

Abrir Anaconda

conda create -n manim python=3.10

Confirme que Procede - SEMPRE

Y, enter

Ativar Manim

conda activate manim

Instalar Manim

conda install -c conda-forge manim

Testar o Manim

manim --version

Indo para o VS

code C:\manim_proj

**PARABENS, VC DEVE ME AGRADECER POR
ISSO**

Animações

1º manim -pql ola_mundo.py OlaMundo

2º círculos.py

```
from manim import *
```

```
class CirculosAparecendo(Scene):
```

```
    def construct(self):
```

```
        grupo = VGroup(*[Circle(radius=0.3).shift(RIGHT*i) for i in range(-3, 4)])
```

```
        self.play(ShowIncreasingSubsets(grupo))
```

```
        self.wait()
```

manim -pql circulos.py CirculosAparecendo

3º estrela.py

```
from manim import *
```

```
class CriarEstrela(Scene):
```

```
    def construct(self):
```

```
        # Cria uma estrela com 5 pontas
```

```
        estrela = Star(color=YELLOW).scale(1.5)
```

```
        # Animação de criação (desenhar a estrela)
```

```
        self.play(Create(estrela))
```

```
        # Espera um pouco
```

```
        self.wait(1)
```

```
# Muda a cor e move a estrela  
  
self.play(estrela.animate.set_fill(ORANGE, opacity=0.6).shift(UP*1.5))  
  
self.wait(1)
```

manim -pql estrela.py CriarEstrela

4º circulo_brilhante.py

```
from manim import *
```

```
class CirculoComBordaAnimada(Scene):
```

```
    def construct(self):
```

```
        # Cria um círculo branco simples
```

```
        circulo = Circle(radius=2, color=WHITE)
```

```
        # Cria o efeito AnimatedBoundary à volta do círculo
```

```
        borda_animada = AnimatedBoundary(circulo, color=YELLOW)
```

```
        # Adiciona o círculo e o efeito à cena
```

```
        self.add(circulo, borda_animada)
```

```
        # Mantém a animação viva durante alguns segundos
```

```
        self.wait(4)
```

manim -pql circulo_brilhante.py CirculoComBordaAnimada

5º seta.py

```
from manim import *

class CriarSeta(Scene):
    def construct(self):
        # Cria uma seta da esquerda para a direita
        seta = Arrow(
            start=LEFT * 3,    # ponto inicial
            end=RIGHT * 3,    # ponto final
            color=RED,         # cor da seta
            buff=0             # sem margem
        )

        # Anima o desenho da seta
        self.play(GrowArrow(seta))

# Espera um segundo antes de encerrar
self.wait(1)
```

manim -pql seta.py CriarSeta

6º

grow_from_edge.py / crescer a partir da borda

```
from manim import *

class CrescerDeBaixo(Scene):
    def construct(self):
        # Cria um quadrado azul
        quadrado = Square(color=BLUE, fill_opacity=0.7)

        # Posição inicial (no centro)
```

```

quadrado.move_to(ORIGIN)

# Anima o quadrado crescendo a partir da borda inferior (DOWN)
self.play(GrowFromEdge(quadrado, edge=DOWN))

# Espera um pouco antes de terminar
self.wait(1)

```

manim -pql grow_from_edge.py CrescerDeBaixo

7°

grow_from_point.py / crescer a partir de um ponto

```
from manim import *
```

```

class EstrelaCrescendo(Scene):
    def construct(self):
        # Cria uma estrela amarela
        estrela = Star(color=YELLOW, fill_opacity=0.8).scale(1.5)

        # Define o ponto de onde ela vai "nascer"
        ponto_origem = DOWN * 2 # dois blocos abaixo do centro

        # Anima a estrela crescendo a partir desse ponto
        self.play(GrowFromPoint(estrela, point=ponto_origem))

        # Espera um segundo
        self.wait(1)

```

manim -pql grow_from_point.py EstrelaCrescendo

8°

circunscrever_miguel.py

```
from manim import *
```

```

class CircunscreverMiguel(Scene):
    def construct(self):
        # Cria o texto com o nome Miguel
        nome = Text("Miguel", font_size=96, color=BLUE)

        # Adiciona o texto na tela
        self.play(Write(nome))
        self.wait(0.5)

        # Circunscreve o nome (animação de destaque)
        self.play(Circumscribe(nome, color=YELLOW, fade_out=True,
                               time_width=1.5))

        # Espera um pouco antes de terminar

```

```
    self.wait(1)
```

```
manim -pql circunscrever_miguel.py CircunscreverMiguel
```

9°

bola_movendo.py

```
from manim import *
```

```
class BolaAoLongoDaReta(Scene):
```

```
    def construct(self):
```

```
        # Cria uma linha horizontal (reta)
```

```
        reta = Line(start=LEFT * 4, end=RIGHT * 4, color=WHITE)
```

```
        # Cria uma bolinha vermelha
```

```
        bola = Dot(color=RED, radius=0.15)
```

```
        # Adiciona ambos à cena
```

```
        self.add(reta, bola)
```

```
        # Move a bolinha ao longo da reta (5 segundos)
```

```
self.play(MoveAlongPath(bola, reta), run_time=5, rate_func=linear)
```

```
        # Espera um segundo no final
```

```
        self.wait(1)
```

```
manim -pql bola_movendo.py BolaAoLongoDaReta
```

10°

transform_quadro_estrela.py

```
from manim import *
```

```
class QuadroViraEstrela(Scene):
```

```
    def construct(self):
```

```
        # Cria o quadrado azul
```

```
        quadrado = Square(color=BLUE, fill_opacity=0.7)
```

```
        # Cria a estrela amarela
```

```
        estrela = Star(color=YELLOW, fill_opacity=0.8).scale(1.2)
```

```
        # Mostra o quadrado
```

```
self.play(Create(quadrado))
```

```
        self.wait(0.5)
```

```
        # Transforma o quadrado na estrela
```

```
self.play(Transform(quadrado, estrela), run_time=2)
```

```
# Espera um pouco para ver o resultado
self.wait(1)
```

manim -pql transform_quadro_estrela.py QuadroViraEstrela

11°

transform_miguel.py

```
from manim import *
```

```
class TransformarMiguel(Scene):
```

```
    def construct(self):
```

```
        # Texto inicial
```

```
        texto_inicial = Text("sou apenas Miguel", font_size=64)
```

```
        # Texto final
```

```
        texto_final = Text("sou incrível Miguel 🌟", font_size=64)
```

```
        # Mostra o texto inicial
```

```
        self.play(Write(texto_inicial))
```

```
        self.wait(1)
```

```
        # Transforma o texto inicial no final
```

```
        self.play(TransformMatchingShapes(texto_inicial, texto_final), run_time=3)
```

```
        # Espera para apreciar o resultado
```

```
        self.wait(1)
```

manim -pql transform_miguel.py TransformarMiguel

12°

spaceout.py / espaçar elementos

```
from manim import *
```

```
class EspacarObjetos(Scene):
```

```
    def construct(self):
```

```
        # Cria uma estrela azul e um círculo vermelho
```

```
        estrela = Star(color=BLUE, stroke_width=4).scale(1)
```

```
        circulo = Circle(color=RED, stroke_width=3).scale(1.8)
```

```
        # Agrupa os dois objetos
```

```
        grupo = VGroup(estrela, circulo)
```

```
        # Mostra o grupo centralizado
```

```
        self.play(Create(grupo))
```

```
        self.wait(1)
```

```
# Anima o grupo se expandindo (espaçando os subobjetos)
self.play(grupo.animate.space_out_submobjects(1.5), run_time=3)

# Espera para visualizar o resultado
self.wait(1)
```

manim -pql spaceout.py EspacarObjetos

13°
grid.py / grade

```
from manim import *

class OrganizarEmGrade(Scene):
    def construct(self):
        # Cria 6 quadrados brancos
        quadrados = VGroup(*[Square(color=WHITE) for _ in range(6)])

        # Coloca todos inicialmente em linha (horizontal)
        quadrados.arrange(RIGHT, buff=0.5)

        # Mostra os quadrados na tela
        self.play(Create(quadrados))
        self.wait(1)

        # Anima reorganizando os quadrados em grade 2x3
        self.play(quadrados.animate.arrange_in_grid(rows=2, cols=3, buff=0.5),
                  run_time=3)

        # Espera no final
        self.wait(1)
```

manim -pql grid.py OrganizarEmGrade