Curso propedeutico

A) Vamos aprender a programar y también la matemática y física detrás

1. Introducción

a) Orientación hacia el problema

b) Creación rápida de prototipos en Python

c) La codificación de Matemáticas luce como Matemáticas

d) Las unidades juegan un papel en los prototipos de la Física

2. Python

2.1. Instalación de Python

1. Windows y Linux
2. Evite el infierno de versiones de Python

2.2. ipython como calculadora

1. Un medio mas tres octavos
   1.1 Enteros
   1.2 Python no redondea los decimales
2. Un radian en grados 4
3. Hay un error porque Python aún no conoce ese valor
4. Asigne un valor y luego asigne un valor nuevo
5. Use la flecha hacia arriba para buscar los comandos anteriores

2.3. Aprender Python

B) Principio básico: aprendes mejor lo que haces todos los días

1. Crear directorio /bin
2. Ejecutar un script
3. Crear una función en un archivo
4. Cargar la función y ejecutarla
5. Navegar por sistema de archivos con comandos cd y ls

3. Importar modulos

3.1. numpy

1. Vamos a usar numpy para guardar arreglos de muchos valores
2. Vetorizacion
3. Posición de numpy en el stack científico de Python
4. Las matrices son una subclase de los arreglos

1. Creación de 1D ndarrays
2. np.e 8.1.1
3. np.pi
4. Funciones clásicas 6.3

3.2. matplotlib

1. Queremos graficar los arreglos de muchos valores
2. Un grafico en un solo subplot 8.1.2
3. plt.plot()
4. plt.show(block=False)

4. Radian

a) Un radian no tiene unidades
b) Un radian en grados
c) Circulo

5. Constantes y Unidades

5.1. Constantes

a) import astropy.constants as cte
b) Ver constantes disponibles
c) Velocidad de la luz
d) Aceleración de gravedad
e) Convertir c y g0
f) Las constantes tienen unidades

5.2. Unidades MKS

a) import astropy.units as u
b) Ver unidades disponibles
c) Crear velocidad de 6 km/h
d) Convertir a m/s
e) Encuentre equivalencia para cada unidad MKS

5.3. Unidades imperiales

a) import astropy.units.imperial as imp
b) Ver unidades disponibles
c) Crear una distancia de 6 pies
d) Convertir aceleración de gravedad 7.1
e) Calcular el perímetro de la Tierra en millas náuticas

6. Matemática

6.1. Trigonometria

1. Definiciones senos y cosenos sobre el circulo unitarion
2. Parámetros oscilatorios
3. Frecuencia
4. Periodo
5. Amplitud

6.2. Álgebra

1. isympy
2. Fracciones
3. fraction-sympy.py

1. Sistema de ecuaciones
2. Cuatro incógnitas
3. 4D.py

6.3. Utilizar funciones

C) Muchos recursos para funciones disponibles

1. Funciones clásicas
2. three-random.py

1. Nuestro propio modelo
2. `guess_a_number.py`

1. Cómo ajustar un polinomio a partir de un conjunto de puntos
2. fit-polynomial.py

6.4. Conjuntos

D) Los conjuntos se usan como base en la fundación de la Matemática

1. set-operations.py
2. Desigualdades

6.5. Graficar la derivada de una función

1. derivative-numpy.py
2.

7. Física

7.1. Cinemática en una dimensión.

1. Aceleración de Galileo
2. freefall.py

7.2. Movimiento de proyectiles

1. Ecuaciones del proyectil
2. projectile.py
3. Compare con resistencia del aire

7.3. Leyes de Newton

1. Plano no inclinado con fricción
2. Plano inclinado con fricción

7.4. Energía

1.

7.5. Espectro EM

1.

8. Ejemplos

8.1. Ejemplos en Matemática

8.1.1. Aproximar e

a) e-aprox.py

8.1.2. Modelar latidos

a) modelar-ciclos.py
b) np.arange()
c) linewidth=, 'r–'

8.2. Ejemplos en Física

8.2.1. Energía

a) cost-of-heat.py
b) gasoline-to-houses.py
c) car_consumption.py

8.2.2. Rotación de la Tierra

a) UTC-datetime.py

8.2.3. Espectro EM

a) limits_EM.py
b) EM.py