

VI Convocatoria para la realización de actividades de formación docente en Centros, Titulaciones y Departamentos.



# Python para Cálculo Científico y Técnico

Fac. de Ciencias. Del 8 al 17 de Septiembre, 2015

**Pedro González Rodelas** ([prodelas@ugr.es](mailto:prodelas@ugr.es))  
**Fco. Miguel García Olmedo** ([@Haskell\\_ETSIIT](#))

Dptos. de Matemática Aplicada y Álgebra de la UGR


# Contenidos

1. Introducción e Instalación de Python.
2. Cálculo simbólico y aproximado.
3. Tipos y estructuras de datos.
4. Programación procedural con Python.
5. Programación orientada a objetos con Python.
6. Representaciones gráficas y visualización de datos.
7. Métodos numéricos con Python.
8. Miscelánea

# Primera Sesión

- 🌐 **Introducción e Instalación de Python:**
  - a) **Introducción e historia de Python.**
  - b) **Por qué es adecuado Python para el cálculo científico.**
  - c) **Distintas distribuciones y entornos de desarrollo de Python.**
  - d) **Proceso de instalación del software en distintos sistemas operativos.**

# Introducción a Python

 [Historia de Python](http://picachu.dmt.upm.es/python/introduccion.html) (extraída de <http://picachu.dmt.upm.es/python/introduccion.html>) :

Se trata de un lenguaje de programación relativamente moderno y en constante, aunque moderada, renovación.

Fue creado por [Guido van Rossum](#) en el año 1991, que consiguió crear un lenguaje sencillo, intuitivo y fácil de aprender.

Siempre ha gozado de cierta popularidad dentro del mundo UNIX porque se le considera una alternativa razonable a [Perl](#) para la realización de todo tipo de scripts.

La explosión de Python llegó entre los años 2003 y 2007 con el auge de las aplicaciones web y posteriormente con la nube.



# Introducción a Python



Historia de Python (extraída de <http://picachu.dmt.upm.es/python/introduccion.html>):

Es uno de los cuatro lenguajes oficiales de **Google** y toda la infraestructura de **Youtube** está programada en Python

Permite a los desarrolladores utilizar el mismo lenguaje para la aplicación, para el middleware (scripts de intercambios entre aplicaciones), así como para la gestión de los equipos.

Se dice que Python es en la actualidad el lenguaje de las startups con más futuro de **Silicon Valley**.

Uno de los motivos del éxito de Python es la gran cantidad de documentación de calidad que existe sobre el lenguaje y sus bibliotecas, empezando por la documentación oficial que encontraremos tanto en la página web <http://python.org> como en los instaladores para cualquier sistema operativo.

# Introducción a Python

 Por qué es adecuado Python para el cálculo científico :

En seguida, en algunos sectores del cálculo científico se lo vió como una más que prometedora alternativa a [Mathematica](#), [Matlab](#) u [Octave](#).

Jim Hugunin puso la primera piedra del castillo, Numeric. No era más que una clase para poder tratar arrays n-dimensionales en Python y algunas rutinas numéricas. El punto de no retorno llegó a partir del año 2007, con los paquetes [numpy](#), [simpy](#) y [scipy](#). Finalmente Python contaba con los bloques básicos para hacer Cálculo Numérico, todos los usuarios usaban lo mismo y sabían dónde compartir sus desarrollos.

Python cuenta en la actualidad con una colección de recursos para Ciencia equivalente a la de Matlab, incluso superior en campos como la visualización, el cálculo simbólico, o el tratamiento y análisis de datos con [panda](#).

Y la gran mayoría de estos recursos son libres y gratuitos. En resumen:

**¡Python es el futuro!**

# Introducción a Python

 Por qué es adecuado Python para el cálculo científico :

En principio, se trata de un **lenguaje interpretado**, no compilado, en la misma línea que Mathematica, Matlab u Octave y al mismo tiempo permite la utilización de entornos de programación integrados (IDLE) adaptados a nuestras preferencias.

Admite los tres principales **estilos de programación**: procedural, funcional y el orientado a objetos.

Se encuentra disponible para los **principales sistemas operativos**: Windows, Mac OSX y Linux, resultando pues multiplataforma.

Gran variedad y **calidad de documentación** y de recursos online, **libres y gratuitos**.



# Introducción a iPython

 Una mención aparte merece iPython:

Entorno interactivo desarrollado por Fernando Pérez más parecido a los notebooks de Mathematica, que a su vez pueden exportarse a formato HTML, PDF, etc.

Se puede visualizar contenido variado (HTML, LaTeX via MathJax, gráficos en línea con matplotlib), editar y ejecutar cálculos desde un simple navegador, de manera que el correspondiente núcleo(s) puede ejecutarse en la misma máquina o en máquinas o clusters remotos, favoreciendo así la **computación paralela**, incluso con **GPUs**.



# Introducción a Python



## Proceso de instalación del software en distintos sistemas operativos :

Todo va a depender del SO que tengamos y de la distribución en la que estemos interesados.

Python está en la actualidad migrando de versión: aunque la mayoría del código escrito en Python sigue las especificaciones de la versión 2 hace ya un tiempo que uno puede descargar y utilizar la versión 3.

Por otro lado, [Sage](#) es un sistema de notebooks parecido al que encontramos en entornos como [Maple](#) o [Mathematica](#), basado en Python. Fue creado por William Stein para cubrir sus necesidades como docente y se podría considerar como un proyecto paralelo, y casi independiente de iPython, que nos permite utilizar Python en la nube a través del navegador. Otras alternativas online podrían ser [Wakari](#) o [CodeSkulptor](#), pero ninguna de estas dos opciones es la óptima de cara a desarrollar un código en Python.

# Introducción a Python

## Distintas distribuciones y entornos de desarrollo de Python:

Aunque en cada instalación de Python viene un pequeño editor llamado [idle](#), no es ni mucho menos el más recomendable.

Hay centenares de entornos de desarrollo para Python; incluso algunos están pensados para parecerse lo máximo posible al entorno de Matlab: como [Enthought Canopy](#), o [Anaconda](#) con [Spyder](#). Otras menos conocidas son [PyCharm](#), [MacPython](#), ...

Es recomendable visitar la [wiki](#) del proyecto Python donde encontraréis una lista actualizada de todos los entornos de desarrollo para Python, tanto libres como comerciales.