

HINTS

- Los comandos de administración son scripts potentes y flexibles, que pueden realizar acciones en su proyecto Django o en la base de datos subyacente.
- En comparación con los scripts de Python normales, el uso del marco de comandos de administración significa que un trabajo de configuración tedioso se realiza automáticamente.

OBSERVACIONES

- Los comandos de gestión se pueden llamar desde:
 - `django-admin <command> [options]`
 - `python -m django <command> [options]`
 - `python manage.py <command> [options]`
 - `./manage.py <command> [options]`
- Para utilizar comandos de gestión con **Cron**:

```
1 */10 * * * * pythonuser /var/www/dev/env/bin/python /var/www/dev/manage.py  
2 <command> [options] > /dev/null
```

- Para obtener la lista de comandos existentes:

```
1 >>> python manage.py help
```

- Si no entiende ningún comando, o busca argumentos opcionales, puede usar el argumento `-h` como éste:

```
1 >>> python manage.py command_name -h
```

- Comandos de gestión incorporados para obtener una lista de todos los comandos disponibles:

```
1 ./manage.py help
```

MEJORES PRÁCTICAS PARA ADMINISTRAR ENTORNOS DE PYTHON

- Usar un entorno virtual: siempre se deben usar, a menos que se emplee Python para la administración del servidor o para ejecutar scripts simples. Hay muchos administradores de paquetes de Python, y entornos virtuales diferentes para elegir, los cuales vienen con una variedad de características. Es importante considerar cuidadosamente el mejor entorno virtual para su proyecto de Python.
 - **Virtualenv**: es posiblemente el entorno virtual de Python más popular y fácil de usar para principiantes. Viene preinstalado con Python 2 como **virtualenv**, y Python 3 como **venv**. La popular extensión **virtualenvwrapper** agrega características adicionales, tales como la finalización de pestañas y un solo comando para cambiar entre entornos.
 - **Pipenv**: es una herramienta de administración de paquetes y entornos virtuales que tiene como objetivo integrar la funcionalidad de Pip y **virtualenv** en una sola herramienta. Posee funciones muy útiles, pero es más complejo de aprender que **virtualenv**.
 - **Poetry**: es una herramienta de Python rica en características para la gestión de dependencias de proyectos. Es más rápido que la mayoría de las herramientas de entorno virtual, y viene con una poderosa CLI para administrar proyectos de Python.
 - **Conda**: es un sistema de administración de entornos y paquetes multipropósito, el cual admite Python y otros lenguajes como Ruby, Scala, R y C/C++. Se utiliza para crear, guardar, cargar y cambiar entre entornos en su máquina local. Es el favorito de los científicos de datos, y viene preinstalado tanto en anaconda como en miniconda.
- Usar archivos **requirements.txt**: la mejor manera de hacer que su trabajo sea reproducible, y de mantener la coherencia de su entorno es incluir un archivo **requirements.txt** en el directorio raíz de su proyecto. Estos contienen una lista de todos los paquetes presentes en un proyecto. Su uso se puede hacer en dos simples pasos:
 - Use **PIP Freeze** para generar paquetes instalados adecuados para un archivo de requisitos.
 - Los archivos **requirements.txt** ayudan a garantizar la coherencia entre instalaciones, implementaciones y desarrolladores.
- Use un entorno virtual separado para cada proyecto.

- Idealmente, debería tener un nuevo entorno virtual para cada proyecto basado en Python en el que trabaje. El objetivo principal de esto es mantener las dependencias de cada proyecto aisladas, tanto del sistema, como entre sí.
- Si tiene varios proyectos que tienen aproximadamente los mismos requisitos, puede parecer una buena idea crear un único entorno virtual en el que ambos proyectos puedan compartir. El problema con esto es que uno de los proyectos podría tener repentinamente requisitos que rompan el otro. El objetivo de los entornos virtuales es aislar cada proyecto de otros proyectos y sus peculiaridades.
- El espacio en disco y la conveniencia ahorrados son marginales, y simplemente no valen la pena. Además, el uso de archivos requirements.txt facilita la configuración de un entorno virtual para un proyecto, e instala lo que necesita con un par de comandos.