

# intro\_gis\_open

July 5, 2018

## 1 Introducción a tecnologías GIS open

- Section [1.1](#)
- Section [1.2](#)
- Section [1.3](#)

### 1.1 Python

Lenguaje interpretado multipropósito y multiplataforma en el que se intentan seguir los siguientes principios, el también llamado *Zen de Python*, que busca un fácil aprendizaje y hacerlo lo más amigable posible.

The Zen of Python, by Tim Peters

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Para introducirse en él hay muchísimos recursos de calidad pero si el inglés no es problema recomiendo seguir la documentación oficial, como por ejemplo esta [Guía inicial](#).

Como lenguaje interpretado que es siempre habrá que lanzar el interprete previamente y pasar como argumento el fichero *.py* que se quiera ejecutar. Si no se pasa ningún fichero entonces se arrancará sesión interactiva en la que podremos probar cualquier sentencia.

```
python hello_world.py
```

Para añadir funcionalidad se usan librerías o paquetes que se gestionan a través de [PyPi](#):

```
pip install geopandas
```

El intérprete guardará los paquetes instalados en paths relativos al directorio de instalación donde reside. A partir de esos paths más el directorio de trabajo donde se encuentra el intérprete se creará el espacio de nombres que es desde donde se podrán importar paquetes y módulos.

```
import lector_csv
lector_csv.leer_csv('usuarios.csv')
```

## 1.2 Conda

Gestor de paquetes y entornos para poder tener distintos intérpretes Python con distintas configuraciones en una misma máquina. En esta [guía inicial](#) se explica cómo realizar la instalación y gestionar entornos a través de la herramienta.

Lo interesante de utilizar conda para gestionar python es que ya viene con una serie de paquetes instalados como *jupyter* y *pandas*, además de poder instalar librerías de otros lenguajes, no sólo de python (*oracle\_instant\_client*, *nodejs*).

Así se instalarán paquetes en el entorno activo:

```
conda install -c conda-forge geopandas
```

## 1.3 Jupyter

Tecnología que permite crear *notebooks*, documentos interactivos que se pueden mostrar en el navegador web y que permiten combinar scripts de código con distintas visualizaciones como [Markdown](#), widgets javascript interactivos, imágenes, vídeos, ...

Para ver una serie de ejemplos podemos emular un entorno con la instalación de jupyter propia o directamente ir [NBViewer](#)