

# Tensorflow and Keras 安装

## 目录

Tensorflow and Keras 安装.....	0
一. Python 环境.....	0
二. 集成 tensorflow .....	0
三. 集成 keras.....	0
四. Ide 开发工具集成 Python.....	0
4.1 安装 .....	1
启动 IDE.....	1
配置 IDE 集成环境 python.....	1
创建 python 新项目 .....	4
4.2 tensorflow 实例 .....	6
使用前先打开 tensorflow 计算环境.....	6
新建 python 文件.....	7
运行 tensorflow 实例文件.....	8

### 一. Python 环境

anaconda

### 二. 集成 tensorflow

```
pip install --upgrade tensorflow
```

### 三. 集成 keras

```
pip install keras -U -pre
```

```
$HOME/.keras/keras.json
```

```
{  
  "image_data_format": "channels_last",  
  "epsilon": 1e-07,  
  "floatx": "float32",  
  "backend": "tensorflow"  
}
```

### 四. Ide 开发工具集成 Python

## 4.1 安装

### 启动 IDE



### 配置 IDE 集成环境 python

第一次启动后

点击[setting]-[installjetbrains plugin]

-搜索 python-[install]-[close]-重启 idea,之后再次启动不用做这一步配置,具体操作如下图所示:



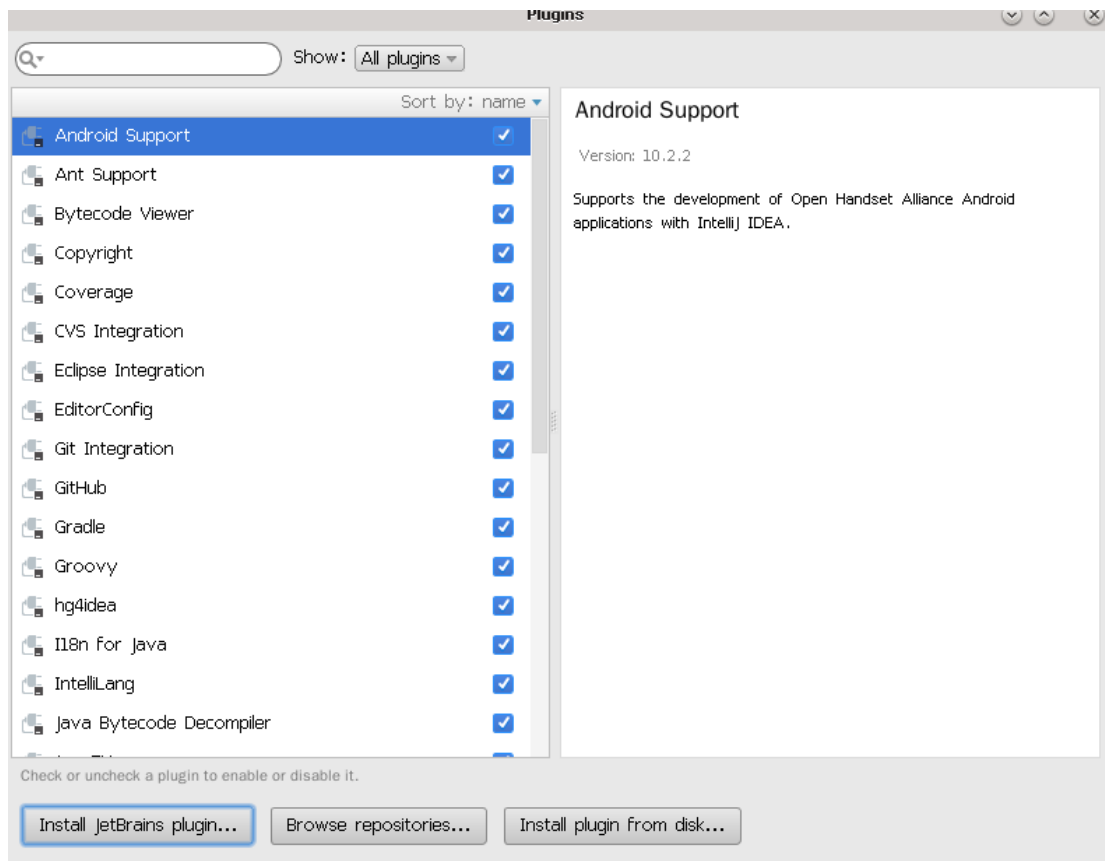
# IntelliJ IDEA

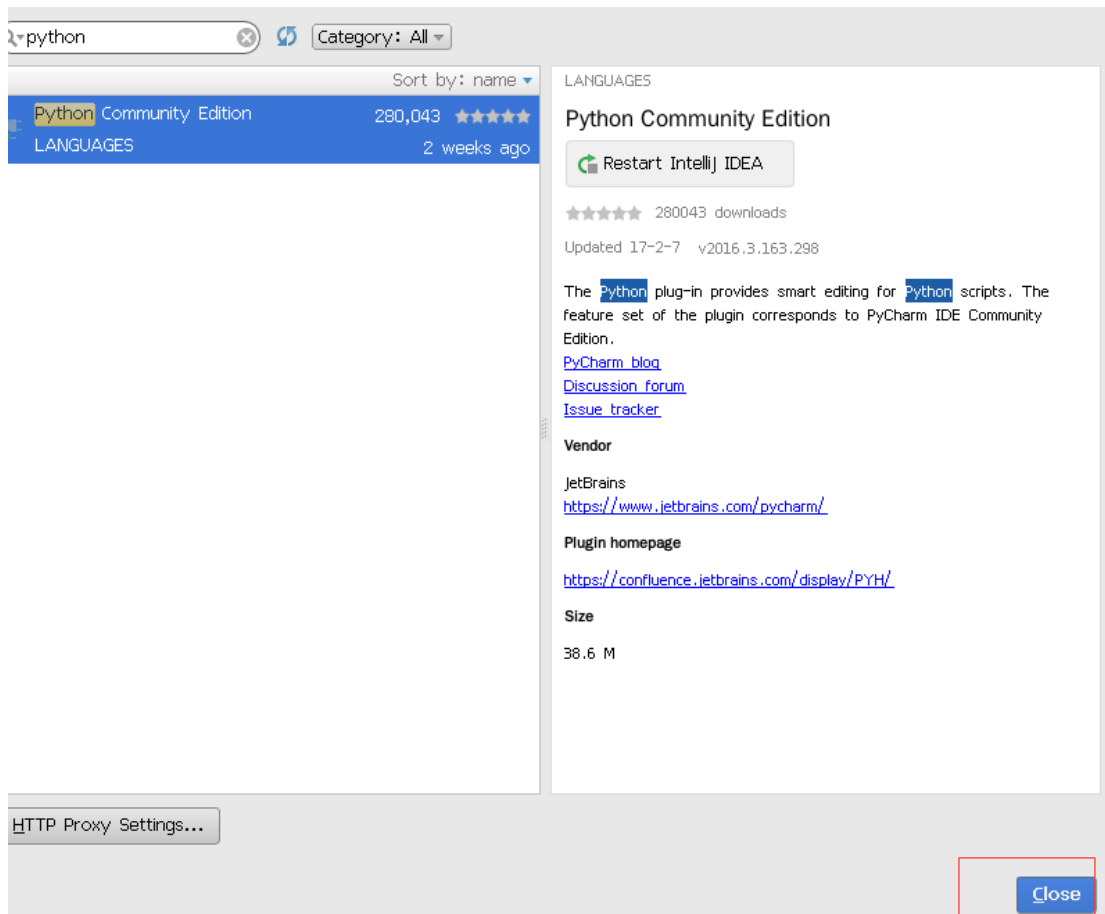
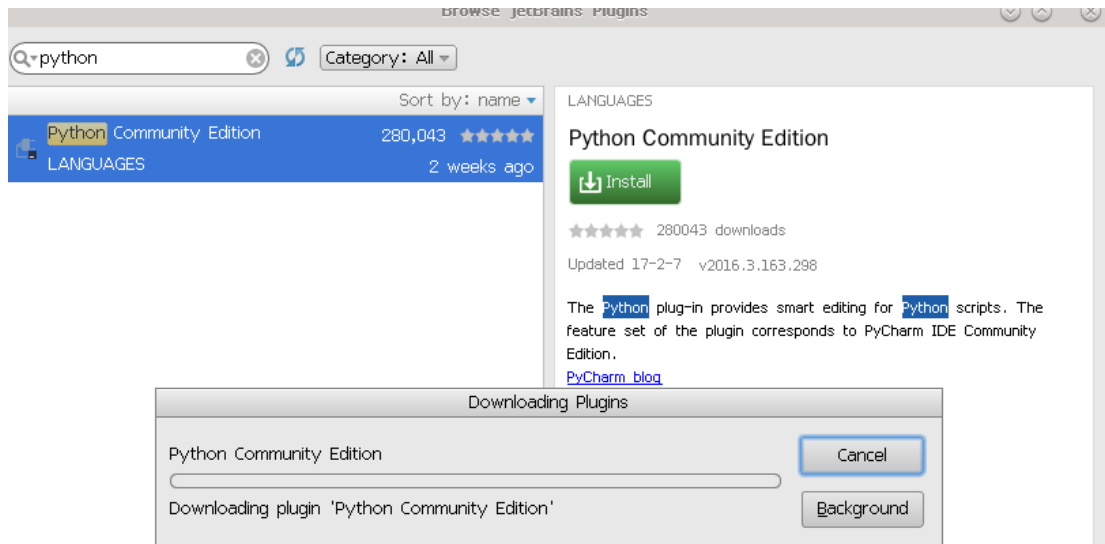
Version 2016.3.4

-  Create New Project
-  Import Project
-  Open
-  Check out from Version Control ▾

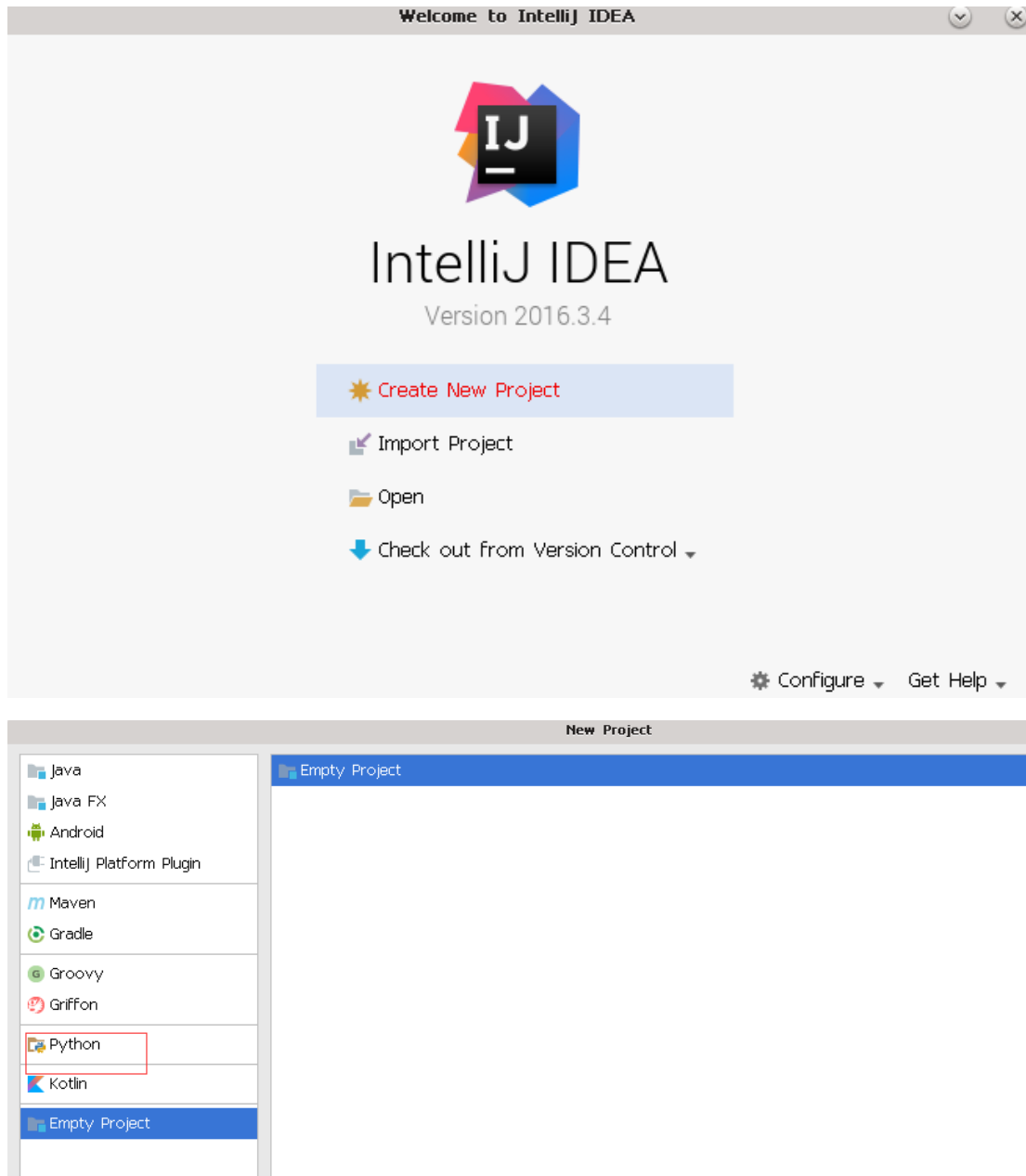
 **Configure** ▾    **Get Help** ▾

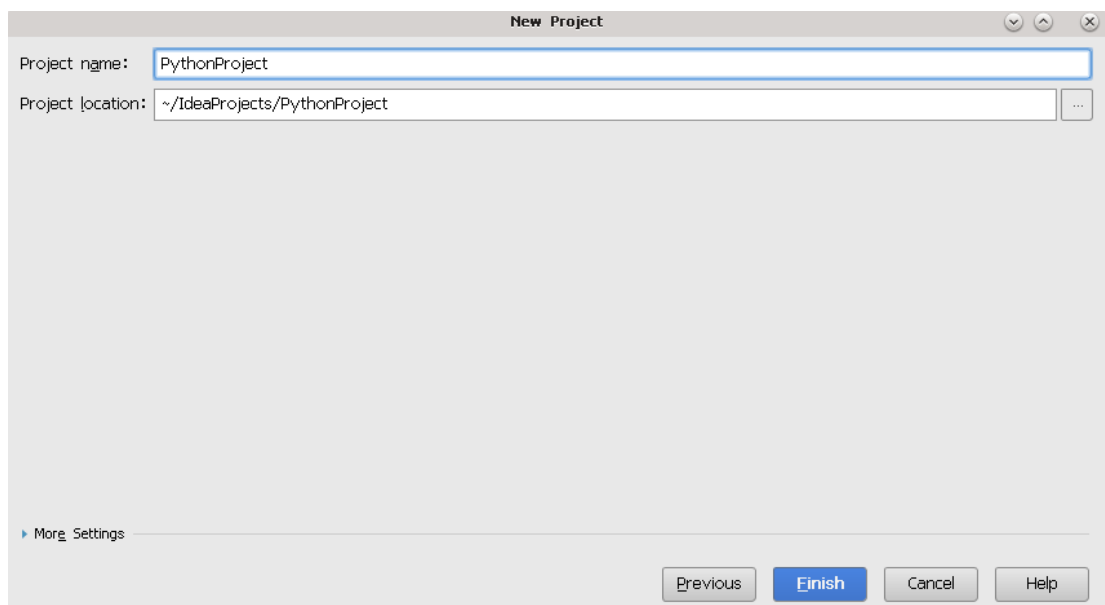
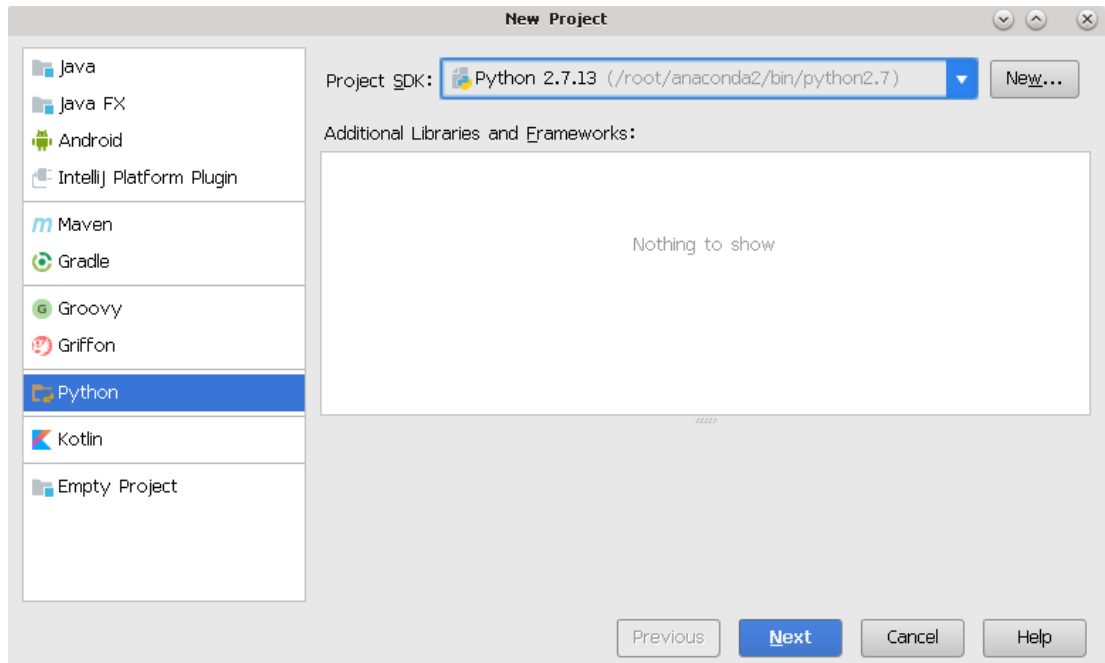
- Settings
- Plugins
- Import Settings
- Export Settings
- Settings Repository...
- Create Desktop Entry
- Check for Update
- Project Defaults ▸





创建 python 新项目

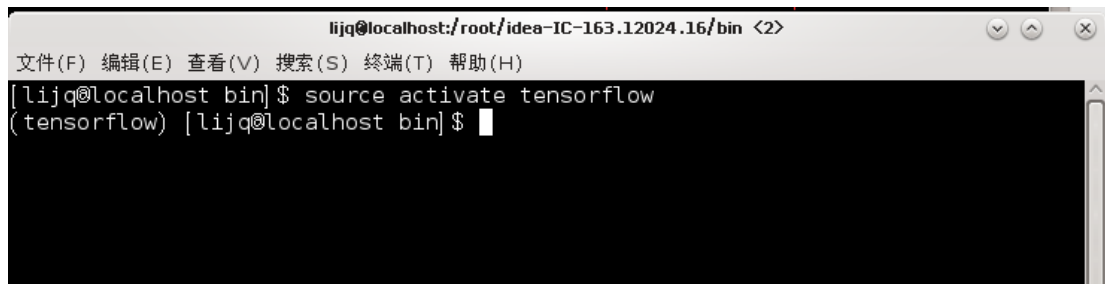




## 4.2 tensorflow 实例

使用前先打开 tensorflow 计算环境

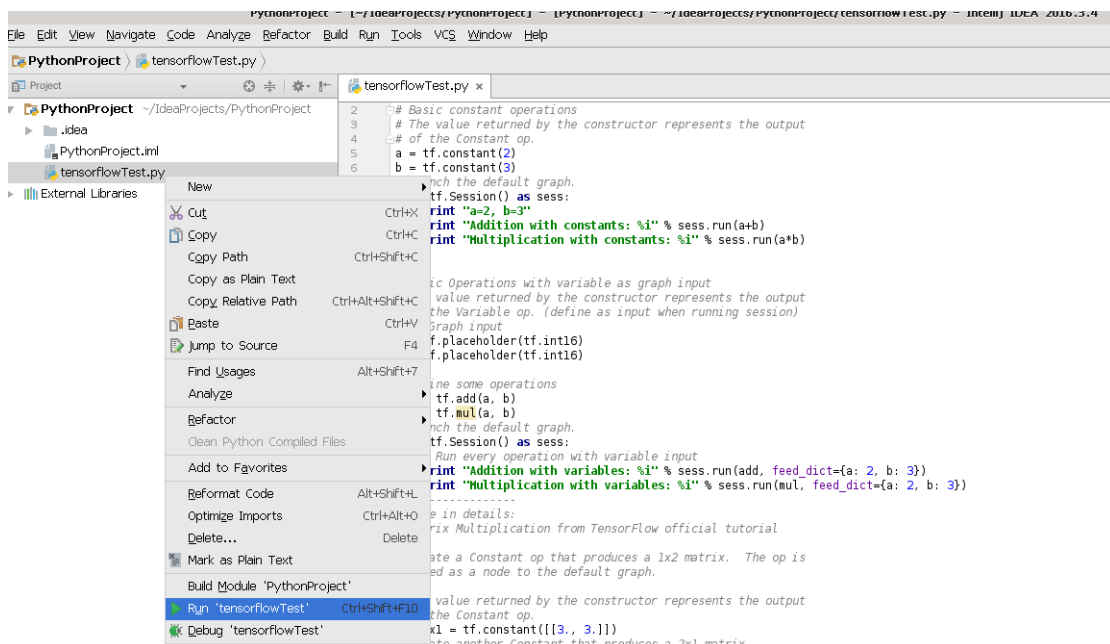
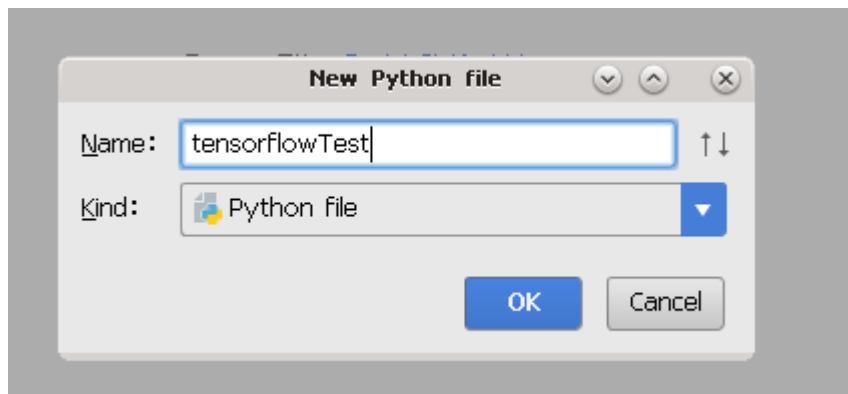
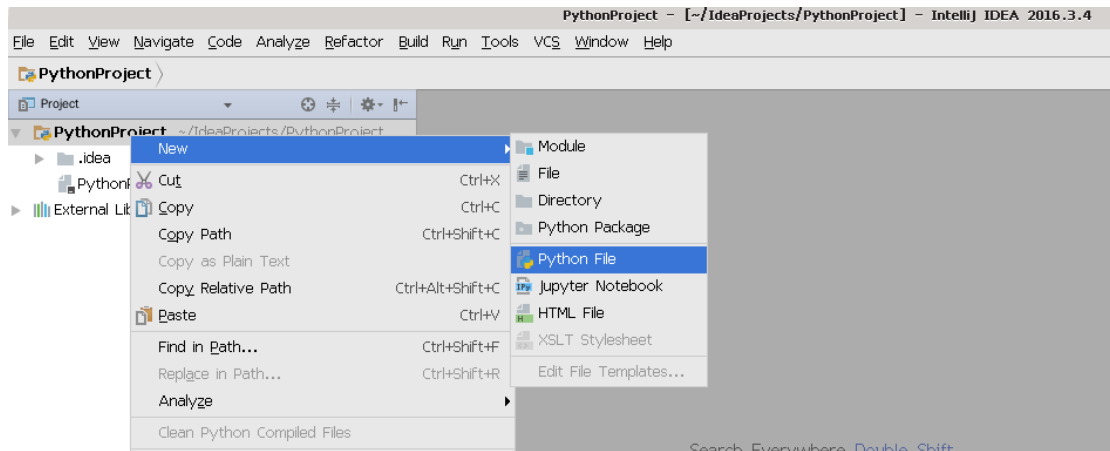
source activate tensorflow



退出环境 source deactivate

## 新建 python 文件

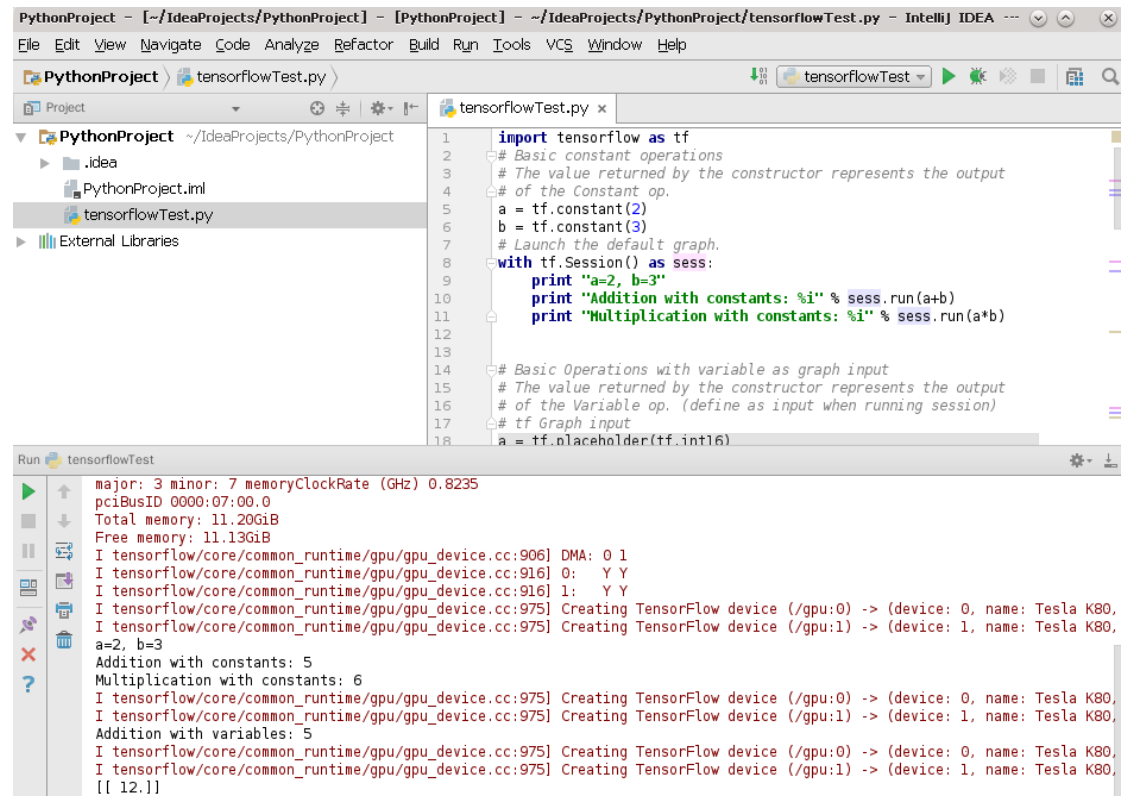
右键[新建]-[python file]-[ok]





## 运行 tensorflow 实例文件

右键[run]后运算过程和结果情况会在下面控制台打印出来



The screenshot displays the IntelliJ IDEA interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The project structure on the left shows a PythonProject with files like .idea, PythonProject.iml, and tensorflowTest.py. The main editor window shows the code for tensorflowTest.py, which imports tensorflow as tf, defines constants a=2 and b=3, and prints the results of a+b and a\*b. The Run console at the bottom shows the execution output, including system information like memory and GPU details, and the results of the TensorFlow operations: 'Addition with constants: 5' and 'Multiplication with constants: 6'.

```
1 import tensorflow as tf
2 # Basic constant operations
3 # The value returned by the constructor represents the output
4 # of the Constant op.
5 a = tf.constant(2)
6 b = tf.constant(3)
7 # Launch the default graph.
8 with tf.Session() as sess:
9     print "a=2, b=3"
10    print "Addition with constants: %i" % sess.run(a+b)
11    print "Multiplication with constants: %i" % sess.run(a*b)
12
13
14 # Basic Operations with variable as graph input
15 # The value returned by the constructor represents the output
16 # of the Variable op. (define as input when running session)
17 # tf Graph input
18 a = tf.placeholder(tf.int16)
```

Run tensorflowTest

```
major: 3 minor: 7 memoryClockRate (GHz) 0.8235
pciBusID 0000:07:00.0
Total memory: 11.20GiB
Free memory: 11.13GiB
I tensorflow/core/common_runtime/gpu/gpu_device.cc:906] DMA: 0 1
I tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 0: Y Y
I tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 1: Y Y
I tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:0) -> (device: 0, name: Tesla K80,
I tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:1) -> (device: 1, name: Tesla K80,
a=2, b=3
Addition with constants: 5
Multiplication with constants: 6
I tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:0) -> (device: 0, name: Tesla K80,
I tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:1) -> (device: 1, name: Tesla K80,
Addition with variables: 5
I tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:0) -> (device: 0, name: Tesla K80,
I tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:1) -> (device: 1, name: Tesla K80,
[[ 12.]]
```