

## 1er Curso de Python Científico

### FUNDAMENTACIÓN Y OBJETIVOS

En la actualidad, para realizar cálculos en muchos casos de la física, entre otras ciencias, es necesario el manejo de cálculos numéricos y con uso directo de computadoras.

Existen distintos métodos de cálculo numérico que nos permiten abarcar un sinnúmero de soluciones a situaciones problemáticas, con aplicaciones en la ciencia.

Dado que estos métodos de cálculo deben ser realizados en computadoras, es absolutamente imprescindible el manejo de lenguajes de programación. Para ello hemos elegido un lenguaje de un gran uso en la actualidad.

Lenguaje que proponemos es Python, pues es un lenguaje moderno, con muchas prestaciones, de software libre y una variedad muy grande de librerías. Preferentemente veremos Python 3.x

El objetivo a alcanzar con este lenguaje, es un manejo suficiente para desarrollar los programas de cálculo que puedan ser de utilidad a lo largo del trayecto universitario y posterior carrera científica

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y un código legible.

Algunas de las razones de aprender Python:

1. Es una buena alternativa para optar por este lenguaje en el mundo de la programación, puesto que es un lenguaje muy sencillo y fácil.
2. Su sintaxis es fácil de entender puesto que es cercana al lenguaje natural, y los programas hechos en Python parecen pseudocódigos, lo cual brinda una gran ayuda en su mantenimiento.
3. Es de código abierto, nos permite crear nuestras aplicaciones e involucrarnos en su desarrollo.
4. Tiene tipado dinámico, esto se refiere a que su tipo de dato se determinará en tiempo de ejecución, según el valor que se le asigne a una variable.
5. Es multiplataforma y está disponible en muchas plataformas, entre las más comunes son Windows, Mac, distribuciones Unix/Linux, Android, etc.
6. Es multiparadigma permitiendo la programación imperativa, orientada a objetos y funcional.
7. Tiene una gran comunidad dispuesta a ayudarnos con nuestro código.
8. Además, puede ser utilizado como script para utilizar otros lenguajes de alto nivel.
9. Se han escrito librerías optimizadas para casi cualquier cosa.
10. Hay interfaces muy amigables para la mayoría de las librerías de uso masivo escritas en Fortran o C.

### CONTENIDOS

1. Presentación
  - a. ¿Qué es Python?
  - b. Características generales de Python
  - c. Ventajas

- d. Desventajas
  - e. Python 2 vs Python 3
- 2. Tipos y estructuras de datos
  - a. Fundamentos de Python
  - b. Ejemplos
- 3. Bloques de código y estructuras de control
  - a. Control de Flujo
  - b. Estructuras de selección
  - c. Estructuras de control iterativas
  - d. Ejemplos
- 4. Funciones y programación estructurada
  - a. Importando Módulos
  - b. Colecciones
  - c. Ejemplos
  - d. Funciones
  - e. Ejemplos
- 5. Operaciones de E/S y manipulación de archivos
  - a. Archivos
  - b. Función Open
  - c. Método close()
  - d. Método write()
  - e. Método read()
- 6. La librería algebraica Numpy
  - a. Arrays
  - b. Indexación de matrices
  - c. Datatypes
  - d. Operaciones Matemáticas en arrays
  - e. Reshape
  - f. Broadcasting (tentativo)
  - g. Transformaciones Array/Matrix
  - h. Álgebra lineal
  - i. Resolución de Ecuaciones Diferenciales Parciales
- 7. La librería de gráficos Matplotlib
  - a. Tamaño de la figura, relación de aspecto y DPI
  - b. Guardando imágenes
  - c. Leyendas, etiquetas y títulos
  - d. Colores y tipos de marcadores de línea
  - e. Rango de trazado de figuras
  - f. Cuadrícula del eje
  - g. Histogramas
  - h. Gráficos de Barras (Bar chart)
  - i. Piecharts
  - j. Curvas de Nivel o Mapas de Contorno
  - k. Subplot
  - l. Figuras en 3D
  - m. Imágenes
  - n. Representación gráfica de funciones y datos
- 8. La librería científica Scipy
  - a. Integración numérica
  - b. Resolución de Ecuaciones Diferenciales Ordinarias

- c. Algebra Lineal
- d. LaTeX
- e. Cálculo de datos
- 9. Cálculo Simbólico con Sympy
  - a. Declaración de símbolos
  - b. Cálculos de Límites
  - c. Derivadas
  - d. Integrales
  - e. Series
  - f. Ecuaciones Algebraicas
  - g. Ecuaciones Ordinarias Simples
  - h. LaTeX
  - i. Matrices
  - j. Gráficos
  - k. Aplicación en electromagnetismo
- 10. El entorno Fenics, cálculo de ecuaciones diferenciales parciales.
  - a. Resolviendo la ecuación de Poisson
  - b. Configuración de múltiples condiciones de Dirichlet
  - c. Aplicación en electromagnetismo

**FECHA** (propuesta)

Semana del 3 al 6 de marzo del 2020

**DURACIÓN**

4 días

**LUGAR**

Laboratorio de Computación (reservada provisoriamente)

**ENTORNO COMPUTACIONAL**

La mayoría de las necesidades de software son inmediatas para la gente acostumbrada al entorno Unix, Mac o Windows

**MODO**

El curso se basará en mostrar ejemplos de código funcionando y las actividades serán las de producir pequeños proyectos modificando ligeramente los mismos.

**REQUERIMIENTOS**

Tener una exposición anterior a la programación de cualquier tipo. Preferentemente tener Numérico I o similar aprobado.

**CERTIFICACIÓN**

Al finalizar el mini curso se le entregará un certificado electrónico.

**CUPOS**



El cupo de participantes dependerá del laboratorio a utilizar. En el caso propuesto es 40 el cupo. De los cuales se propone que 30 sean para alumnos de grado de la Universidad Nacional de Córdoba o estudiantes de doctorado de Nuestra Facultad, y 10 de otro entorno.

### **COSTOS**

Para los alumnos de grado de la Universidad de Nacional de Córdoba o estudiantes de doctorado de nuestra Facultad, será sin costos.

Para otros alumnos, el costo propuesto es de \$ 2500,00