

**Propuesta capacitación Python**  
**Nelson R. Salinas**  
**IDEAM - SMByC**

## **Objetivo**

Introducir a los asistentes a Python 2.7 y su empleo en la realización de operaciones computacionales básicas. Dada la planificación de contenidos, no es necesario haber tenido experiencia previa con Python o cualquier otro lenguaje de programación para tomar la capacitación.

## **Esquema**

Se propone dictar la capacitación a lo largo de 5 sesiones teórico-prácticas (una hora teórica, luego dos horas prácticas) para máximo 20 estudiantes. Las sesiones serán realizadas el 3, 4, 9, 10 y 23 de abril, 2018.

## **Requerimientos**

**Equipo:** Portátil con procesador de 64bits, al menos 600 MB de espacio libre en el disco duro y permiso de administrador.

**Software:** El curso requiere la instalación de Python 2.7 y varias librerías adicionales (Pandas, Numpy, Matplotlib, Jupyter). Una de las formas más sencillas es instalar todo el software necesario es a través del sistema de distribución [Conda](#). Antes de descargar Anaconda debe decidir cual versión de Conda desea instalar: una versión resumida (Miniconda), que sólo contiene Python y el sistema de distribución; o una versión extendida (Anaconda), que contiene Python, el sistema de distribución y varias librerías, entre ellas todas aquellas que usaremos en la capacitación. La versión extendida ocupa 3 GB de espacio en el disco duro, la versión resumida 400 MB. Las librerías de Python necesarias para el curso serán instaladas durante la primera sesión, dado el caso que se haya instalado la versión resumida (Miniconda).

**Instrucciones de instalación en Windows:** [oficial](#) y [no oficial](#) pero detallada.

**Instrucciones de instalación en Mac:** [oficial](#).

## **Contenido**

1. Introducción. Origen de Python, aplicabilidad.

1.1. Herramientas: Editor de texto, intérprete de Python, distribuciones, comentarios.

1.2. Tipos de datos: Escalares (int, float, bool, None), no escalares (list, tuples, dictionary, string, unicode).

1.3. Operaciones clásicas: len, indexing, slicing, print

1.4. Operadores, expresiones.

1.5. Variables, asignación de variables, palabras reservadas.

## 2. Control de flujo

2.2. Pruebas condicionales

2.3. if, elif, else

2.4. for, while, break, continue

## 3. Funciones

3.1. Definición, argumentos, valores de retorno

3.2. Ámbito, variables internas

## 4. Programación orientada a objetos

4.1. Clases, atributos.

4.2. Instancias.

## 5. Modulos de importancia

5.1. codecs: manipulación eficiente de textos con acentos diacríticos.

5.2. numpy: operaciones matemáticas y estadísticas.

5.3. pandas: manipulación efectiva de tablas.

5.4 matplotlib.pyplot: visualización de datos.

5.5. re: manipulación de texto y expresiones regulares.



