pyenv管理python多版本和jupyter的使用

1、使用pyenv来管理Python环境

仅用于开发环境，生产环境都是单一python环境

```
sudo yum -y install git
curl -L https://raw.githubusercontent.com/yyuu/pyenv-installer/master/bin/pyenv-installer | bash
```

安装完`pyenv-installer`之后就会让配置环境变量

环境变量:`vi ~/.bash_profile`

```
export PATH="/home/clg/.pyenv/bin:$PATH"
eval "$(pyenv init -)"
eval "$(pyenv virtualenv-init -)"
```

环境变量加入完成之后生效

```
source ~/.bash_profile
```

2、安装python

centos版本

安装编译工具

```
sudo yum -y install gcc make patch
```

安装依赖项

```
sudo yum -y install gdbm-devel openssl-devel sqlite-devel readline-devel zlib-devel bzip2-devel
```

pyenv安装python

如果网络不行install下载包的时候一直卡主，可以先下载源码包.tar.xz的，然后放在~/.pyenv/cache目录下，cache需要手工创建，然后执行下面命令

```
pyenv install 3.5.2
```

pyenv查看所有的python版本：其中system就是系统自带的

```
[clg@Pracl ~]$ pyenv versions
* system (set by /home/clg/.pyenv/version)
  3.5.2
```

使用pyenv安装的python都在以下目录中进行管理，并且只对当前用户有用，换了用户之后可以在该用户下在安装一套pyenv。

```
[clg@Pracl ~]$ cd ~/.pyenv/versions/;ls
3.5.2
```

3、pyenv的常见使用

pyenv local的使用

```
[clg@Pracl ~]$ python -V
Python 2.7.5
[clg@Pracl ~]$ ^C
[clg@Pracl ~]$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[clg@Pracl ~]$ mkdir workspace
[clg@Pracl ~]$ ls
Desktop    Downloads  Pictures  Templates  workspace
Documents  Music      Public    Videos
[clg@Pracl ~]$ cd workspace/
ls
[clg@Pracl workspace]$ ls
[clg@Pracl workspace]$ pyenv local 3.5.2
[clg@Pracl workspace]$ python -V
Python 3.5.2
[clg@Pracl workspace]$ ls
[clg@Pracl workspace]$ cd ..
[clg@Pracl ~]$ python -V
Python 2.7.5
[clg@Pracl ~]$ cd workspace/
[clg@Pracl workspace]$ ls
[clg@Pracl workspace]$
```

子目录会继承父目录的版本

```
[clg@Pracl workspace]$ python -V
Python 3.5.2
[clg@Pracl workspace]$ mkdir subworkspace
[clg@Pracl workspace]$ cd subworkspace/
[clg@Pracl subworkspace]$ python -V
Python 3.5.2
```

版本恢复

```
[clg@Pracl subworkspace]$ pyenv version
3.5.2 (set by /home/clg/workspace/.python-version)
[clg@Pracl subworkspace]$ pyenv local system
[clg@Pracl subworkspace]$ pyenv version
system (set by /home/clg/workspace/subworkspace/.python-version)
[clg@Pracl subworkspace]$ ll -a
total 4
drwxrwxr-x. 2 clg clg 28 Dec  3 15:48 .
drwxrwxr-x. 3 clg clg 47 Dec  3 15:15 ..
-rw-rw-r--. 1 clg clg  7 Dec  3 15:48 .python-version
[clg@Pracl subworkspace]$ vi .python-version
```

可以发现.python-version文件中写入的是system，如果需要恢复版本，直接删除这个文件就行

```
[clg@Prac1 subworkspace]$ vi .python-version
[clg@Prac1 subworkspace]$ rm -rf .python-version
[clg@Prac1 subworkspace]$ pyenv
pyenv 1.0.4
Usage: pyenv <command> [<args>]

Some useful pyenv commands are:
   commands    List all available pyenv commands
   local       Set or show the local application-specific Python version
   global      Set or show the global Python version
   shell       Set or show the shell-specific Python version
   install     Install a Python version using python-build
   uninstall   Uninstall a specific Python version
   rehash      Rehash pyenv shims (run this after installing executables)
   version     Show the current Python version and its origin
   versions    List all Python versions available to pyenv
   which       Display the full path to an executable
   whence      List all Python versions that contain the given executable

See `pyenv help <command>' for information on a specific command.
For full documentation, see: https://github.com/yyuu/pyenv#readme
[clg@Prac1 subworkspace]$ pyenv version
3.5.2 (set by /home/clg/workspace/.python-version)
```

删除之后继续继承父目录的版本

global的使用：绝对不要使用global

4、python第三方包管理：pyenv的虚拟环境

用途：隔离项目间使用的第三包。每个项目最好创建一个虚拟环境。

每个项目可能会依赖相同的第三方包，但是第三方包的版本可能不太一样

```
1.  python virtualenv 3.5.2 ShanghaiMonitor
```

即创建了一个名为ShanghaiMonitor的虚拟环境

```
[clg@Prac1 ~]$ pyenv virtualenv 3.5.2 ShanghaiMonitor
Ignoring indexes: https://pypi.python.org/simple
Requirement already satisfied (use --upgrade to upgrade): setuptools in /home/clg/.pyenv/versions/3.5.2/envs/ShanghaiMonitor/lib/python3.5/site-packages
Requirement already satisfied (use --upgrade to upgrade): pip in /home/clg/.pyenv/versions/3.5.2/envs/ShanghaiMonitor/lib/python3.5/site-packages
[clg@Prac1 ~]$ cd ~/.pyenv/
bin/        cache/      completions/ .git/       libexec/    plugins/    pyenv.d/    shims/      src/        test/       versions/
[clg@Prac1 ~]$ cd ~/.pyenv/versions/
2.6.6/          2.7.12/         3.5.2/          ShanghaiMonitor/
[clg@Prac1 ~]$ cd ~/.pyenv/versions/
[clg@Prac1 versions]$ ls
2.6.6  2.7.12  3.5.2  ShanghaiMonitor
[clg@Prac1 versions]$ pyenv versions
* system (set by /home/clg/.pyenv/version)
  2.6.6
  2.7.12
  3.5.2
  3.5.2/envs/ShanghaiMonitor
  ShanghaiMonitor
[clg@Prac1 versions]$
```

关于virtualenv的解释如下：

```
[clg@Pracl versions]$ pyenv versions
* system (set by /home/clg/.pyenv/version)
  2.6.6
  2.7.12
  3.5.2
  3.5.2/envs/ShanghaiMonitor
  ShanghaiMonitor
[clg@Pracl versions]$ ls
2.6.6  2.7.12  3.5.2  ShanghaiMonitor
[clg@Pracl versions]$ ll
total 0
drwxr-xr-x. 6 clg clg 52 Dec  3 15:08 2.6.6
drwxr-xr-x. 6 clg clg 52 Dec  3 15:12 2.7.12
drwxr-xr-x. 7 clg clg 63 Dec  3 15:35 3.5.2
lrwxrwxrwx. 1 clg clg 52 Dec  3 15:35 ShanghaiMonitor -> /home/clg/.pyenv/versions/3.5.2/envs/ShanghaiMonitor
[clg@Pracl versions]$
```

在3.5.2里面创建了一个虚拟环境，版本是3.5.2/envs/ShanghaiMonitor，然后在versions目录下创建了一个软链接ShanghaiMonitor

卸载虚拟环境的方法：

```
[clg@Pracl versions]$ pyenv --help
Usage: pyenv <command> [<args>]

Some useful pyenv commands are:
   commands    List all available pyenv commands
   local       Set or show the local application-specific Python version
   global      Set or show the global Python version
   shell       Set or show the shell-specific Python version
   install     Install a Python version using python-build
   uninstall   Uninstall a specific Python version
   rehash      Rehash pyenv shims (run this after installing executables)
   version     Show the current Python version and its origin
   versions    List all Python versions available to pyenv
   which       Display the full path to an executable
   whence      List all Python versions that contain the given executable

See `pyenv help <command>' for information on a specific command.
For full documentation, see: https://github.com/yyuu/pyenv#readme
[clg@Pracl versions]$ pyenv uninstall ShanghaiMonitor
pyenv-virtualenv: remove /home/clg/.pyenv/versions/3.5.2/envs/ShanghaiMonitor? y
[clg@Pracl versions]$
```

5、安装ipython

```
pip install --upgrade pip
pip install ipython
```

ipython：Python交互shell的增强工具，作为一个帮助文档和帮助工具

6、安装jupyter

```
pip install jupyter
jupyter notebook --ip=0.0.0.0 --no-browser

#后台运行
> nohup.out&&nohup jupyter notebook --ip=0.0.0.0 --no-browser &
```

安装完成之后需要关闭防火墙才能在浏览器打开：

```
1.  iptables -F && setenforce 0
```

浏览器地址：ip/8888