

# Curso de Python para astrónomos aficionados... ¡o no!



## Introducción

---

S. Alonso [a.k.a. Zerjillo] - [zerjioi@ugr.es](mailto:zerjioi@ugr.es) y  
Javier Flores - [javierfloresmartin1992@gmail.com](mailto:javierfloresmartin1992@gmail.com)

Abril - Mayo 2022

Licencia de la presentación: CC BY-NC-SA 4.0 (Atribución-NoComercial-CompartirIgual 4.0 Internacional)

# ¿Quiénes proponemos el curso?



Zerjillo (S. Alonso)

Dpto. Lenguajes y Sistemas

Informáticos (UGR)

Sociedad Astronómica Granadina (SAG)



Javier Flores

Sociedad Astronómica Granadina (SAG)

Con la colaboración de la FAES y la SEA.

# Temporización

## Sesiones

- En sábado (conciliación laboral)
- De 9:30 a 14:00 (con una pausa de 30 minutos)
- Se grabarán para su uso posterior

## Calendario

- Sesión 1: 12 de marzo
- Sesión 2: 19 de marzo
- Sesión 3: 9 de abril
- Sesión 4: 23 de abril
- Sesión 5: 30 de abril

# Sobre los participantes del curso

- Más de 200 solicitudes (decenas el primer día de su publicación)
- Interés tanto de aficionados como de profesionales
- De toda la geografía nacional (¡y parte del extranjero!)
- De todos los rangos de edad

# ¿Qué queremos conseguir?

- Aprender rudimentos de programación usando Python
- Aplicar Python en astronomía:
  - Procesamiento básico de imágenes
  - Obtención automática de información desde Internet (web scrapping)
  - Cálculo de efemérides
  - Análisis de datos
  - Fotometría
  - Manejo de instrumental astronómico
  - ...
- ¡Crear comunidad!

# ¿Por qué Python?

- Código abierto y multi-plataforma

# ¿Por qué Python?

- Código abierto y multi-plataforma
- Fácil de aprender

# ¿Por qué Python?

- Código abierto y multi-plataforma
- Fácil de aprender - ¡Mentira! (opinable)



# ¿Por qué Python?

- Código abierto y multi-plataforma
- Fácil de aprender - ¡**Mentira!** (opinable)
- Se puede comunicar con Fortran o C++ (para usar código antiguo o muy eficiente)

# ¿Por qué Python?

- Código abierto y multi-plataforma
- Fácil de aprender - ¡**Mentira!** (opinable)
- Se puede comunicar con Fortran o C++ (para usar código antiguo o muy eficiente)
- Existen muchos paquetes (bibliotecas) para multitud de tareas

# ¿Por qué Python?

- Código abierto y multi-plataforma
- Fácil de aprender - ¡**Mentira!** (opinable)
- Se puede comunicar con Fortran o C++ (para usar código antiguo o muy eficiente)
- Existen muchos paquetes (bibliotecas) para multitud de tareas
- Facilidad para graficar datos

# ¿Por qué Python?

- Código abierto y multi-plataforma
- Fácil de aprender - ¡**Mentira!** (opinable)
- Se puede comunicar con Fortran o C++ (para usar código antiguo o muy eficiente)
- Existen muchos paquetes (bibliotecas) para multitud de tareas
- Facilidad para graficar datos
- Se ha convertido en un estándar

# Materiales a utilizar

- Disponibles en [esta web](#).
- Se irán “descubriendo” poco a poco.
- En el futuro quedarán disponibles en un repositorio de GitHub.
- Se añadirán las grabaciones de las sesiones (posiblemente “censuradas” :-P).
- Con licencia [CC BY-NC-SA 4.0](#)  
(Atribución-NoComercial-CompartirIgual 4.0 Internacional)

# Distribución y entorno de desarrollo que usaremos

## Anaconda

Una plataforma de distribución de Python.



## JupyterLab

Un entorno de desarrollo interactivo basado en web. Muy útil para pequeños proyectos, scripts y para docencia.



Dicho todo esto...

**¡Manos a la obra!**