Python3.7在CentOS6的系统中源码包编译安装流程：

由于在Python3.7中要求OpenSSL版本1.0.2（含）版本以上，CentOS6中默认安装的是OpenSSL 1.0.1版本，提前安装OpenSSL，在CentOS7系统中不需要升级

首先安装OpenSSL（如果不安装后面无法使用pip3命令在线升级Python模块）

1.下载源码包

2.tar xf解压

3.yum install zlib-devel openssl-devel libffi-devel安装依赖软件包，libffi-devel模块建议线上安装，或者下载到本地使用rpm命令进行安装

4../configure配置

5.make

6.make install

几个依赖包文件的作用可以使用yum info zlib-devel

Python3的调用

/usr/local/Python3.7/bin/python3 直接使用配置过程中定义的安装路径

>>> quit() 退出Python环境

python 直接使用Python命令打开的是Python2

可以通过修改PATH变量直接执行Python3命令不需要绝对路径

echo 'PATH=/usr/local/Python3.7/bin:$PATH' >> /etc/profile

source /etc/profile

python3 直接执行命令即可

默认Python命令不支持tab补全和命令提示，使用很不便

pip3 install ipython

ipython3 可以使用ipython3命令

pip更换更新源，默认情况下是从Python包括其他模块的官网（国外网站）下载，速度不是很理想

**pip国内的一些镜像**

  阿里云 <http://mirrors.aliyun.com/pypi/simple/>

  中国科技大学 <https://pypi.mirrors.ustc.edu.cn/simple/>

  豆瓣(douban) <http://pypi.douban.com/simple/>

  清华大学 <https://pypi.tuna.tsinghua.edu.cn/simple/>

  中国科学技术大学 <http://pypi.mirrors.ustc.edu.cn/simple/>

**修改源方法：**

***临时使用：***

可以在使用pip的时候在后面加上-i参数，指定pip源

eg: pip install scrapy -i <https://pypi.tuna.tsinghua.edu.cn/simple>

***永久修改：***

**linux:**

修改 ~/.pip/pip.conf (没有就创建一个)， 内容如下：

[global]
index-url = https://pypi.tuna.tsinghua.edu.cn/simple

**windows:**

直接在user目录中创建一个pip目录，如：C:\Users\xx\pip，新建文件pip.ini，内容如下

[global]
index-url = https://pypi.tuna.tsinghua.edu.cn/simple

mkdir ~/pip

vim ~/pip/pip.conf

Linux Shell中语句有开头和结束标记

for xxx in 循环语句开始

do 语句内容

done 语句结束

Python中语句没有开头和结尾，每个不同语句是通过制表位(TAB)进行分隔

vim 01-示例脚本.py Python脚本的后缀一般都是以py为后缀

print("Hello Wrold") 写入这个内容，print屏幕输出后面()中内容，中间内容使用引号保护(双引号和单引号都可以)

第一种执行方式：

python3 01-示例脚本.py 使用python2或python3都能执行

第二种执行方式：

支持shell脚本是执行

chmod +x 01-示例脚本.py 添加执行权限

vim 01-示例脚本.py

#!/usr/local/Python3.7/bin/python3 添加解析器，否则会以shell命令方式执行会报错

./01-示例脚本.py 和shell一样执行

每部分内容是需要缩进，下面直接执行会报错

print("Hello Wrold")

def main():

print("HELLO WORLD") 这行属于上面配置的子内容，必须缩进

Python注释：

单行注释：以#开头，这一行是注释

多行注释：'''开头和'''结尾，中间的所有内容都是注释

#!/usr/local/Python3.7/bin/python3

print("Hello Wrold")

#def是定义函数的命令

def main():#main是函数名称

'''

这是一个多行注释

下面的内容是属于main函数内容

定义了print打印内容

'''

print("HELLO WORLD")

当脚本文件中含有中文，Python2将不能再执行，Python3正常

Python2中默认语言使用的ASCII编码默认只能支持英文中127个字符（2个字节存储），由于其他国家（中国等）编码比较多，使用UTF-8的编码，Python3默认采用就是UTF-8编码，无需定义，Python2需要定义编码格式

#encoding=UTF-8 定义编码格式，Python2需要这个编码才能执行中文内容，Python3有没有无所谓，如果有脚本希望Python2和Python3都能执行以要加上这个内容

python可以定义变量

In [1]: a=11 定义变量

In [2]: b=22

In [3]: a 引用变量

Out[3]: 11

In [4]: b

Out[4]: 22

In [5]: num1,num2=33,44 同时定义两个变量

In [6]: num1

Out[6]: 33

In [7]: num2

Out[7]: 44

In [8]: num1+num2

Out[8]: 77

Python中变量的类型属于弱类型，不需要特意定义

In [14]: a=1 定义数字int类型

In [15]: b='2' 定义字符类型使用引号

In [16]: a+b int和字符不能相加，字符相加是直接写多个字符

In [21]: type(a) 查看类型

Out[21]: int

In [22]: type(b) 查看类型

Out[22]: str

变量命名规则：第一个字符必须是字母或下划线，其他字符可以是字母或数字或下划线

变量命名要取的有意义能够根据名称识别意思

小驼峰：第一个单词首字母小写，其他大写myName

大驼峰：第一个单词首字母大写，其他也大写MyName

\_方式：单词之间使用\_分隔，my\_name

Python中有一些保留字有自身含义不能做变量名

In [23]: import keyword 引入keyword

In [24]: keyword.kwlist 查看keyword列表

Python输出

print支持表达式

print输出替换

In [25]: a=10

In [26]: b=20

In [27]: print("a的是数值:%d"%a) %d代表字符类型

In [28]: print("a的数值是:%d,b的数值是:%d"%(a,b)) 同时调用多个变量，%a对应第一个%d,%b对应第二个%d

a的数值是:10,b的数值是:20

In [29]: print("a的数值是:%d,b的数值是:%d,a+b的数值是:%d"%(a,b,a+b))

a的数值是:10,b的数值是:20,a+b的数值是:30

In [30]: name='张三'

In [31]: print("我的名字是:%d"%name)

---------------------------------------------------------------------------

TypeError Traceback (most recent call last)

<ipython-input-31-131e22a14635> in <module>()

----> 1 print("我的名字是:%d"%name)

TypeError: %d format: a number is required, not str

In [32]: print("我的名字是:%s"%name) 字符类型必须使用%s，数字类型也能用%s,全部使用%s也是可以的

我的名字是:张三

In [33]: '%2d-%02d'%(1,3) 定义输出格式%2d代表输出两位，不足两位前面补空格，02d也是如果不足两位以0补充

In [37]: '%.2f'%3.1415926 小数点后保留两位

>>> name=input("请输入您的姓名:")

请输入您的姓名:1+1 Python2允许输入表达式

>>> name

2

In [1]: name=input("请输入您的姓名:")

请输入您的姓名:1+1

In [2]: name python3中输入的所有内容全部是字符处理，更安全

Out[2]: '1+1'

Python3中不再支持表达式的方式写入，Python2中input支持表达式的方式，Python2中raw\_input是不支持表达式

vim 02-input方法.py

#encoding=utf-8

name=input("请输入您的姓名:")

age=input("请输入您的年龄:")

print("您的姓名是:%s,您的年龄是:%s,两年后的年龄是:%s"%(name,age,int(age)+2)) #默认input定义的是字符类型，字符不能使用+加数字，可以通过int将输入的改为数字类型

[4]: a=10

In [5]: a\*=2+3-4 \*=是一次赋值，将后面的表达式算完了再给前面赋值

In [6]: a

Out[6]: 10