



# Métodos numéricos

Mauricio Suárez Durán  
Unidad 2, Clase 1  
Introducción a Python

Departamento de Física y Geología  
Universidad de Pamplona  
I Semestre, 2020





# Python: programación básica

- Objetivo:
  - Escribir un código en Python



# Python: programación básica

- ¿Por qué Python?



# Python: programación básica

- ¿Por qué Python?
  - Es gratis
  - Fácil de programar => fácil aprender
  - Usado por miles de personas => Montones de desarrollos realizados en Python
  - Librerías poderosas para análisis de datos
  - Amplio soporte



# Python: programación básica

- Versión que usaremos Python 3.5
  - Python es un lenguaje interpretado de alto nivel, *no se necesita compilación*.
  - Ejemplo:
    - `print("hola")`



# Python: programación básica

- Variables y asignaciones:



# Python: programación básica

- Variables y asignaciones:
  - En python no se requiere especificar el tipo de variable (entero, flotante, string, etc).
  - Los nombres de las variables no deben empezar con un número, ni contener símbolos ni espacios.
  - Python distingue minúsculas de mayúsculas.



# Python: programación básica

- Tipos de variables:
  - Variables string (cadena de caracteres):
    - `x = "Solo caracteres"`
  - Enteros: valores enteros incluyendo los negativos: 0, 1, -10, 10, etc.





# Python: programación básica

- Tipos de variables:
  - Flotantes: valores reales, 3.14159...,  $6.63 \times 10^{\{34\}}$ , 1.0.
  - Complejos: valores complejos, ejemplo:  $1+2j$ ;  $-3.5 - 0.4j$ . En Python,  $j$  es la unidad imaginaria.



# Python: programación básica

- Variables:
  - Podemos usar notación científica para definir una variable:
    - `x = 1.2e2; y = 1e-10`
  - El tipo de variable puede cambiar durante la ejecución:
    - `x = 1; x = 1.5;` Lo que no ocurre en lenguajes como C y C++
  - Para crear una variable como flotante:
    - `x = 1.`



# Python: programación básica

- Aritmética:

- $x+y$

- $x-y$

- $x*y$

- $x/y$

- $x**y$



# Python: programación básica

- Aritmética:
  - $x//y$ , división entera
  - $x\%y$ , modulo; cuando un número es divisible por otro.
  - $x+y$ , el resultado depende del tipo de variables que se estén operando.
    - Pruebe operando combinaciones de enteros con flotantes.



# Python: programación básica

- Aritmética:
  - En el caso de la operación (/) siempre se obtiene un flotante.
- Combinando operaciones:
  - $x+2*y$
  - $x-y/2$
  - $2*x**3$
  - $x/2*y$
  - Multiplicación y división se realizan antes que las sumas y restas



# Python: programación básica

- Otras asignaciones:
  - `x = x + 1`
  - Ejemplo:
    - `x = 0; x = x**2 - 2`
  - `x += 1`
  - `x -= 1`
  - `x *= -2.6`
  - `x /= 5*y`
  - `x //= 3.4`



# Python: programación básica

- Otras asignaciones:
  - $x, y = 1, 2.5$
  - $x, y = 2*z+1, (x+y)/3$ 
    - Primero evalúa el lado derecho antes de asignar los respectivos valores.
  - Útil para intercambiar valores:
    - $x, y = y, x$



# Python: programación básica

- Ejercicios:
  - Realice operaciones con números imaginarios y verifique.
  - Ejemplo:
    - `print(a*b)`





# Python: programación básica

- Estamentos de entrada y salida:



# Python: programación básica

- Estamentos de entrada y salida:
  - Salida
    - `print(x, y, sep="...")`
    - `print(x, y, sep="")`



# Python: programación básica

- Estamentos de entrada y salida:
  - Entrada
    - `x = input ("Entre el valor para x: ")`
    - `x = input ("")`



# Python: programación básica

- Estamentos de entrada y salida:
  - Entrada
    - `x = input("Entre el valor para x: ")`
    - `x = input("")`
  - Lo que se ingresa es considerado como un string y no como un número. Para convertirlo en número:
    - `nonumber = input("Entre el valor para x: ")`
    - `number = float(nonnumber)`



# Python: programación básica

- Estamentos de entrada y salida:
  - Otra forma:
    - `x = float(input("Entre el valor para x: "))`
  - Ejecute la linea anterior y asigne un entero, qué ocurre?



# Python: programación básica

- Un ejemplo de código:



# Python: programación básica

- Un ejemplo de código:
  - Caída libre:
    - $y = 0.5 * g * t^{**2}$
  - Escriba un código que calcule la posición en  $\hat{j}$  para un objeto en caída libre, luego de un tiempo  $t$  y una altura inicial  $h$ ; ambos valores ingresados por consola.



# Python: programación básica

- Ejercicio:
  - Satélite orbitando la Tierra.
    - 1) Calcule la altura sobre la superficie ( $h$ ) terrestre a la que debe estar un satélite que órbita la Tierra con período  $T$ .
    - 2) Escriba un programa que estime  $h$  a partir de un período dado.
    - 3) Estime la altura para un  $T$  de un día (geosincrónico), 90 min y 45 min.
    - 4) Técnicamente, el período  $T$  de un satélite geosincrónico es por día sideral (23.93 h). Cuánta es la diferencia en  $h$ , para un período de 24 h?