**Linux下virtualenv的使用与pycharm的基本配置**

Virtualenv（更详细的使用参照官方文档https://virtualenv.pypa.io/en/stable/）

创建一个virtualenv环境

不包含系统的python包，新的环境里面只有pip, setuptools和wheel这些包，许多包要用pip重新安装：

virtualenv + 路径

也可以指定python版本

Virtualenv –p python3 +路径

包含系统的python包，系统中安装过的包可以在新的环境中直接使用不用重新安装：

virtualenv --system-site-packages + 路径

激活virtualenv环境

source 路径/bin/activate

激活只对当前终端有效，如果新打开了一个终端的话，重新运行上面的命令。

激活后终端前面会多一个(\*\*\*\*)的东西，提示当前virtualenv的名称。

激活后可以在当前终端通过python + 文件名.py的方式运行python脚本，如果脚本中使用了当前环境中没有的包，而且没有使用“—system-site-packages”的话，将会报错, 可以在激活环境后使用pip安装对应的包。注意不用使用sudo，因为包不会安装到系统当中去，而是安装到了当前的virtualenv对应的目录中。

退出virtualenv环境：

deactivate

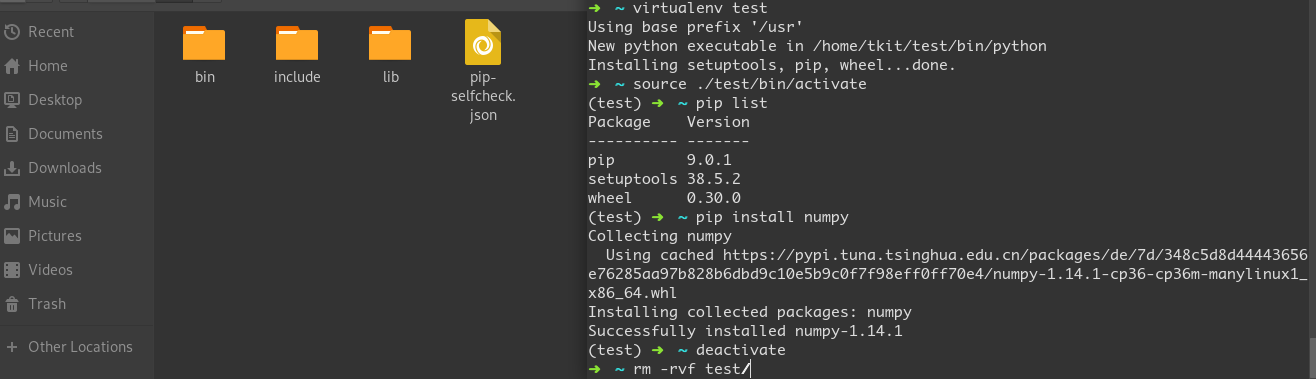
或者直接关闭当前终端

删除virtualenv环境

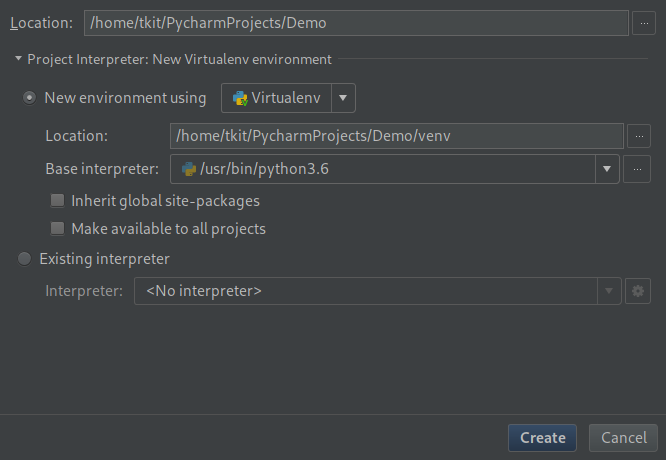
直接删除对应目录即可删除virtualenv环境，不会对系统产生任何影响。所以在virtualenv中可以放心操作

rm -rvf + 路径

可参考下图的操作



pycharm配置



New environment using Virtualenv:

将在项目的目录下创建一个virtualenv环境，然后使用它当作当前项目的python解释器，默认不包含系统的python包。相当于：

virtualenv + 路径

location:

为新建的环境的位置，默认为当前工程下的venv。

Base interpreter:

基于系统中的python版本，新建的环境中的python版本与此一致，可以选择python2或者python3, 取决于项目的需要，相当于virtualenv –p python版本 +路径。

勾选Inherit global site-packages，包含系统的python包，相当于：

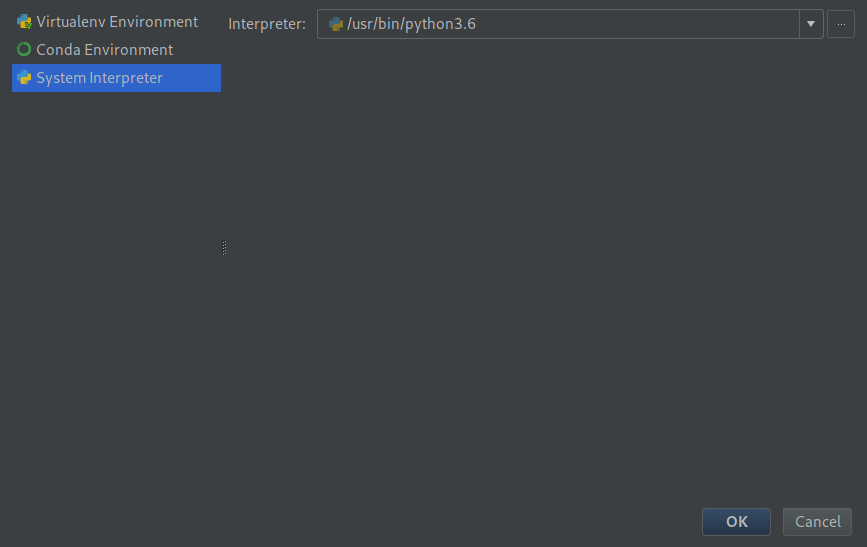
virtualenv --system-site-packages + 路径

勾选Make available to all projects，下次新建项目的时候会在Existing interpreter中找到这个环境， 可以重复使用这个环境。

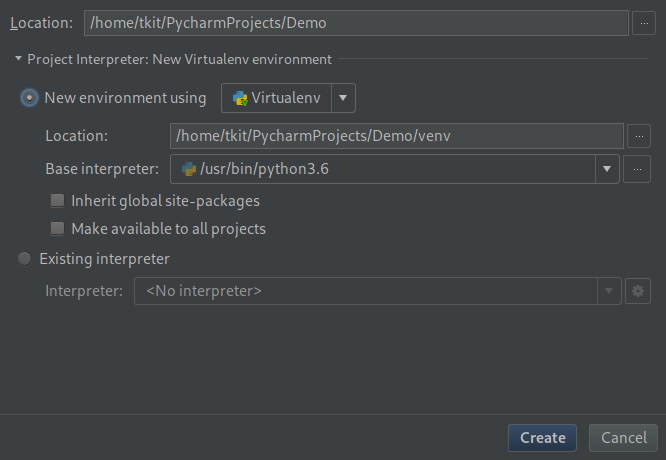
Existing interpreter

使用已有的python环境，点击后会出现后面的设置会出现这个界面，分别是virtualenv, conda和系统的python环境。可以选择已有的virtualenv环境，或者直接使用系统的python解释器。

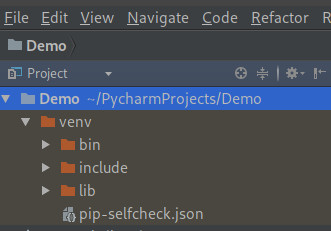
Conda是anaconda(一个科学计算的python发行版)的包管理器，也可以用来建立python环境。



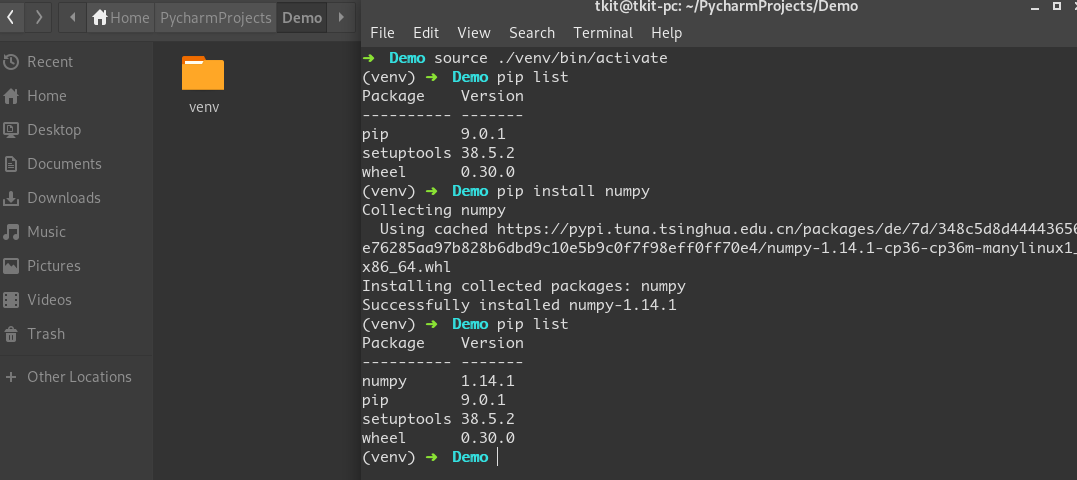
使用如下的配置新建一个项目：



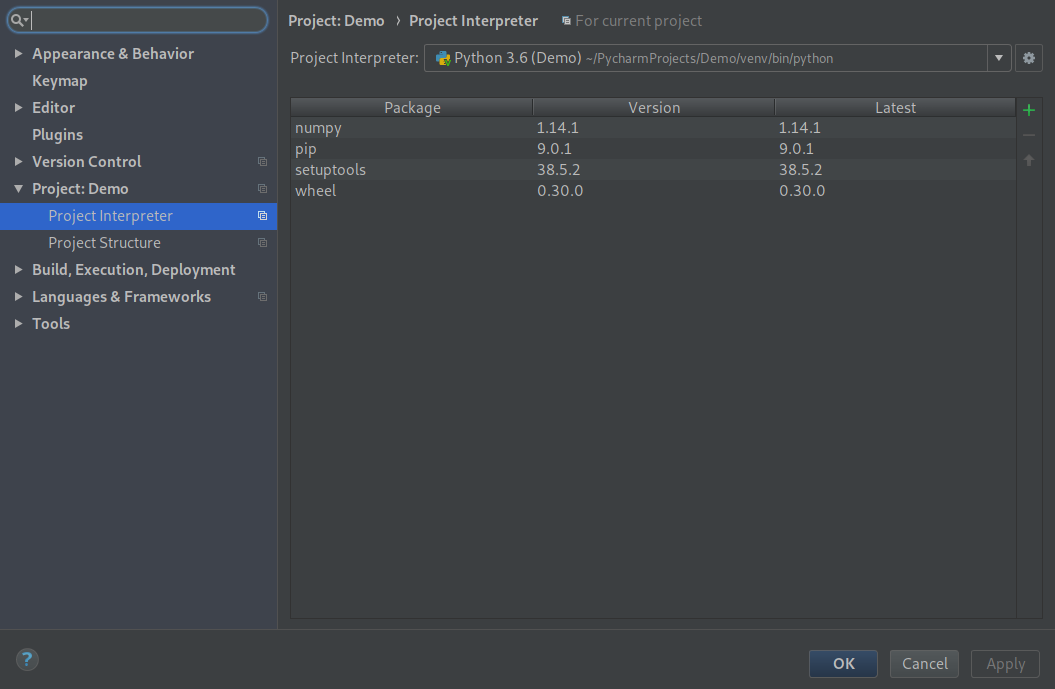
会发现生成的项目中有一个叫venv的文件夹，它实质上和直接用virtualenv创建的一样。



可以用virtualenv的管理方法管理它，比如安装numpy，安装之后可以在pycharm正常使用。（注意在virtualenv中不要使用sudo）



也可以在pycharm中使用 file🡪settings🡪project🡪project interpreter中管理环境中的python包，可以对该环境下的python包进行删除和安装。



如果你系统环境的python损坏了，可以试着用virtualenv或者pycharm新建一个新的不包含系统包的python环境，然后在这个环境下安装新的python包，以后在pycharm中都用这个环境，或者在终端通过source +路径的方式激活该环境，然后运行程序，应该可以正常运行。