Punktzahl oder einen anderen Zahlenwert zuweisen möchte, mit dem gerechnet werden kann.  
Beispiel: (`set: $punkte to 5`).
- **Text:** Diesem Typ kann irgendeine Zeichenkette zugewiesen werden, z. B. ein Lösungswort, ein Name oder ein mitgenommener Gegenstand.  
Beispiel: (`set: $password to "Rosebud"`).
- **wahr/falsch:** Variablen dieses Typs werden auf `true` oder `false` gesetzt. Dies ist dann sinnvoll, wenn die Variable nur wie ein Schalter angeben soll, ob etwas der Fall ist oder nicht (ja/nein, an/aus).  
Beispiel: (`set $lichtschalter to true`).

## Variablen anzeigen

Um den aktuellen Wert einer Variablen im Spiel anzuzeigen, kann man ihn mit der Funktion `(print:)` ausgeben lassen:

```
(set: $name to "Harry")
Hallo, (print: $name), wie geht es dir?
```

## Variablen verändern

Mit der Funktion `(set:)` kann man einer Variablen jederzeit einen neuen Wert zuweisen:

- `(set: $geld to 10)`
- Du hast `(print: $geld)` Euro.
- Du kaufst dir für 2 Euro ein Eis.
- `(set: $geld to $geld - 2)`
- Du hast jetzt noch `(print: $geld)` Euro.

**Du hast 10 Euro.**

**Du kaufst dir für 2 Euro ein Eis.**

**Du hast jetzt noch 8 Euro.**

Neben der in der Abbildung gezeigten Subtraktion sind auch Addition (+), Multiplikation (\*) und Division (/) möglich. Weitere mathematische Anwendungen sind möglich, werden im Rahmen dieser Anleitung aber nicht behandelt.

## Variablen eingeben lassen

Die Funktion `(prompt:)` ermöglicht es den Spieler\_innen, einer Variablen selbst einen Wert zuzuweisen, indem sie diesen in eine Eingabebox tippen:

```
(set: $name to (prompt: "Gib bitte deinen Namen ein:", ))
```

## 4.2 Bedingungen

Wirklich sinnvoll sind Variablen erst dann, wenn sie im Spielverlauf auch ausgewertet werden, wenn also z.B geprüft wird, ob man den Schlüssel zur Tür hat, genug Geld besitzt oder das Zauberwort weiß. Dazu dienen die Funktionen (if:) und (else:).

Nehmen wir an, an einer Stelle des Spiels könne man wählen, ob man eine Fackel mitnehmen möchte oder nicht:

```
[[Fackel mitnehmen]]  
[[Fackel liegenlassen]]
```

Die erste Option führt zu einem Abschnitt wie diesem:

```
Du nimmst die Fackel mit und setzt deinen Weg fort.  
(set: $fackel to true)
```

Die zweite Option führt zu einem Hinweis, dass man die Fackel nicht mitnimmt. In diesem Fall wird die Variable \$fackel) nicht gesetzt. Wenn man nun später im Verlauf der Geschichte eine dunkle Höhle betritt, kann Twine über den Status der Variablen prüfen, ob man die Fackel hat oder nicht:

```
(if: $fackel is true)[Im Licht deiner Fackel erkennst du  
eine Inschrift an der Wand.]  
(else:)[Es ist zu dunkel, um etwas zu sehen.]
```

Wer die Fackel mitgenommen hat (if: \$fackel is true), kann also hier vielleicht eine wichtige Information an der Wand lesen, die später hilfreich sein wird. Andernfalls (else:) bleibt der Hinweis verborgen.

### Tipp

Man kann sogar Auswahllinks in die Optionen schreiben. Dies bedeutet, dass bestimmte Handlungsalternativen nicht wählbar sind, wenn eine bestimmte Bedingung nicht erfüllt ist:

```
(if: $fackel is true)[Im Licht deiner Fackel  
erkennst du eine [[Geheimtür]] in der Wand.]
```

Die Geheimtür kann nur durchquert werden, wenn sie im Licht der Fackel sichtbar wird.

## 4.3 Zufall

Die Verwendung von zufälligen Texten kann eine Geschichte noch lebendiger wirken lassen. Außerdem kann mit Zufallszahlen ein Würfelwurf oder ein anderes Zufallsereignis simuliert werden.

### Zufälliger Text

Die Funktion `either`: ermöglicht es, Teile des Textes zufällig von Twine bestimmen zu lassen:

```
Im Baum zwitschert ein (either: "Buchfink",  
"Wintergoldhähnchen", "Fitis").
```

## Zufallszahlen

Zufallszahlen werden mit der Funktion `(random:)` ermittelt, wobei die beiden Zahlen das minimale und maximale Ergebnis angeben:

```
Du würfelst eine (random: 1,6).
```

Die ermittelten Zufallszahlen kann man entweder mit der Funktion `(put:)` einer Variablen zuweisen oder direkt auswerten lassen:

```
(put: (random: 1,6) into $wurf)
Du hast eine (print: $wurf) gewürfelt.
(if: $wurf < 4)[Leider verloren!]
(else:)[Gewonnen!]

(if: (random: 1,10) < 2)[Der Schlüssel bricht ab. Du musst
[[zurück]].]
(else:)[Du öffnest das Schloss und gehst [[weiter]].]
```

## 4.4 Layoutänderungen

Da Twine HTML-Dateien erzeugt, steht erfahrenen Nutzer\_innen die ganze Palette an HTML- und CSS-Anweisungen zur Verfügung, um die Geschichte an eigene Layoutvorstellungen anzupassen. An dieser Stelle sollen nur wenige Effekte beschrieben werden, die zur Hervorhebung von Textstellen nützlich sind.

### Schriftschnitt

Um Text **fett** oder *kursiv* zu setzen, kann man folgende Schreibweisen verwenden:

```
**fetter Text**
*kursiver Text*
```

## Textfarbe

Um Text einzufärben, gibt es die Funktion `text-color`:

```
(text-color:"red")[roter Text]
```

### Tipp

Viele weitere Farbnamen finden sich in der Farbtabelle unter [https://en.wikipedia.org/wiki/X11\\_color\\_names](https://en.wikipedia.org/wiki/X11_color_names).

## 5 Schlussbemerkung

Twine bietet weit mehr an Möglichkeiten, als in dieser Anleitung (Version 0.9) behandelt werden. Es ist geplant, dieses Dokument nach und nach zu erweitern – bis es soweit ist, empfehle ich einen Blick in das Twine-Wiki (<http://twinery.org/wiki/harlowe:reference>) und das Twine-Cookbook (<http://www.twinery.org/cookbook/>), in dem häufig gesuchte Funktionen zu finden sind. So wird z. B. erklärt, wie man einen Timer implementiert, Links versteckt oder Bild und Ton einfügen kann.