🡪

什么是Pipenv?

[Pipenv](https://link.zhihu.com/?target=https%3A//docs.pipenv.org/)是Kenneth Reitz在2017年1月发布的Python依赖管理工具，现在由PyPA维护。你可以把它看做是pip和virtualenv的组合体，而它基于的[Pipfile](https://link.zhihu.com/?target=https%3A//github.com/pypa/pipfile)则用来替代旧的依赖记录方式（requirements.txt）。

Pipenv会自动帮你管理虚拟环境和依赖文件，并且提供了一系列命令和选项来帮助你实现各种依赖和环境管理相关的操作。简而言之，它更方便、完善和安全

🡪

Pipenv与virtualenv对比：

1.

Virtualenv需要我们创建一个虚拟环境，然后再激活activate进入该虚拟环境。

而pipenv在执行命令时，如果虚拟环境没有被激活，则会默认激活并且执行命令。如果虚拟环境不存在，则会自动为我们创建虚拟环境并激活，不需要手动调节。

2.

Pip采用requirements.txt记录自己的依赖，但是当我们安装新的包或者卸载包以后需要手动执行pip freeze更新requirements.txt。如果要区分development和production环境还要手动操作。

Pipenv自动帮你管理依赖。Pipenv会在你创建虚拟环境时自动创建Pipfile和Pipfile. lock文件（如果不存在），并且会在你使用pipenv install和pipenv uninstall命令安装和卸载包时自动更新Pipfile和Pipfile.lock。并且执行pipenv install时加入--dev修饰，则会特别的写入Pipfile的dev-packages一节中进行区分。

🡪

常用命令：

pipenv install [python包名] # 在虚拟环境中使用pip安装指定的python包

pipenv install # 根据目录下的Pipfile中的包进行安装

pipenv run [命令] # 在python虚拟环境中运行命令。从而可以不显示进入python虚拟环境直接让pipenv帮我们把命令带到python虚拟环境中执行。

pipenv shell # 激活并且进入python虚拟环境

pipenv graph # 显示依赖图示

pipenv lock # 锁定版本

例子：

我们使用pipenv install [Flask]的时候，是pipenv自动在python虚拟环境中安装Flask python包。安装完成，只有该python虚拟环境才有该Flask包。所以pipenv run pip list installed 可以看到Flask包安装。而直接pip list installed则Flask包不会存在。

🡪