

Snowman Cheatsheet 1.0

Created by Dan Cox (@videlais)

Based on bitbucket.org/klembot/snowman-2

General Macro and Script Usage

```
<% blocks %>
```

Comments

```
/* inline comments */
```

```
// line comments
```

<div> Elements

```
[  
  This appears onscreen.  
]{#id.className}
```

Empty <div> Elements

```
[  
{.empty}
```

Initially Hiding Elements (using minus sign)

```
[  
  This message is hidden!  
{-.surprise}
```

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.

window.story Functions

Name	Paramaters	Return Type	Description
s.start()	None	None	Begins playing this story.
s.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.
s.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.
s.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.
s.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.
s.saveHash()	None	{String} Hash	Returns a hash value representing the current state of the story.
s.save()	None	{String} Hash	Sets the URL's hash property to the hash value created by saveHash().
s.restore()	{String} Hash	{Boolean} Whether the restore succeeded	Tries to restore the story state from a hash value generated by saveHash().

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.passage Functions			
Name	Paramaters	Return Type	Description
<i>render()</i>	{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.
<i>renderEl()</i>	{String} nodeName element's node name, e.g. 'div' or 'span'; {String} source inner source code of the element; {String} selector a string selector	{String} HTML source code	A helper function that converts markup like [this]{#id.class} into HTML source for a DOM element.
<i>readyFunc()</i>	None	{Object} jQuery object, as with jQuery()	A helper function that is connected to passage templates as \$. It acts like the jQuery \$ function, running a script when the passage is ready in the DOM. The function passed is also bound to div#passage for convenience. If this is *not* passed a single function, then this acts as a passthrough to jQuery's native \$ function.