

# 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. Python como calculadora

B) ipython

- 1. Un medio mas tres octavos [5.2](#)
  - 1.1 Enteros
  - 1.2 Números con decimales
- 2. Un radian en grados [3](#)

### 2.3. Importar modulos

- 1. numpy
  - 1.1 Creación de 1D ndarrays
  - 1.2 np.e [7.1.1](#)
  - 1.3 np.pi
  - 1.4 Funciones clásicas [5.3](#)
- 2. matplotlib [7.1.2](#)
  - 2.1 Un grafico en un solo subplot
  - 2.2 plt.plot()
  - 2.3 plt.show(block=False)

## 3. Radian

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



## 4. Constantes y Unidades

### 4.1. Constantes

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

### 4.2. 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) Convertir velocidad de la luz a Km/h

### 4.3. 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 [6.1](#)
- e) Calcular el perímetro de la Tierra en millas náuticas

## 5. Matemática

### 5.1. Trigonometria

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

### 5.2. Álgebra

1. `isympy`
2. Fracciones
3. `fraction-sympy.py`
1. Sistema de ecuaciones
2. Cuatro incógnitas
3. `4D.py`



### 5.3. 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. Física

### 6.1. Cinemática en una dimensión.

1. Aceleración de Galileo
2. `freefall.py`

### 6.2. Movimiento de proyectiles

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

### 6.3. Leyes de Newton

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

## 7. Ejemplos

### 7.1. Ejemplos en Matemática

#### 7.1.1. Aproximar e

- a) `e-aprox.py`

#### 7.1.2. Modelar latidos

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



## 7.2. Ejemplos en Fisica

### 7.2.1.

*a)* cost-of-heat.py