[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY01/CASE/01/index.html" \l "page_top_case)

# NSD Python1 DAY01

1. [案例1：准备python开发环境](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY01/CASE/01/index.html" \l "case1)
2. [案例2：模拟用户登陆](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY01/CASE/01/index.html" \l "case2)

## 1 案例1：准备python开发环境

### 1.1 问题

1. 下载最新版本的python3
2. 下载pycharm社区版
3. 安装python3，使其支持Tab键补全
4. 配置pycharm，使其符合自己的习惯

### 1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：下载最新版python3

首先去python官网下载python3的源码包，网址：https://www.python.org/

进去之后点击导航栏的Downloads，也可以鼠标放到Downloads上弹出菜单选择Source code，表示源码包，这里选择最新版本3.6.4，这里选择第一个下载即可，下载的就是源码包：Python-3.6.4.tar.gz，下载好之后上传到linux系统，准备安装，如图-1所示：



图-1

步骤二：Linux下安装python3

1)python安装之前需要一些必要的模块，如果没有这些模块后来使用会出现一些问题，输入以下命令提前预装依赖包：

1. [root@localhost ~]# yum install -y gcc gcc-c++ zlib-devel openssl-devel readline-devel libffi-devel sqlite-devel tcl-devel tk-devel

释放文件：

1. [root@localhost python]# tar -xzf Python-3.6.4.tar.gz

2)进入Python-3.6.4目录：

1. [root@localhost python]# cd Python-3.6.4
2. [root@localhost Python-3.6.4]#ls        #此时Python-3.6.4文件夹中没有makefile文件

3)配置安装目录：

configure是用来进行用户个性配置的工具，--prefix是说软件安装目录设置在哪里， =/usr/local就是你给出的安装目录

1. [root@localhost Python-3.6.4]# ./configure --prefix=/usr/local
2. [root@localhost Python-3.6.4]# ls     #此时Python-3.6.4文件夹中生成了makefile文件
3. aclocal.m4 Doc Makefile PCbuild python-config.py
4. build Grammar Makefile.pre Programs python-gdb.py
5. config.guess Include Makefile.pre.in pybuilddir.txt README.rst
6. config.log install-sh Misc pyconfig.h setup.py
7. config.status Lib Modules pyconfig.h.in Tools
8. config.sub libpython3.6m.a Objects python
9. configure LICENSE Parser Python
10. configure.ac Mac PC python-config

4)接下来编译源码：

1. [root@localhost Python-3.6.4]# make

5)执行安装：

1. [root@localhost Python-3.6.4]# make install

整个过程大约5-10分钟，安装成功

步骤三：下载并安装Pycharm社区版

网址：https://www.jetbrains.com/pycharm/download，这里选择下图红框下载即可，下载好之后上传到linux系统，准备安装，如图-2所示：

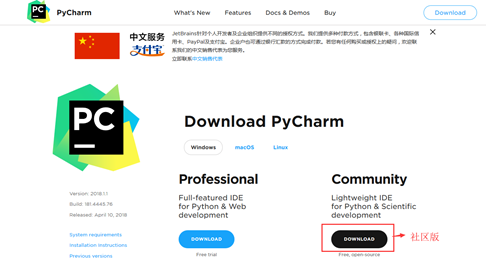


图-2

2)释放文件：

1. [root@localhost ~]# tar -xzf pycharm-community-2018.1.1.tar.gz

3)运行下面的命令进入PyCharm 目录：

1. [root@localhost pycharm-community-2018.1.1]# cd pycharm-community-2018.1.1/bin

4)通过运行下面的命令来运行PyCharm进入图形化安装界面：

1. [root@localhost bin]# sh pycharm.sh &

5)Pycharm打开后，如果你需要导入之前安装版本的设置的话，可以选择第一个选项，如果没有的话，选择(Do not import settings)默认不导入设置，点击/同意，就可以进入pycharm进行配置，如图-3所示：

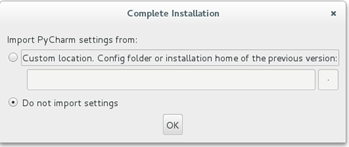


图-3

6)激活Pycharm：在弹出的激活窗口中，选择“License serveer”输入激活服务器地址“http://127.0.0.1:1017”，之后点击‘Activate’，完成pycharm激活，如图-4所示：

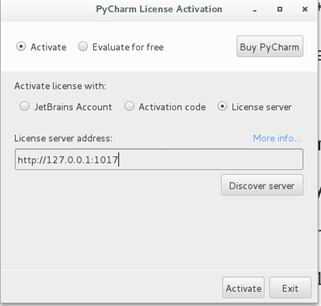


图-4

7)启动完成进入欢迎界面，如图-5所示：



图-5

## 2 案例2：模拟用户登陆

### 2.1 问题

编写login.py脚本，实现以下目标：

1. 创建名为login.py的程序文件
2. 程序提示用户输入用户名
3. 用户输入用户名后，打印欢迎用户

### 2.2 方案

编写程序时，很多情况下都需要程序与用户交互。在python3中，主要通过input()获取用户输入信息，使用print()打印信息。

通常当想看变量内容时，会在代码中使用print()语句输出。不过在交互式解释器中，可以用print语句显示变量的字符串表示，或者仅使用变量名查看该变量的原始值。

从用户那里得到数据输入的最容易的方法是使用input()内建函数。它读取标准输入，并将读取到的数据赋值给指定的变量。需要注意的是，input()函数读入的数据全部是以字符串的方式存储的。如果用户输的是数字，那么python也将其保存为字符串，当将字符串与数字做数学运算是将会出现TypeError的错误。

初学者在需要显示信息或得到用户输入时，很容易想到使用print()语句和input()内建函数。不过在此建议函数应该保持其清晰性，也就是它只应该接受参数，返回结果。从用户那里得到需要的数据， 然后调用函数处理， 从函数得到返回值，然后显示结果给用户。这样你就能够在其它地方也可以使用你的函数而不必担心自定义输出的问题。这个规则的一个例外是，如果函数的基本功能就是为了得到用户输出，或者就是为了输出信息，这时在函数体使用print()语句或input()也未尝不可。更重要的，将函数分为两大类，一类只做事，不需要返回值（比如与用户交互或设置变量的值）， 另一类则执行一些运算，最后返回结果。如果输出就是函数的目的，那么在函数体内使用 print()语句也是可以接受的选择。

### 2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：编写脚本

本次练习的脚本文件是/root/bin/login.py。

1. [root@localhost day01]# vim login.py
2. #!/usr/bin/env python3
3. username = input('username: ') #使用变量username接收用户输入的字符
4. print('Welcome', username) #输出欢迎信息，字符串和变量名之间用逗号
5. #隔开，两者之间自动会加上空格

步骤二：测试脚本执行

1. [root@localhost day01]# python3 login.py
2. username: bob #输入用户名
3. Welcome bob