

Jupyter - Sympy

Autor:

Juan David Argüello Plata - Ingeniero Mecánico

Profesor tutor:

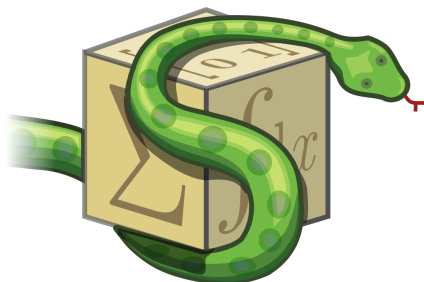
Jairo René Martínez Morales - Químico PhD

CENIVAM

Universidad Industrial de Santander

Introducción

Sympy es una librería de *Python* que se utiliza para la solución de **ecuaciones simbólicas**. Los resultados de Sympy son en nomenclatura \LaTeX , por lo que es ideal emplearlo en Jupyter.



SymPy

Objetivos

Vamos a ver cómo resolver...

- Sistema de ecuaciones algebraicas.
- Derivadas.
- Integrales
- Ecuaciones diferenciales.

Instalar

Para instalar Sympy, sólo debes abrir la consola y...

Instalar

```
pip install sympy
```

Si presentas problemas, esto podría funcionar...

¿Problemas?

```
python -m pip install sympy
```

Por si acaso...

¿Desesperado?

```
python -m pip install sympy --user
```

O...

Porque valoramos tu vida

```
python3 -m pip install sympy --user
```

¿Instalar? (paréntesis)

Es hora de avanzar al siguiente nivel: muchos de los trabajos que encontremos en GitHub emplean librerías que, posiblemente, no tengamos preinstaladas. O incluso, cuando queramos **compartir** a nuestros compañeros, profesores o para publicar nuestro trabajo *online*, debemos saber **cómo informarles** cuáles librerías necesitan para poder correr nuestro software.

En el mundo del *código abierto*, es común encontrar simples ficheros de texto (requirements.txt) que contienen las librerías necesarias para correr el software.

Un ejemplo del contenido de *requirements.txt*:

```
attrs==19.1.0
backcall==0.1.0
bleach==3.1.0
cyclr==0.10.0
decorator==4.4.0
defusedxml==0.6.0
```

...

¿Instalar? - requirements.txt (paréntesis)

Cuando queramos instalar las librerías empleadas por otros, es simplemente encontrar el fichero, copiar la dirección y:

Instalar

```
pip install -r requirements.txt
```

Listo!

¿Instalar? - virtualenv (paréntesis) I

Cuando queramos **compartir** software, debemos generar nuestro propio fichero. Obviamente, no vamos a escribir manualmente las librerías que utilizamos. Vamos a hacer que Python lo haga por nosotros, para ello (desde la dirección del software):

1. Instalamos virtualenv.

Virtualenv ^{Primera vez}

```
pip install virtualenv
```

2. Generamos nuestro propio “ambiente virtual”.

Generación del ambiente ^{Primera vez*}

```
virtualenv env
```

```
(si no funciona...) python -m venv env
```

¿Instalar? - virtualenv (paréntesis) II

3. Entramos al ambiente:

Entrar

```
.\env\Scripts\activate
```

4. Instalamos las librerías manualmente:

Instalamos...

```
pip install <librerías de nuestra aplicación>
```

5. Generamos el archivo.

Archivo .txt

```
pip freeze > requirements.txt
```

6. Salimos del ambiente virtual.

Salir

```
deactivate
```


Empecemos

A partir de aquí, el aprendizaje será didáctico. Abre la carpeta “Jupyter”(en la dirección de esta presentación), copia el enlace y:

En la dirección...

```
jupyter lab
```