

Python 学习环境的搭建_Linux 篇

前言.....	2
安装/升级 Python 库.....	2
Vim 版本升级	10
安装 Jedi-vim 插件	14
初始化 Vim for Python 开发环境.....	15
项目创建和测试.....	17

本文介绍 Python 学习环境的搭建。

Python 在 Linux 下的开发环境无法做到像 windows 下那么傻瓜化,但是却非常灵活。Linux 平台软件遵从了 Linux 极简实用主义的哲学。全部采用内核+插件的可插拔安装形式。需要什么安装什么, 非常透明。

Linux 下的 Python 开发环境亦是如此。可以选用任何自己顺手的编辑器进行开发, 比如 Linux 自带的编辑器 Vim, gredit, 其他如 emacs 等。辅之于各种插件, 熟悉之后, 效率相比 win 下有过之而无不及。

搭建环境:

OS: CentOS 6.5 X86_64

Python IDE: Vim7.4+Jedi-vim 插件

Python Library: Python2.7.8

SSH 工具: putty

前言

Linux 平台的环境搭建, 需要掌握一些基础的 Linux 操作命令。下面的安装过程有些繁琐, 请保持耐心。

CentOS6.5 自带的 Python 库为 2.6.6, 实际上这并不影响对 python 的学习, 可以直接使用。

```
drwxr-xr-x. 2 root 4096 May 28 07:05 videos
drwxr-xr-x. 2 yangshuang yangshuang 4096 Jun  3 02:58 Yang
[root@YangShuang_CentOS ~]# vim
[root@YangShuang_CentOS ~]# python
Python 2.6.6 (r266:84292, Nov 22 2013, 12:16:22)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> c
```

如果不在乎这些, 并且对 Linux 下 Python 环境和开发环境的搭建无兴趣, 请点右上角的 X。

24 服务器上 Python in Linux 环境已经搭好, 有兴趣体验的用 ssh 登录工具登录到 24 即可。
ssh 客户端可选择 putty 或者 xshell 等工具。

服务器地址: 192.168.10.24

用户名: canyou

密码: 123456

对环境搭建有兴趣的继续往下看。

安装/升级 Python 库

升级 Python library 到 2.7.8

因 Python 官方无 RPM 安装包放出, 并且国内多数 centos 镜像站的 yum 源, 最新版本仍然

是 2.6.6,Centos 默认的 python 环境已经安装好了所有常见依赖包，python 升级，对应的扩展库版本有可能也要做升级。若对 python 库升级感兴趣继续往下看，不感兴趣可跳过此节。

```
[root@YangShuang CentOS ~]# yum update python
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
Setting up Update Process
Resolving Dependencies
--> Running transaction check
--> Package python.x86_64 0:2.6.6-51.el6 will be updated
--> Package python.x86_64 0:2.6.6-52.el6 will be an update
--> Processing Dependency: python-libs(x86-64) = 2.6.6-52.el6 for package: python-2.6.6-52.el6.x86_64
--> Running transaction check
--> Package python-libs.x86_64 0:2.6.6-51.el6 will be updated
--> Package python-libs.x86_64 0:2.6.6-52.el6 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch             Version           Repository         Size
=====
Updating:
python                 x86_64           2.6.6-52.el6      updates            74 k
Updating for dependencies:
python-libs            x86_64           2.6.6-52.el6      updates            5.3 M
=====
```

所以，升级 Python 库采用源码编译安装。

Python2.7.8 源码下载链接

<https://www.python.org/ftp/python/2.7.8/>

Google 鸟哥的 Linux 私房菜 ... RPM resource libnss... 应该知道的Linux技巧 ... 28个Unix/Linux的命... crz

Index of /ftp/python/2.7.8/

../	30-Jun-2014 02:41	10525244
Python-2.7.8.tar.xz	30-Jun-2014 03:58	836
Python-2.7.8.tar.xz.asc	30-Jun-2014 02:42	14846119
Python-2.7.8.tgz	30-Jun-2014 03:58	836
Python-2.7.8.tgz.asc	30-Jun-2014 03:18	20772301
python-2.7.8-macosx10.3.dmg	30-Jun-2014 03:24	473
python-2.7.8-macosx10.3.dmg.asc	30-Jun-2014 03:41	20469575
python-2.7.8-macosx10.5.dmg	30-Jun-2014 03:44	473
python-2.7.8-macosx10.5.dmg.asc	30-Jun-2014 04:13	20370972
python-2.7.8-macosx10.6.dmg	30-Jun-2014 05:41	473
python-2.7.8-macosx10.6.dmg.asc	01-Jul-2014 00:18	25355330
python-2.7.8-pdb.zip	01-Jul-2014 00:17	23561282
python-2.7.8.amd64-pdb.zip	30-Jun-2014 23:58	17231872
python-2.7.8.amd64.msi	01-Jul-2014 00:11	16703488
python278.chm	01-Jul-2014 00:18	6062806

光标移向 Python-2.7.8.tgz 链接，右键复制下载链接。

使用 root 账户登录到 Linux 服务器，并切换到/usr/local/src 目录

```
cd /usr/local/src/
```

执行命令 `wget https://www.python.org/ftp/python/2.7.8/Python-2.7.8.tgz`
`--no-check-certificate`

下载源码包，`wget` 后跟的是上一步 `tgz` 压缩包的连接地址。`--no-check-certificate` 是不进行安全验证，`no` 前跟两个减号。

解压 `python2.7.8.tgz` 压缩包

在原目录执行以下命令 `tar -zxvf Python-2.7.8.tgz`

```
Python-2.7.8/Misc/RPM/
Python-2.7.8/Misc/RPM/README
Python-2.7.8/Misc/RPM/python-2.7.spec
Python-2.7.8/Misc/valgrind-python.supp
Python-2.7.8/Misc/README.klocwork
[root@YangShuang_CentOS src]# ll
total 14504
drwxr-xr-x 17 1000 1000      4096 Jun 30 10:05 Python-2.7.8
-rw-r--r--  1 root    0 14846119 Jun 30 10:42 Python-2.7.8.tgz
[root@YangShuang_CentOS src]#
```

创建新 Python 库的安装目录。执行以下指令

```
mkdir /usr/local/Python2.7.8
```

执行命令 `cd /usr/local/src/Python-2.7.8`，进入解压缩的源码包

使用 `pwd` 命令确认当前所在工作目录。

```
[root@YangShuang_CentOS src]# cd /usr/local/src/Python-2.7.8
[root@YangShuang_CentOS Python-2.7.8]# pwd
/usr/local/src/Python-2.7.8
[root@YangShuang_CentOS Python-2.7.8]#
```

编译 Python2.7.8 源码，执行以下指令

```
./configure --prefix=/usr/local/Python2.7.8
```

```
checking for build directories... done
configure: creating ./config.status
config.status: creating Makefile.pre
config.status: creating Modules/Setup.config
config.status: creating Misc/python.pc
config.status: creating Modules/ld_so_aix
config.status: creating pyconfig.h
creating Modules/Setup
creating Modules/Setup.local
creating Makefile
[root@YangShuang_CentOS Python-2.7.8]# ./configure --prefix=/usr/local/Python2.7.8
```

--prefix 指定我们创建的新 Python 库的安装目录。

程序开始编译，时间持续 20s 左右。如果编译成功，会生成安装脚本，并且不会有任何异常抛出。如上图，编译成功。

执行此条指令，可能会出现找不到 gcc 编译器。

保证机器联网，使用 `yum install gcc` 安装编译器，重新执行以上编译过程即可。

执行 `pwd` 命令，确认当前目录为源码解压包

```
[root@YangShuang_CentOS Python-2.7.8]# pwd
/usr/local/src/Python-2.7.8
[root@YangShuang_CentOS Python-2.7.8]#
```

执行命令 `make && make install`

安装 python 库，你将看到安装过程如下。整个安装过程时间持续 2 分钟左右

```
gcc -pthread -c -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapv -O3 -Wall -W
-DPy_BUILD_CORE -o Objects/longobject.o Objects/longobject.c
gcc -pthread -c -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapv -O3 -Wall -W
-DPy_BUILD_CORE -o Objects/dictobject.o Objects/dictobject.c
gcc -pthread -c -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapv -O3 -Wall -W
-DPy_BUILD_CORE -o Objects/memoryobject.o Objects/memoryobject.c
gcc -pthread -c -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapv -O3 -Wall -W
-DPy_BUILD_CORE -o Objects/methodobject.o Objects/methodobject.c
gcc -pthread -c -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapv -O3 -Wall -W
-DPy_BUILD_CORE -o Objects/moduleobject.o Objects/moduleobject.c
gcc -pthread -c -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapv -O3 -Wall -W
-DPy_BUILD_CORE -o Objects/object.o Objects/object.c
gcc -pthread -c -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapv -O3 -Wall -W
-DPy_BUILD_CORE -o Objects/obmalloc.o Objects/obmalloc.c
gcc -pthread -c -fno-strict-aliasing -g -O2 -DNDEBUG -g -fwrapv -O3 -Wall -W
```

如果没有任何异常信息显示在控制台，安装成功。最后看到的应该是这样

```
running install_egg_info
Writing /usr/local/Python2.7.8/lib/python2.7/lib-dynload/Python-2.7.8-py2
rm /usr/local/Python2.7.8/lib/python2.7/lib-dynload/_sysconfigdata.py*
if test -f /usr/local/Python2.7.8/bin/python -o -h /usr/local/Python2.7.8
then rm -f /usr/local/Python2.7.8/bin/python; \
else true; \
fi
(cd /usr/local/Python2.7.8/bin; ln -s python2 python)
rm -f /usr/local/Python2.7.8/bin/python2
(cd /usr/local/Python2.7.8/bin; ln -s python2.7 python2)
rm -f /usr/local/Python2.7.8/bin/python2-config
(cd /usr/local/Python2.7.8/bin; ln -s python2.7-config python2-config)
rm -f /usr/local/Python2.7.8/bin/python-config
(cd /usr/local/Python2.7.8/bin; ln -s python2-config python-config)
test -d /usr/local/Python2.7.8/lib/pkgconfig || /usr/bin/install -c -d -m
rm -f /usr/local/Python2.7.8/lib/pkgconfig/python2.pc
(cd /usr/local/Python2.7.8/lib/pkgconfig; ln -s python-2.7.pc python2.pc)
rm -f /usr/local/Python2.7.8/lib/pkgconfig/python.pc
(cd /usr/local/Python2.7.8/lib/pkgconfig; ln -s python2.pc python.pc)
rm -f /usr/local/Python2.7.8/share/man/man1/python2.1
(cd /usr/local/Python2.7.8/share/man/man1; ln -s python2.7.1 python2.1)
rm -f /usr/local/Python2.7.8/share/man/man1/python.1
(cd /usr/local/Python2.7.8/share/man/man1; ln -s python2.1 python.1)
[root@YangShuang_CentOS Python-2.7.8]#
```

执行命令 `cd /usr/local/Python2.7.8` 进入新 Python 库的安装目录

```
[root@YangShuang_CentOS Python-2.7.8]# cd /usr/local/Python2.7.8
[root@YangShuang_CentOS Python2.7.8]# ll
total 16
drwxr-xr-x 2 root 0 4096 Aug 14 20:49 bin
drwxr-xr-x 3 root 0 4096 Aug 14 20:49 include
drwxr-xr-x 4 root 0 4096 Aug 14 20:49 lib
drwxr-xr-x 3 root 0 4096 Aug 14 20:49 share
[root@YangShuang_CentOS Python2.7.8]#
```

`cd ./bin` 进入 bin 文件

再执行 `./python`。可以看到下图，新 python2.7.8 库已经安装成功。

```
total 6112
-rwxr-xr-x 1 root 0 113 Aug 14 20:48 2to3
-rwxr-xr-x 1 root 0 111 Aug 14 20:48 idle
-rwxr-xr-x 1 root 0 96 Aug 14 20:48 pydoc
lrwxrwxrwx 1 root 0 7 Aug 14 20:49 python -> python2
lrwxrwxrwx 1 root 0 9 Aug 14 20:49 python2 -> python2.7
-rwxr-xr-x 1 root 0 6221527 Aug 14 20:48 python2.7
-rwxr-xr-x 1 root 0 1686 Aug 14 20:49 python2.7-config
lrwxrwxrwx 1 root 0 16 Aug 14 20:49 python2-config -> python2.7-config
lrwxrwxrwx 1 root 0 14 Aug 14 20:49 python-config -> python2-config
-rwxr-xr-x 1 root 0 18559 Aug 14 20:48 smtpd.py
[root@YangShuang_CentOS bin]# ./python
Python 2.7.8 (default, Aug 14 2014, 20:48:17)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Python2.7.8 解析器已经可以使用。

提及系统自带 python 库为 2.6.6，下面 python 多版本的问题。

执行命令 `whereis python`，查看当前系统 path 中 python 指令调用的 python 解析器在那个目录下

```
>>> exit()
[root@YangShuang_CentOS bin]# whereis python
python: /usr/bin/python /usr/bin/python2.6 /usr/lib/python2.6 /usr/lib64/
python2.6 /usr/share/man/man1/python.1.gz
[root@YangShuang_CentOS bin]#
```

`cd /usr/bin/python`，进入系统自带 python 解析器的默认安装目录

`ls -ah | grep python` 查看 bin 下有哪些程序与 python 关联

```
[root@YangShuang_CentOS bin]# ll -ah | grep pytoh
[root@YangShuang_CentOS bin]# ll -ah | grep python
-rwxr-xr-x. 1 root 0 11K Nov 23 2013 abrt-action-analyze-python
-rwxr-xr-x 2 root 0 4.8K Jan 22 2014 python
lrwxrwxrwx 1 root 0 6 Aug 14 20:19 python2 -> python
-rwxr-xr-x 2 root 0 4.8K Jan 22 2014 python2.6
[root@YangShuang_CentOS bin]#
```

执行以下命令备份老版本的 python，命名为 python2.6.6 以作区分。

```
mv python python2.6.6
```

停留在原目录，创建新 python 解析器连接文件，指向/usr/local/python2.7.8/bin/python.

执行以下命令

```
ln -s /usr/local/Python2.7.8/bin/python /usr/bin/python
```

重新执行 `ls -ah |grep python` 查看连接文件创建是否成功

```
[root@YangShuang_CentOS bin]# ls -ah |grep python
abrt-action-analyze-python
python
python2
python2.6
python2.6.6
[root@YangShuang_CentOS bin]#
```

输入 python 命令，查看 python 版本是否升级成功

```
[root@YangShuang_CentOS bin]# python
Python 2.7.8 (default, Aug 14 2014, 20:48:17)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)] on linux2
Type "help", "copyright", "credits" or "license" for more information
>>>
```

可以看到 python 库已经升级成功。到这里 python 库安装结束。

Tips: python 升级以后，yum 功能异常，因为 yum 依赖于旧的 python 解析器

```
[root@YangShuang_CentOS bin]# yum
There was a problem importing one of the Python modules
required to run yum. The error leading to this problem was:

  No module named yum

Please install a package which provides this module, or
verify that the module is installed correctly.

It's possible that the above module doesn't match the
current version of Python, which is:
2.7.8 (default, Aug 14 2014, 20:48:17)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)]

If you cannot solve this problem yourself, please go to
the yum faq at:
http://yum.baseurl.org/wiki/Faq
```

按照如下办法解决。将 yum 配置文件中调用的 python 解析器指向旧的 python 解析器即可。

执行命令 `vim /usr/bin/yum` 编辑该文件

按 a 进入编辑模式，光标定位第一行，修改 python 为 python2.6.6


```

1 #!/usr/bin/python2.6.6
2 import sys
3 try:
4     import yum
5 except ImportError:
6     print >> sys.stderr, """\
7 There was a problem importing one of the Python modules
8 required to run yum. The error leading to this problem was:
9
10    %s
11
12 Please install a package which provides this module, or
13 verify that the module is installed correctly.
14
15 It's possible that the above module doesn't match the
16 current version of Python, which is:
17 %s
18
19 If you cannot solve this problem yourself, please go to
20 the yum faq at:
21 http://yum.baseurl.org/wiki/Faq
22
23 """ % (sys.exc_value, sys.version)
24 sys.exit(1)
25

```

按 esc 结束编译，ctrl+space 进英文半角。

Shift+; 输入英文冒号

wq 指令保存并退出。

```

29     yummain.user_ma
30 except KeyboardInterrupt
31     print >> sys.st
32     sys.exit(1)
:wq

```

再次执行 yum 指令，因 python 导致 yum 无法使用的问题修复。

可以在控制台通过交互模式学习 python 的基本语法了。但是行编辑有问题

```

Python 2.7.8 (default, Aug 14 2014, 20:48:17)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)] on linux2
Type "help", "copyright", "credits" or "license" for more informat
>>> def add ()^[[D^[[D

```

安装 readline-devel 和 readline 行编辑支持包即可，注意要安装和机器对应版本的支持包。

yum install readline-devel.86_64

yum install readline


```

[root@YangShuang_CentOS ~]# yum search readline-devel
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
* base: mirrors.aliyun.com
* extras: mirrors.aliyun.com
* updates: mirrors.aliyun.com
===== N/S Matched: readline-devel =====
readline-devel.i686 : Files needed to develop programs which use readline library
readline-devel.x86_64 : Files needed to develop programs which use readline library

```

安装完毕应该就可以进行控制台行编辑了。

在安装完以上两个库以后，如果此时还无法进行行编辑，那么如下操作

打开 python 源码目录下的/Module/Setup 文件。

```

[root@YangShuang_CentOS Python-2.7.8]# vim /usr/local/src/Python-2.7.8/Modules/Setup

```

/readline 命令，定位到 readline

```

165
166 readline readline.c -lreadline -ltermcap
167

```

166 行去掉#注释，然后按照上面的步骤重新编译 python。完成这以后，行编辑可用。

```

[root@YangShuang_CentOS Python-2.7.8]# python
Python 2.7.8 (default, Aug 14 2014, 22:09:33)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> def add(x,y):
...     print "你输入的两个参数为%s ,%s" %(x,y)
...
>>> add(1,2)
你输入的两个参数为1 ,2
>>>

```

有行编辑支持，控制台下学习 python 如虎添翼。

如果还不行，请私信我。

#####2014-8-16 补充#####

按照以上编译安装 python 以后，在安装一些 python 软件时，无法安装，原因是因为使用 python 脚本安装软件，无法调用依赖库。比如 ssl, bz2, zlib 等。Pip 包管理器的安装依赖于 zlib 库。

如果发现无法使用新版本 python 解析器执行 py 安装脚本，如 python setup.py install 报错。

则需要手动安装依赖库，常见的有 openssl-devel、tk-devel、libbz2-devel、sqlite3-devel 等安装依赖包以后，需要再次进行 python 源码的编译安装。

当然，如果够细心，执行 `make && makinstall` 安装结束后，滚动向上输出窗口,大概 100 行左右，找到 `running install`，能看到 `python` 库编译的时候，输出提示有部分公用模块库未找到。

```
running install
running build
running build_ext
building dbm using gdbm

Python build finished, but the necessary bits to build these modules were not found:
bsddb185      bz2           dl
imageop       sunaudiodev
To find the necessary bits, look in setup.py in detect_modules() for the module's name.

running build_scripts
running install_lib
copying build/lib.linux-x86_64-2.7/_sysconfigdata.py => /usr/local/lib/python2
```

这个地方能看到缺少的依赖库。使用 `yum install` 安装即可，缺少少模块不需要全部安装，有些模块(如 `dl,imageop` 均过时)已经被废弃或被替代了，一般说安装 `openssl, tk-devel,bzip2` 就可以了。

Tips: 这些依赖模块不要使用源码编译安装，极易出错

```
#####
```

Vim 版本升级

Python 库升级结束，要进行 `vim` 的升级，因为采用 `jedi-vim` 插件，该插件能和 `vs` 里面一样，对库中的函数，模块中函数进行自动补全和提示，真正的将 `vim` 改造成适合 `python` 编程的 `ide`。

`Jedi-vim` 良好工作需要 `vim7.3+`以上版本支持。

`centos6.5` 默认 `vim` 版本 `7.2`，安装 `jedi` 插件，无法良好工作。

```

VIM - Vi IMproved

        version 7.2.411
        by Bram Moolenaar et al.
        Modified by <bugzilla@redhat.com>
        Vim is open source and freely distributable


        Sponsor Vim development!
type   :help sponsor<Enter>      for information

type   :q<Enter>                  to exit
type   :help<Enter> or <F1>       for on-line help
type   :help version7<Enter>     for version info

```

备份老版本的 vim 程序

```
mv /usr/bin/vim /usr/bin/vim7.3
```

如何编译安装在 python library 升级中已做了很详细的说明，以下升级说明从简。

Root 账户登录，依次执行以下命令

```
cd /user/local/src #进入 src 目录
```

```
wget ftp://ftp.vim.org/pub/vim/unix/vim-7.4.tar.bz2 #下载 vim7.4 安装包
```

```
tar -vxf vim-7.4.tar.bz2 #解压缩
```

```
cd vim74 #进入源码包
```

```
mkdir /usr/local/vim74 #创建 vim 安装目录
```

```
./configure --prefix=/usr/local/vim74 --enable-multibyte --enable-pythoninterp
```

这里编译 vim 需要注意，

--enable-multibyte 为 vim 编辑器启用多字符支持，否则有字符为乱码，

--enable-pythoninterp, vim 编辑器提供 python 支持，集成 python，这个是后面 jedi-vim 插件安装所必须的，也是 vim 作为 python ide 最关键的一步。编译持续时间 1 分多种。

原目录，make && make install 安装 vim，安装持续时间 2 分钟。

这里很容易出问题，到此步有各种各样的问题，不一一说明，有问题私信我。

编译安装结束以后，创建 vim7.4 的链接文件

```

vim      vimdiff  vimtutor
[root@YangShuang_CentOS bin]# ln -s /usr/local/vim7.4/bin/vim /usr/bin/vim
[root@YangShuang_CentOS bin]# vm
-bash: vm: command not found
[root@YangShuang_CentOS bin]# vi
vi      vigr      vim7.2      vimtutor      vino-preferen

```

```
[root@YangShuang_CentOS bin]# ll -ah | grep vim
lrwxrwxrwx. 1 root      0   3 May 28 06:08 ex -> vim
lrwxrwxrwx. 1 root      0   3 May 28 06:08 rvim -> vim
lrwxrwxrwx. 1 root      0  25 Aug 14 23:35 vim -> /usr/local/vim/.4/bin/vim
-rwxr-xr-x. 1 root      0 1.9M Apr  5 2012 vim7.2
lrwxrwxrwx. 1 root      0   3 May 28 06:08 vimdiff -> vim
-rwxr-xr-x. 1 root      0 2.1K Apr  5 2012 vimtutor
[root@YangShuang_CentOS bin]#
```

键入 vim 命令，vim 成功升级到 vim7.4

```

      VIM - Vi IMproved

      version 7.4
      by Bram Moolenaar et al.
Vim is open source and freely distributable

      Become a registered Vim user!
type  :help register<Enter>   for information

type  :q<Enter>               to exit
type  :help<Enter> or <F1>    for on-line help
type  :help version7<Enter>  for version info

      Running in Vi compatible mode
type  :set nocp<Enter>        for Vim defaults
type  :help cp-default<Enter> for info on this
```

升级 vim 以后，发现，使用 vim 编辑器编辑文件，语法高亮丢失。
并且编辑文本出现异常，不能删除，方向键不起作用了。

接下来需要新增和修改一些配置文件。

```
root@tangshuang_CentOS:/usr/bin
```

```
if &t_Co > 2 || has("gui_running")
    syntax on
    set hlsearch
endif

filetype plugin on

if &term=="xterm"
    set t_Co=8
    set t_Sb=^[[4%dm
    set t_Sf=^[[3%dm
endif

" Don't wake up system with blinking cursor:
" http://www.linuxpowertop.org/known.php
let &guicursor = &guicursor . ",a:blinkon0"

"add by ys on 2014-6-13 ,add some principles
set number "行号"
syntax on "语法高亮
set autoindent " 自动对齐
set smartindent
set showmatch " 括号匹配
set ruler " 显示光标位置的行
set incsearch " 快速查找
filetype on

if has ("vms")
    set nobackup
```

略感头疼，执行以下命令

```
cp /etc/vimrc /root/.vimrc
```

拷贝一个 vimrc 文件过来到 root 家目录，保存为.vimrc。隐藏文件。

如果是其他用户使用 vim 编辑器，语法高亮丢失，同样，cp 一个配置文件到对应 home 目录下。

说到 vim 的配置文件，作为一个编辑器，需要行号，语法高亮，编辑文件名提示，光标所在列，文档显示百分比，等基本信息的提示，向/etc/vimrc 文件中添加以下内容

```

set number "行号"A
syntax on "语法高亮
set autoindent " 自动对齐
set smartindent
set showmatch " 括号匹配
set ruler " 显示光标位置的行
set incsearch " 快速查找
filetype on

```

等 jedi 插件安装完毕，还会再次调整 vimrc 配置文件。

如上配置以后，vim 编辑器初步效果

```

11 operator:x:11:0:operator:/root:/sbin/nologin
12 games:x:12:100:games:/usr/games:/sbin/nologin
13 gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
14 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
15 nobody:x:99:99:Nobody:./:/sbin/nologin
16 dbus:x:81:81:System message bus:./:/sbin/nologin
17 usbmuxd:x:113:113:usbmuxd user:./:/sbin/nologin
18 vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
19 rtkit:x:499:497:RealtimeKit:/proc:/sbin/nologin
20 avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
21 abrt:x:173:173:./etc/abrt:/sbin/nologin
22 haldaemon:x:68:68:HAL daemon:./:/sbin/nologin
23 gdm:x:42:42:./var/lib/gdm:/sbin/nologin
24 ntp:x:38:38:./etc/ntp:/sbin/nologin
25 apache:x:48:48:Apache:/var/www:/sbin/nologin
26 saslauthd:x:498:76:"Saslauthd user"/var/empty/saslauthd:/sbin/nologin
27 postfix:x:89:89:./var/spool/postfix:/sbin/nologin
28 pulse:x:497:496:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
29 sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
30 tcpdump:x:72:72:./:/sbin/nologin
31 yangshuang:x:500:500:CentOS:/home/yangshuang:/bin/bash
32 mysql:x:496:493:MySQL server:/home/mysql:/bin/bash
"/etc/passwd" 36L, 1713C

```

到这里为止，已经可以用 vim 来写 python 代码了。只不过用 vim 编辑器，没有代码提示和补全功能。

安装 Jedi-vim 插件

Jedi-vim 是 vim 作为 python ide 的绝佳利器。

jedi 插件的安装需要两个模块的支持，一个是 git，一个是包管理器 pathogen

root 账户登录

yum install git 安装 git

只需要 git 的基础功能，能拉取到 github 上的文件即可。

mkdir ~/.vim

创建存储 vim 插件的存放目录。隐藏

```
mkdir -p ~/.vim/autoload ~/.vim/bundle
curl -LSso ~/.vim/autoload/pathogen.vim https://tpo.pe/pathogen.vim
```

创建包管理器存放目录。创建插件存放目录，下载插件包管理插件 pathogen.vim

下载 jedi 插件

```
cd ~/.vim/bundle/ && git clone --recursive https://github.com/davidhalter/jedi-vim.git
```

初始化 Vim for Python 开发环境

修改/etc/vimrc,针对 python 的语法特性，做以下配置，配置自动缩进，检索字背景高亮，开启插件支持

```
"vim 开启插件支持
set expandtab
set tabstop=8
set shiftwidth=4
set softtabstop=4
set autoindent
let Tlist_Ctags_Cmd='/usr/bin/ctags'
execute pathogen#infect()
filetype plugin indent on
filetype plugin on
set ofu=syntaxcomplete#Complete
highlight OverLength ctermfg=red ctermfg=white guibg=#592929
match OverLength /\%81v.\%+/\
```

执行命令 `cp -R ~/.vim /home/youhomenamename`

拷贝一份 vim 插件包到需要使用 vim 作为 python 开发的用户的家目录。

各个有需要的用户同步完 vimrc 配置文件以后，可调整全局的 vim 配置文件 /etc/vimrc,关闭插件支持。

也可以针对指定用户，配置.vimrc，具体配置内容同上。

在正确配置以后，使用 vim，编写 python 脚本，效果如下

项目创建和测试

普通用户账户登录，命令较多，不一一敲了。创建一个简单的 shell 脚本，初始化项目骨架。

家目录下执行 `touch ~/initprojectdir.sh`

写入以下命令

```
#!/usr/bin/bash
# add by ys on 2014-8-14
# 项目目录初始化脚本

cd ~
#项目父路径
basepath=~ /Projects"
#项目根目录不存在，则创建
if [ ! -x "$basepath"];then
    mkdir "$basepath"
fi
#创建项目相关文件和文件夹
cd Projects
#项目根目录
mkdir htMis
cd htMis
mkdir bin
#模块文件夹
mkdir module
#模块单元测试文件夹
mkdir tests
#文档文件夹
mkdir docs

# 初始化配置文件
cd ~/Projects/htMis
touch module/__init__.py
touch tests/__init__.py

#创建单元测试文件
cd tests
touch module_tests.py
```

`:wq` 保存。执行 `sh ~/initprojectdir.sh`
完成项目骨架的创建

执行 `ls -R htMis/` 查看项目根文件下的目录和文件结构，应该如下

```
[yangshuang@cyCentOS Projects]$ ls -R htMis/  
htMis/:  
bin  docs  module  tests
```

htMis/bin:

htMis/docs:

htMis/module:

helloworld.py __init__.py

htMis/tests:

__init__.py module_tests.py

最后一步，可以写 python 代码了。

```
[yangshuang@cyCentOS Projects]$ cd htMis/  
[yangshuang@cyCentOS htMis]$ ll  
total 16  
drwxrwxr-x. 2 yangshuang yangshuang 4096 Aug 15 19:34 bin  
drwxrwxr-x. 2 yangshuang yangshuang 4096 Aug 15 19:34 docs  
drwxrwxr-x. 2 yangshuang yangshuang 4096 Aug 15 19:36 module  
drwxrwxr-x. 2 yangshuang yangshuang 4096 Aug 15 19:36 tests  
[yangshuang@cyCentOS htMis]$
```

```
cd ~/Projects/htMis/module
```

```
touch Helloworld.py
```

```
vim Helloworld.py
```

写一段脚本，定义一个 student 类，编写构造函数__init__和普通函数 get_personinfo

```
1 S.capitalize() -> string
2
3 Return a copy of the string S with only its first character
4 capitalized.
Scratch] [Preview]
2 # coding=utf8
3 # add by ys on 2014-8-14
4 # module测试
5
6 class Student(object):
7     def __init__(self, name):
8         print "Student对象"
9         self.name = name
10        self.age = age
11
12    def get_studentinfo(self):
13        '''注释'''
14        print "name:%s\n age:%r" % (self.name, self.age)
15
16 student = Student("canyou")
17 student.get_studentinfo()
18 #student.
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666

```