# **1安装pyenv**

1、安装git

$ yum install git curl -y

2、python编译依赖如下

$ yum install -y gcc make patch gdbm-devel openssl-devel sqlite-devel readline-devel zlib-devel bzip2-devel

3、创建普通用户

$ useradd python

4、切换到python用户：

$ su - python

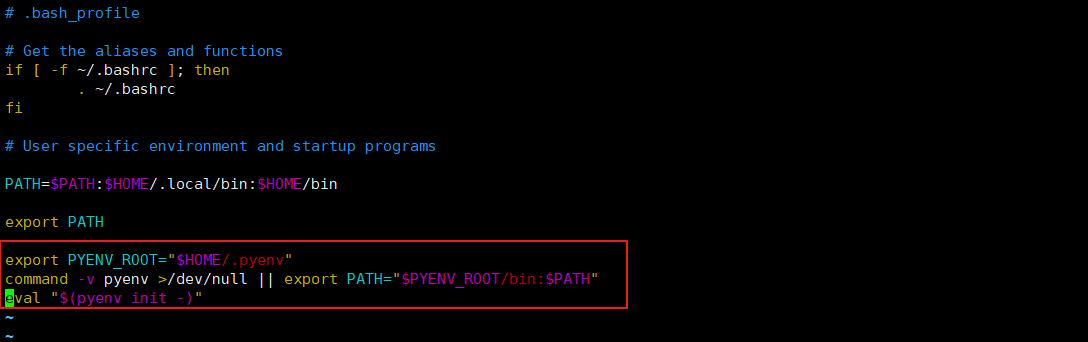
$ curl -L https://github.com/pyenv/pyenv-installer/raw/master/bin/pyenv-installer | bash

5 编辑~/.bash\_profile

$ vim ~/.bash\_profile

添加如下内容

|  |
| --- |
| export PYENV\_ROOT="$HOME/.pyenv"  command -v pyenv >/dev/null || export PATH="$PYENV\_ROOT/bin:$PATH"  eval "$(pyenv init -)" |



$ source ~/.bash\_profile

这样当用户启动的时候，会执行用户的.bash\_profile中的脚本，就会启动pyenv。 安装好的pyenv

就在~/.pyenv中

$ pyenv

# **2 pyenv缓存**

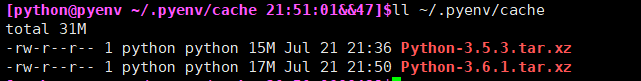
在pyenv中安装各个版本，可以要使用pyenv安装，由于连接国外网址比较慢，所以可以利用pyenv的包缓存，提前下载安装包，然后直接安装。

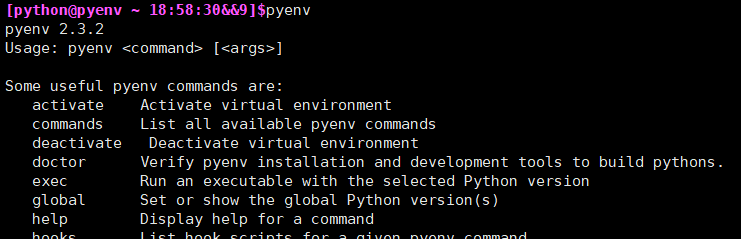
# 缓存目录

~/.pyenv/cache

此cache目录不存在，要自行创建。

mkdir ~/.pyenv/cache





安装python多版本

pyenv install -l       # 列出所有可以安装的版本

pyenv install 3.5.3 -vvv         # -vvv 列出安装过程，安装时会先去缓存文件中找安装包，找不到再去下载

pyenv version   # 查看当前的python版本

pyenv versions  # 查看所有安装的python版本

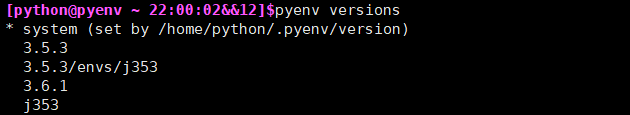
# **3.0 pyenv使用**

### 3.0.1 pyenv 虚拟环境

pyenv 可以将一个python版本及相关库封装成一个独立的“虚拟环境”，然后将这个虚拟环境应用到某个目录，此目录及其子目录就会继承这个虚拟环境，单个虚拟环境的配置不会影响到其他版本的虚拟环境。

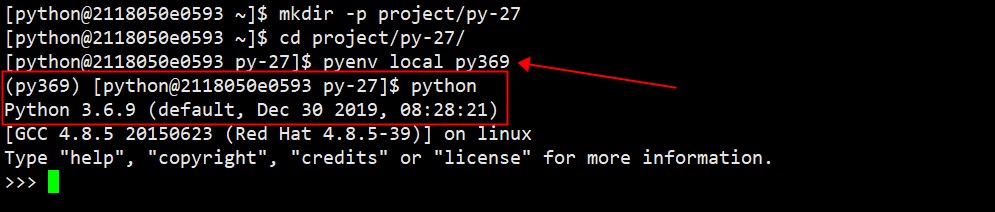
$pyenv virtualenv 3.5.3 j353  # 先创建一个虚拟环境，j353是别名

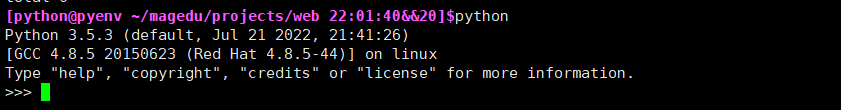




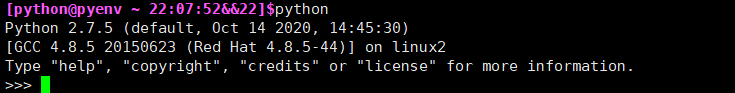
然后切换到项目目录，应用这个j353的环境

$pyenv local j353





可以看到版本已经切换到了 3.5.3，这个python版本只对此目录以及其子目录生效，再目录外面，python版本依然是 系统默认的python版本



这种虚拟环境可以同时创建多个，互不影响。

### **3.0.2 删除指定virtualenv（三种方法）**

**法一：删除文件**

直接删除$(pyenv root)/versions和$(pyenv root)/versions/{version}/envs中的指定目录

**法二：使用uninstall命令**

pyenv uninstall <名称>

**法三：使用virtualenv-delete命令**

pyenv virtualenv-delete <名称>

# **4.0 pip 包管理工具**

pip是Python包管理器，以后安装Python第三方包都需要它，它从3.x开始就集成在Python安装包里面了。 相当于yum。

查看pip版本，可以看到这个版本只是再j353 这个虚拟环境中的版本。

### 4.0.1 pip 切换国内源



**修改pip配置文件**

windows配置文件： ~/pip/pip.ini 。windows家目录，在“运行”中键入 .

Linux配置文件： ~/.pip/pip.conf

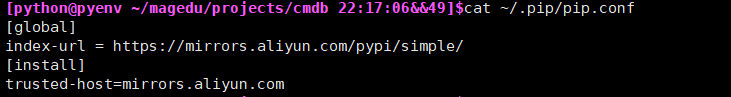
内容，可参照 [http://mirrors.aliyun.com的pypi帮助](http://mirrors.aliyun.xn--compypi-0m1l996f447c/)

$mkdir ~/.pip

$vim ~/.pip/pip.conf

|  |
| --- |
| [global]  index-url = <https://mirrors.aliyun.com/pypi/simple/>  [install]  trusted-host=mirrors.aliyun.com |

cat ~/.pip/pip.conf



更新pip

pip install --upgrade pip

更新setuptools

pip install --upgrade setuptools

pip install redis  #在不同的虚拟环境中，安装redis包，使用pip list看看效果。

 pip install pkgname 命令，是以后经常要使用的安装python包的命令

**ipython** 是增强的交互式Python命令行工具

$ pip install ipython

$ ipython

**Jupyter** 是基于WEB的交互式笔记本，其中可以非常方便的使用Python。 安装Jupyter，也会安装

ipython的

$ pip install jupyter

$ jupyter notebook help

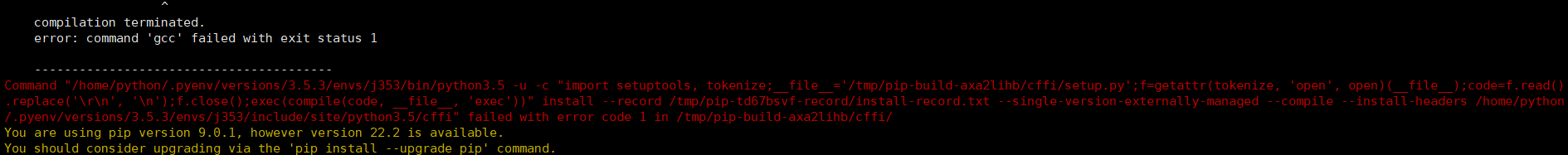
$jupyter notebook password

$ jupyter notebook --ip=0.0.0.0 --no-browser

$ ss -tanl

**报错**

error: command 'gcc' failed with exit status 1



**解决办法**

yum install gcc libffi-devel python-devel openssl-devel -y

<https://www.hugbg.com/archives/1406.html>