

Seminario de Aplicaciones de Cómputo Animación Programática (con Python, Manim y Programación Orientada a Objetos)

M. en C. Diego Alberto Barceló Nieves
Facultad de Ciencias, Universidad Nacional Autónoma de México

Objetivo general

Aprender a visualizar y animar conceptos matemáticos de forma precisa utilizando la librería **Manim** de **Python**.

Objetivos específicos

1. Aprender la sintaxis básica del lenguaje de programación Python.
2. Entender el paradigma de Programación Orientada a Objetos y poder aplicarlo, creando clases o atributos nuevos cuando sea necesario.
3. Aprender a realizar animaciones precisas (programadas mediante un *script* de Python) en temas fundamentales de las matemáticas tales como geometría analítica, álgebra lineal, cálculo diferencial e integral, ecuaciones diferenciales, etc.
4. Entender principios de diseño del *software* Manim.
5. Aprender a utilizar, modificar y contribuir a proyectos de *software* de código abierto (como **Manim**).

0. Introducción a Python

1. Sintaxis básica, arreglos, variables y diccionarios. (Mutabilidad e inmutabilidad)
2. Funciones con cantidades variables de parámetros. (***args** y ****kwargs**)
3. Clases y objetos. (Parámetros y atributos)
4. Subclases. (Herencia y polimorfismo)

1. Introducción a Manim

1. Instalación, uso básico y clases fundamentales.
2. Escenas. (Introducción a la clase **Scene**)
3. Objetos animables vectorizados. (**Mobject** \rightarrow **VObject**¹ y ejemplos como **Triangle**, **Rectangle** y **Circle**)
4. Escritura de texto y \LaTeX . (**Mobject** \rightarrow **VObject** \rightarrow **SVGObject** \rightarrow {**Text** y **SingleStringMathTex** \rightarrow **MathTex**}²)
5. Configuración. (Introducción a la clase **ManimConfig**)

¹Utilizamos el formato **A** \rightarrow **B** para indicar que **B** es una subclase directa de **A** en el sentido de la Programación Orientada a Objetos.

²El formato **A** \rightarrow {**B**, **C** y **D**} indica que **B**, **C** y **D** son subclases directas de **A**.

2. Escenas geométricas en dos dimensiones

1. Plano cartesiano, puntos y líneas (`NumberPlane`, `Dot` y `VObject` \rightarrow `TypableVObject` \rightarrow `{Line y Arc}`)
2. Polígonos. (`VObject` \rightarrow `Polygram` \rightarrow `Polygon`)
3. Sistemas coordenados. (`ParametricFunction` \rightarrow `NumberLine`, `NumberPlane` \leftarrow `{Axes y CoordinateSystems}`³ y `NumberPlane` \rightarrow `ComplexPlane`)
4. Curvas y gráficas.

3. Cámaras móviles en dos dimensiones

1. Movimiento de cámara. (`Scene` \rightarrow `MovingCameraScene` y `Camera` \rightarrow `MovingCamera`)
2. Acercamiento y alejamiento de cámara. (`MovingCameraScene` \rightarrow `ZoomedScene`)
3. Múltiples cámaras. (`MovingCamera` \rightarrow `MultiCamera`)

4. Escenas vectoriales en dos dimensiones

1. Flechas y vectores flecha. (`Line` \rightarrow `Arrow` \rightarrow `{DoubleArrow y Vector}`)
2. Escenas vectoriales. (`Scene` \rightarrow `VectorScene`)
3. Transformaciones lineales. (`VectorScene` \rightarrow `LinearTransformationScene`)
4. Campos vectoriales. (`VObject` \rightarrow `SVGPathMobject` \rightarrow `VectorField` \rightarrow `{StreamLines y ArrowVectorField}`)

5. Escenas en tres dimensiones

1. Escena y cámara tridimensionales (`Scene` \rightarrow `ThreeDScene` y `Camera` \rightarrow `ThreeDCamera`)
2. Puntos, curvas y superficies en 3D.
3. Líneas y flechas en 3D.
4. Texto en 3D.

Bibliografía básica

1. [Documentación de Manim Community Edition](#).
2. Deitel y Deitel, *Intro to Python for Computer Science and Data Science* (2021).

Bibliografía complementaria

1. Repositorio de GitHub [animathica/seminario](#).
2. Lista de reproducción [Tutorial de Manim en español](#) del canal de YouTube “El teorema de Beethoven”.
3. [Documentación de Python](#).
4. Página web [GeeksforGeeks](#).
5. [Documentación de Jupyter](#).

³Utilizamos el formato `Z <- {W, X y Y}` para indicar que `W`, `X` y `Y` son supclases (o “súper clases”) directas de `Z`.