

SugarCube v2 文档翻译

Alrcatraz/Nanaly

2024 年 3 月 27 日

目录

1	SugarCube 基本知识	4
1.1	简介	4
1.2	标记	4
1.3	TwineScript	8
1.4	宏	9
1.5	函数	10
1.6	方法	11
1.7	特殊名称	12
1.8	CSS 样式表	13
1.9	HTML	14
1.10	事件	15
2	API	16
2.1	配置 API	16
2.2	对话 API	17
2.3	引擎 API	18
2.4	全屏 API	19
2.5	加载屏幕 API	20
2.6	宏 API	21
2.6.1	宏文本 API	21
2.7	通道 API	22
2.8	保存 API	23
2.9	设置 API	24
2.10	简单音频 API	25
2.10.1	音频 API	25
2.10.2	音频播放器 API	25
2.10.3	音频列表 API	25
2.11	状态 API	26
2.12	故事 API	27
2.13	模板 API	28
2.14	用户界面 API	29
2.15	UIBar API	30

目录	3
----	---

3 指南	31
3.1 指南：状态、会话和保存	31
3.2 指南：小贴士	32
3.3 指南：媒体段落	33
3.4 指南：从 Harlowe 到 SugarCube	34
3.5 指南：测试模式	35
3.6 指南：TypeScript	36
3.7 指南：安装	37
3.8 指南：代码更新	38
3.9 指南：本地化	39

1 SugarCube 基本知识

1.1 简介

本文档为 [SugarCube](#) 的参考，SugarCube 是 [Twine/Twee](#) 的免费故事格式。

提示： 本文档是单个网页，因此你可以使用浏览器的页面查找功能（Ctrl+F 或 F3）来搜索特定术语。**注意：** 如果您认为您在 SugarCube 中发现了错误，或者只是想提出建议，您可以在 [源代码存储库](#) 中 [创建一个新问题](#)。

本文档的贡献者包括：

- **Chaper** ([TwineLab](#), [GitHub](#))

1.2 标记

注意： 除非另有说明，否则所有标记都可用。v2.0.0

裸变量 除了使用其中一个打印宏（`<<print>>`，`<<==>>`，`<<=>>`）来打印 TwineScript 变量的值以外，SugarCube 的裸变量标记还允许简单地将它们包含在普通段落文本中来打印它们——即段落文本中的变量值会转化为字符串的形式。

裸变量标记支持以下形式：

类型	语法	样例
简单变量	<code>\$variable</code>	<code>\$name</code>
属性访问 (点表示法)	<code>\$variable.property</code>	<code>\$thing.name</code>
索引/属性访问 (方括号表示法)	<code>\$variable[numericIndex]</code> <code>\$variable["property"]</code> <code>\$variable['property']</code> <code>\$variable[\$indexOrPropertyVariable]</code>	<code>\$thing[0]</code> <code>\$thing["name"]</code> <code>\$thing['name']</code> <code>\$thing[\$member]</code>

如果需要打印任何更复杂的内容（如使用计算或方法调用），则仍需要使用其中一个打印宏。

例如：

```
1      /* 通过 <<print>> 宏显式打印 $name 的值 */
2      Well hello there, <<print $name>>.
3
4      /* 通过裸变量标记隐式打印 $name 的值 */
5      Well hello there, $name.
6
7      /* 假设 $name 设置为 “Mr. Freeman”，两者都应产
8      生以下结果 */
9      Well hello there, Mr. Freeman.
```

因为段落文本中的变量会自动转换为它们的值，如果你真的想按原样输出变量——即不插值。对于教程、调试输出或其他内容，您需要以某种方式对其进行转义。例如：

```
1      /* 使用 nowiki 标记： """...""" （三个双引号）
2      */
3      The variable """$name""" is set to: $name
4
5      /* 使用 nowiki 标记： <nowiki>...</nowiki> */
6      The variable <nowiki>$name</nowiki> is set to:
7      $name
8
9      /* 使用双美元符号标记（转义 $ 符号）： $$ */
10     The variable $$name is set to: $name
11
12     /* 假设 $name 设置为 “Mr. Freeman”，以上所有内
13     容都应产生以下结果 */
14     The variable $name is set to: Mr. Freeman
```

此外，您可以使用内联代码标记来转义变量，尽管它也会将转义变量包装在元素中，因此它可能最适合用于示例和教程。例如：

```
1      /* 使用内联代码标记： {{{...}}} （三重花括号） */
2      The variable {{{$name}}} is set to: $name
3
```

```

4      /* 假设 $name 设置为 “Mr. Freeman”，它应该产生
        以下结果 */
5      The variable <code>$name</code> is set to: Mr.
        Freeman

```

链接 SugarCube 的链接标记由必需组件和可选组件组成。组件可以是纯文本，也可以是任何有效的 TwineScript 表达式，这些表达式将在早期（即最初处理链接时）进行处理。该组件仅适用于段落链接，必须是 «set» 宏变体的有效 TwineScript 表达式，该表达式将在后期（即单击链接时）进行计算。

组件值可以是段落的标题，也可以是资源（本地或远程）的任何有效 URL。

除了用于分离和组件的标准管道分离器 (|) 外，SugarCube 还支持箭头分离器 (&)。特别是对于箭头分隔符，箭头的方向决定了元件的顺序，箭头始终指向元件——即，右箭头的工作方式类似于管道分隔符，而左箭头则相反。

警告 (Twine2)：由于 Twine 2 自动段落创建功能目前的工作方式，对组件使用任何 TwineScript 表达式都会导致创建以表达式命名的段落，该段落需要删除。为了避免此问题，建议您在需要使用表达式时使用 Twine 2 中 «set» 宏的单独参数形式。

以下是一个例子：

语法	样例
[[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”]]”

图像 SugarCube 的图像标记由必需的组件和可选的、和组件组成。、和组件可以是纯文本，也可以是任何有效的 TwineScript 表达式，这些表达式将尽早计算，即在最初处理链接时进行计算。该组件仅适用于段落链接，必须是 **«set»** 宏变体的有效 TwineScript 表达式，该表达式将在后期（即单击链接时）进行计算。

组件值可以是图像资源（本地或远程）的任何有效 URL，也可以是 **嵌入图像段落的标题（仅限 Twine 1 和 Tweego）**。组件值可以是段落的标题，也可以是资源（本地或远程）的任何有效 URL。

除了用于分离和组件的标准管道分离器（**|**）外，SugarCube 还支持箭头分离器（**&**）。特别是对于箭头分隔符，箭头的方向决定了元件的顺序，箭头始终指向元件，即，右箭头的工作方式类似于管道分隔符，而左箭头则相反。

警告 (twine2): 由于 Twine 2 自动段落创建功能目前的工作方式，对组件使用任何 TwineScript 表达式都会导致创建以表达式命名的段落，该段落需要删除。为了避免此问题，建议您在需要使用表达式时使用 Twine 2 中 **«link»** 宏的单独参数形式。

1.3 TwineScript

1.4 宏

1.5 函数

1.6 方法

1.7 特殊名称

1.8 CSS 样式表

1.9 HTML

1.10 事件

2 API

2.1 配置 API

2.2 对话 API

2.3 引擎 API

2.4 全屏 API

2.5 加载屏幕 API

2.6 宏 API

2.6.1 宏文本 API

2.7 通道 API

2.8 保存 API

2.9 设置 API

2.10 简单音频 API

2.10.1 音频 API

2.10.2 音频播放器 API

2.10.3 音频列表 API

2.11 状态 API

2.12 故事 API

2.13 模板 API

2.14 用户界面 API

2.15 **UIBar API**

3 指南

3.1 指南：状态、会话和保存

3.2 指南：小贴士

3.3 指南：媒体段落

3.4 指南：从 Harlowe 到 SugarCube

3.5 指南：测试模式

3.6 指南: TypeScript

3.7 指南：安装

3.8 指南：代码更新

3.9 指南：本地化

参考文献

Edwards, T. M. (2021, December 22). *Sugarcube v2 documentation* [Sugarcube v2 documentation]. Retrieved March 27, 2024, from <https://www.motoslave.net/sugarcube/2/docs/>