

*Manim*

*Jianfeng He*

# 目录

|                      |   |
|----------------------|---|
| 第一章 Quick Start      | 2 |
| § 1.1 Scene          | 2 |
| 1.1.1 Introduction   | 2 |
| 1.1.2 Method         | 2 |
| § 1.2 Mobject        | 2 |
| 1.2.1 TextMobject    | 2 |
| 1.2.2 TexMobject     | 3 |
| § 1.3 Other Function | 3 |
| § 1.4 Constant       | 3 |
| 1.4.1 长度常数           | 3 |
| § 1.5 VGroup         | 4 |

# 第一章 Quick Start

## § 1.1 Scene

### 1.1.1 Introduction

**Scene** 是最基础和最主要的类, 几乎其它所有的类都由其构造而来

### 1.1.2 Method

- 1). **Scene.play()** : 播放动画, 如:

```
1 .play(ShowCreation(circle))
2 .play(FadeOut(circle))
3 .play(GrowFromCenter(square))
4 .play(Transform(square,triangle))
```

- 2). **.add()** : places a mobject on screen at the start of the scene

- 3). **arrow.next\_to()** : 使用相对位置进行移动, 如:

```
1 arrow = Arrow(LEFT,UP)
2 arrow.next_to(circle,DOWN+LEFT)
```

- 4). **.move\_to()** : move the object to a specific location on the screen, (使用绝对位置?)

- 5). **.wait(<number>)** : 控制停止的秒数, 默认为 1 秒

## § 1.2 Mobject

### 1.2.1 TextMobject

Text rendering is based on Latex so you can use many Latex typesetting features, which means if you already known the rule of  $\text{\LaTeX}$ , it will be easily to use equation. One should be noted that the backslash should be tranfered by another backslash. A more convenient way is use Python's raw string (add **r** before the first quotation mark) e.g., those are two valid using methods:

```
1 eq1=TextMobject("$\\vec{X}_0 \\cdot \\vec{Y}_1 = 3$")
2 eq1=TextMobject(r"$\\vec{X}_0 \\cdot \\vec{Y}_1 = 3$")
```

1. **.scale(<number>)**

缩放文字

2. **.set\_color()**

设置文字的颜色. 颜色可以采用默认提供的一些常数. 具体的定义参见 **constants.py**.

3. `.set_color_by_gradient(<color_1>, <color_2>, <color_3>)` 括号内的参数为一串以逗号分隔的颜色序列, 效果是生成按照该颜色序列进行的渐变色.

4. `.match_color(<obj>)`

In addition to setting the color, you can also match the color to another object. This is convenient since you can easily keep the color of two different objects as same without define extral variable.

5. `.to_edge()` , `.to_corner()`

You can align mobjects with the center of the edge of the screen by telling `to_edge()` whether you want the object to be `UP` , `DOWN` , `LEFT` , or `RIGHT` . You can also use `to_corner()` , in which case you need to combine two directions such as `UP+LEFT` to indicate the corner.

6. `.get_corner()`

Each mobject has a bounding box that indicates the outermost edges of the mobject and you can get the coordinates of the corners of this bounding box using `get_corner()` and specifying a direction. Thus `get_corner(DOWN+LEFT)` will return the location of the lower left corner of a mobject.

7. `.shift()`

变换物体的位置, 为达成缓慢上升的效果:

```
1 self.play(ApplyMethod(my_first_text.shift,3*UP))
```

注意到其中 `ApplyMethod()` 的重要性. 若没有该函数, 则上升瞬间完成. Notice the arguments of `ApplyMethod()` is a pointer to the method (in this case `my_first_text.shift` without any parentheses) followed by a comma and then the what you would normally include as the argument to the `shift()` method. In other words, `ApplyMethod(my_first_text.shift,3*UP)` will create an animation of shifting `my_first_text` three MUnits up.

8. `.rotate()`

旋转文字. 其中可调用常量 `TAU` , 一个 `TAU` 相当于  $2\pi$ .

### 1.2.2 TexMobject

Similar to `TextMobject` , the only different is you can ignore the enclosed `$` symbol, which made it more convenient to type  $\text{L}^{\text{T}}\text{E}^{\text{X}}$  equation. There is a good example:

```
1 eq2=TexMobject(r"\vec{F}_{net} = \sum_i \vec{F}_i")
```

## § 1.3 Other Function

1. `wait()` 控制停止的秒数, 默认 1 秒

## § 1.4 Constant

### 1.4.1 长度常数

`UP` , `DOWN` , `LEFT` , `RIGHT` . 这些长度常数可以被当成向量来进行计算, 也就是说可以直接进行加减. (实际上它们本来也是通过 `numpy` 的数组进行定义的)

## § 1.5 VGroup

The `VGroup` class allows you to combine multiple mobjects into a single vectorized math object. The using of the class is like:

```
1 label2_group=VGroup(label2,label2.bg)
```

This statement combine the object `label2` and `label2.bg` then generate a new object.