

# manim 常见问题

鹤翔万里 & catfish

MANIM-KINDERGARTEN

版本：3.1

更新：2021 年 2 月 1 日



# 一切之前

manim 是一个使用 python 制作视频的动画引擎。学习它你首先要会一点 python，至少要学会 python 的基础语法、模块的调用以及类的基础知识（如果想要阅读源码，还需要掌握更多 python 面向对象的知识）。没有 python 的知识学 manim 是毫无意义的，会非常吃力，也会遇到非常多的问题。学会了 python 之后便可以少走非常多的弯路，这也是我们的忠告。因此，加入我们 MANIM-KINDERGARTEN<sup>1</sup>首先要会使用 python，我们会在入群问题<sup>2</sup>中进行一个最基础的检测。

最后，本常见问题文档基于 manim 的 cairo-backend 分支版本<sup>3</sup>，不包含新版 manim 中出现的新问题。

## 如何提问：

- I. 在 mk 群里提问，首先需要阅读完本常见问题文档。  
如果群友对你说“常见问题”或“RTFM”，那么就说明你问的问题在本文档里已经有明确的解答。
- II. 确保你问的问题不是由于 python 基础语法问题而造成的错误。  
如，不要问 IndentationError，尽量不要问由于拼写导致的错误等。当群友劝你学习 python 或者说“STFW”时，那么就说明你的问题属于没有掌握 python 基础语法而导致的问题，而且你可以在网络上轻松搜索到解决方案。（但这时更建议你认真打牢 python 基础）
- III. 如果在 manim 使用过程中出现了报错：
  1. 确保问题不在本文档中
  2. 将你的代码和完整的报错信息全部发送到群中，这样会方便群友为你解答
- IV. 如果你想要实现某个效果，但不知道如何操作。请将你想要达到的效果详细易懂地描述出来。
- V. 语气友善，态度谦虚，避免造成不必要的纠纷。

---

<sup>1</sup>QQ 群：862671480

<sup>2</sup><https://b23.tv/KmAvsG>

## 教程推荐：

### I. Python 教程：

1. 《Python 编程：从入门到实践》，ISBN 978-7-115-42802-8
2. 《流畅的 Python》，ISBN 978-7-115-45415-7（适合掌握 Python 基础知识后进阶）
3. 菜鸟教程-Python3 教程 <https://www.runoob.com/python3/python3-tutorial.html>

### II. manim 教程：

1. manim 教程文档（制作中）：<https://manim.ml/>
2. MK 制作的系列视频教程（制作中）
  - <https://space.bilibili.com/171431343/favlist?fid=947158443>
3. MK 制作的视频源码 (videos/) 和常用自定义类 (utils/)
4. 群主 cigar666 的 B 站专栏
  - <https://www.bilibili.com/read/readlist/rl82339>
5. pdcxs 大大转载的 manim 教程
  - <https://www.bilibili.com/video/av64023740>
  - 源码 <https://github.com/Elteoremadebeethoven/AnimationsWithManim>
6. GitHub 上 cai-hust 的中文教程
  - <https://github.com/cai-hust/manim-tutorial-CN>
7. 看 manim 源码

### III. 新版 manim 教程：

1. 新版 manim 官方文档：<https://3b1b.github.io/manim/>
2. 新版 manim 中文文档：<https://manim.ml/shaders/>

### IV. OpenGL 及 shaders 教程（Grant 亲自推荐）

1. The Book of Shaders: <https://thebookofshaders.com/>
2. Python & OpenGL for Scientific Visualization: <https://www.labri.fr/perso/nrougier/python-opengl/>
3. Geometry Shader: <https://learnopengl.com/Advanced-OpenGL/Geometry-Shader>

# 目录

<b>1</b>	<b>安装问题</b>	<b>6</b>
1.1	Python 问题 . . . . .	6
<b>2</b>	<b>运行时问题</b>	<b>8</b>
2.1	import 问题 . . . . .	8
2.2	L <sup>A</sup> T <sub>E</sub> X 问题 . . . . .	8
2.3	dvisvgm 问题 . . . . .	9
2.4	中文显示问题 . . . . .	11
2.5	文字问题 . . . . .	11
2.6	素材引用问题 . . . . .	14
<b>3</b>	<b>其它问题</b>	<b>15</b>
Q1:	没有 manim 源码 . . . . .	15
Q2:	群友用的 manim 都是什么版本 . . . . .	15
Q3:	如何使用傅里叶级数作图 . . . . .	15
Q4:	傅里叶级数作图如何调整时长 . . . . .	15
Q5:	svg 用什么软件制作 . . . . .	16
Q6:	动画怎么显示旋转一个物体 . . . . .	16
Q7:	Transform 和 ReplacementTransform 有什么区别 . . . . .	16
Q8:	怎么控制物体移动或者 Transform 的速率 . . . . .	16
Q9:	数学符号/公式用 L <sup>A</sup> T <sub>E</sub> X 怎么打 . . . . .	17
Q10:	一些特殊 L <sup>A</sup> T <sub>E</sub> X 的外部包 . . . . .	17

Q11: 使用 $\text{\LaTeX}$ 外部包, 编译错误或者无显示 . . . . .	18
Q12: 一些比较复杂, 操纵东西比较多的动画怎么做 . . . . .	19
Q13: 一个 <code>self.play</code> 里写两个 <code>ApplyMethod</code> 只对一个起作用 怎么办 . . . . .	19
Q14: 如何解决二维画面中的图层问题 . . . . .	20
Q15: 如何导出 gif 文件 . . . . .	21
Q16: 如何导出透明的图片或者视频 . . . . .	21
Q17: 渲染视频的画质和帧率怎么调整 . . . . .	21
Q18: 有没有什么好的场景例子供学习 . . . . .	22
Q19: 新版本 <code>manim</code> 是什么, 和旧版有什么区别 . . . . .	22
<b>4 注意</b>	<b>23</b>
<b>5 附: ChangeLog</b>	<b>24</b>

# 1 安装问题

安装时最好不要看 README.md 自己研究，推荐一视数学卷毛杨的两个教程，和教程文档中的安装指南<https://manim.ml/installation>:

- <https://www.bilibili.com/video/av38126904>
- <https://www.bilibili.com/read/cv4139851>

## 1.1 Python 问题

**Q1:** 使用 anaconda, 命令行输入 python 无反应或报错

考虑 path 环境变量是否填全<sup>4</sup>, path 变量里应该有:

```
<your_path>\Anaconda3;  
<your_path>\Anaconda3\Scripts;  
<your_path>\Anaconda3\Library\bin;
```

**Q2:** pip install ... 时满屏红字报错, 或者安装过慢

更换国内镜像源, 使用

```
pip install -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
```

代替<sup>5</sup>

```
pip install -r requirements.txt
```

**Q3:** pip 安装 pycairo 总是失败

下载 pycairo 对应版本的 whl 包<sup>6</sup> 并手动安装

---

<sup>4</sup>安装 anaconda 时是否勾选添加到 path 变量

<sup>5</sup>临时换源

<sup>6</sup>可在<https://www.lfd.uci.edu/~gohlke/pythonlibs/#pycairo>中下载, 注意 Python 版本和系统版本是否均合适

```
pip install pycairo.....whl
```

#### Q4: pip 安装过包，但运行时提示没有模块

考虑电脑上是否有多个 Python，确定 pip 把包装到了需要使用的 Python 上面。

#### Q5: 关于 scipy 有报错

可能是版本不对，使用 `pip uninstall scipy` 后重新 `pip install scipy`

## 2 运行时问题

**注** 在出现以下问题时, 请确保你正在使用最新版 `cairo-backend` 分支的 `manim`

### 2.1 import 问题

**Q1: 没有模块** `big_ol_pile_of_manim_imports`

将文件中的

```
from big_ol_pile_of_manim_imports import *
```

改成

```
from manimlib.imports import *
```

**Q2: 缺少模块** `pygments7`

手动安装 `pip install pygments`

### 2.2 L<sup>A</sup>T<sub>E</sub>X 问题

**Q1: 报错** `Latex error converting to dvi`

先不要管错误在哪, 先把 `manimlib/constants.py` 中的 `TEX_USE_CTEX` 改成 `True` 再运行

**Q2: 报错** `xelatex error converting to xdv`

若为 Windows 系统, 先把 `manimlib/constants.py` 的第 29 行:

```
MEDIA_DIR = "./media"
```

---

<sup>7</sup>已在[#1147](#)中修复



改成<sup>8</sup>

```
MEDIA_DIR = os.path.join(os.getcwd(), "media")
```

再进行尝试。如果仍然出错,尝试将 `ctex_template.tex` 中的 `\usepackage{ctex}` 提到该文件的第二行再进行尝试<sup>9</sup>。还出错误的话, 向下继续按步骤进行:

### I. 若安装的 $\text{T}_{\text{E}}\text{X}$ 发行版为 $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$

1.  $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$  的有关路径是否添加到环境变量中
2. 是否有包没有装全

对于 2., 可以正常运行一遍 `WriteStuff` 场景, 看是否有框弹出提示 `install` 什么东西, 如果有, 则 `install`, 并重复运行安装运行安装... 直到不报错为止。

或者使用  $\text{T}_{\text{E}}\text{X}$  编辑器 `TEXStudio` 等并使用 `xelatex` 手动编译 `media/TeX` 文件夹中的 `.tex` 文件, 查看是否有包没有安装。

对于没有 1. 和 2. 问题却依旧报错的, 可以选择重新安装新版  $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$  或者安装 `TEXLive-full` 版 (推荐)。

### II. 若安装的 $\text{T}_{\text{E}}\text{X}$ 发行版为 `TEXLive`

1. `TEXLive` 有关路径是否添加到环境变量中
2. 安装的是否为 `full` 版本

### III. 若安装的 $\text{T}_{\text{E}}\text{X}$ 发行版不为以上两款

建议换成 `TEXLive-full` 版或者  $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ , 并且注意在重新安装前删除旧版

## 2.3 `dvisvgm` 问题

**Q1: 报错** `OSError: No file matching .svg in image directory`

清空 `media/TeX` 文件夹内全部内容, 再次运行带文字的场景, 查看 `TeX` 文件夹中的内容:

<sup>8</sup>已在#689中修复

<sup>9</sup>已在#1187中修复

- I. 若含有 .tex 文件, 但没有 .xdv 文件, 按照 2.2 中方法处理
- II. 若含有 .xdv 文件但没有 .svg 文件
  1. 检查 dvisvgm 是否添加到环境变量, 可以使用 `dvisvgm --version` 观察是否由报错来检查
  2. dvisvgm 版本是否过低, 若 `dvisvgm --version` 的输出版本号小于 2.4, 请更换新版 dvisvgm<sup>10</sup>, 并注意将含有 dvisvgm 的文件夹添加到环境变量中
  3. 若 dvisvgm 的版本高于 2.4, 可能是你的 dvisvgm 暂不支持 PostScript 请按照 Q2 中指导操作

## Q2: 如何让 dvisvgm 支持 PostScript<sup>11</sup>

打开终端, 输入 `dvisvgm -l` 检查有没有 `ps dvips PostScript specials` (如果有, 则已经支持了 PostScript); 输入 `dvisvgm -h` 检查有没有 `--libgs=filename`。接下来按照以下处理

1. 若 `dvisvgm -h` 输出中没有 `--libgs`, `dvisvgm -l` 中没有 `ps`
  - 你安装的 dvisvgm 无法支持 PostScript, 请换一个安装再试
2. 若 `dvisvgm -h` 输出中含有 `--libgs`

这说明你的 dvisvgm 需要 Ghostscript 才能支持 PostScript, 按下述操作:

  - a. 查找 Ghostscript 库
    - 如果是 Windows32 位系统, 则需要 `gsdll32.dll` (可能位于 `C:\Windows\System32\` 文件夹中)
    - 如果是 Windows64 位系统, 则需要 `gsdll64.dll` (可能位于 `C:\Windows\System32\` 文件夹中)
    - 如果是 Linux 系统, 则需要 `libgs.so` (可能位于 `/usr/local/lib/` 或 `/usr/lib/` 文件夹中)
    - 如果是 MacOS 系统, 则需要 `libgs1.dylib` (可能位于 `/usr/local/lib/` 或 `/opt/local/lib/` 文件夹中)
  - b. 添加 Ghostscript 库, 可以通过以下三种方法:

<sup>10</sup> 上网下载、或者使用群文件中的版本

<sup>11</sup> 这部分解决方案来自 dvisvgm 的 FAQ: <https://dvisvgm.de/FAQ/>

- 把上述文件位于的文件夹添加到 PATH 环境变量中
- 把上述文件的完整路径（包括目录和文件名）设置为 LIBGS 环境变量的值
- 终端输入 `dvisvgm --libgs=" 文件完整位置（包括文件名）"`

如上操作后,再输入 `dvisvgm -l`,如果含有 ps 则成功支持了 PostScript。

## 2.4 中文显示问题

**Q1:** 含有中文的 `TextObject` 编译报错, `Latex error converting to dvi`

将 `manimlib/constants.py` 中的 `TEX_USE_CTEX` 改成 `True` 再尝试

**Q2:** 英文可以正常显示, 中文不报错, 但不显示

考虑使用的是否为 `TextObject` 而不是 `TexObject`

## 2.5 文字问题

**Q1:** `TextObject` 和 `TexObject` 有什么区别

`TextObject` 和 `TexObject` 使用的都是  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  语法

其中 `TextObject` 文字模式相当于直接在  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  环境下书写

`TexObject` 公式模式使用的是  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  的 `\begin{align*}` 环境或者可以看成加了 `$$` 的环境

使用 `TextObject` 与 `TexObject` 书写公式时:

<code>TextObject("文字\$公式\$")</code> $\iff$ <code>TexObject("\\text{文字}公式")</code>
---

## Q2: TextMobject 中怎么改字体样式

TextMobject 中只能使用 L<sup>A</sup>T<sub>E</sub>X 的字体样式

字体常用样式命令见表：

字体样式	L <sup>A</sup> T <sub>E</sub> X 代码	字体样式	L <sup>A</sup> T <sub>E</sub> X 代码
roman	<code>\textrm{...}</code>	<b>bold face</b>	<code>\textbf{...}</code>
sans serif	<code>\textsf{...}</code>	medium weight	<code>\textmd{...}</code>
typewriter	<code>\texttt{...}</code>	<i>italic</i>	<code>\textit{...}</code>
SMALL CAPS	<code>\textsc{...}</code>	<i>slanted</i>	<code>\textsl{...}</code>
upright	<code>\textup{...}</code>		

严格地讲中文字体并没有衬线、无衬线、等宽、斜体等概念

## Q3: 想自定义字体怎么办

使用新版 manim 特有的 Text() 类, 方法如下 Text("文字", font="字体"), 其中字体要填写在计算机内存储的格式<sup>12</sup>, 但是不能使用 L<sup>A</sup>T<sub>E</sub>X 语法书写公式

## Q4: 想用自定义字体写公式怎么办

可以使用 cigar666 编写的 MyText() 类, 源码地址: [https://github.com/manim-kindergarten/manim\\_sandbox/blob/master/utils/mobjects/MyText.py](https://github.com/manim-kindergarten/manim_sandbox/blob/master/utils/mobjects/MyText.py)

## Q5: TexMobject 中换行是什么

四个右划线\\ \\ \\ \\, Python 转义右划线, 所以涉及到\的均要写成两个\\, 而换行在 L<sup>A</sup>T<sub>E</sub>X 中是两个右划线, 所以要写成四个<sup>13</sup>

<sup>12</sup>例如: Microsoft YaHei, Source Han Sans CN(Windows 可以打开 C:/Windows/Fonts 中的字体文件查看名称)

<sup>13</sup>或者在字符串前加 r, 正常书写

## Q6: 公式怎么对齐

- I. 直接在 `TexMobject` 中使用 `&` 对齐
- II. 两个 `mobject` 对齐,使用 `obj2.next_to(obj1, DOWN, aligned_edge=LEFT)` 使 `obj2` 在 `obj1` 下方, 并左对齐
- III. `VGroup` 内对齐, 使用 `group.arrange(DOWN, aligned_edge=LEFT)` 使 `VGroup` 中的子元素依次向下排开, 并左对齐

写公式的示例:

[https://github.com/Elteoremadebeethoven/AnimationsWithManim/blob/master/English/3\\_text\\_like\\_arrays/scenes.md](https://github.com/Elteoremadebeethoven/AnimationsWithManim/blob/master/English/3_text_like_arrays/scenes.md)

## Q7: TexMobject 上色问题的处理办法

- I. 将上色的字符分开, 使用 `text[i].set_color(color)` 来上色
- II. 将上色的字符分开,使用 `text.set_color_by_tex_to_color_map(t2c)` 传入 `t2c` 字典来对相同的字符串上色
- III. 只传入一个字符串, 但同时传入 `tex_to_color_map=t2c` 来自动拆分上色 (容易出问题)
- IV. 只传入一个字符串, 使用 `text[0][i]` 来对细小的路径上色 (一般是一个字符一个下标)

## Q8: TexMobject 的下标怎么分析

- I. 使用 `debugTeX`<sup>14</sup>, 先 `self.add(tex)` 然后再 `debugTeX(self, tex)`, 导出最后一帧<sup>15</sup>, 观察每段字符上的标号, 即为下标
- II. 使用自带的函数 `get_submobject_index_labels` 获取下标的 `VGroup`, 然后添加

关于 `Tex(t)Mobject` 的结构, 详细可以看视频<https://www.bilibili.com/video/BV1CC4y1H7kp>

<sup>14</sup>[https://github.com/manim-kindergarten/manim\\_sandbox/blob/master/utils/functions/debugTeX.py](https://github.com/manim-kindergarten/manim_sandbox/blob/master/utils/functions/debugTeX.py)

<sup>15</sup>-s 选项

## Q9: TexObject 使用 \frac 拆分时出错

这个是 Grant 写 `tex_file_writing.py` 的一个 bug, 建议使用 `{分子 \over 分母}` 来代替 `\frac{分子}{分母}`

## Q10: 使用 \left\{ ... \right. 报错

```
TexObject(r"\left\{\begin{matrix} a+b \\ b+a \\ \end{matrix} \right.")
```

`matrix` 这样的写法在 `manim` 中会报错, 无法生成 `dvi`, 原因是 `manim` 会自动寻找相对应的括号来匹配, 但这里并没有右大括号, 而是.

所以推荐使用 `cases` 环境, 效果是一样的: 
$$\begin{cases} a + b \\ b + a \end{cases}$$

```
TexObject(r"\begin{cases} a+b \\ b+a \\ \end{cases}")
```

## 2.6 素材引用问题<sup>16</sup>

### Q1: 使用 SVGObject 找不到 svg 文件

- I. 直接使用绝对路径引用 `svg` 文件
- II. 将 `svg` 文件放到 `assets/svg_images/` 文件夹中

### Q2: 如何使用 jpg 或者 png 文件

- I. 直接使用绝对路径引用, 并使用 `ImageObject`
- II. 将 `jpg/png` 文件放到 `assets/raster_images/` 文件夹中

### Q3: 能否导入 gif 文件

可以使用 `ImageObject` 导入, 但是只保留第一帧, 不会显示动图

<sup>16</sup>关于插入素材 (图片), 详细可以看视频<https://www.bilibili.com/video/BV1CC4y1H7kp>

## 3 其它问题

### Q1: 没有 manim 源码

最好不要使用 `pip install manimlib` 来装 manim, 请在 GitHub 上 clone 下来 manim 的全部内容

### Q2: 群友用的 manim 都是什么版本

一般使用的都是 GitHub 上的最后一版 `cairo-backend` 分支上的源码, 少部分使用的是 `master` 分支上的新版本源码 (不在本文档讨论范围内)

### Q3: 如何使用傅里叶级数作图

套用 Grant 写好的文件 (有部分代码 `import` 部分路径不对, 请自行调整)

```
from_3b1b/active/diffyq/part2/fourier_series.py
from_3b1b/active/diffyq/part4/fourier_series_scenes.py
from_3b1b/active/diffyq/part4/long_fourier_series.py
```

### Q4: 傅里叶级数作图如何调整时长

CONFIG 中 `run_time` 无法控制, 使用 `slow_factor` 和 `n_cycles` 来控制

$\frac{1}{\text{slow\_factor}}$  为一个循环的时间, `n_cycles` 为循环的个数

只需要更换 `svg` 素材即可<sup>17</sup>

---

<sup>17</sup>自己制作, 或者使用这里的 `svg` 素材:[https://github.com/manim-kindergarten/manim\\_sandbox/tree/master/assets/svg\\_images](https://github.com/manim-kindergarten/manim_sandbox/tree/master/assets/svg_images)

## Q5: svg 用什么软件制作

Adobe Illustrator(简称 AI, 推荐) 或者 inkscape(简称 ink, 不推荐)。而且不要使用网页版编辑器

目前 manim 对 SVG 的解析很局限, 推荐使用 AI<sup>18</sup>

## Q6: 动画怎么显示旋转一个物体

使用 `Rotate` 和 `Rotating`, 区别在群文件中有视频

## Q7: `Transform` 和 `ReplacementTransform` 有什么区别

1. `Transform(A, B)` 在画面上 A 变成了 B 的样子, 但是画面上的物体名字还叫 A
2. `ReplacementTransform(A, B)` 在画面上 A 变成了 B 的样子, 并且画面上的物体名字叫 B

所以以下两个效果相同

```
self.play(Transform(A, B))
```

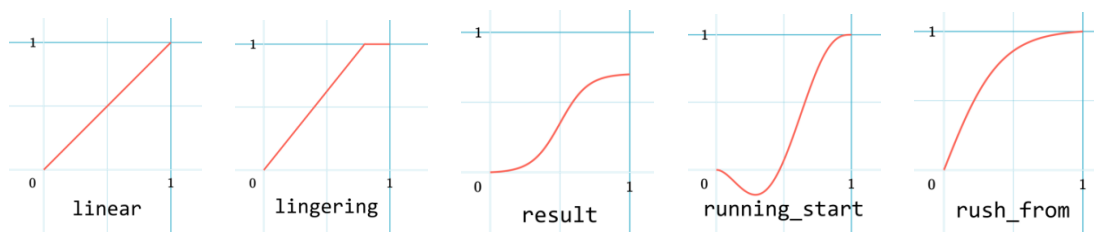
```
self.play(Transform(A, C))
```

```
self.play(ReplacementTransform(A, B))
```

```
self.play(ReplacementTransform(B, C))
```

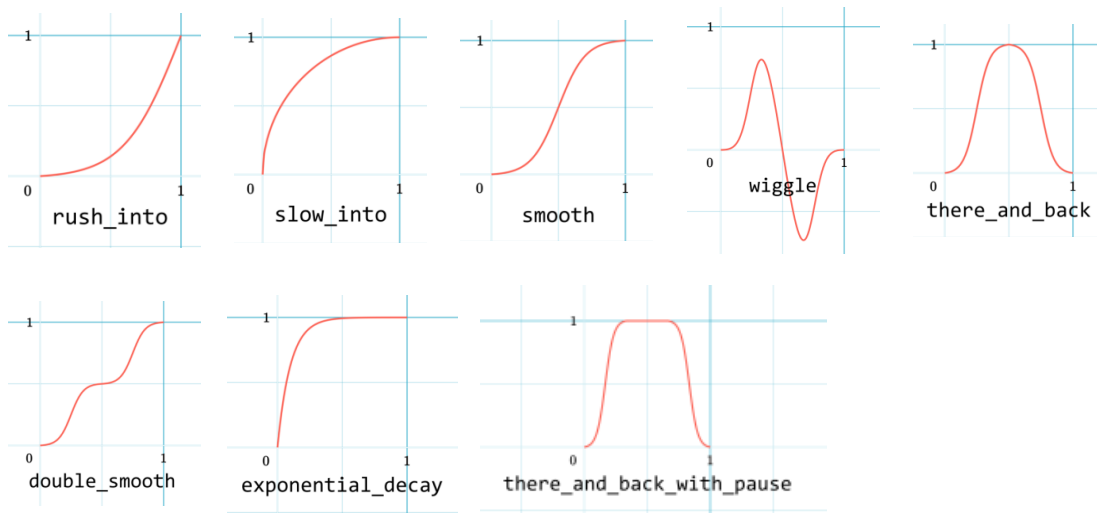
## Q8: 怎么控制物体移动或者 `Transform` 的速率

使用 `rate_func`, 一些 manim 中已经定义的在群文件中有视频



<sup>18</sup>并且使用“另存为 →SVG”的方式, 不要使用导出





## Q9: 数学符号/公式用 $\text{\LaTeX}$ 怎么打

请见 <https://www.luogu.com.cn/blog/IowaBattleship/latex-gong-shi-tai-quan>

推荐妈咪叔维护的<https://www.latexlive.com/>

## Q10: 一些特殊 $\text{\LaTeX}$ 的外部包



如何使用 `manim` 画出上面的音符，或怎么使用这些包？

在 `manimlib` 目录下的 `ctex_template.tex` 或者 `tex_template.tex` 文件中添加外部包的名称<sup>19</sup>

就拿上面的音符为例，因为是在 `harmony` 包中的，所以在 `tex` 文件中添加 `\usepackage{harmony}`<sup>20</sup>

然后新建一个 `py` 文件，写入代码

```
from manimlib.imports import *
class TestHarmony(Scene):
    def construct(self):
        # harmony具体用法请百度
```

<sup>19</sup>修改 `TEX_USE_CTEX` 为 `True` 的，可以只在 `ctex_template.tex` 中添加

<sup>20</sup>不需要使用的时候记得改回来哦

```

harmony = TextMobject(r"\Ganz \Halb \Vier \Acht \Sech
    \Zwdr")
self.play>ShowCreation(harmony))
self.wait()

```

运行 py 文件即可

## Q11: 使用 L<sup>A</sup>T<sub>E</sub>X 外部包, 编译错误或者无显示

首先, 并不是所有外部包都能在 manim 中顺利使用, 大多都不支持 xelatex 编译, 所以建议需要使用外部包时只用 latex 编译<sup>21</sup>

至于有些群友常用 TikZ 这个外部包, 也是使用 latex 才能顺利运行, 在 xelatex 用 \draw 会无法显示, 需要修改 tex\_template.tex 文件<sup>20</sup>, 修改成如下:

```

\documentclass[preview, dvisvgm]{standalone}
\usepackage{tikz}

```

新建 py 文件, 写入代码来画一条线: \_\_\_\_\_

```

class TestTikz(Scene):
    def construct(self):
        tikz = TextMobject(
            # tikz具体用法请百度
            r"\tikz{\draw (-1, 0) -- (1, 0);}",
            color=WHITE,
            stroke_width=1,
            stroke_opacity=1,
        )
        self.play>ShowCreation(tikz))
        self.wait()

```

运行 py 文件即可

---

<sup>21</sup>即把 TEX\_USE\_CTEX 改为 False

## Q12: 一些比较复杂, 操纵东西比较多的动画怎么做

使用外部剪辑软件, 例如 Adobe Premiere Pro 或者达芬奇

## Q13: 一个 `self.play` 里写两个 `ApplyMethod` 只对一个起作用怎么办

去掉 `ApplyMethod`, 例如:

```
self.play(ApplyMethod(mob.scale, 2), ApplyMethod(mob.shift, DOWN))
```

改成

```
self.play(mob.scale, 2, mob.shift, DOWN)
```

## Q14: 如何解决二维画面中的图层问题

可以使用 `pdcsxs` 添加的 `plot_depth`, 具体更改见下图<sup>22</sup>

MK fork 的版本已经做了修改: <https://github.com/manim-kindergarten/manim>

```
3 manimlib/camera/camera.py
@@ -48,6 +48,7 @@ class Camera(object):
48 48     # round z coordinate to nearest hundredth when comparing
49 49     "z_buff_func": lambda m: np.round(m.get_center()[2], 2),
50 50     "cairo_line_width_multiple": 0.01,
51 +     "open_plot_depth": True,
51 52     }
52 53
53 54     def __init__(self, background=None, **kwargs):
@@ -207,6 +208,8 @@ def extract_mobject_family_members(
207 208         method = Mobject.family_members_with_points
208 209         else:
209 210             method = Mobject.get_family
211 +         if self.open_plot_depth:
212 +             mobjects.sort(key=lambda m: m.plot_depth)
210 213     return remove_list_redundancies(list(
211 214         it.chain(*[method(m) for m in mobjects])
212 215     ))
```

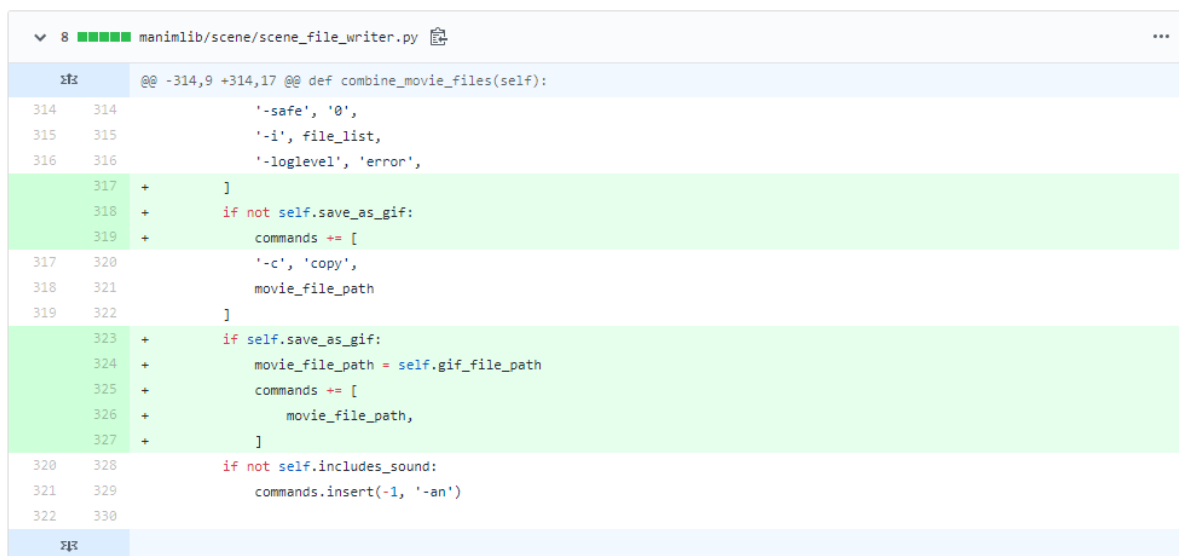
```
11 manimlib/mobject/mobject.py
@@ -34,6 +34,7 @@ class Mobject(Container):
34 34     "name": None,
35 35     "dim": 3,
36 36     "target": None,
37 +     "plot_depth": 0,
37 38     }
38 39
39 40     def __init__(self, **kwargs):
@@ -1120,6 +1121,16 @@ def throw_error_if_no_points(self):
1120 1121         caller_name = sys.getframe(1).f_code.co_name
1121 1122         raise Exception(message.format(caller_name))
1122 1123
1124 +     def set_plot_depth(self, plot_depth_value):
1125 +         self.plot_depth = plot_depth_value
1126 +
1127 +     def get_plot_depth(self):
1128 +         return self.plot_depth
1129 +
1130 +     def set_plot_depth_by_z(self):
1131 +         # to update the plot_depth by the value of Z coordinate
1132 +         z_value = self.get_center()[-1]
1133 +         self.set_plot_depth(z_value)
1123 1134
1124 1135     class Group(Mobject):
1125 1136         def __init__(self, *mobjects, **kwargs):
```

<sup>22</sup>`plot_depth` 的值越大, 运行出来的物体就越在上面

## Q15: 如何导出 gif 文件

在最后一个 `cairo-backend` 版本中，`manim` 导出 `gif` 已经失效，可以导出 `mp4`，后用 `ffmpeg` 转换。也可以按照下图修改源码

MK fork 的版本已经做了修改：<https://github.com/manim-kindergarten/manim>



```
@@ -314,9 +314,17 @@ def combine_movie_files(self):
314 314         '-safe', '0',
315 315         '-i', file_list,
316 316         '-loglevel', 'error',
317 +         ]
318 +         if not self.save_as_gif:
319 +             commands += [
317 320             '-c', 'copy',
318 321             movie_file_path
319 322         ]
323 +         if self.save_as_gif:
324 +             movie_file_path = self.gif_file_path
325 +             commands += [
326 +                 movie_file_path,
327 +             ]
320 328         if not self.includes_sound:
321 329             commands.insert(-1, '-an')
322 330
```

改过后，在输入命令时加上 `-i` 选项，就能导出 `gif` 了

## Q16: 如何导出透明的图片或者视频

在运行命令的时候加上 `-t` 选项

- 如果是 `-s` 保存图片，则会存储为背景透明的 `png` 图片
- 如果是 `-l/-m/-w` 保存视频，则会存储为背景透明的 `mov` 视频文件，方便 `pr` 中的剪辑

## Q17: 渲染视频的画质和帧率怎么调整

`manim` 的默认画质有四种

- `-l` 最低画质 480P15

- `-m` 中等画质 720P30
- `--high_quality`<sup>23</sup> 高画质 1080P60
- `-w` 导出 (最高) 画质 1440P60(2K)
- `-uhd` 超高清 4K120fps(B 站最高)<sup>24</sup>

不加画质选项,默认使用 `-w` 最高画质<sup>25</sup>。可以通过修改 `constants.py` 中对应的画面长宽和帧率来修改<sup>26</sup>

一般把 `-w` 最高画质修改成 1080P60

## Q18: 有没有什么好的场景例子供学习

1. GitHub 上 `manim-kindergarten/manim_sandbox` 中的 `demo` 和 `videos` 文件夹中的代码
2. Grant 的代码<sup>27</sup>对应 3B1B 的视频, 可能会有报错, 需要魔改
3. 群文件里 “manim 相关的 python 代码及视频结果”
4. 群里几个 B 站 up 主的 GitHub 库对应他们的代码
  - cigar666 [https://github.com/cigar666/my\\_manim\\_projects](https://github.com/cigar666/my_manim_projects)
  - 鹤翔万里 [https://github.com/TonyCrane/manim\\_projects](https://github.com/TonyCrane/manim_projects)
  - pdcxs <https://github.com/pdcxs/ManimProjects>
  - 有一种悲伤叫颓废 <https://github.com/136108Haumea/my-manim>

## Q19: 新版本 manim 是什么, 和旧版有什么区别

新版 manim 使用 OpenGL 和 `moderngl` 来进行 GPU 渲染, 会有更快的速度, 也支持实时渲染和交互操作。更改了很多底层逻辑和结构, 也调整了一些小的用法。详细在针对新版的常见问题中叙述。

关于三个版本的 manim 的简要说明在[#1243](#)

<sup>23</sup>没有缩写

<sup>24</sup>仅限 MK 版本 manim

<sup>25</sup>比如 `-p`(虽然很多人把 `-p` 当成了 `-w...`)

<sup>26</sup>`manimlib/constants.py` 的 118 行开始

<sup>27</sup>`from_3b1b` 文件夹中

## 4 注意

如果有以上之外的问题, 可以在群里提出, 也可以在 **GitHub** 上提出 issue, 或者按照下图操作



也请注意群规第 3,4 条

- 3. 虽为 **manim** 交流群, 但不要一有问题就提出来, 简单的问题能自己解决最好, 不能解决时再寻求帮助
- 4. 群主和管理员平时较忙, 有时若不能及时回复敬请谅解

最后, 祝大家好运 (\* ^ - ^ \*)

## 5 附：ChangeLog

### v3.0

- 使用了 ElegantL<sup>A</sup>T<sub>E</sub>XNote 主题
- 增加了 master 分支和 shaders 分支的区别

### v3.1

- 修复了代码段字符间距大且包含空格的 bug
- 增加了文档首页“一切之前”部分
- 增加了针对新旧版 manim 的区别和新版教程链接
- 增加了由于 dvisvgm 问题导致 OSError 的解决方案
- 增加了 ChangeLog 部分