

# 开始 Python 之旅

## 一、实验介绍

### 1.1 实验内容

本课程基于 *Python for you and me* 教程翻译制作，其中参考了 *Python tutorial* 和 *The Python Standard Library*，并对原教程的内容进行了改进与补充。

相关链接地址如下：

- *Python tutorial* : <http://www.pythondoc.com/pythontutorial3/index.html>  
(<http://www.pythondoc.com/pythontutorial3/index.html>)
- *Python for you and me* : <http://pymbook.readthedocs.io/en/latest/>  
(<http://pymbook.readthedocs.io/en/latest/>)
- *The Python Standard Library* : <https://docs.python.org/3/library/index.html>  
(<https://docs.python.org/3/library/index.html>)

*Python for you and me* 的法律声明有如下内容：

Permission is granted to copy, distribute, and/or modify this document under the terms of the Open Publication Licence, Version 1.0, or any later version.

### 1.2 实验知识点

- 使用交互模式的 Python3 解释器
- 简单使用 vim 编写 Python3 脚本
- 执行 Python3 脚本
- Python3 代码风格建议
- Python3 中使用注释
- 认识 Python3 模块

## 1.3 实验环境

- python3.5
- Xfce终端
- Vim

## 1.4 适合人群

本课程属于初级级别课程，不仅适用于那些有其它语言基础的同学，对没有编程经验的同学也非常友好。

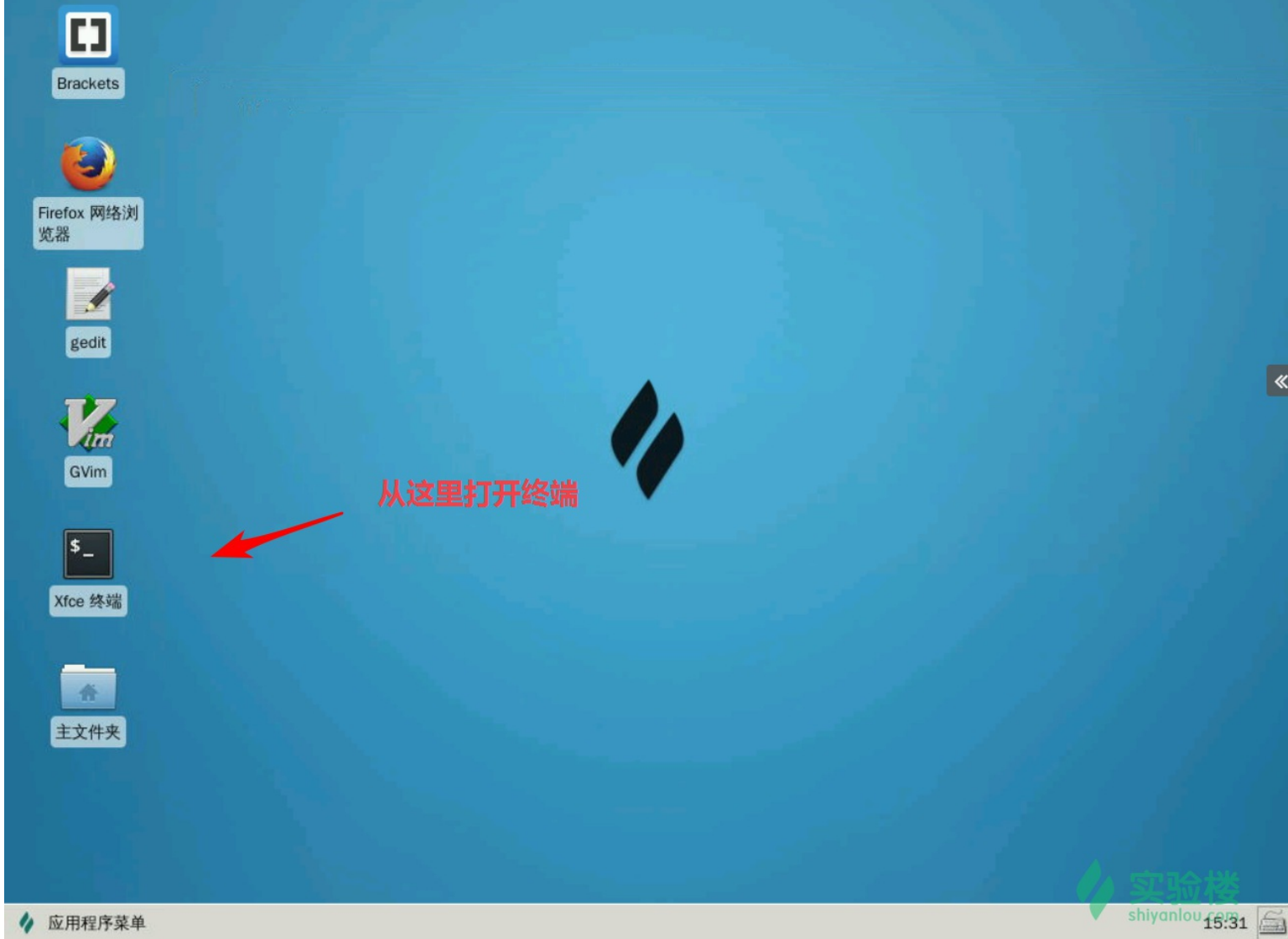
# 二、实验步骤

---

## 2.1 解释器

Python 是一个脚本语言，你可以在 Python 解释器中直接写代码或者将代码写到一个文件里，然后执行这个文件（即脚本文件）。

首先，我们从使用解释器开始，打开终端，键入 python3 并回车。这个时候解释器便工作于交互模式。



```
Terminal 终端 - python3
文件(E) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
shiyanlou:~/ $ python3 [16:13:38]
Python 3.5.2 (default, Jul 17 2016, 00:00:00)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

作为我们第一个 Python 代码，我们将打印经典的 "Hello World!"，如下所示：

```
>>> print("Hello World!")
Hello World!
```

```
Terminal 终端 - python3
文件(E) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
shiyanlou:~/ $ python3 [16:17:39]
Python 3.5.2 (default, Jul 17 2016, 00:00:00)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World")
Hello World
>>>
```

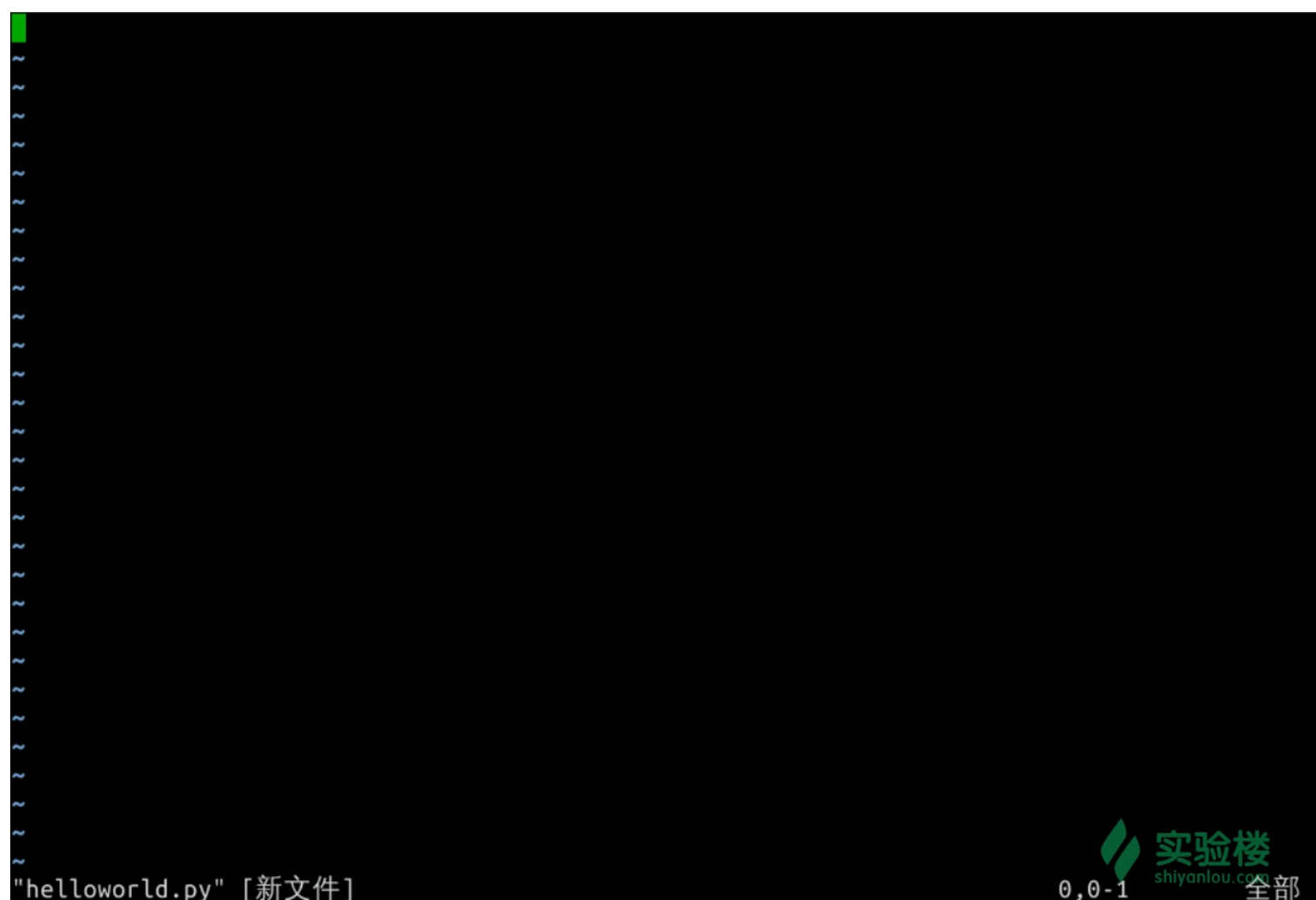
欣赏完自己的杰作后，按 `Ctrl + D` 输入一个 EOF 字符来退出解释器，你也可以键入 `exit()` 来退出解释器。

## 2.2 脚本文件

作为正式的程序员，你可能想要将上述的代码写到一个源文件。那么我们来创建一个 `helloworld.py` 文件，用你喜欢的任何编辑器来创建这个文件，我使用 Vim，你可以使用基于 GUI 的工具，如实验楼环境桌面上的 `gedit`。

我们建议使用 Vim，在这里我们简单的使用它，你可以在实验楼的课程 Vim 编辑器 (<https://www.shiyanlou.com/courses/2>) 里深入学习。

键入 `vim helloworld.py` 来启动 Vim 并编辑 `helloworld.py`，启动后不要乱按键。



然后按 `i` 键进入插入模式，此时你可以看到左下角有“插入”两个字，现在你可以键入下面的代码了。

```
#!/usr/bin/env python3
print("Hello World!")
```

其中第一行的前两个字符 `#!` 称为 *Shebang*，目的是告诉 shell 使用 Python 解释器执行其下面的代码。

```
#!/usr/bin/env python3
print("Hello World!")
```

-- 插入 --

2,22 实验楼  
shiyancelou.com 全部

然后按 `Esc` 键退出插入模式，再键入 `:wq` 回车，Vim 就会保存文件并退出。

要运行脚本文件 `helloworld.py`，还要为文件添加可执行权限：

```
$ chmod +x helloworld.py
```

然后执行脚本文件：

```
$ ./helloworld.py
```

```
shiyancelou:~/ $ chmod +x helloworld.py
shiyancelou:~/ $ ./helloworld.py
Hello World!
shiyancelou:~/ $
```

[17:12:38]  
[17:12:40]  
实验楼  
shiyancelou.com [17:12:56]

## 2.3 代码风格建议

在 Python 中，空格很重要，我们使用空格将标识符区分开，行开始处的空格我们称之为缩进，如果你的缩进是错误的，Python 解释器会抛出错误。例如下面情形：

```
>>> a = 12
>>>   a = 12
      File "<stdin>", line 1
        a = 12
        ^
IndentationError: unexpected indent
```

这是因为第二行开始处出现了多余的空格，所以需要经常检查代码的缩进。

如果你在代码中混用制表符和空格，这种缩进错误会很常见。

所以如果是用空格，就一直用空格缩进，不要使用制表符。

建议遵守以下约定：

- 使用 4 个空格来缩进
- 永远不要混用空格和制表符
- 在函数之间空一行
- 在类之间空两行
- 字典，列表，元组以及参数列表中，在 ， 后添加一个空格。对于字典，：后面也添加一个空格
- 在赋值运算符和比较运算符周围要有空格（参数列表中除外），但是括号里侧不加空格：a = f(1, 2) + g(3, 4)

## 2.4 注释

注释是一些文本用来解释此处代码是做什么的，写注释是为了使他人更容易理解代码。

Python 的注释以 # 字符开始的，在 # 字符到行尾之间的所有东西都被程序忽略为注释，也就说对程序没有任何影响。

```
>>> # 这是一个注释
>>> # 下面这一行是求两数之和
>>> a = 12 + 34
>>> print(a) # 这里也是注释 :)
```

注释主要面向的人群是开发人员和维护代码库的人员，所以如果你写了一些复杂代码，你应该写足够的注释在里面，以便让其他任何人能够通过你的注释读懂代码。你应该总是在 `#` 后跟一个空格，然后再写注释。你也可以做一些标准的注释，像下面这样。

```
# FIXME -- fix these code later
# TODO -- in future you have to do this
```

## 2.5 模块

模块是包含了我们能复用的代码的文件，包含了不同的函数定义，变量。模块文件通常以 `.py` 为扩展名。

Python 本身在默认安装时就带有大量的模块。我们之后将会用到其中的一部分。在使用模块前先导入它。

```
>>> import math
>>> print(math.e)
2.71828182846
```

我们会在模块这个实验里学习更多有关模块的知识。

## 三、总结

完成本实验后，应当明白如何使用交互模式的 Python3 解释器，如何简单使用 vim 编写脚本文件并执行这个脚本文件，要特别注意的是，Python 使用缩进来定义语句块，缩进是 Python 语法的一部分，这与其它大多数语言不同。代码风格在本课程的第14个实验 *PEP8 代码风格指南* 有更为详细的讲述。注释的话，能使用英文建议使用英文。模块同样会在之后的实验里深入学习。

\*本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播。

下一节：变量和数据类型 (</courses/596/labs/2037/document>)