



Entornos

Programación II
Grado en Business Analytics



¿Por qué trabajar con entornos?



¿Por qué trabajar con entornos?

Por defecto, pip instala todos los paquetes en un entorno compartido a todos los proyectos.



¿Por qué trabajar con entornos?

¿Qué pasa si tengo proyectos que usan diferentes versiones de módulos o paquetes para su funcionamiento?



¿Por qué trabajar con entornos?

¿Qué pasa si tengo proyectos que usan diferentes versiones de módulos o paquetes para su funcionamiento?

Proyecto1: tf1.0

Proyecto2: tf2.x

Un entorno es un mecanismo que va a facilitar aislar las dependencias de un proyecto de python a nivel de librerías e intérpretes.

- Cada entorno puedes pensarlo como un contenedor en tiempo de desarrollo.



¿Cómo gestionar entornos?

Aunque existen varios gestores de entornos, algunos muy potentes como Anaconda o POETRY, nosotros utilizaremos **venv**

`https://docs.python.org/es/3/library/venv.html`


```
>> virtualenv /ruta/de/proyecto/.env
```

```
>> source .env/bin/activate
```

Creo un entorno

**Lo activo: ahora cuando instale
paquetes**

**Con pip estos paquetes estarán
instalados dentro de mi entorno**

```
>> deactivate
```

```
>> rm -rf /ruta/de/proyecto/.env
```

Salgo del entorno

Borro el entorno

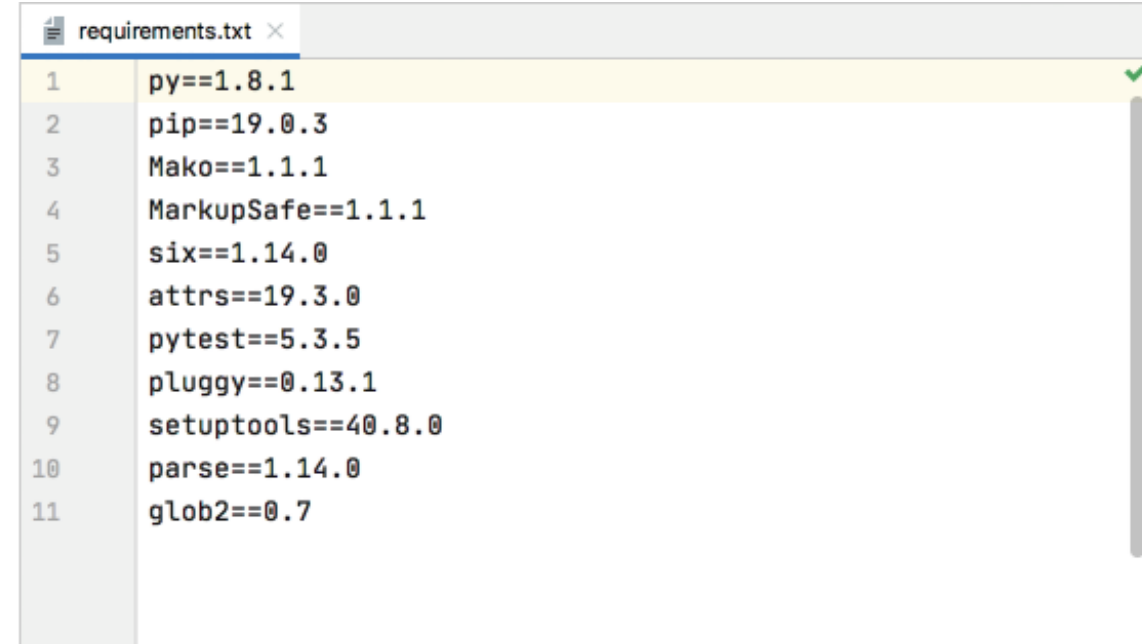


Requirements file

Los paquetes y sus versiones específicas se definen en un fichero que, por convención, denominaremos `requirements.txt`

Sobre el directorio en el que esté el fichero:

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



```
requirements.txt x
1 py==1.8.1 ✓
2 pip==19.0.3
3 Mako==1.1.1
4 MarkupSafe==1.1.1
5 six==1.14.0
6 attrs==19.3.0
7 pytest==5.3.5
8 pluggy==0.13.1
9 setuptools==40.8.0
10 parse==1.14.0
11 glob2==0.7
```



Entornos

Programación II
Grado en Business Analytics