

小书匠语法说明之 codeChunk

小书匠

目录

code chunks 语法	1
语法格式	1
参数	1
注意	4

小册子

code chunks 语法

元数据: `grammar_codeChunk`

该语法特点主要受 [markdown](#) 和 [markdown-preview-enhanced](#) 的启发, 代码实现上主要参考了 [markdown-preview-enhanced](#)。主要用途就是实现直接执行代码块内的代码。目前该功能只能通过客户端实现。代码的执行受到操作系统, 软件版本等多种因素的限制。

想使用该语法, 需要要 **设置> 语法扩展** 里打开 `codeChunk` 语法, 也可以在每篇文章的元数据头里通过添加标识 `grammar_codeChunk` 来打开该语法

语法格式

```
```{lang [参数1:参数值1,参数2:参数值2...]}
...
```
```

代码段.png

除了 `lang` 参数用来指明代码块内的语言, 系统还提供了几个内置参数 `cmd`, `line_number/linenums`, `fancy`, `title`, `hide`, `auto`, `first_line`, `output`, `args`, `stdin`

参数

1. lang

代码块内的语言, 用于代码高亮. 如果 `cmd` 参数没有指定, 则默认使用这个 `lang` 做为 `cmd` 来执行

2. cmd

代码块被执行时使用的命令, 不同的操作系统会不同
比如我想列出当前用户主目录下的文件, 可以这样执行

```
```{bash·cmd:'bash'}~
ls ~/.~
```
```

1483600161379.jpg

3. line_number/linenums

用于代码高亮时是否显示行号, 如果是具体数值, 表示从指定行号开始

4. first_line

代码高亮时指定行号起始值

5. fancy

指定具体哪几行可以被高亮显示, 多行用逗号分开

```
```{javascript cmd:'node',fancy:'1,3,4', linenums:true}
var x = 1;
var y = 2;
var z = 3;
var u = 4;
var v = 5;
console.log(x + y + z + u + v)
```
```

javascript 代码片段

显示效果



代码块执行显示效果1

6. hide

是否隐藏代码高亮。如果是隐藏代码高亮，代码块将会被直接执行。

注：出于安全考虑，如果用户没有在全球打开非安全模式，该参数将失效。

7. auto

是否自动执行代码块。

注：出于安全考虑，如果用户没有在全球打开非安全模式，该参数将失效。

8.title

代码高亮时显示的标题，隐藏代码块时，该参数失效。

8. output

执行完代码块后，返回给小书匠系统的输出格式。目前支持 `html`，`png`，`none`，及不填

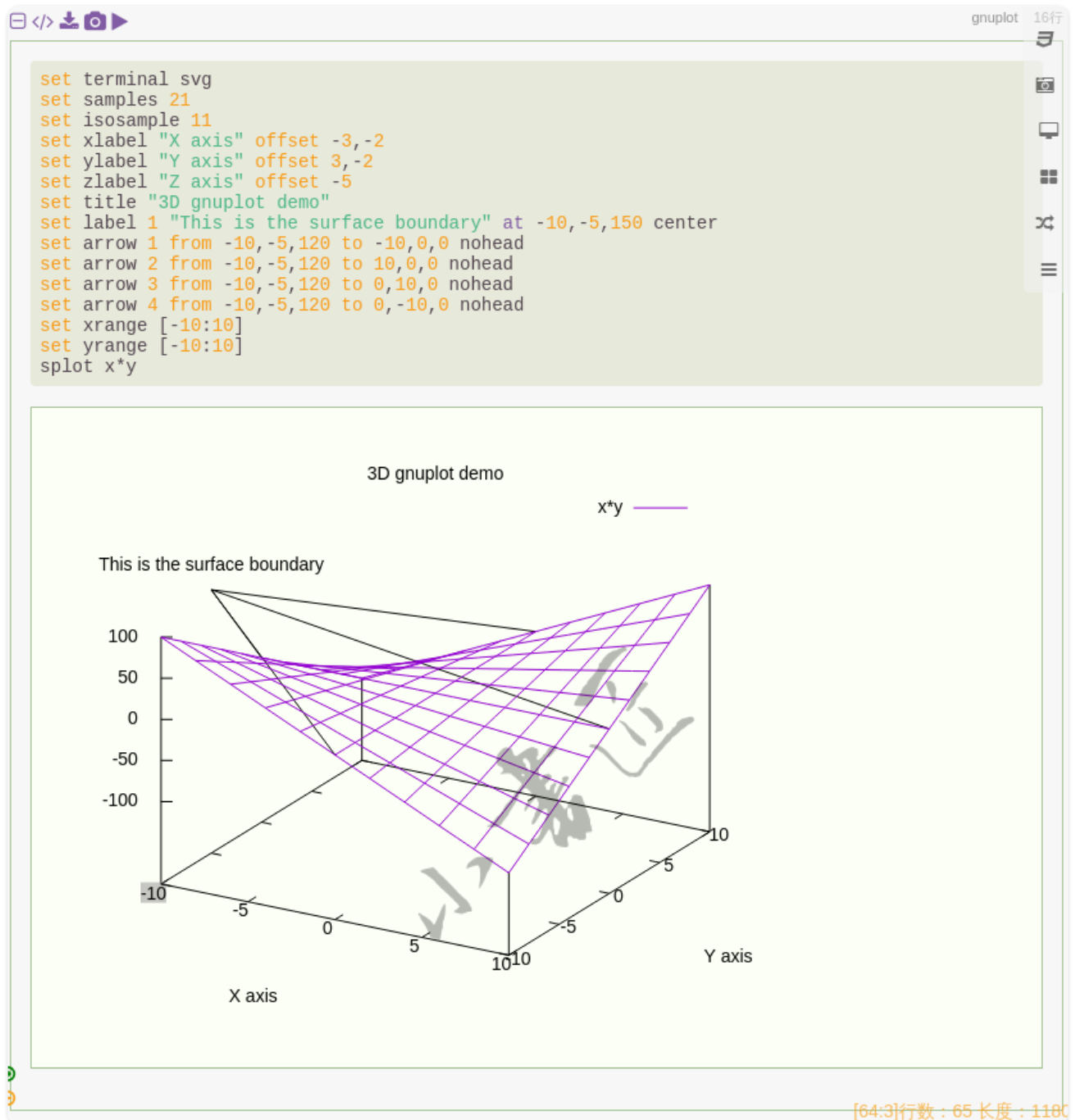
- `html` 返回 `html` 代码片段
- `png` 返回 `base64` 格式的图片
- `none` 无返回，隐藏输出结果
- `text` 返回命令行的输出结果
- 不填 返回命令行的输出结果

测试代码

```
```{gnuplot output:"html"}
set terminal svg
set samples 21
set isosample 11
set xlabel "X axis" offset -3,-2
set ylabel "Y axis" offset 3,-2
set zlabel "Z axis" offset -5
set title "3D gnuplot demo"
set label 1 "This is the surface boundary" at -10,-5,150 center
set arrow 1 from -10,-5,120 to -10,0,0 nohead
set arrow 2 from -10,-5,120 to 10,0,0 nohead
set arrow 3 from -10,-5,120 to 0,10,0 nohead
set arrow 4 from -10,-5,120 to 0,-10,0 nohead
set xrange [-10:10]
set yrange [-10:10]
splot x*y
```
```

gnuplot.png

输出效果



输出 html 片段效果

10. args

args 参数可以追加自定义的参数

测试代码


```

python args: ["-v"]
print("Verbose will be printed first")

```

python.png

输出效果



```
print('Verbose will be printed first')
```

```
# installing zipimport hook import zipimport # builtin # installed zipimport hook #  
/usr/lib64/python2.7/site.pyc matches /usr/lib64/python2.7/site.py import site # precompiled from  
/usr/lib64/python2.7/site.pyc # /usr/lib64/python2.7/os.pyc matches /usr/lib64/python2.7/os.py import os #  
precompiled from /usr/lib64/python2.7/os.pyc import errno # builtin import posix # builtin #  
/usr/lib64/python2.7/posixpath.pyc matches /usr/lib64/python2.7/posixpath.py import posixpath #  
precompiled from /usr/lib64/python2.7/posixpath.pyc # /usr/lib64/python2.7/stat.pyc matches  
/usr/lib64/python2.7/stat.py import stat # precompiled from /usr/lib64/python2.7/stat.pyc #  
/usr/lib64/python2.7/genericpath.pyc matches /usr/lib64/python2.7/genericpath.py import genericpath #
```

自定义 args 参数效果

11. stdin

true 或者 false, 为true时, 表示将代码块里的内容做为命令行输入, 否则通过临时文件输入

注意

1. **codeChunk** 语法会执行代码段里面的代码, 存在被恶意代码段的风险, 不要在不了解的情况下轻意指行来历不明的代码块
2. 系统默认处在安全模式下, 想打开非安全模式, 可以到 关于 界面里设置为非安全。考虑到安全原因, 用户下次再打开小书匠时, 该设置将会被重置。
3. 在导出时, 系统只会导出用户手动执行过的代码段。想要让系统导出未执行过的代码段, 需要用户打开非安全模式。