







# Houdini







Houdini  [VEX snippets](#)  Python  VEX 

HScript 	<a href="#">expression functions</a> 	
Python	<a href="#">Houdini Object Model</a>  	API  Python 
VEX		

 HScript  [Python Parameter](#)  [expression](#) .





-  
-  [Font node](#)  Variables are expanded 

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

... `frame00001.pic`, `frame00002.pic`, 

 [expressions in filenames](#) 



 Houdini 

- `FF` (\$F) (the current frame number) `TT` (\$T) (the current time in seconds). [list of global variables](#) `XXXXXXXXXX`
- `XXXXXXXXXX` HSc `@P` @pscale `XXXX` pscale (`XXXX`) `XXXX`
- `P` (position) `@P` .x/ .y/. z `XXXX` .1/. 2/. 3 `XXXX` .r/. g/. b `XXXX`
- `XXXXXXXXXX` @ptnum `XX` Point `XXXXXXXXXX`
- `FF` \$HIP `XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX`

-  HSc @xtt @pscale III pscale ( )

- [P](position) [ @P. x [I. x/ y/ z [I. 1/ 2/ 3 [ . r/ g/ b [

- 

- 

**□ □ / □ □ □ □**

HScript

ch

```
spare parameters.  
```

⋮	
	1. Copy parameter. 2. Paste relative reference. Houdini ch( "" )
	HScript ch ch Z ch("tz")  lamp X ch("/obj/lamp/tx")  grid1 Y ch("../grid1/ry")  (chs.)
	Name



Script Wrangle Attribute Wrangle VEX snippet

VEX

Window ▶ HScript Textport

```
echo `expression`
```

Houdini Network

Print geometry node Y

Position Y @P.y + rand( @Frame \* @ptnum)

...

# HScript



' is not expanded. Text inside double quote " 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 [REDACTED]

Updated 9 October 2018 05:56:42 by [REDACTED]