Houdini 表达式基础

语言

Houdini可以让你使用两种语言写表达式：传统 HScript 表达式和 Python。VEX 是一种编译语言，可以让你编写

HScript 表达式	使用 expression functions（表达式函数）编写表达式的传统方法
Python	更强大但更繁琐的使用 Houdini Object Model API 和任何 Python 本身
VEX	一种快速编译的语言，只用在特定节点的特定参数上

默认使用 HScript 表达式语言。如果要使用 Python，请参照 Python expressions.

字符串参数

- 在数字参数（例如 位置，缩放，旋转）中，参数中填写的文本作为表达式计算。
- 在字符串参数（例如 文件名，由 Python 创建的文本），参数中填写的文本被视为文字。Variables are expanded，但是若要通过表达式生成文字内容，需要把表达式放

```
frame`padzero(5, $F)`.pic
```

...可以 `frame00001.pic`，`frame00002.pic`，这样的文件名，诸如此类。

（浏览 Expressions in filenames 了解在文件名中使用表达式和变量）。

变量和属性

你的表达式可以使用 Houdini 提供的变量：

- 全[$F (the current frame number) [$T (the current time in seconds). 详情见t of global variables （全局变量表格）。
- 几何体属性的 HSc @[引 @pscale 用 pscale (点缩放) 属性。
- [P](position) 点 @P. x 表[. x/. y/. z][. 1/. 2/. 3 [. r/. g/. b 。
- 本地变量，节点通常会提[@ptnum]的变量输出详情请参阅节点帮助文档的“局部变量”部分。节点帮助文档会列出所有可用变量（点）的节点会提供
- 环[$HIP （场景文件所在的目录）也可以被表达式使用。

通道/参数引用

在 HScript 中你可以用[引用节点中的参数的值。这可以用来让一个节点中的数字随着另外一个节点发生改变。

> 你[创建自己[参数并引用它[详[rs.（备用参数）

要...	这样做
自动创建参数引用	1. 源参[上右[选[parameter.（复制参数） 2. 目[Pa 数[t 键选择elative reference.（粘贴对应参数） Houdini [ch("") 语法让后一个参数引用前一个参数的值。

手动创建通道引用	用 HScrpt 表达式[得[用一个参数的值。
	ch 会从当前节点的（可能的相对）路径到参数。比如，要得到当前节点的 Z 变换属
	`ch("tz")`
	[lamp 物体的 X 变换属性:
	`ch("/obj/lamp/tx")`
	[grid1 络[点的 Y 旋转属性:
	`ch("../grid1/ry")`
	(要获得字符串的值，使用

查询参数的内部名称	要手动设[参数[的[me 查[内部名称，你可以将鼠标悬停在参数框上方 或 通过参数[

提示

在几何节点中使用 HScript 表达式和本地变量或参数来修改几何体很容易。但是，在 VEX snippet 中修改几何体的性能更高。

未来的版本中会更鼓励使用 VEX 短代码修改几何体。

你可以在 **Window ▸ HScript Textport** 输入下面的命令查看一个表达式的值：

```
echo `expression`
```

Houdini 总是用网络（Network）的“配方”“从头开始”创建每一帧。表达式总是使用初始值而不是前一帧的值来操作数值。

例如，如果在一个 Geometry node（点几何节点）给网格中的每个点添加随机的 Y 偏移值：

Position Y

```
@P.y + rand(@Frame * @ptnum)
```

...当你回放时，点将上下随机跳跃，而不是平滑运动。这是因为每一帧时，网格都会从平整的网格开始，然后再应用随机移动给它。这个表达式不会从前一帧的几何体状态开始运行。

HScript表达式的括号用法

字符串

在 `' '` 中的字符is not expanded. Text inside double quotes `" "` has variables expanded. A double-quoted string is considered one argument.

A backslash character (`\`) escapes the next character. For example, to use double-quotes in a string:

```
"I had a \"great\" time."
```

When a string doesn’t require variable expansion, use single quotes to speed up evaluation.

If you have two quoted strings next to each other with no spaces, they are considered a single argument. In this example...

```
set foo = "Hello world"
echo '$foo='"$foo"
$foo=Hello world
```

...the echo command has one argument: `'$foo=Hello world'`.

Embedding

In the HScript command language, text inside backticks is evaluated as an expression. For example:

```
echo `strlen("$foo")`
```

Tip

Scripting using the HScript command language is deprecated. You should use Python instead.

The string parser cannot decode nested quotes such as in the following (horribly contrived) example:

```
echo `system("echo `ls`")`
```

...however, it is possible to accomplish this with very careful usage of backquotes (and sometimes multiple backquotes in a row) to protect quote symbols from various levels of evaluation:

```
echo `system('echo \`ls\`')`
```

Revision #5
Created 29 August 2018 07:48:50 by □□□□□□□□
Updated 9 October 2018 05:56:42 by □□□□□□□□