

<p>Snowman Cheatsheet 1.3 Created by Dan Cox (@videlais)</p> <p>Based on bitbucket.org/klembot/snowman-2</p> <p>Functionality from: jQuery Underscore</p>

Underscore.js Template Usage

<% Arbitrary Code block %>

Interpolate (Write-to-Page) Value
--

<%= Value.to.write %>

HTML-escaped

<%- Value.HTML %>

Special Variables from Snowman within Underscore code blocks

s = window.story.state

\$ = jQuery selector or, if function, is run when the passage appears onscreen, with it bound to the passage DOM element
--

Element Creation
[Create element][#elementID.className]

Hiding Elements (using minus sign)
[This message is hidden!][-.classname]

window.story Properties		
Name	Type	Description
name	String	The name of the story.
startPassage	Number	The ID of the first passage to be displayed.
creator	String	The program that created this story.
creatorVersion	String	The version of the program used to create this story
history	Array	An array of passage IDs, one for each passage viewed during the current session.
state	Object	An object that stores data that persists across a single user session. Any other variables will not survive the user pressing back or forward.
checkpointName	String	The name of the last checkpoint set. If none has been set, this is an empty string.
ignoreErrors	Boolean	If set to true, then any JavaScript errors are ignored -- normally, play would end with a message shown to the user
errorMessage	String	The message shown to users when there is an error and ignoreErrors is not true. Any %s in the message will be interpolated as the actual error message.
passages	Array	An array of all passages, indexed by ID.
userScripts	Array	An array of user-specific scripts to run when the story is begun
userStyles	Array	An array of user-specific style declarations to add when the story is begun.
atCheckpoint	Boolean	If currently at a checkpoint or not

window.passage Properties		
Name	Type	Description
id	Number	The numeric ID of the passage
name	String	The name of the passage
tags	Array	The tags of the passage
source	String	The passage source code

window.story Functions			
Name	Parameters	Return Type	Description
checkpoint()	{String} Checkpoint name, appears in history, optional	(None.)	Tries to add an entry in the browser history for the current story state. Remember, only variables set on this story's state variable are stored in the browser history.
passage()	{String or Number} ID or name of the passage	Object or Null	Returns the Passage object corresponding to either an ID or name. If none exists, then it returns null
render()	{String or Number} ID or name of the passage	{String} HTML source code	Returns the HTML source for a passage. This is most often used when embedding one passage inside another. In this instance, make sure to use <%= %> instead of <%- %> to avoid incorrectly encoding HTML entities
restore()	{String} Hash	{Boolean} Whether the restore succeeded	Tries to restore the story state from a hash value generated by saveHash().
save()	(None.)	{String} Hash	Sets the URL's hash property to the hash value created by saveHash()
saveHash()	(None.)	{String} Hash	Returns a hash value representing the current state of the story.
show()	{String or Number} ID or name of the passage; {Boolean} if true, then this will not be recorded in the story history	(None.)	Displays a passage on the page, replacing the current one. If there is no passage by the name or ID passed, an exception is raised.
start()	(None.)	(None.)	Begins playing this story.

window.passage Functions			
Name	Parameters	Return Type	Description
render()	{String} Source	{String} HTML source	Returns an HTML-rendered version of this passage's source. This first runs the source code through the Underscore template parser, then runs the result through a Markdown renderer, and then finally converts bracketed links to passage links