Grahic Model Media

Windows Media Player Microsoft Expression Encoder Audacity Manim

Manim of 3blue1brown Knowledge Base of 3b1b Manim

TOC Getting Start Manim Constant Object Useful Info

Object container.py Scene Mobject Animation Camera Utils

Mobject VMobject SVG Mobject Function

mobject.py point_cloud_mobject.py image_mobject.py

mobject.py Class Mobject

Draft for Information Only

Content

Manim Mobject

Under Construction

Home 5
Business
Management
HBR 3
Information
Recreation
Hobbies 8
Culture
Chinese 1097
English 337
Reference 68
Computer
Hardware 151
Software
Application 203
Digitization 25
Latex 34
Manim 171
Numeric 19
Programming
Web 285
Unicode 504
HTML 65
CSS 59
ASP.NET 194
OS 395
DeskTop 7
Python 56
Knowledge

A mobject, or called mathematical object, is a general name used to name a physical object used in Manim. A `Mobject` object is the fundamental element used in Manim as a dummy mobject container with base mobject manipulating functions. A `Mobject` object focuses only on the internal structural design of a `Mobject` object. The three natural features of a mathematical object are

- `m.points`, an Nx3 `numpy.array`, for specifying how to draw `m`
- `m`'s attributes for specifying the properties of `m`
- `m.submobjects`, a list of `Mobject` instances, for specifying the child objects linked to `m`

A `Group` object is simply a Mobject wrapper that `Mobject`s are grouped together as one single `Mobject`.

Codes in Mobject.py

Available codes defined in manimlib.mobject.mobject.py

```
from functools import reduce
import copy
import itertools as it
import operator as op
import os
import random
import sys

from colour import Color
import numpy as np

import manimlib.constants as consts
```

Two classes, `Mobject` and `Group` are defined.

Import

```
from functools import reduce
import copy
import itertools as it
import operator as op
import os
import random
import sys

from colour import Color
import numpy as np

import manimlib.constants as consts
from manimlib.constants import *
from manimlib.container.container import Container
from manimlib.utils.color import color_gradient
from manimlib.utils.color import interpolate_color
from manimlib.utils.iterables import list_update
from manimlib.utils.iterables import remove_list_redundancies
from manimlib.utils.paths import straight_path
from manimlib.utils.simple_functions import get_parameters
from manimlib.utils.space_ops import angle_of_vector
from manimlib.utils.space_ops import get_norm
from manimlib.utils.space_ops import rotation_matrix
```

Class Mobject(Container)

class manimlib.mobject.mobject.Mobject(Container) version 19Dec2019

Configuration of Mobject

```
    CONFIG = {
        "color": WHITE,
        "name": None,
        "dim": 3,
        "target": None,
    }
```

Functions

Functions defined in class `Mobject` are

- Initializing
  - def __init__(self, **kwargs)
  - def __str__(self)
  - def reset_points(self)
  - def init_colors(self)
  - def generate_points(self)
- def get_array_attrs(self)
- def digest_mobject_attrs(self)
- def apply_over_attr_arrays(self, func)
- #Displaying Operations
- def get_image(self, camera=None)
- def show(self, camera=None)
- def save_image(self, name=None)
- def generate_target(self, use_deepcopy=False)

- #Updating Operations
- def update(self, dt=0, recursive=True)
- def get_time_based_updaters(self)
- def has_time_based_updater(self)
- def get_updaters(self)
- def get_family_updaters(self)
- def add_updater(self, update_function, index=None, call_updater=True)
- def remove_updater(self, update_function)
- def clear_updaters(self, recursive=True)
- def match_updaters(self, mobject)
- def suspend_updating(self, recursive=True)
- def resume_updating(self, recursive=True)
-
- def apply_to_family(self, func)
- def apply_function(self, function, **kwargs)
- def apply_function_to_position(self, function)
- def apply_function_to_submobject_positions(self, function)
- def apply_matrix(self, matrix, **kwargs)
- def apply_complex_function(self, function, **kwargs)
    - def R3_func(point)
- def reverse_points(self)
- def repeat(self, count)
- #In Place Operations (much of these are now redundant)
- def apply_points_function_about_point(self, func, about_point=None, about_edge=None)
-
- def is_off_screen(self)
- def space_out_submobjects(self, factor=1.5, **kwargs)

- def replace(self, mobject, dim_to_match=0, stretch=False)
- def surround(self, mobject, dim_to_match=0, stretch=False, buff=MED_SMALL_BUFF)
-
- ##
- def save_state(self, use_deepcopy=False)
- def restore(self)
- ##
- def reduce_across_dimension(self, points_func, reduce_func, dim)
- def nonempty_submobjects(self)
- def get_merged_array(self, array_attr)
- def get_all_points(self)
-
-
-
- #Transforming
    - Resizing
        - def scale(self, scale_factor, **kwargs)
        - def scale_in_place(self, scale_factor, **kwargs) # Redundant with default behavior of scale now
        - def scale_about_point(self, scale_factor, point) # Redundant with default behavior of scale now
        - def stretch(self, factor, dim, **kwargs)
        - def stretch_about_point(self, factor, dim, point)
        - def stretch_in_place(self, factor, dim) # Now redundant with stretch
        - def rescale_to_fit(self, length, dim, stretch=False, **kwargs)
        - def stretch_to_fit_width(self, width, **kwargs)
        - def stretch_to_fit_height(self, height, **kwargs)

        - def stretch_to_fit_depth(self, depth, **kwargs)
        - def set_width(self, width, stretch=False, **kwargs)
        - def set_height(self, height, stretch=False, **kwargs)
        - def set_depth(self, depth, stretch=False, **kwargs)
    - Rotating
        - def rotate_about_origin(self, angle, axis=OUT, axes=[])
        - def rotate(self, angle, axis=OUT, **kwargs)
        - def rotate_in_place(self, angle, axis=OUT) # redundant with default behavior of rotate now
        - def flip(self, axis=UP, **kwargs)
    - Distorting
        - def wag(self, direction=RIGHT, axis=DOWN, wag_factor=1.0)
        - def pose_at_angle(self, **kwargs)
        - def put_start_and_end_on(self, start, end)
- #Positioning
    - def center(self)
    - def align_on_border(self, direction, buff=DEFAULT_MOBJECT_TO_EDGE_BUFFER)
    - def to_corner(self, corner=LEFT + DOWN, buff=DEFAULT_MOBJECT_TO_EDGE_BUFFER)
    - def to_edge(self, edge=LEFT, buff=DEFAULT_MOBJECT_TO_EDGE_BUFFER)
    - def next_to(self, mobject_or_point, direction=RIGHT, buff=DEFAULT_MOBJECT_TO_MOBJECT_BUFFER, aligned_edge=ORIGIN, submobject_to_align=None, index_of_submobject_to_align=None, coor_mask=np.array([1, 1, 1]),)
    - def shift_onto_screen(self, **kwargs)
    - def shift(self, *vectors)
    - def set_coord(self, value, dim, direction=ORIGIN)

  - def set_x(self, x, direction=ORIGIN)
  - def set_y(self, y, direction=ORIGIN)
  - def set_z(self, z, direction=ORIGIN)
  - def move_to(self, point_or_mobject, aligned_edge=ORIGIN, coor_mask=np.array([1, 1, 1]))
- Background Rectangle
  - def add_background_rectangle(self, color=BLACK, opacity=0.75, **kwargs)
  - def add_background_rectangle_to_submobjects(self, **kwargs)
  - def add_background_rectangle_to_family_members_with_points(self, **kwargs)
- Coloring
  - def set_color(self, color=YELLOW_C, family=True)
  - def set_color_by_gradient(self, *colors)
  - def set_colors_by_radial_gradient(self, center=None, radius=1, inner_color=WHITE, outer_color=BLACK)
  - def set_submobject_colors_by_gradient(self, *colors)
  - def set_submobject_colors_by_radial_gradient(self, center=None, radius=1, inner_color=WHITE, outer_color=BLACK)
  - def to_original_color(self)
  - def fade_to(self, color, alpha, family=True)
  - def fade(self, darkness=0.5, family=True)
- Attributes
  - Boundary
    - def get_points_defining_boundary(self)
    - def get_boundary_point(self, direction)
    - def get_start(self)
    - def get_end(self)
    - def get_start_and_end(self)

        - def point_from_proportion(self, alpha)
    - Critical Points
        - def get_critical_point(self, direction)
        - def get_edge_center(self, direction)
        - def get_corner(self, direction)
        - def get_center(self)
        - def get_top(self)
        - def get_bottom(self)
        - def get_right(self)
        - def get_left(self)
        - def get_zenith(self)
        - def get_nadir(self)
    - Coordinate
        - def get_coord(self, dim, direction=ORIGIN)
        - def get_x(self, direction=ORIGIN)
        - def get_y(self, direction=ORIGIN)
        - def get_z(self, direction=ORIGIN)
        - def get_z_index_reference_point(self)
    - Measurement
        - def get_extremum_along_dim(self, points=None, dim=0, key=0)
        - def length_over_dim(self, dim)
        - def get_width(self)
        - def get_height(self)
        - def get_depth(self)
- Properties
    - def get_color(self)
    - def get_center_of_mass(self)
    - def has_points(self)

    - def has_no_points(self)
    - def get_num_points(self)
- def get_pieces(self, n_pieces)
-
- #Match Other Mobject Properties
    - def match_color(self, mobject)
    - def match_dim_size(self, mobject, dim, **kwargs)
    - def match_width(self, mobject, **kwargs)
    - def match_height(self, mobject, **kwargs)
    - def match_depth(self, mobject, **kwargs)
    - def match_coord(self, mobject, dim, direction=ORIGIN)
    - def match_x(self, mobject, direction=ORIGIN)
    - def match_y(self, mobject, direction=ORIGIN)
    - def match_z(self, mobject, direction=ORIGIN)
    - def align_to(self, mobject_or_point, direction=ORIGIN, alignment_vect=UP)
- Submobject
    - def add(self, *mobjects)
    - def add_to_back(self, *mobjects)
    - def remove(self, *mobjects)
    - def push_self_into_submobjects(self)
    - def add_n_more_submobjects(self, n)
    - def copy(self)
    - def deepcopy(self)
    - def repeat_submobject(self, submob)
    -
- #Family Matter
- def __getitem__(self, value)
- def __iter__(self)

- def __len__(self)
- def get_group_class(self)
- def split(self)
- def get_family(self)
- def family_members_with_points(self)
- def arrange(self, direction=RIGHT, center=True, **kwargs)
- def arrange_in_grid(self, n_rows=None, n_cols=None, **kwargs)
- def sort(self, point_to_num_func=lambda p: p[0], submob_func=None)
- def shuffle(self, recursive=False)
- # Just here to keep from breaking old scenes.
- def arrange_submobjects(self, *args, **kwargs)
- def sort_submobjects(self, *args, **kwargs)
- def shuffle_submobjects(self, *args, **kwargs)
- #Alignment
- def align_data(self, mobject)
- def get_point_mobject(self, center=None)
- def align_points(self, mobject)
- def align_points_with_larger(self, larger_mobject)
- def align_submobjects(self, mobject)
- def null_point_align(self, mobject)
- def interpolate(self, mobject1, mobject2, alpha, path_func=straight_path)
- def interpolate_color(self, mobject1, mobject2, alpha)
- def become_partial(self, mobject, a, b)
- def pointwise_become_partial(self, mobject, a, b)
- def become(self, mobject, copy_submobjects=True)
- #Error
- def throw_error_if_no_points(self)

Class Group(Mobject)

class manimlib.mobject.mobject.Group(Mobject) version 19Dec2019

Functions

Functions defined in class `Group` are

- def __init__(self, *mobjects, **kwargs)

Source and Reference

https://github.com/3b1b/manim version 19Dec2019

Latest Updated Links

- Windows 8.1 Knowledge Base Networking Network Shell Netsh mbn (last updated On 3/28/2021)
- Windows 8.1 Knowledge Base Networking Network Shell Netsh interface portproxy (last updated On 3/27/2021)
- Windows 8.1 Knowledge Base Networking Network Shell Netsh http (last updated On 3/26/2021)
- Windows 8.1 Knowledge Base Networking Network Shell Netsh (last updated On 3/25/2021)
- Manim Knowledge Base Getting Started Useful Information Geometry VMobject TipableVMobject Line Number_line.py (last updated On 3/24/2021)
- Manim Knowledge Base Getting Started Useful Information Numbers DecimalNumber, Integer (last updated On 3/23/2021)
- Manim Knowledge Base Getting Started Useful Information Numbers (last updated On 3/22/2021)
- Manim Knowledge Base Getting Started Useful Information Geometry VMobject TipableVMobject Line DashedLine, TangentLine, Arrow, Vector, DoubleArrow (last updated On 3/21/2021)
- Manim Knowledge Base Getting Started Useful Information Geometry VMobject TipableVMobject Arc Circle, Dot, SmallDot, Ellipse, Annulus (last updated On 3/20/2021)
- Manim Knowledge Base Getting Started Useful Information Geometry VMobject TipableVMobject Arc ArcBetweenPoints, CurvedArrow, CurvedDoubleArrow (last updated On 3/19/2021)
- Manim Knowledge Base Getting Started Useful Information Geometry VMobject TipableVMobject Line (last updated On 3/18/2021)