**Help file –  
Twine 2/Twee 3 (+SugarCube2)**

Philip Sutton 5 April 2022 / 10:35 AM / Version 21

This file also has information about using Tweego, VSCode and the Twee3 VSCode extension

**Contents**

[History of the evolution of the twee/twine ecosystem 2](#_Toc100047962)

[Browser support 2](#_Toc100047963)

[Location of Twine 2 windows installer file 2](#_Toc100047964)

[Code repository for Twine 2 3](#_Toc100047965)

[A catalogue of Twine story and proofing formats 3](#_Toc100047966)

[Story format repositories for Twine 2 3](#_Toc100047967)

[Guides to Story formats for Twine 2 3](#_Toc100047968)

[Installing the latest version of SugarCube for Twine 2 & repository 3](#_Toc100047969)

[Proofing formats for Twine 2 4](#_Toc100047970)

[Installing Tweego 4](#_Toc100047971)

[Installing Twee3 Language Tools 4](#_Toc100047972)

[Using Tweego 4](#_Toc100047973)

[### - Work flow: Twine to Tweego/VSCode/Twee3 extension 5](#_Toc100047974)

[Regex syntax for use in vscode of twine2 7](#_Toc100047975)

[Twine users/related tools that have use-cases more like what RSTI is trying to do 8](#_Toc100047976)

[**Branching scenarios in training/elearning, story-based teaching** 8](#_Toc100047977)

[**Building databases in twine** 9](#_Toc100047978)

[**Managing complex/large twine projects** 9](#_Toc100047979)

[Information about Twine (all versions) 9](#_Toc100047980)

[CSS & html guides/templates for Twine 10](#_Toc100047981)

[Additional Twine / SugarCube code resources 10](#_Toc100047982)

[General guides for Twine 2 10](#_Toc100047983)

[Guides for Web languages 10](#_Toc100047984)

[Special passages that are automatically displayed in the final HTML 11](#_Toc100047985)

[Special passages that are NOT displayed in the final HTML 11](#_Toc100047986)

[Special Twine 1 passages NOT used in Twine 2 12](#_Toc100047987)

[Special Twine 2 passages NOT used in SugarCube 12](#_Toc100047988)

[Special tags with predefined roles 12](#_Toc100047989)

[Special tags that area NOT used any more 13](#_Toc100047990)

[Preventing the passage of ‘private’ information into the final .html file – a work around 13](#_Toc100047991)

[Formatting text / Markup 13](#_Toc100047992)

[**Links** 13](#_Toc100047993)

[**Heading format** 13](#_Toc100047994)

[**Basic formatting** https://www.motoslave.net/sugarcube/2/docs/#**markup-style** 14](#_Toc100047995)

[**Characters/ Symbols** 14](#_Toc100047996)

[**Text colour** 14](#_Toc100047997)

[**HTML tags** 14](#_Toc100047998)

[**List format** 16](#_Toc100047999)

[**Blockquote** 16](#_Toc100048000)

[**Special formatting codes** 16](#_Toc100048001)

[**Code within-a passage** 16](#_Toc100048002)

[**Non-active emulation text** 16](#_Toc100048003)

[**Verbatim text in TwineScript and html** 17](#_Toc100048004)

[**Commenting out – does nor appear in the visible html display – but is passed into the HTML file as a comment** 18](#_Toc100048005)

[Supported variable types 18](#_Toc100048006)

[Inbuilt Twine 2 + SugarCane MACROS 19](#_Toc100048007)

[CUSTOM MACROS made for SugarCane 26](#_Toc100048008)

[Functions 26](#_Toc100048009)

[Methods 28](#_Toc100048010)

[Expressions 28](#_Toc100048011)

[TwineScript conditional (evaluation) operators 28](#_Toc100048012)

[JavaScript conditional (evaluation) operators 29](#_Toc100048013)

[TwineScript assignment operators 29](#_Toc100048014)

[Transferring data between Twine projects 30](#_Toc100048015)

[SugarCube documentation – (on the SugarCube site) 30](#_Toc100048016)

[Twine 2 advice from the Twine Cookbook 32](#_Toc100048017)

[Merging Twine 2 files 35](#_Toc100048018)

[Developer and resource curator contact details 35](#_Toc100048019)

[Migrating Twine 1 stories/documents to Twine 2 36](#_Toc100048020)

[Useful things in Twine 1 that are NOT available in Twine 2/SugaCube 36](#_Toc100048021)

[**Sending people links to specific passages in a Twine .html file The hashbang method** 37](#_Toc100048022)

[**Special passage: StoryIncludes** 37](#_Toc100048023)

[**Special tags: Twine.private** 37](#_Toc100048024)

[Related tools 37](#_Toc100048025)

[(the rest in the old Twine 1 help file??) 37](#_Toc100048026)

[Joey’s backup method 37](#_Toc100048027)

# History of the evolution of the twee/twine ecosystem

A short history of the evolution of the twee/twine ecosystem:

<https://videlais.com/2019/06/08/an-oral-history-of-twee/>

# Browser support

[https://twinery.org/cookbook/twine1/editor/](https://twinery.org/cookbook/twine1/editor/browser_support.html)**[browser\_support](https://twinery.org/cookbook/twine1/editor/browser_support.html)**[.html](https://twinery.org/cookbook/twine1/editor/browser_support.html)

# Location of Twine 2 windows installer file

Twine 2 - v2.3.15 download link:

<https://twinery.org/>

There might be a new version at:

[https://github.com/klembot/twinejs/releases](https://github.com/klembot/twinejs)

As of Jan 2022, Twine 2.4 is in development (with a beta available from the Twine 2 repository.

# Code repository for Twine 2

<https://github.com/klembot/twinejs>

# A catalogue of Twine story and proofing formats

[http://mcdemarco.net/tools/hyperfic/twine/**catalog**/](http://mcdemarco.net/tools/hyperfic/twine/catalog/)

# Story format repositories for Twine 2

[https://github.com/tmedwards/](https://github.com/tmedwards/sugarcube-2)**[sugarcube](https://github.com/tmedwards/sugarcube-2)**[-2](https://github.com/tmedwards/sugarcube-2)

[https://github.com/klembot/**chapbook**](https://github.com/klembot/chapbook)

[https://foss.heptapod.net/games/**harlowe**](https://foss.heptapod.net/games/harlowe)

[https://github.com/videlais/**snowman**](https://github.com/videlais/snowman)

# Guides to Story formats for Twine 2

**Sugarcube 2** guide  
[http://www.motoslave.net/sugarcube/2/](http://www.motoslave.net/sugarcube/2/docs/)**[docs](http://www.motoslave.net/sugarcube/2/docs/)**[/](http://www.motoslave.net/sugarcube/2/docs/)

<http://www.motoslave.net/sugarcube/2/>

**Chapbook** guide:  
[https://klembot.github.io/**chapbook**/guide/](https://klembot.github.io/chapbook/guide/)

**Harlowe** guide:  
<https://twine2.neocities.org>

**Snowman 1** guide:  
[https://videlais.github.io/**snowman**/1/](https://videlais.github.io/snowman/1/)

**Snowman 2** guide:  
[https://videlais.github.io/**snowman**/2/](https://videlais.github.io/snowman/2/)

# Installing the latest version of SugarCube for Twine 2 & repository

***Instructions:***

From: [http://twinery.org/forum/discussion/7903/**how-can-i-install-sugarcube-2-x-as-a-format**](http://twinery.org/forum/discussion/7903/how-can-i-install-sugarcube-2-x-as-a-format)

+ Download the current local/offline version of SugarCube 2.x for Twine 2.

[https://github.com/tmedwards/sugarcube-2/**releases**/](https://github.com/tmedwards/sugarcube-2/releases/)

+ Extract the archive to a safe location on your computer and take note of the path to it. Make sure to keep the files together if you move them out of the included directory.

+ Launch Twine 2.

+ Click on the Formats link in the Twine 2 sidebar.

+ In the dialog that opens, click on the Add a New Format tab.

+ Finally, paste a file URL to the **format.js** file, based on the path from step #2, into the textbox and click the +Add button (see below for examples).

<https://github.com/tmedwards/sugarcube-2>

# Proofing formats for Twine 2

**Illume**

<http://www.maximumverbosity.net/twine/Illume/>

<http://www.maximumverbosity.net/twine/Illume/format.js>

[https://www.christytuckerlearning.com**/illume-proofing-format-for-twine**/](https://www.christytuckerlearning.com/illume-proofing-format-for-twine/)

**Poof**

[https://github.com/ChapelR/**poof**](https://github.com/ChapelR/poof)

<https://cdn.jsdelivr.net/gh/chapelr/poof@latest/docs/use/format.js>

**DotGraph**

[http://mcdemarco.net/tools/scree/**dotgraph**/](http://mcdemarco.net/tools/scree/dotgraph/)

# Installing Tweego

Download from:

<https://www.motoslave.net/tweego/>

Installation instructions and installer from:

[https://github.com/ChapelR/tweego-setup/blob/tweego-2/docs/**installing-tweego**.md](https://github.com/ChapelR/tweego-setup/blob/tweego-2/docs/installing-tweego.md)

<https://github.com/ChapelR/tweego-installer>

Tweego is a command-line application, meaning it is run from your computer's terminal or command prompt. Simply clicking on the executable will not really do anything. Further reading: <https://en.wikipedia.org/wiki/Command-line_interface>

The best way to use Tweego is to add it to your PATH environment variable. How you do that depends on your operating system. If you are not confident doing that, there are scripts and installation packages available online you can use, like this one (made by me) for Windows: <https://github.com/ChapelR/tweego-installer>

Actually using Tweego does not necessarily require it be added to your PATH, but it is suggested you do so. After installing it as above, open a terminal or command prompt and type tweego -v.

If the command is recognized and the version is reported, congratulations, you installed Tweego!

# Installing Twee3 Language Tools

[https://marketplace.visualstudio.com/items?itemName=**cyrusfirheir.twee3-language-tools**](https://marketplace.visualstudio.com/items?itemName=cyrusfirheir.twee3-language-tools)

<https://code.visualstudio.com/docs/editor/extension-marketplace>

[https://intfiction.org/t/recommended-**ide-for-sugarcube**/49905/13](https://intfiction.org/t/recommended-ide-for-sugarcube/49905/13)

# Using Tweego

**Documentation**

<https://www.motoslave.net/tweego/docs/>

[https://github.com/ChapelR/**tweego-setup**](https://github.com/ChapelR/tweego-setup)

# ### - Work flow: Twine to Tweego/VSCode/Twee3 extension

[https://github.com/cyrusfirheir/**twee3-language-tools**](https://github.com/cyrusfirheir/twee3-language-tools) (VSCode extension – with map)

<https://joshuagrams.github.io/tiny-qbn/doc/tweego.html> - linking tweego with VSCode

[https://github.com/cyrusfirheir/twee3-language-tools/blob/master/docs/**getting-started.md**](https://github.com/cyrusfirheir/twee3-language-tools/blob/master/docs/getting-started.md)

<https://dev.to/lazerwalker/a-modern-developer-s-workflow-for-twine-4imp>

**The forward conversion flow**: The original twine project file is converted to twee format by tweego. Then Joeyrsp ‘s python code is run to split the twee file into single passage files. Later tweego is used to compile a new output .html file.

Tweego needs to use the story format type and version to match that used in the Twine file to be converted. You can find out what version you have loaded by running the command below though the windows command prompt or vscode’s terminal:

tweego.exe --list-formats

To get the latest SugarCube format files go to the link below and choose the .zip file to download:

<https://github.com/tmedwards/sugarcube-2/releases>

(If the latest sugarcube release is a later version than the one used for the twine file, then update the twine file first, then update tweego.

Then go to the tweego installation folder and copy the up-to-date sugarcube files over the top of the out of date sugarcube files:

C:\Program Files\Tweego\storyformats\sugarcube-2

***Using Tweego to convert a Twine2 file to a Twee3 file***

Steps: Open the Windows Command Prompt (or the vscode terminal). In the Command Prompt (or the terminal) change the directory with this command:

cd "C:\Tweego projects\Projects\Climate Rescue Adviser\CRA Repo\src"

This will point tweego to create the conversion twee file in that directory.

Then run this command through the Command Prompt:

tweego -d -o Adviser\_twee\_format.txt "C:/Tweego projects/Projects/Climate Rescue Adviser/Twine2 source/Draft Adviser\_T2-SC.html"

This command above needs double quotes around the file path (and if necessary the file name) to allow tweego to deal with any white spaces, and the file source path has to have the **direction of the slashes reversed**. Note: the twee file has a .txt extension so that tweego doesn’t accidentally try to compile it into an output .html file.

Then use Joeyrsp’s Python code to split the multi-passage Twee file into a set of single passage files. This can be run through the command prompt, set to the same folder as used when running tweego.

py "C:\Tweego projects\Utilities\split.py" .\Adviser\_twee\_format.txt

The single passage .twee files will be placed in a folder called “split”.

After successfully converting the twine file to a twee file and then splitting that file, delete the copied twine source file and the unsplit twee file to make sure that old versions of the Adviser are not left in the system. Rename the ‘split’ folder to ‘Unsorted’.  
  
The files in the Unsorted folder (renamed from ‘split’) need to be placed (manually) in the correct cluster folders within the ‘src’ folder.

***Compiling the .html file through the terminal***

Make sure the vscode Terminal is pointing to the CRA Repo folder:

cd "C:\Tweego projects\Projects\Climate Rescue Adviser\CRA Repo"

Then for a once off compilation of an .html output file, run this tweego command:

tweego -o docs/index.html src

Tweego can be set up in watch mode to automatically update the output file when changes are made:

tweego -o docs/index.html -w src

The long version is:

tweego -o "C:/Tweego projects/Projects/Climate Rescue Adviser/CRA Repo/docs/index.html" -w "C:/Tweego projects/Projects/Climate Rescue Adviser/CRA Repo/src"

For running in vscode terminal

./watch.bat

End the watch session using ctrl-c when the focus is in the terminal.

***Automating the Tweego watch and compile function via a VSCode task***

In the CRA REPO folder>.vscode: place a text file called tasks.json containing:

{

// See https://go.microsoft.com/fwlink/?LinkId=733558

// for the documentation about the tasks.json format

"version": "2.0.0",

"tasks": [

{

"label": "tweego watch",

"type": "shell",

"command": "./watch.bat",

"runOptions": {"runOn": "folderOpen"}

}

]

}

For advice on how to activate the task to run automatically, see:

<https://www.roboleary.net/vscode/2020/10/19/vscode-task-onstartup.html>

Go to command palette (ctrl-shift-p), run the option: “Tasks: Allow Automatic Tasks in Folder” and then choose the option: “Allow automatic tasks in folder”. Then the automatic watch task should work after VSCode has been rebooted.

***T3LT one time set up:***

ctrl-shift-p then select command option: **Add all unrecognised macros to Definition file**

Now vscode and the T3LT extension can be used to edit the twee files.

***Some tips for using vscode/T3LT extension***

ctrl-shift-p vscode shortcut to open all current registered commands (ie. vscode & extension)

ctrl-p in vscode (pure) – search for and select file

ctrl-p with > ….opens the command pallette

ctrl-` opens the terminal in vscode (ctrl-back tick)

The terminal panel space has several tabs – the first “problems” shows errors in code and if a problem is clicked on then the focus jumps to the error.

ctrl-r - open recent

ctrl-z undo!!! ☺

ctrl-, (ie. ctrl-comma) – opens settings for vscode.

ctrl-f – find in file

ctrl-shift-f find in workspace

alt-z to toggle word wrap in vscode

in search and replace: add folder names to limit which files are acted on

Alt-click - creates multiple cursors

# Regex syntax for use in vscode of twine2

In the search box choose the .\* search mode option

|  |  |
| --- | --- |
| . | any character except newline |
| ^ | When doing a multiline search means any character including line feed |
| \d | any numerical character |
| \w | alphanumeric character plus underline |
| \s | any whitespace character |
| \n | line break |
| \* | qualifies the preceding character or token to equal zero or any number of cases |
| + | qualifies the preceding character or token to equal one or any number of cases |
| | | match either side of the pipe – no limit to number of pipes/choices (an ‘or’ statement |
| [] | matches any character within |
| [-] | matches an alphanumeric range [f-k] or [4-7] |
| ^ | start of a string |
| $ | end of a string |
| \b | word boundary \bword , word\b, \bword\b |
| \t | tab |
| \n | linefeed |
| \r | carriage return |
| \ | when followed by itself or an non special character |

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions>

<https://regexr.com>

<https://regex101.com>

Passage name link find regex:

find at: <https://github.com/Joeyrsp/twee-tools/blob/main/link%20regex.txt>

Paste in the passage name (with special characters escaped) in the place of REPLACE.

In the vscode replace box insert:

find at: <https://github.com/Joeyrsp/twee-tools/blob/main/link%20regex.txt>

and then paste the new passage name over REPLACE.

If the new passage name contains a $ sign then that would need to be escaped using the $ sign eg. $$2

**Special case: To find broken first lines:**

Search: (::.+\{.+\})(.)

Replace: $1\n$2

To auto insert the regex into the vscode search, type **pbrokensearch//** and in the replace field type: **pbrokenrepair//**

(The problem being corrected is when the T3LT saves moved passage tiles incorrectly when moved in the map. The problem occurs when the first and second lines are merged. The correction inserts a line break where it should be after the map location json information.

**Special case: Cascading passage name changes**

Search: (\[(?:img\[[^\[\]]+\])?\[(?:[^\[\]]+(?:\||->))?)(REPLACE)((?:<-[^\[\]]+)?\](?:\[[^\[\]]+\])?\])

Replace: $1REPLACE$3

To auto insert the regex into the vscode search, type **pnamesearch//** and in the replace field type: **pnamereplace//**

# Twine users/related tools that have use-cases more like what RSTI is trying to do

====================================================

**Branching scenarios in training/elearning, story-based teaching**

[https://www.christytuckerlearning.com/**twine-makes-branching-scenarios-easier**/](https://www.christytuckerlearning.com/twine-makes-branching-scenarios-easier/)

[https://www.syniadlearning.com/portfolio/**branching-scenario-example/**](https://www.syniadlearning.com/portfolio/branching-scenario-example/)

[https://www.christytuckerlearning.com/**creative-solutions-with-twine-presentation**/](https://www.christytuckerlearning.com/creative-solutions-with-twine-presentation/)

[https://www.christytuckerlearning.com/**dont-restart-scenario-based-learning-go-back**/](https://www.christytuckerlearning.com/dont-restart-scenario-based-learning-go-back/)

[https://www.christytuckerlearning.com/**branching-scenario-prototype-in-twine**/](https://www.christytuckerlearning.com/branching-scenario-prototype-in-twine/)

[https://www.monash.edu/learning-teaching/TeachHQ/**learning-technologies/twine**](https://www.monash.edu/learning-teaching/TeachHQ/learning-technologies/twine)

[https://blog.training-pros.com/software-tools/**twine-branching-scenarios**/](https://blog.training-pros.com/software-tools/twine-branching-scenarios/)

[https://tedcurran.net/2018/10/**use-twine-for-branching-learning-scenarios**/](https://tedcurran.net/2018/10/use-twine-for-branching-learning-scenarios/)

<https://www.youtube.com/watch?v=arzpprGT9SE> Design a branching elearning scenario with Twine (part 1)

[https://blogs.sussex.ac.uk/tel/2019/10/01/**twine-a-choose-your-own-teaching-adventure/**](https://blogs.sussex.ac.uk/tel/2019/10/01/twine-a-choose-your-own-teaching-adventure/)

[https://ecampusontario.pressbooks.pub/techtoolsforteaching/chapter/11-**twine-interactive-storytelling/**](https://ecampusontario.pressbooks.pub/techtoolsforteaching/chapter/11-twine-interactive-storytelling/)

[http://twinery.org/questions/2034/**can-twine-be-used-for-educational-research**](http://twinery.org/questions/2034/can-twine-be-used-for-educational-research)

[https://blog.cathy-moore.com/2013/07/](https://blog.cathy-moore.com/2013/07/when-do-you-need-a-branching-scenario/)**[when-do-you-need-a-branching-scenario](https://blog.cathy-moore.com/2013/07/when-do-you-need-a-branching-scenario/)**[/](https://blog.cathy-moore.com/2013/07/when-do-you-need-a-branching-scenario/)

[https://blog.cathy-moore.com/2011/07/**sample-branching-scenario-cool-tool**/](https://blog.cathy-moore.com/2011/07/sample-branching-scenario-cool-tool/)

[https://blog.cathy-moore.com/tag/](https://blog.cathy-moore.com/tag/branching-scenarios/)**[branching-scenarios](https://blog.cathy-moore.com/tag/branching-scenarios/)**[/](https://blog.cathy-moore.com/tag/branching-scenarios/)

[https://blog.cathy-moore.com/training-design-toolkits/**how-to-design-scenario-based-training**/](https://blog.cathy-moore.com/training-design-toolkits/how-to-design-scenario-based-training/)

[https://blog.cathy-moore.com/training-design-toolkits/](https://blog.cathy-moore.com/training-design-toolkits/toolkit-partner-from-the-start/)**[toolkit-partner-from-the-start](https://blog.cathy-moore.com/training-design-toolkits/toolkit-partner-from-the-start/)**[/](https://blog.cathy-moore.com/training-design-toolkits/toolkit-partner-from-the-start/)

[https://blog.cathy-moore.com/2013/09/**12-cool-ways-to-use-scenarios**/](https://blog.cathy-moore.com/2013/09/12-cool-ways-to-use-scenarios/)

[https://blog.cathy-moore.com/2014/05/**branching-scenarios-how-many-decision-points**/](https://blog.cathy-moore.com/2014/05/branching-scenarios-how-many-decision-points/)

[https://blog.cathy-moore.com/2013/11/**4-ideas-you-should-steal-from-interactive-fiction**/](https://blog.cathy-moore.com/2013/11/4-ideas-you-should-steal-from-interactive-fiction/)

[https://blog.cathy-moore.com/2018/04/**3-ways-to-help-people-learn-from-mistakes-in-branching-scenarios/**](https://blog.cathy-moore.com/2018/04/3-ways-to-help-people-learn-from-mistakes-in-branching-scenarios/)

[https://blog.cathy-moore.com/2018/05/**3-cool-ideas-to-steal-for-your-training-scenarios**/](https://blog.cathy-moore.com/2018/05/3-cool-ideas-to-steal-for-your-training-scenarios/)

<http://wbt24.de/?cat=4373> (Cathy Moore describes the development of branching scenario with variables

<https://www.amazon.com.au/Map-hands-strategic-training-design-ebook/dp/B075RDL1SJ> “Map it” book by Cathy Moore

[https://elearningart.com/blog/**scenario-tips**/](https://elearningart.com/blog/scenario-tips/)

**Christy Tucker**, CDT (GPS)

Syniad Learning

Christy Tucker <christy@syniadlearning.com>

919-238-9807

https://www.syniadlearning.com

https://www.christytuckerlearning.com

**Building databases in twine**

[https://www.insidehighered.com/blogs/gradhacker/**twine-storytelling-study-aid**](https://www.insidehighered.com/blogs/gradhacker/twine-storytelling-study-aid)

**Managing complex/large twine projects**

***Em Lazer-Walker - A Modern Developer's Workflow For Twine***

<https://dev.to/lazerwalker/a-modern-developer-s-workflow-for-twine-4imp>

https://lazerwalker.com

hi@lazerwalker.com

# Information about Twine (all versions)

Information about Twine (all versions) has moved from the Twinery Wiki to the Twine Cookbook.

<https://twinery.org/cookbook/>

[https://twinelab.net/](https://twinelab.net/twine-resources)**[twine-resources](https://twinelab.net/twine-resources)**

[https://github.com/ChapelR/twine-](https://github.com/ChapelR/twine-resources)**[resource](https://github.com/ChapelR/twine-resources)**[s](https://github.com/ChapelR/twine-resources) (the Twine Resources repository)

<https://intfiction.org/c/authoring/twine/46> (active – read/write)

[https://**discord**app.com/invite/n5dJvPp](https://discordapp.com/invite/n5dJvPp)

[https://www.**reddit**.com/r/twinegames/](https://www.reddit.com/r/twinegames/) The Twine subreddit

<http://twinery.org/forum> (read only since 2017)

<https://twinery.org/questions/> (read only since 2019)

# CSS & html guides/templates for Twine

<https://gcbaccaris.itch.io/grimoire-one>

[https://itch.io/c/1625036/**coding-templates-resources**](https://itch.io/c/1625036/coding-templates-resources)

[https://www.w3schools.com/colors/](https://www.w3schools.com/colors/colors_names.asp)**[colors\_names](https://www.w3schools.com/colors/colors_names.asp)**[.asp](https://www.w3schools.com/colors/colors_names.asp)

[https://developer.mozilla.org/en-US/docs/Web/](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference)**[CSS/Reference](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference)**

# Additional Twine / SugarCube code resources

**Chapel's Custom Macro Collection**

[https://twinelab.net/**custom-macros-for-sugarcube-2**/#/](https://twinelab.net/custom-macros-for-sugarcube-2/#/)

**HiEv**

<https://qjzhvmqlzvoo5lqnrvuhmg-on.drv.tw/UInv/Sample_Code.html#Main%20Menu>

(Includes code for linking to passages externally using a url)

**SugarCube author (Thomas Edwards) add-on code**

<http://www.motoslave.net/sugarcube/2/#downloads>

**cyrusfirheir**

<https://github.com/cyrusfirheir/cycy-wrote-custom-macros/tree/master/live-update>

[https://github.com/cyrusfirheir/**twee3-language-tools**](https://github.com/cyrusfirheir/twee3-language-tools) (VSCode extension – with map)

**A list of sources for additional resources:**

<https://twinelab.net/twine-resources/#/?id=sugarcube-libraries>

**Kira’s sugarcube coding guide**

<https://pastebin.com/Ddrtuk2X>

# General guides for Twine 2

[https://www.adamhammond.com/](https://www.adamhammond.com/twineguide/)**[twineguide](https://www.adamhammond.com/twineguide/)**[/](https://www.adamhammond.com/twineguide/)

# Guides for Web languages

[https://developer.**mozilla**.org/en-US/docs/Web](https://developer.mozilla.org/en-US/docs/Web)

[https://www.**w3schools**.com](https://www.w3schools.com)

# Special passages that are automatically displayed in the final HTML

|  |  |
| --- | --- |
| Passage name | Explanation |
| “Start” | The first passage to display (in Twine 2 this passage can have **any** name – Twine 2 allows the developer to assign the “start” role to any passage. |
| StoryAuthor | Text in this passage inserts the story author’s name in the side bar (menu). Used to populate the authorial byline area in the menu (element ID: story-author). |
| StoryBanner | Can be used to include a graphic or logo at the top of the menu |
| StoryCaption | May be, and often is, used to add additional story UI elements and content to the menu. |
| StoryDisplayTitle | Sets the story's display title in the browser's titlebar and the menu (element ID: story-title). If omitted, the story title will be used instead. |
| StorySubtitle | Sets the story's subtitle in the menu (element ID: story-subtitle). |
| StoryMenu | The story menu only displays links—specifically, anything that creates an anchor element (<a>). While it renders content just as any other passage does, instead of displaying the rendered output as-is, it sifts through the output and builds its menu from the generated links contained therein. |
| StoryShare | Used to populate the contents of the Share dialog. Intended for social media links. NOTE: The Share dialog only displays links—specifically, anything that creates an anchor element (<a>). While it renders content just as any other passage does, instead of displaying the rendered output as-is, it sifts through the output and builds its contents from the generated links contained within. |
| StoryInterface | Used to replace SugarCube's default menu. Its contents are treated as raw HTML markup—i.e., none of SugarCube's special HTML processing is performed. It must contain, at least, an element with the ID passages that will be the main passage display area.  Additional elements, aside from the #passages element, may include either the data-init-passage or data-passage content attribute, whose value is the name of the passage used to populate the element—the passage will be processed as normal, meaning that markup and macros will work as expected. The data-init-passage attribute causes the element to be updated once at initialization, while the data-passage attribute causes the element to be updated upon each passage navigation. See:  <https://www.motoslave.net/sugarcube/2/docs/#special-passage-storyinterface> for links to more information in how to set up the alternative user interface. |
| PassageHeader | Prepended to each rendered passage.  Roughly equivalent to the :passagestart event. |
| PassageFooter | Appended to each rendered passage.  Roughly equivalent to the :passagerender event. |

[https://www.motoslave.net/sugarcube/2/docs/#**special-names**](https://www.motoslave.net/sugarcube/2/docs/#special-names)

[https://twinery.org/cookbook/sidebar\_left/sugarcube/sugarcube\_**sidebar\_left**.html](https://twinery.org/cookbook/sidebar_left/sugarcube/sugarcube_sidebar_left.html)

# Special passages that are NOT displayed in the final HTML

|  |  |
| --- | --- |
| Passage name | Explanation |
| StoryInit | Used for pre-story-start initialization tasks, like variable initialization (happens at the beginning of story initialization). Generates no visible output. |
| PassageReady | Used for pre-passage-display tasks, like redoing dynamic changes (happens before the rendering of each passage). Generates no visible output.  Roughly equivalent to the :passagestart event. |
| PassageDone | Used for post-passage-display tasks, like redoing dynamic changes (happens after the rendering and display of each passage). Generates no output visible.  Roughly equivalent to the :passagedisplay event. |

[https://www.motoslave.net/sugarcube/2/docs/#**special-names**](https://www.motoslave.net/sugarcube/2/docs/#special-names)

[https://twinery.org/cookbook/sidebar\_left/sugarcube/sugarcube\_**sidebar\_left**.html](https://twinery.org/cookbook/sidebar_left/sugarcube/sugarcube_sidebar_left.html)

# Special Twine 1 passages NOT used in Twine 2

|  |  |
| --- | --- |
| Passage name | Explanation |
| StoryTitle | Replaced by StoryDisplayTitle |
| StoryIncludes | Include other paths/URLs for .tws or .twee files to include in the build of this file |

# Special Twine 2 passages NOT used in SugarCube

|  |  |
| --- | --- |
| Passage name | Explanation |
| StorySettings | WARNING: Not used by SugarCube. The Config API serves the same basic purpose. |

# Special tags with predefined roles

|  |  |
| --- | --- |
| **Special tags**  (case sensitive) | **Explanation** |
| bookmark | Registers the passage into the Jump To menu. |
| nobr | Causes leading/trailing newlines to be removed and all remaining sequences of newlines to be replaced with single spaces before the passage is rendered. Equivalent to wrapping the entire passage in a <<nobr>> macro. See the Config.passages.nobr setting for a way to apply the same processing to all passages at once. |
| init | Registers the passage as an initialization passage. Used for pre-story-start initialization tasks, like variable initialization (happens at the beginning of story initialization). Generates no output. **(Must not be added to the Special Passage StoryInit.)** |
| Twine.audio | Registers the passage as an audio passage. See Guide: Media Passages for more information. |
| Twine.image | Registers the passage as an image passage. See Guide: Media Passages for more information. |
| Twine.video | Registers the passage as a video passage. See Guide: Media Passages for more information. |
| Twine.vtt | Registers the passage as a VTT passage. See Guide: Media Passages for more information. |
| widget | Registers the passage as <<widget>> macro definitions, which are loaded during startup. |

# Special tags that area NOT used any more

|  |  |
| --- | --- |
| **Special tags**  (case sensitive) | Explanation |
| Twine.private | Passages with this tag are **not** passed to the final HTML file. (This tag works in Tweego.) |
| annotation | Large area display in the story map, and passages with this tag are **not** passed to the final HTML file. |

Passages tagged with Twine.private can be stripped out of the .html file if the Twine 2 published file is run through Tweego to produce a new .html file – because Tweego recognises the Twine.private tag in the way it works in Twine 1. See the instructions near the end of this file.

# Preventing the passage of ‘private’ information into the final .html file – a work around

It should be possible to strip information in passages tagged with Twine.private by passing the Twine 2 published HTML through Tweego, since it supports the `Twine.private` tag.

eg., `tweego -o final.html twine2\_published.html

# Formatting text / Markup

[http://www.motoslave.net/sugarcube/2/docs/#](http://www.motoslave.net/sugarcube/2/docs/#markup)**[markup](http://www.motoslave.net/sugarcube/2/docs/#markup)**

|  |  |
| --- | --- |
| **Links** | |
| **Markup** | **Output** |
| SugarCube's link markup consists of a required **Link** component and optional **Text** and **Setter** components. The Link and Text components may be either plain text or any valid TwineScript expression, which will be evaluated early—i.e., when the link is initially processed. The Setter component, which only works with passage links, must be a valid TwineScript expression, of the <<set>> macro variety, which will be evaluated late—i.e., when the link is clicked on.  For the following examples assume: $go is "Grocery" and $show is "Go buy milk" | |
| [[Link]] | [[Grocery]]  [[$go]] |
| [[Text|Link]] | [[Go buy milk|Grocery]]  [[$show|$go]] |
| [[Link][Setter]] | [[Grocery][$bought to "milk"]]  [[$go][$bought to "milk"]] |
| [[Text|Link][Setter]] | [[Go buy milk|Grocery][$bought to "milk"]]  [[$show|$go][$bought to "milk"]] |

See <<link>> macros in the section on macros ie. “Inbuilt Twine 2 + SugarCane MACROS”.

|  |  |
| --- | --- |
| **Heading format** | |
| **Markup** | **Output** |
| ! | **Level 1 Heading** |
| !! | **Level 2 Heading** |
| !!! | **Level 3 Heading** |
| !!!! | **Level 4 Heading** |
| !!!!! | **Level 5 Heading** |
| !!!!!! | **Level 6 Heading** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Basic formatting** [https://www.motoslave.net/sugarcube/2/docs/#**markup-style**](https://www.motoslave.net/sugarcube/2/docs/#markup-style) | | | |
| **Type** | **Markup** | **Rendered as html** | **Output** |
| **Bold** | ''sample text''  (ie. two single quotation marks at each end) | <strong>sample text</strong> | **sample text** |
| **Italics** | //sample text// | <em>sample text</em> | *sample text* |
| **Underline** | \_\_sample text\_\_ | <u>sample text</u> | sample text |
| **Strikethrough** | ==sample text== | <s>sample text</s> | ~~sample text~~ |
| **Superscript** | sample ^^text^^ | sample<sup> text</sup> | sample text |
| **Subscript** | sample ~~text~~ | sample<sub> text</sub> | sample text |
| **Horizontal line** | ---- |  |  |
|  |  |  |  |

| **Characters/ Symbols** | |
| --- | --- |
| **Description** | **Code** |
| double dash – | &ndash; |
| triple dash — | &mdash; |
| Bullet code | &#8226; |
| Bullet code | &bull; |
| Size controlled bullet, ready to paste | <font size=3>&#8226;</font> |
| < symbol | &lt; |
| > symbol | &gt; |
| [ symbol | &#91; |
| ] symbol | &#93; |
| **|** (vertical bar) | &#124; |
| Non-breaking space | &nbsp; |
| \* asterisk | &#42; |
| ∗ Asterisk not recognised by code as an asterisk!! (called a ‘lowast’ – low asterisk?) | &#8727; &lowast; |
| Ready to paste degree symbol º | &#176; |
| Double chevrons (that don’t trigger macro behaviour in Twine) | « » |
| Double angle brackets (that don’t trigger macro behaviour in Twine) | ≪ ≫ |

| **Text colour** | |
| --- | --- |
| **Description** | **Code** |
| Xxxxxxx | <font color=red>Xxxxxxx</font>  Colours that work: yellow, cyan, red, blue, white (plus?) |

| **HTML tags** | | |
| --- | --- | --- |
| **Purpose** | **HTML** | **Comment** |
| **Image call** | <IMG SRC="http://URL"> |  |
| **Centre text** | <center>Text</center> |  |
| **Tooltips** | <abbr title="ToolTipText">MainText</abbr> |  |
| **Forced break** | <br> | A line break (best used in a <<nobr>> block) or in a nobr tagged passage |
| **Smaller tex** | <small>Text</small> | To shrink font |
| **Bigger text** | <big>Text</big> | To enlarge font |

**HTML code for tables**

An example table:

<table style="width: 70%; border-collapse: collapse;">

<tr>

<td width="33.3%">&nbsp;</td>

<td class="grayed" style="width: 33.3%; border-style: solid;"><center><strong>Earth system level</strong></center></td>

<td class="grayed" style="width: 33.3%; border-style: solid;"><center><strong>Local level</strong></center></td>

</tr>

<tr>

<td class="grayed" style="border-style: solid;"><strong>&nbsp;&nbsp;Prevention</strong></td>

<td style="border-style: solid;"><center><strong><span style="font-family: Wingdings;">&uuml;</span></strong></center></td>

<td style="border-style: solid;"><center>?</center></td>

</tr>

</table>

Make sure that tables are bracketed with <<nobr>><</nobr>> so that Tweego with SugarCube doesn’t add <br> above and below the text for each row.

Also make sure that para coding is not added: <p></p>

Use: class="grayed" in the style tag of grayed cells in a table. The grey colour will then auto-toggle depending on whether the Adviser is in dark mode or light mode. The table display responds to the CSS code added to the Story Stylesheet passage.

Another way to format tables is:

<style>

table {

border: 1px solid black;

}

</style>

<table>

<tr>

<td>text</td>

</tr>

</table>

| **List format** | | | |
| --- | --- | --- | --- |
| **Type** | **Syntax & Example** | **Rendered As** | **Displays As (roughly)** |
| **Unordered** | \* A list item \* Another list item  \*\* **Nested** one level down  \*\*\* Nested two levels  (Unordered lists have only 3 levels of unique bullets. | <ul> <li>A list item</li> <li>Another list item</li> </ul> | * A list item * Another list item |
| **Ordered** | # A list item # Another list item  ##  ###  (numbered lists seem to be nested endlessly) | <ol> <li>A list item</li> <li>Another list item</li> </ol> | 1. A list item 2. Another list item |

| **Blockquote** | | |
| --- | --- | --- |
| **Syntax & Example** | **Rendered As** | **Displays As (roughly)** |
| >Line 1  >Line 2  >>Nested 1  >>Nested 2 | <blockquote>Line 1<br>  Line 2<br>  <blockquote>Nested 1<br>  Nested 2<br>  </blockquote></blockquote> | Line 1 Line 2  Nested 1 Nested 2 |

| **Special formatting codes** | | |
| --- | --- | --- |
| **Type** | **Syntax** | **Rendered As** |
| **Inline** | ---- | Full width horizontal line |
| **Non-breaking line** | The rain in Spain falls¬  \ mainly on the plain. | The rain in Spain falls mainly on the plain. |
|  | The various no-break features—[<<nobr>> macro](https://www.motoslave.net/sugarcube/2/docs/#macros-macro-nobr), [nobr special tag](https://www.motoslave.net/sugarcube/2/docs/#special-tag-nobr), and [Config.passages.nobr setting](https://www.motoslave.net/sugarcube/2/docs/#config-api-property-passages-nobr)—all perform a similar, though slightly different, function. | |

| **Code within-a passage** | | |
| --- | --- | --- |
| **Type** | **Syntax & Example** | **Rendered As** |
| **Inline** | {{{Code}}} | Code |
| **Block** | <pre><code>Code  More code  Even more code  </code></pre> | Code  More code  Even more code |

**Non-active emulation text**

(Using a character set that is not recognised by Twine as control text.)

&#8727; &lowast; ∗ Asterisk not recognised by code as an asterisk!! (called a ‘lowast’ – low asterisk?)

〚Link-text|http׃⁄⁄url.url〛

«actions»

«back»

«choice»

«print»

«return»

«set»

«textarea»

«update»

≪actions≫

≪back≫

≪display≫

≪print≫

≪return≫

≪set≫

≪textarea≫

≪update≫

**The <<Back>> macro**

&lt;&lt; back &gt;&gt;

<IMG SRC="http://www.green-innovations.asn.au/Twine/Back.gif">

**The <<Choice>> macro**

&lt;&lt; choice &gt;&gt;

<IMG SRC="http://www.green-innovations.asn.au/Twine/Choice-macro.gif">

**The <<Display>> macro**

&lt;&lt; display &gt;&gt;

<IMG SRC="http://www.green-innovations.asn.au/Twine/Display-macro.gif">

**Warning**: These methods, below, for making text verbatim don’t seem to prevent the text from having active effect in all cases. Not sure what the failure pattern is. Combining triple quotes with chevrons instead of angle brackets seems to render everything literal and inactive.

| **Verbatim text in TwineScript and html** | | |
| --- | --- | --- |
| **Type** | **Syntax & Example** | **Displays As (roughly)** |
| **Triple double quotes** | """No //format//""" | No //format// |
| **<nowiki> tag** | <nowiki>No //format//</nowiki> | No //format// |
| <**html**><**/html**> |  | |
| A set of opening and closing <html> tags—i.e., <html></html>—defines the verbatim HTML markup. The verbatim HTML markup disables processing of all markup contained within—both SugarCube and HTML—passing its contents directly into the output as HTML markup for the browser. Thus, you should only use plain HTML markup within the verbatim markup—meaning using none of SugarCube's special HTML attributes or directives.  The verbatim text markup disables processing of all markup contained within—both SugarCube and HTML—passing its contents directly into the output as plain text. | | |

| **Commenting out – does nor appear in the visible html display – but is passed into the HTML file as a comment** |
| --- |
| **SugarCube passage text** |
| **/\*** Singe line comment. **\*/** |
| **/\*** Multi-line  comment. **\*/** |
| **/%** Single line or multi-line **%/** |
| **<!**-- Single line or multi-line comment. --> |
| **Javascript** |
| **//** This is a single line Javascript comment – anything following the operator to the end of he line..  /\* This is a multi-line  Javascript comment \*/ |
| **CSS** |
| **/\*** Single line or multi-line CSS comment. **\*/** |

[https://www.motoslave.net/sugarcube/2/docs/#**markup-comment**](https://www.motoslave.net/sugarcube/2/docs/#markup-comment)

I haven’t been able to find way to stop comment text being included in the published file or to make a passage private.  (Passages tagged with Twine.private can be stripped out of the .html file if the Twine 2 published file is run through Tweego to produce a new .html file – because Tweego recognises the Twine.private tag rin the way it works in Twine 1.)

# Supported variable types

**Primitives**

Booleans—e.g., true & false

Numbers—e.g., 42, 3.14, & -17.01

Strings—e.g., "I like pie" & 'You like pie'

null

undefined

**Objects**

Array

Date

Map

Set

Generic objects

|  |  |  |
| --- | --- | --- |
| **Notation types** | **Syntax** | **Example** |
| Simple variable | $variable | $name |
| Property access, dot notation | $variable.property | $thing.name |
| Index/property access, square bracket notation | $variable[numericIndex]  $variable["property"]  $variable['property']  $variable[$indexOrPropertyVariable] | $thing[0]  $thing["name"]  $thing['name']  $thing[$member] |

**Some array use cases :**

***Create empty array.***

<<set $*array* to []>>

***Set array to values***

This creates or resets an array with two items with the value 0 each.

<<set $*array* to [0,0]>>

***Changing specific array values***

This changes the first array member to the value 1.

<<set $*array* [0] to 1>>

***Adding items to an array*** *(at the end)*

This .push method adds an item to the end of an array.

<<set $*array*.push(“cat”)>>

***Deleting all items of a type from an array***

This .delete method removes all instances of one or more item types.

<<run $*array*.delete(4,6)>>

***Deleting items at defined index locations of an array***

This .delete method removes all instances of one or more item types.

<<run $*array*.delete(4,6)>>

***To find the first instance of an item in an array***

If the item is not in the array, $index will be set to -1, otherwise it will be set to the index value if the first instance of the item, counting from the left.

<<set $index = $array.indexOf(“cat”)>>

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/**indexOf**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/indexOf)

***Creating a new array using Joey’s .without() method***

<<set $newArray = $initialArray.without(“cat”,”horse”)>>

# Inbuilt Twine 2 + SugarCane MACROS

|  |  |
| --- | --- |
| **Variables Macros** | <https://www.motoslave.net/sugarcube/2/docs/#macros-variables> |
| **<<capture>>** | <http://www.motoslave.net/sugarcube/2/docs/#macros-macro-capture> |
| **<<set>>** | <<set expression>>  Sets story $variables and temporary \_variables based on the given expression.  → Using the TwineScript "to" operator  <<set $cheese to "a nice, sharp cheddar">> → Assigns "a nice, sharp cheddar" to story variable $cheese  <<set $chestEmpty to true>> → Assigns boolean true to story variable $chestEmpty  <<set $gold to $gold + 5>> → Adds 5 to the value of story variable $gold  <<set \_counter to \_counter + 1>> → Adds 1 to the value of temporary variable \_counter  → Using standard JavaScript operators  <<set $cheese = "a nice, sharp cheddar">> → Assigns "a nice, sharp cheddar" to story variable $cheese  <<set $chestEmpty = true>> → Assigns boolean true to story variable $chestEmpty  <<set $gold += 5>> → Adds 5 to the value of story variable $gold  <<set \_counter += 1>> → Adds 1 to the value of temporary variable \_counter  See the list of “See the list of “TwineScript assignment operators”  <http://www.motoslave.net/sugarcube/2/docs/#macros-macro-set> |
| **<<unset>>** | <<unset variableList>>  Unsets story $variables and temporary \_variables.  eg.  <<unset $cheese, $chestEmpty, $gold>>  <<unset \_someTempVar>>  [http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**unset**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-unset) |
| **Scripting Macros** | <https://www.motoslave.net/sugarcube/2/docs/#macros-scripting> |
| **<<run>>** | [http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**run**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-run) |
| **<<script>>** | [http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**script**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-script) |
| **Display Macros** | <https://www.motoslave.net/sugarcube/2/docs/#macros-display> |
| **<<=>>** | <<= expression>>  Outputs a string representation of the result of the given expression. This macro is an alias for <<print>>. |
| **<<->>** | <<- expression>>  Outputs a string representation of the result of the given expression. This macro is functionally identical to <<print>>, save that it also encodes HTML special characters in the output. |
| **<<include>>** | <<include [[passageName]] [elementName]>>  <<include passageName [elementName]>>  <<include linkMarkup [elementName]>>  <<include "Go West">> → Include the passage "Go West"  <<include [[Go West]]>> → Include the passage "Go West"  <<include "Go West" "div">> → Include the passage "Go West", wrapping it within a <div>  <<include [[Go West]] "div">> → Include the passage "Go West", wrapping it within a <div>  Replaces Twine 1 <<display>> |
| **<<nobr>>** | <<nobr>> <</nobr>>  Non-breaking block  <<endnobr>> is deprecated. |
| **<<print>>** | <<print expression>>  Outputs a string representation of the result of the given expression. |
| **<<silently>>** | [http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**silently**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-silently) |
| **<<type>>** |  |
| **Control Macros** | <https://www.motoslave.net/sugarcube/2/docs/#macros-control> |
| **<<if>>** | <<if conditional>> … [<<elseif conditional>> …] [<<else>> …] <</if>>  <<if $daysUntilLoanDue is 0>><<include "Panic">><</if>>  <<if $cash lt 5>>  I'm sorry, ma'am, but you don't have enough for the pie.  <<else>>  <<set $cash -= 5, $hasMeatPie = true>>  One meat pie, fresh out of the oven, coming up!  <</if>>  <<if $affection gte 75>>  I love you!  <<elseif $affection gte 50>>  I like you.  <<elseif $affection gte 25>>  I'm okay with you.  <<else>>  Get out of my face.  <</if>>  <<if $hullBreached>>  <<if $wearingHardSuit>>  <<include "That was close">>  <<elseif $wearingSoftSuit>>  <<include "Hole in suit">>  <<else>>  <<include "You die">>  <</if>>  <</if>>  See the list of ***English-based*** “TwineScript conditional (evaluation) operators” below or see here:.  [http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**if**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-if)  See the list of ***mathematical*** “JavaScript conditional (evaluation) operators” below or see here:.  [http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**if**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-if)  <<endif>> is deprecated. |
| **<<for>>** |  |
| **<<break>>** |  |
| **<<continue>>** |  |
| **<<switch>>** |  |
| **Interactive Macros** | <https://www.motoslave.net/sugarcube/2/docs/#macros-interactive> |
| **<<button>>** | <<button “linkText” [passageName]>> … <</button>>  <<button [[linkMarkup]]>> … <</button>>  <<button imageMarkup>> … <</button>>  Creates a button that silently executes its contents when clicked, optionally forwarding the player to another passage. May be called with either the link text and passage name as separate arguments, a link markup, or an image markup.  Examples: → Without forwarding: a very basic statistic setting example  Strength: <<set $pcStr to 10>><span id="stats-str"><<print $pcStr>></span> \  ( <<button "[+]">><<set $pcStr++>><<replace "#stats-str">><<print $pcStr>><</replace>><</button>> \  | <<button "[-]">><<set $pcStr-->><<replace "#stats-str">><<print $pcStr>><</replace>><</button>> )  → With forwarding: execute a script, then go to the specified passage  <<button "Onward, Reginald!" "On with the story">><<script>>/\* (code) \*/<</script>><</button>>  <<button [[Onward, Reginald!|On with the story]]>><<script>>/\* (code) \*/<</script>><</button>>  <<button [img[onward.jpg][On with the story]]>><<script>>/\* (code) \*/<</script>><</button>>  [http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**button**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-button) |
| **<<checkbox>>** | <<checkbox receiverName uncheckedValue checkedValue [autocheck|checked]>>  Creates a checkbox, used to modify the value of the variable with the given name.  **receiverName**: The name of the variable to modify, which must be quoted—e.g., "$foo". Object and array property references are also supported—e.g., "$foo.bar", "$foo['bar']", & "$foo[0]".  **uncheckedValue**: The value set by the checkbox when unchecked.  **checkedValue**: The value set by the checkbox when checked.  **autocheck**: (optional) Keyword, used to signify that the checkbox should be automatically set to the checked state based on the current value of the receiver variable. NOTE: Automatic checking may fail on non-primitive values—i.e., on arrays and objects.  **checked**: (optional) Keyword, used to signify that the checkbox should be in the checked state.  What pies do you enjoy?  \* <<checkbox "$pieBlueberry" false true autocheck>> Blueberry?  \* <<checkbox "$pieCherry "" "cherry" autocheck>> Cherry?  \* <<checkbox "$pieCoconutCream" false true autocheck>> Coconut cream?  What pies do you enjoy?  \* <<checkbox "$pieBlueberry" false true checked>> Blueberry?  \* <<checkbox "$pieCherry" "" "cherry">> Cherry?  \* <<checkbox "$pieCoconutCream" false true checked>> Coconut cream? |
| **<<cycle>>** |  |
| **<<link>>** | <<link linkText [passageName]>> … <</link>>  <<link linkMarkup>> … <</link>>  <<link imageMarkup>> … <</link>>  Creates a link that silently executes its contents when clicked, optionally forwarding the player to another passage. May be called with either the link text and passage name as separate arguments, a link markup, or an image markup.  → Without forwarding: a very basic statistic setting example  Strength: <<set $pcStr to 10>><span id="stats-str"><<print $pcStr>></span> \  ( <<link "[+]">><<set $pcStr++>><<replace "#stats-str">><<print $pcStr>><</replace>><</link>> \  | <<link "[-]">><<set $pcStr-->><<replace "#stats-str">><<print $pcStr>><</replace>><</link>> )  → With forwarding: execute a script, then go to the specified passage  <<link "Onward, Reginald!" "On with the story">><<script>>/\* (code) \*/<</script>><</link>>  <<link [[Onward, Reginald!|On with the story]]>><<script>>/\* (code) \*/<</script>><</link>>  <<link [img[onward.jpg][On with the story]]>><<script>>/\* (code) \*/<</script>><</link>> |
| **<<linkappend>>** |  |
| **<<linkprepend>>** |  |
| **<<linkreplace>>** |  |
| **<<listbox>>** |  |
| **<<numberbox>>** |  |
| **<<radiobutton>>** | <<radiobutton receiverName checkedValue [autocheck|checked]>>  **receiverName**: The name of the variable to modify, which must be quoted—e.g., "$foo". Object and array property references are also supported—e.g., "$foo.bar", "$foo['bar']", & "$foo[0]".  **checkedValue**: The value set by the radio button when checked.  **autocheck**: (optional) Keyword, used to signify that the radio button should be automatically set to the checked state based on the current value of the receiver variable. NOTE: Automatic checking may fail on non-primitive values—i.e., on arrays and objects.  **checked**: (optional) Keyword, used to signify that the radio button should be in the checked state.  NOTE: Only one radio button in a group—i.e., those using the same receiver variable—should be so checked.  What's your favorite pie?  \* <<radiobutton "$pie" "blueberry" autocheck>> Blueberry?  \* <<radiobutton "$pie" "cherry" autocheck>> Cherry?  \* <<radiobutton "$pie" "coconut cream" autocheck>> Coconut cream?  What's your favorite pie?  \* <<radiobutton "$pie" "blueberry" checked>> Blueberry?  \* <<radiobutton "$pie" "cherry">> Cherry?  \* <<radiobutton "$pie" "coconut cream">> Coconut cream?  <<radio>> no longer works |
| **<<textarea>>** | <<textarea “receiverName” defaultValue [autofocus]>>  **receiverName**: The name of the variable to modify, which must be quoted—e.g., "$foo". Object and array property references are also supported—e.g., "$foo.bar", "$foo['bar']", & "$foo[0]".  **defaultValue**: The default value of the text block.  **autofocus**: (optional) Keyword, used to signify that the text block should automatically receive focus. Only use the autofocus keyword once per page; attempting to focus more than one element is undefined behavior.  → Creates a text block that modifies $pieEssay  Write a short essay about pies:  <<textarea "$pieEssay" "">>  → Creates an automatically focused text block that modifies $pieEssay  Write a short essay about pies:  <<textarea "$pieEssay" "" autofocus>>  [http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**textarea**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-textarea) |
| **<<textbox>>** | <<textbox receiverName defaultValue [passage] [autofocus]>>  **receiverName**: The name of the variable to modify, which must be quoted—e.g., "$foo". Object and array property references are also supported—e.g., "$foo.bar", "$foo['bar']", & "$foo[0]".  **defaultValue**: The default value of the text box.  **passage**: (optional) The name of the passage to go to if the return/enter key is pressed. May be called either with the passage name or with a link markup.  **autofocus**: (optional) Keyword, used to signify that the text box should automatically receive focus. Only use the keyword once per page; attempting to focus more than one element is undefined behaviour.  → Creates a text box that modifies $pie  What's your favorite pie? <<textbox "$pie" "Blueberry">>  → Creates an automatically focused text box that modifies $pie  What's your favorite pie? <<textbox "$pie" "Blueberry" autofocus>>  → Creates a text box that modifies $pie and forwards to the "Cakes" passage  What's your favorite pie? <<textbox "$pie" "Blueberry" "Cakes">>  → Creates an automatically focused text box that modifies $pie and forwards to the "Cakes" passage  What's your favorite pie? <<textbox "$pie" "Blueberry" "Cakes" autofocus>>  [http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**textbox**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-textbox) |
| **Links Macros** | <https://www.motoslave.net/sugarcube/2/docs/#macros-links> |
| **<<actions>>** |  |
| **<<choice>>** |  |
| **<<return>>** |  |
| **<<back>>** |  |
| **DOM Macros** | <https://www.motoslave.net/sugarcube/2/docs/#macros-dom> |
| **<<addclass>>** |  |
| **<<append>>** |  |
| **<<copy>>** |  |
| **<<prepend>>** |  |
| **<<remove>>** |  |
| **<<removeclass>>** |  |
| **<<replace>>** |  |
| **<<toggleclass>>** |  |
| **Audio Macros** | <https://www.motoslave.net/sugarcube/2/docs/#macros-audio> |
| **<<audio>>** |  |
| **<<cacheaudio>>** |  |
| **<<createaudiogroup>>** |  |
| **<<createplaylist>>** |  |
| **<<masteraudio>>** |  |
| **<<playlist>>** |  |
| **<<removeaudiogroup>>** |  |
| **<<removeplaylist>>** |  |
| **<<waitforaudio>>** |  |
| **Miscellaneous Macros** | <https://www.motoslave.net/sugarcube/2/docs/#macros-miscellaneous> |
| <<**done**>> |  |
| <<**goto**>> | <<goto passageName>>  <<goto linkMarkup>>  **passageName**: The name of the passage to go to.  Link markup form  **linkMarkup**: The link markup to use (regular syntax only, no setters).  → Passage name form  <<goto "Somewhere over yonder">>  <<goto $selectedPassage>>  → Link markup form  <<goto [[Somewhere over yonder]]>>  <<goto [[$selectedPassage]]>>  [http://www.motoslave.net/sugarcube/2/docs/#macros-**macro-goto**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-goto) |
| <<**repeat**>> |  |
| <<**stop**>> |  |
| <<**timed**>> |  |
| <<**widget**>> |  |

|  |  |  |
| --- | --- | --- |
| **Deprecated Macros –**These macros currently work but should be replaced | | |
| **Old** | **Replaced by:** | **Link** |
| **<<display>>** | <<**include**>> | <https://www.motoslave.net/sugarcube/2/docs/#macros-macro-display> |
| **<<forget>>** | **forget**() | <https://www.motoslave.net/sugarcube/2/docs/#functions-function-forget> |
|  | <<**forget** $variable>>  << forget $variable =numerical expression>>  << forget $variable ="text string">> | |
| **<<remember>>** | **memorize**() or  **recall**() | <https://www.motoslave.net/sugarcube/2/docs/#functions-function-memorize>  <https://www.motoslave.net/sugarcube/2/docs/#functions-function-recall> |
|  | <<**remember** $variable>>  <<remember $variable =numerical expression>>  <<remember $variable ="text string">> | |
| **<<click>>** |  |  |
| **<<setplaylist>>** |  |  |

|  |  |
| --- | --- |
| **Twine 1 macros that have been dropped or replaced** | |
| **Twine 1 macros** | **Relationship to Twine 2-SugarCube macros** |
| <<**display**>> | Replaced by <<**include**>> |
| <<**forget** >> | Replaced by the **forget**() function |
| <<**$variable**>> | Replaced by the naked variable: **$variable** |
| <<**PassageName**>> | Replaced by the <<**include**>> variable. |
| <<**radio**>> | Replaced by <<**radiobutton**>> |
| <<**remember**>> | Replaced by the **memorize**() or **recall**()the functions. |
| <<**textinput**>> | Replaced by <<**textarea**>> and <<**textbox**>> |

# CUSTOM MACROS made for SugarCane

|  |  |
| --- | --- |
| **Custom Macros** | **Usage & source** |
| **<<clearvariables>>** | To clear ***all*** story (project) variables. Made by Joey SP |
| **<<clearremember>>** | To clear ***all*** variables stored to the remember cache using <<remember>> macro. Made by Joey SP |
| **<<clearmetadata>>** | To clear ***all*** variables stored in the metadata cache using <<memorize>> macro. Made by Joey SP |

# Functions

<https://www.motoslave.net/sugarcube/2/docs/#functions>

| **Function** | **Advice** |
| --- | --- |
| Any Javascript built-in function is available to Twine | |
| **memorize()** | memorize(key, value)  Sets the specified key and value within the story metadata store, which causes them to persist over story and browser restarts. To update the value associated with a key, simply set it again.  Parameters:   * key: (string) The key that should be set. * value: (any) The value to set.   Examples:  <<run memorize('$TestVariable', 29)>>  <<run memorize('$TestVariable', ‘horse’)>>  // Sets 'achievements', with the given value, in the metadata store.  <<run memorize('achievements', { ateYellowSnow : true })>>  // Sets 'ngplus', with the given value, in the metadata store.  <<run memorize('ngplus', true)>>  [http://www.motoslave.net/sugarcube/2/docs/#functions-function-**memorize**](http://www.motoslave.net/sugarcube/2/docs/#functions-function-memorize) |
| **recall()** | recall(key [, defaultValue]) → any  Returns the value associated with the specified key from the story metadata store or, if no such key exists, the specified default value, if any.  Parameters:   * key: (string) The key whose value should be returned. * defaultValue: (optional, any) The value to return if the key doesn't exist.   Examples:  <<set $TestVariable to recall('$TestVariable ')>>  // Set setup.achievements to the 'achievements' metadata or an empty generic object.  <<set setup.achievements to recall('achievements', {})>>  // Set setup.ngplus to the 'ngplus' metadata, with no default.  <<set setup.ngplus to recall('ngplus')>>  [http://www.motoslave.net/sugarcube/2/docs/#functions-function-**recall**](http://www.motoslave.net/sugarcube/2/docs/#functions-function-recall) |
| **forget()** | forget(key)  Removes the specified key, and its associated value, from the story metadata store.  <<run forget('$TestVariable')>>  <<run forget('achievements')>>  [http://www.motoslave.net/sugarcube/2/docs/#functions-function-**forget**](http://www.motoslave.net/sugarcube/2/docs/#functions-function-forget) | |
| **clone()** |  | |
| **either()** |  | |
| **hasVisited()** |  | |
| **lastVisited()** |  | |
| **importScripts()** |  | |
| **importStyles()** |  | |
| **passage()** | [http://www.motoslave.net/sugarcube/2/docs/#functions-function-**passage**](http://www.motoslave.net/sugarcube/2/docs/#functions-function-passage) | |
| **previous()** | <<if previous() is "Café">>…the most recent non-active passage is the Café passage…<</if>>  [http://www.motoslave.net/sugarcube/2/docs/#functions-function-**previous**](http://www.motoslave.net/sugarcube/2/docs/#functions-function-previous) | |
| **random()** | [http://www.motoslave.net/sugarcube/2/docs/#functions-function-**random**](http://www.motoslave.net/sugarcube/2/docs/#functions-function-random) | |
| **randomFloat()** |  | |
| **setPageElement()** |  | |
| **tags()** |  | |
| **temporary()** |  | |
| **time()** |  | |
| **turns()** |  | |
| **variables()** |  | |
| **visited()** |  | |
| **visitedTags()** |  | |
|  |  | |

**Issues with memorize(), recall() and forget():**

[https://www.reddit.com/r/twinegames/comments/q0jpmp/**achievements\_with\_memorize**/](https://www.reddit.com/r/twinegames/comments/q0jpmp/achievements_with_memorize/)

[https://intfiction.org/t/**a-persisting-scoreboard-for-sugarcube**/48311](https://intfiction.org/t/a-persisting-scoreboard-for-sugarcube/48311)

[https://intfiction.org/t/**help-with-memorize-function-twine-2-sugarcube-2**-30-0/45743](https://intfiction.org/t/help-with-memorize-function-twine-2-sugarcube-2-30-0/45743)

| **Built in browser functions** | **Advice** |
| --- | --- |
| Any Javascript built-in function is available to Twine | |
| confirm("String") | Used inside a <<set>> macro. Displays an OK/Cancel dialog box. If you click Yes, the function's value is true. Otherwise, it is false. The provided string is the question shown to the player.  [https://twinery.org/cookbook/twine1/functions/**confirm**.html](https://twinery.org/cookbook/twine1/functions/confirm.html) |
| alert("message") | Used inside a <<set>> macro. Displays an alert box, with the given text string displayed. It has no value - feel free to use it by simply writing.  [https://twinery.org/cookbook/twine1/functions/**alert**.html](https://twinery.org/cookbook/twine1/functions/alert.html) |
| open("url") | Used inside a <<set>> macro. When given a URL in string form, it opens a new browser tab containing that web page. It has no value.  [https://twinery.org/cookbook/twine1/functions/**open**.html](https://twinery.org/cookbook/twine1/functions/open.html) |

|  |  |
| --- | --- |
| **Twine 1 functions that have been dropped or replaced** | |
| **Twine 1 function** | **Relationship to Twine 2-SugarCube functions** |
| **parameters()** |  |
| [https://twinery.org/cookbook/twine1/functions/**parameter**.html](https://twinery.org/cookbook/twine1/functions/parameter.html) | |
| **prompt()** | Replaced by <<textbox>> macro. |

# Methods

[https://www.motoslave.net/sugarcube/2/docs/#**methods**](https://www.motoslave.net/sugarcube/2/docs/#methods)

# Expressions

<http://www.motoslave.net/sugarcube/2/docs/#twinescript-expressions>

You will, in all likelihood, use **expressions** most often within macros—e.g., <<set>>, <<print>>, <<if>>, <<for>>

# TwineScript conditional (evaluation) operators

For use with <<if>> statements:

| **Operator** | **Function** | **Example** |
| --- | --- | --- |
| **is** | Evaluates to true if both sides are strictly equal. | <<if $bullets is 6>> |
| **isnot** | Evaluates to true if both sides are strictly not equal. | <<if $pie isnot "cherry">> |
| **eq** | Evaluates to true if both sides are equivalent. | <<if $bullets eq 6>> |
| **neq** | Evaluates to true if both sides are not equivalent. | <<if $pie neq "cherry">> |
| **gt** | Evaluates to true if the left side is greater than the right side. | <<if $cash gt 5>> |
| **gte** | Evaluates to true if the left side is greater than or equal to the right side. | <<if $foundStars gte $neededStars>> |
| **lt** | Evaluates to true if the left side is less than the right side. | <<if $shoeCount lt ($peopleCount \* 2)>> |
| **lte** | Evaluates to true if the left side is less than or equal to the right side. | <<if $level lte 30>> |
| **not** | Flips a true evaluation to false, and vice versa. | <<if not $hungry>> |
| **and** | Evaluates to true if all subexpressions evaluate to true. | <<if $age gte 20 and $age lte 30>> |
| **or** | Evaluates to true if any subexpressions evaluate to true. | <<if $friend is "Sue" or $friend is "Dan">> |
| **def** | Evaluates to true if the right side is defined. | <<if def $mushrooms>> |
| **ndef** | Evaluates to true if the right side is not defined. | <<if ndef $bottlecaps>> |

The **def** and **ndef** operators have very low precedence, so it is strongly recommended that if you mix them with other operators, that you wrap them in parentheses—e.g., (def $style) and ($style is "girly").

# JavaScript conditional (evaluation) operators

For use with <<if>> statements:

| **Operator** | **Function** | **Example** |
| --- | --- | --- |
| **===** | Evaluates to true if both sides are strictly equal. | <<if $bullets === 6>> |
| **!==** | Evaluates to true if both sides are strictly not equal. | <<if $pie !== "cherry">> |
| **==** | Evaluates to true if both sides are equivalent. | <<if $bullets == 6>> |
| **!=** | Evaluates to true if both sides are not equivalent. | <<if $pie != "cherry">> |
| **>** | Evaluates to true if the left side is greater than the right side. | <<if $cash > 5>> |
| **>=** | Evaluates to true if the left side is greater than or equal to the right side. | <<if $foundStars >= $neededStars>> |
| **<** | Evaluates to true if the left side is less than the right side. | <<if $shoeCount < ($peopleCount \* 2)>> |
| **<=** | Evaluates to true if the left side is less than or equal to the right side. | <<if $level <= 30>> |
| **!** | Flips a true evaluation to false, and vice versa. | <<if !$hungry>> |
| **&&** | Evaluates to true if all subexpressions evaluate to true. | <<if $age >= 20 && $age <= 30>> |
| **||** | Evaluates to true if any subexpressions evaluate to true. | <<if $friend === "Sue" || $friend === "Dan">> |

# TwineScript assignment operators

For use with <<set>>:

|  |  |  |  |
| --- | --- | --- | --- |
| **TwineScript** | | | |
| **Operator** | | **Function** | **Example** |
| **to** | | Assigns the value on the right-hand side of the operator to the left-hand side. | <<set $apples to 6>> |
| **JavaScript assignment operators:** *(not an exhaustive list)* | | | |
| **Operator** | **Function** | | **Example** |
| **=** | Assigns the value on the right-hand side of the operator to the left-hand side. | | <<set $apples = 6>> |
| **+=** | Adds the value on the right-hand side of the operator to the current value on the left-hand side and assigns the result to the left-hand side. | | <<set $apples += 1>> |
| **-=** | Subtracts the value on the right-hand side of the operator from the current value on the left-hand side and assigns the result to the left-hand side. | | <<set $apples -= 1>> |
| **\*=** | Multiplies the current value on the left-hand side of the operator by the value on the right-hand side and assigns the result to the left-hand side. | | <<set $apples \*= 2>> |
| **/=** | Divides the current value on the left-hand side of the operator by the value on the right-hand side and assigns the result to the left-hand side. | | <<set $apples /= 2>> |
| **%=** | Divides the current value on the left-hand side of the operator by the value on the right-hand side and assigns the remainder to the left-hand side. | | <<set $apples %= 10>> |

# Transferring data between Twine projects

<http://twinery.org/questions/1102/how-to-transfer-variables-from-one-twine-story-to-another>

# SugarCube documentation – (on the SugarCube site)

(very detailed)

<https://www.motoslave.net/sugarcube/2/docs/> ….(home page)

[https://www.motoslave.net/sugarcube/2/docs/#**introduction**](https://www.motoslave.net/sugarcube/2/docs/#introduction)

[https://www.motoslave.net/sugarcube/2/docs/#**markup**-naked-variable](https://www.motoslave.net/sugarcube/2/docs/#markup-naked-variable)

[https://www.motoslave.net/sugarcube/2/docs/#**twinescript**](https://www.motoslave.net/sugarcube/2/docs/#twinescript) …(variables, expressions, functions, operators, etc.)

[https://www.motoslave.net/sugarcube/2/docs/#**macros**](https://www.motoslave.net/sugarcube/2/docs/#macros)

[https://www.motoslave.net/sugarcube/2/docs/#**functions**](https://www.motoslave.net/sugarcube/2/docs/#functions)

[https://www.motoslave.net/sugarcube/2/docs/#**methods**](https://www.motoslave.net/sugarcube/2/docs/#methods)

[https://www.motoslave.net/sugarcube/2/docs/#**special-names**](https://www.motoslave.net/sugarcube/2/docs/#special-names) …(special names to give ‘objects’ special functions

[https://www.motoslave.net/sugarcube/2/docs/#**css**](https://www.motoslave.net/sugarcube/2/docs/#css)

[https://www.motoslave.net/sugarcube/2/docs/#**html**](https://www.motoslave.net/sugarcube/2/docs/#html)

[https://www.motoslave.net/sugarcube/2/docs/#**events**](https://www.motoslave.net/sugarcube/2/docs/#events)

[https://www.motoslave.net/sugarcube/2/docs/#**config-api**](https://www.motoslave.net/sugarcube/2/docs/#config-api)

[https://www.motoslave.net/sugarcube/2/docs/#**dialog-api**](https://www.motoslave.net/sugarcube/2/docs/#dialog-api)

[https://www.motoslave.net/sugarcube/2/docs/#**engine-api**](https://www.motoslave.net/sugarcube/2/docs/#engine-api)

[https://www.motoslave.net/sugarcube/2/docs/#**fullscreen-api**](https://www.motoslave.net/sugarcube/2/docs/#fullscreen-api)

[https://www.motoslave.net/sugarcube/2/docs/#**loadscreen-api**](https://www.motoslave.net/sugarcube/2/docs/#loadscreen-api)

[https://www.motoslave.net/sugarcube/2/docs/#**macro-api**](https://www.motoslave.net/sugarcube/2/docs/#macro-api)

[https://www.motoslave.net/sugarcube/2/docs/#**macrocontext-api**](https://www.motoslave.net/sugarcube/2/docs/#macrocontext-api)

[https://www.motoslave.net/sugarcube/2/docs/#**passage-api**](https://www.motoslave.net/sugarcube/2/docs/#passage-api)

[https://www.motoslave.net/sugarcube/2/docs/#**save-api**](https://www.motoslave.net/sugarcube/2/docs/#save-api)

[https://www.motoslave.net/sugarcube/2/docs/#**setting-api**](https://www.motoslave.net/sugarcube/2/docs/#setting-api)

[https://www.motoslave.net/sugarcube/2/docs/#**simpleaudio-api**](https://www.motoslave.net/sugarcube/2/docs/#simpleaudio-api)

[https://www.motoslave.net/sugarcube/2/docs/#**audiotrack-api**](https://www.motoslave.net/sugarcube/2/docs/#audiotrack-api)

[https://www.motoslave.net/sugarcube/2/docs/#**audiorunner-api**](https://www.motoslave.net/sugarcube/2/docs/#audiorunner-api)

[https://www.motoslave.net/sugarcube/2/docs/#**audiolist-api**](https://www.motoslave.net/sugarcube/2/docs/#audiolist-api)

[https://www.motoslave.net/sugarcube/2/docs/#**state-api**](https://www.motoslave.net/sugarcube/2/docs/#state-api)

[https://www.motoslave.net/sugarcube/2/docs/#**story-api**](https://www.motoslave.net/sugarcube/2/docs/#story-api)

[https://www.motoslave.net/sugarcube/2/docs/#**template-api**](https://www.motoslave.net/sugarcube/2/docs/#template-api)

[https://www.motoslave.net/sugarcube/2/docs/#**ui-api**](https://www.motoslave.net/sugarcube/2/docs/#ui-api)

[https://www.motoslave.net/sugarcube/2/docs/#**uibar-api**](https://www.motoslave.net/sugarcube/2/docs/#uibar-api)

[https://www.motoslave.net/sugarcube/2/docs/#**guide-state-sessions-and-saving**](https://www.motoslave.net/sugarcube/2/docs/#guide-state-sessions-and-saving)

[https://www.motoslave.net/sugarcube/2/docs/#**guide-tips**](https://www.motoslave.net/sugarcube/2/docs/#guide-tips)

[https://www.motoslave.net/sugarcube/2/docs/#**guide-media-passages**](https://www.motoslave.net/sugarcube/2/docs/#guide-media-passages)

[https://www.motoslave.net/sugarcube/2/docs/#**guide-test-mode**](https://www.motoslave.net/sugarcube/2/docs/#guide-test-mode)

[https://www.motoslave.net/sugarcube/2/docs/#**guide-typescript**](https://www.motoslave.net/sugarcube/2/docs/#guide-typescript)

[https://www.motoslave.net/sugarcube/2/docs/#**guide-installation**](https://www.motoslave.net/sugarcube/2/docs/#guide-installation)

[https://www.motoslave.net/sugarcube/2/docs/#**guide-code-updates**](https://www.motoslave.net/sugarcube/2/docs/#guide-code-updates)

# Twine 2 advice from the Twine Cookbook

[https://twinery.org/cookbook/questions/**links**.html](https://twinery.org/cookbook/questions/links.html)

[https://twinery.org/cookbook/questions/**style\_text**.html](https://twinery.org/cookbook/questions/style_text.html)

[https://twinery.org/cookbook/questions/**stories\_saved**.html](https://twinery.org/cookbook/questions/stories_saved.html)

[https://twinery.org/cookbook/terms/**terms\_stories**.html](https://twinery.org/cookbook/terms/terms_stories.html)  (especially the IFID identifier)

<https://twinery.org/cookbook/terms/terms_passages.html> (mainly about links)

[https://twinery.org/cookbook/terms/**terms\_variables**.html](https://twinery.org/cookbook/terms/terms_variables.html)

[https://twinery.org/cookbook/terms/terms\_**markdown\_and\_markup**.html](https://twinery.org/cookbook/terms/terms_markdown_and_markup.html)

[https://twinery.org/cookbook/terms/terms\_**storyformats**.html](https://twinery.org/cookbook/terms/terms_storyformats.html)

[https://twinery.org/cookbook/terms/terms\_**macros**.html](https://twinery.org/cookbook/terms/terms_macros.html)

[https://twinery.org/cookbook/terms/terms\_**css**.html](https://twinery.org/cookbook/terms/terms_css.html)

[https://twinery.org/cookbook/terms/terms\_**javascript**.html](https://twinery.org/cookbook/terms/terms_javascript.html)

[https://twinery.org/cookbook/terms/terms\_**html**.html](https://twinery.org/cookbook/terms/terms_html.html)

[https://twinery.org/cookbook/introduction/twine2\_**story\_listing**.html](https://twinery.org/cookbook/introduction/twine2_story_listing.html)

[https://twinery.org/cookbook/introduction/twine2\_**passages\_view**.html](https://twinery.org/cookbook/introduction/twine2_passages_view.html)

[https://twinery.org/cookbook/introduction/**passages**.html](https://twinery.org/cookbook/introduction/passages.html)

[https://twinery.org/cookbook/introduction/**story\_formats**.html](https://twinery.org/cookbook/introduction/story_formats.html)

[https://twinery.org/cookbook/starting/twine2/**firststory**.html](https://twinery.org/cookbook/starting/twine2/firststory.html)

[https://twinery.org/cookbook/starting/twine2/**storyformat**.html](https://twinery.org/cookbook/starting/twine2/storyformat.html)  (choosing story formats)

[https://twinery.org/cookbook/starting/twine2/**creatinglinks**.html](https://twinery.org/cookbook/starting/twine2/creatinglinks.html)

[https://www.motoslave.net/**sugarcube**/2/docs/#**markup-link**](https://www.motoslave.net/sugarcube/2/docs/#markup-link)

[https://twinery.org/cookbook/starting/twine2/**review**.html](https://twinery.org/cookbook/starting/twine2/review.html) (reviewing stories)

[https://twinery.org/cookbook/**html/reviewing**.html](https://twinery.org/cookbook/html/reviewing.html)

[https://twinery.org/cookbook/html/storyformats/**sugarcube**.html](https://twinery.org/cookbook/html/storyformats/sugarcube.html) ….( SugarCube HTML)

[https://twinery.org/cookbook/**css/selectors**.html](https://twinery.org/cookbook/css/selectors.html)

[https://twinery.org/cookbook/**css**/storyformats/**sugarcube**.html](https://twinery.org/cookbook/css/storyformats/sugarcube.html) …….(SugarCube CSS)

[https://twinery.org/cookbook/**js/reviewing**.html](https://twinery.org/cookbook/js/reviewing.html)

[https://twinery.org/cookbook/**js**/storyformats/**sugarcube**.html](https://twinery.org/cookbook/js/storyformats/sugarcube.html)

<https://www.motoslave.net/sugarcube/2/> …….(go to the Add-Ons section)

[https://twinery.org/cookbook/addingfunctionality/sugarcube/sugarcube\_](https://twinery.org/cookbook/addingfunctionality/sugarcube/sugarcube_adding_functionality.html)**[adding\_functionality](https://twinery.org/cookbook/addingfunctionality/sugarcube/sugarcube_adding_functionality.html)**[.html](https://twinery.org/cookbook/addingfunctionality/sugarcube/sugarcube_adding_functionality.html)

[https://www.motoslave.net/sugarcube/2/docs/#**macro-api**](https://www.motoslave.net/sugarcube/2/docs/#macro-api)

[https://twinery.org/cookbook/arrays/sugarcube/**sugarcube\_arrays**.html](https://twinery.org/cookbook/arrays/sugarcube/sugarcube_arrays.html)

[https://twinery.org/cookbook/audio/sugarcube/sugarcube\_**audio**.html](https://twinery.org/cookbook/audio/sugarcube/sugarcube_audio.html)

[https://twinery.org/cookbook/conditionalstatements/sugarcube/sugarcube\_**conditionalstatements**.html](https://twinery.org/cookbook/conditionalstatements/sugarcube/sugarcube_conditionalstatements.html)

[https://twinery.org/cookbook/cssselectors/sugarcube/sugarcube\_**cssselectors**.html](https://twinery.org/cookbook/cssselectors/sugarcube/sugarcube_cssselectors.html)

[https://twinery.org/cookbook/passagetags/sugarcube/sugarcube\_**passagetags**.html](https://twinery.org/cookbook/passagetags/sugarcube/sugarcube_passagetags.html)

[https://twinery.org/cookbook/cycling/sugarcube/sugarcube\_**cycling**.html](https://twinery.org/cookbook/cycling/sugarcube/sugarcube_cycling.html) …….(cycling choices)

[https://twinery.org/cookbook/dateandtime/sugarcube/sugarcube\_**dateandtime**.html](https://twinery.org/cookbook/dateandtime/sugarcube/sugarcube_dateandtime.html)

[https://twinery.org/cookbook/delayedtext/sugarcube/sugarcube\_**delayedtext**.html](https://twinery.org/cookbook/delayedtext/sugarcube/sugarcube_delayedtext.html)

[https://twinery.org/cookbook/deletingvariables/sugarcube/sugarcube\_**deletingvariables**.html](https://twinery.org/cookbook/deletingvariables/sugarcube/sugarcube_deletingvariables.html)

[https://twinery.org/cookbook/dicerolling/sugarcube/sugarcube\_**dicerolling**.html](https://twinery.org/cookbook/dicerolling/sugarcube/sugarcube_dicerolling.html)

[https://twinery.org/cookbook/fairmath/sugarcube/sugarcube\_**fairmath**.html](https://twinery.org/cookbook/fairmath/sugarcube/sugarcube_fairmath.html)

[https://twinery.org/cookbook/geolocation/sugarcube/sugarcube\_**geolocation**.html](https://twinery.org/cookbook/geolocation/sugarcube/sugarcube_geolocation.html)

[https://twinery.org/cookbook/googlefonts/sugarcube/sugarcube\_**googlefonts**.html](https://twinery.org/cookbook/googlefonts/sugarcube/sugarcube_googlefonts.html)

[https://twinery.org/cookbook/headersandfooters/sugarcube/sugarcube\_**headersandfooters**.html](https://twinery.org/cookbook/headersandfooters/sugarcube/sugarcube_headersandfooters.html)

[https://twinery.org/cookbook/hiddenlink/sugarcube/sugarcube\_**hiddenlink**.html](https://twinery.org/cookbook/hiddenlink/sugarcube/sugarcube_hiddenlink.html)

[https://twinery.org/cookbook/images/sugarcube/sugarcube\_**images**.html](https://twinery.org/cookbook/images/sugarcube/sugarcube_images.html)

[https://twinery.org/cookbook/importexternaljs/sugarcube/sugarcube\_**importexternaljs**.html](https://twinery.org/cookbook/importexternaljs/sugarcube/sugarcube_importexternaljs.html)

[https://twinery.org/cookbook/keyboard/sugarcube/sugarcube\_**keyboard**.html](https://twinery.org/cookbook/keyboard/sugarcube/sugarcube_keyboard.html) ….(keyboard events)

[https://twinery.org/cookbook/sidebar\_left/sugarcube/sugarcube\_](https://twinery.org/cookbook/sidebar_left/sugarcube/sugarcube_sidebar_left.html)**[sidebar\_left](https://twinery.org/cookbook/sidebar_left/sugarcube/sugarcube_sidebar_left.html)**[.html](https://twinery.org/cookbook/sidebar_left/sugarcube/sugarcube_sidebar_left.html)

[https://www.motoslave.net/sugarcube/2/docs/#**special-names**](https://www.motoslave.net/sugarcube/2/docs/#special-names)

[https://twinery.org/cookbook/clamping\_numbers/sugarcube/sugarcube\_**clamping\_numbers**.html](https://twinery.org/cookbook/clamping_numbers/sugarcube/sugarcube_clamping_numbers.html)

[https://twinery.org/cookbook/loadscreen/sugarcube/sugarcube\_**loadscreen**.html](https://twinery.org/cookbook/loadscreen/sugarcube/sugarcube_loadscreen.html)

[https://twinery.org/cookbook/lockandkey\_variable/sugarcube/sugarcube\_**lockandkey\_variable**.html](https://twinery.org/cookbook/lockandkey_variable/sugarcube/sugarcube_lockandkey_variable.html)

[https://twinery.org/cookbook/looping/sugarcube/sugarcube\_**looping**.html](https://twinery.org/cookbook/looping/sugarcube/sugarcube_looping.html)

[https://twinery.org/cookbook/modal/sugarcube/sugarcube\_**modal**.html](https://twinery.org/cookbook/modal/sugarcube/sugarcube_modal.html) …….("Modal (Pop-up Window)")

[https://twinery.org/cookbook/modularity/sugarcube/sugarcube\_**modularity**.html](https://twinery.org/cookbook/modularity/sugarcube/sugarcube_modularity.html) ......Modularity, widgets, and <<include>> (like <<display>>)

[https://twinery.org/cookbook/dungeonmoving/sugarcube/sugarcube\_**dungeonmoving**.html](https://twinery.org/cookbook/dungeonmoving/sugarcube/sugarcube_dungeonmoving.html)

[https://twinery.org/cookbook/passageevents/sugarcube/sugarcube\_**passage\_events**.html](https://twinery.org/cookbook/passageevents/sugarcube/sugarcube_passage_events.html)

[https://twinery.org/cookbook/passagesinpassages/sugarcube/sugarcube\_**passagesinpassages**.html](https://twinery.org/cookbook/passagesinpassages/sugarcube/sugarcube_passagesinpassages.html).....(like <<display>>)

[http://www.motoslave.net/sugarcube/2/docs/#macros-macro-**include**](http://www.motoslave.net/sugarcube/2/docs/#macros-macro-include)

[https://twinery.org/cookbook/passagetransitions/sugarcube/sugarcube\_**passagetransitions**.html](https://twinery.org/cookbook/passagetransitions/sugarcube/sugarcube_passagetransitions.html)

[https://twinery.org/cookbook/passagevisits/sugarcube/sugarcube\_**passagevisits**.html](https://twinery.org/cookbook/passagevisits/sugarcube/sugarcube_passagevisits.html)

[https://twinery.org/cookbook/playerstatistics/sugarcube/sugarcube\_**player\_statistics**.html](https://twinery.org/cookbook/playerstatistics/sugarcube/sugarcube_player_statistics.html)

[https://twinery.org/cookbook/programmaticundo/sugarcube/sugarcube\_**programmaticundo**.html](https://twinery.org/cookbook/programmaticundo/sugarcube/sugarcube_programmaticundo.html)

[https://twinery.org/cookbook/passagetoelement/sugarcube/sugarcube\_**passagetoelement**.html](https://twinery.org/cookbook/passagetoelement/sugarcube/sugarcube_passagetoelement.html%20) .....(render passage to element)

[https://twinery.org/cookbook/savinggames/sugarcube/sugarcube\_**savinggames**.html](https://twinery.org/cookbook/savinggames/sugarcube/sugarcube_savinggames.html)

<https://twinery.org/cookbook/settingandshowing/sugarcube/sugarcube_settingandshowing.html> ….(setting and showing variables)

[https://twinery.org/cookbook/spaceexploration/sugarcube/sugarcube\_**space\_exploration**.html](https://twinery.org/cookbook/spaceexploration/sugarcube/sugarcube_space_exploration.html)

[https://twinery.org/cookbook/statichealthbars/sugarcube/sugarcube\_**statichealthbars**.html](https://twinery.org/cookbook/statichealthbars/sugarcube/sugarcube_statichealthbars.html)

[https://twinery.org/cookbook/storyandpassage/sugarcube/sugarcube\_**storyandpassage**.html](https://twinery.org/cookbook/storyandpassage/sugarcube/sugarcube_storyandpassage.html)

…(story and passage API for external date access)

[https://twinery.org/cookbook/markup/sugarcube/sugarcube\_**markup**.html](https://twinery.org/cookbook/markup/sugarcube/sugarcube_markup.html) ….(style markup)

[https://www.motoslave.net/sugarcube/2/docs/#**markup**](https://www.motoslave.net/sugarcube/2/docs/#markup)

[https://twinery.org/cookbook/template\_type/sugarcube/sugarcube\_**templates**.html](https://twinery.org/cookbook/template_type/sugarcube/sugarcube_templates.html)

[https://twinery.org/cookbook/timedprogressbars/sugarcube/sugarcube\_**timed\_progress\_bars**.html](https://twinery.org/cookbook/timedprogressbars/sugarcube/sugarcube_timed_progress_bars.html)

[https://twinery.org/cookbook/turncounter/sugarcube/sugarcube\_**turncounter**.html](https://twinery.org/cookbook/turncounter/sugarcube/sugarcube_turncounter.html)

[https://twinery.org/cookbook/typewriter/sugarcube/sugarcube\_**typewriter**.html](https://twinery.org/cookbook/typewriter/sugarcube/sugarcube_typewriter.html)

[https://twinery.org/cookbook/storystyling/sugarcube/sugarcube\_**storystyling**.html](https://twinery.org/cookbook/storystyling/sugarcube/sugarcube_storystyling.html) …(variable story styling)

[https://twinery.org/cookbook/usingaddons/sugarcube/sugarcube\_**usingaddons**.html](https://twinery.org/cookbook/usingaddons/sugarcube/sugarcube_usingaddons.html)

<https://www.motoslave.net/sugarcube/2/#downloads> …..(see Add-Ons)

# Merging Twine 2 files

Twine 2 doesn’t have the Twine 1 StoryIncludes functionality

**Work around:**

The Twee command line compiler, Tweego, can read Twine 2 **published** HTML files, so you can use Tweego to merge multiple Twine 2 HTML output files, post-publish.

[https://www.motoslave.net/**tweego**/](https://www.motoslave.net/tweego/)

[https://www.motoslave.net/tweego/**docs**/](https://www.motoslave.net/tweego/docs/) Documentation

[https://github.com/tmedwards/**tweego**](https://github.com/tmedwards/tweego) - source code repository

Multiple Twine 2 files can be merged using Grunt and Entwine running together

[https://github.com/klembot/**grunt-entwine**-quickstart](https://github.com/klembot/grunt-entwine-quickstart)

It might also be possible to do this with Extwee as well.

<https://www.npmjs.com/package/extwee>

<https://github.com/videlais/extwee>

There are various tools to make command line interface (CLI) compilers nicer to work with—e.g.,

Visual Studio Code + Twee 3 Language Tools (T3LT) extension.

# Developer and resource curator contact details

**Twine 2**

Chris Klimas, cpb8010, Leon Arnott, Thomas M Edwards, Valentin Rocher, Jordi Mallach, Ross Smith, Colin McMillan, Moritz Rebbert, Shitake molingyu.

**SugarCube**

Thomas Edwards <https://github.com/tmedwards> / Email: <tmedwards@motoslave.net> / <thomasmedwards@gmail.com> / Guide: <http://www.motoslave.net/sugarcube/2/>

**SugarCube Add-Ons**

??

**Twine Resources**

ChapelR Email: <twinelab.net@gmail.com>

[https://twinelab.net/](https://twinelab.net/twine-resources)**[twine-resources](https://twinelab.net/twine-resources)**

[https://github.com/ChapelR/](https://github.com/ChapelR/twine-resources)[twine-](https://github.com/ChapelR/twine-resources)**[resource](https://github.com/ChapelR/twine-resources)**[s](https://github.com/ChapelR/twine-resources) (the Twine Resources repository)

**RSTI Adviser**

Philip Sutton, Joey Sutton Preece, Daniel Sutton,

?? (Glenn Todd?, Carolyn Hicks?)

# Migrating Twine 1 stories/documents to Twine 2

[https://twinery.org/forum/discussion/8423/**transferring-my-twine-stories-to-another-version-of-twine**](https://twinery.org/forum/discussion/8423/transferring-my-twine-stories-to-another-version-of-twine) 2017 (Advice from Grey Elf)

The Twine 2 application does not currently support the importing of either a Twine 1 Story .tws file or a Story .html file created using Twine 1's build option, but there is a manual process that you can use to convert your existing story into something that the Twine 2 application will recognise.

The following lists the issues that need to be overcome and the steps to do so.

**1. Sugarcane is not supported on Twine 2.**

You can get around this issue by switching your existing Twine 1 project over to using SugarCube v1, which is similar to Sugarcane although there are differences that will **need to be resolved later**.

1a. Download SugarCube 1 [Adding Formats section](http://www.motoslave.net/sugarcube/1/) of the Twine 1 Guide explains how to install the contents of the ZIP file you downloaded in point 1a. <https://www.motoslave.net/sugarcube/1/>

1c. Make a copy of your Story .tws file then open that copy with Twine 1 and use the **Story** > **Story Format** menu items to change it to use SugarCube.

1d. Use the **Build** > **Build Story** menu items to create a new Story .html file, give this HTML file a **different** file name than you would normally use so you can distinguish it from any other story HTML file you have previously created.

**2. Converting Twine 1 Story HTML file to Twine 2 archive file.**

In the **Utilities** section of the web-site you downloaded SugarCube from there is a link for downloading a tool to do this conversion.

2a. Download the ZIP file and extract its contents onto your hard-disk. <http://www.motoslave.net/sugarcube/download.php/extras/t1-compiled-to-t2-archive.zip>

2b. Open the contained HTML file in your web-browser and follow the instructions.

**3. Importing generated Archive file.**

Run the Twine 2 application and use it's **Import From File** option to add the Story Project contained within the Archive file (you just created) to your list of Stories.

**4. Fixing the differences between Sugarcane and SugarCube**

Without knowing exactly what code you have written in your story I can't tell you exactly what may need to be fixed, I can only suggest you test your new story project and use these forums to resolve any issues you find.

**5. Upgrading to** [**SugarCube**](http://www.motoslave.net/sugarcube/2/)

This is an optional but recommended step because as explained on the SugarCube 1 web-site the 1.x series has reached the end of it's development cycle and is now only getting bug/security fixes if they are needed.

# Useful things in Twine 1 that are NOT available in Twine 2/SugaCube

**Sending people links to specific passages in a Twine .html file  
The hashbang method**

Eg.

https://climate-rescue-global.org/#!Output\_Report

HiEv has created a custom replacement:

[https://qjzhvmqlzvoo5lqnrvuhmg-on.drv.tw/UInv/Sample\_Code.html#**Linking%20to%20Passages%20by%20URL**](https://qjzhvmqlzvoo5lqnrvuhmg-on.drv.tw/UInv/Sample_Code.html#Linking%20to%20Passages%20by%20URL)

**Special passage:** **StoryIncludes**

The StoryIncludes was used to store url links to additional Twine or Twee files that would be incorporated into the compiled html output. A work around is available for Twine 2 (using Tweego). (See the section “**Merging Twine 2 files**” that is just a few major sections above.

**Special tags: Twine.private**

Passages with this tag would not be incorporated into the Twine html output file. Passages with the Twine.private tag can be stripped using Tweego.

# Related tools

<https://www.christytuckerlearning.com/tools-for-building-branching-scenarios/>

<https://articulate.com/360/storyline>

<https://miro.com>

[https://community.articulate.com/discussions/**building-better-courses/writing-and-designing-branching-scenarios**](https://community.articulate.com/discussions/building-better-courses/writing-and-designing-branching-scenarios)

# (the rest in the old Twine 1 help file??)

The Twine 1 help file might be a useful guide to content types to build up in this help file (adapted of course to suit the latest Twine 2 with the latest SugarCube format.

# Joey’s backup method