

第一个程序

你应该在习题 0 上花了不少的时间，学会了如何安装文本编辑器，运行文本编辑器，以及如何运行终端。如果你还没有完成这些练习，请不要继续往下进行了，否则你不会觉得很好过的。写在习题开头警告你不要跳过前面内容的警示本书中仅此一次，切记切记。

将下面的内容写到一个文件中，取名为 `ex1.py`。这种命名方式很重要，Python 文件最好以 `.py` 结尾。

`ex1.py`

```
1 print "Hello World!"
2 print "Hello Again"
3 print "I like typing this."
4 print "This is fun."
5 print 'Yay! Printing.'
6 print "I'd much rather you 'not'."
7 print 'I "said" do not touch this.'
```

如果你使用的是 Mac OSX 下的 TextWrangler，那你的文本编辑器大致是图 1-1 所示的这个样子。

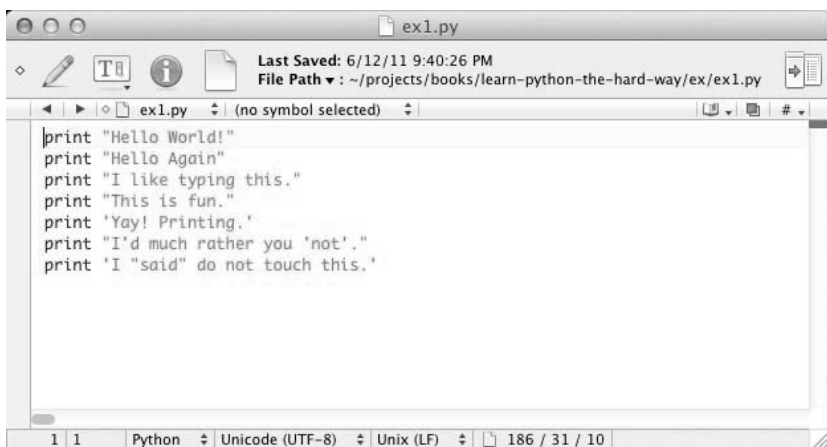


图 1-1

如果你是在 Windows 下使用 Notepad++，那你看到的应该是图 1-2 所示的这个样子。

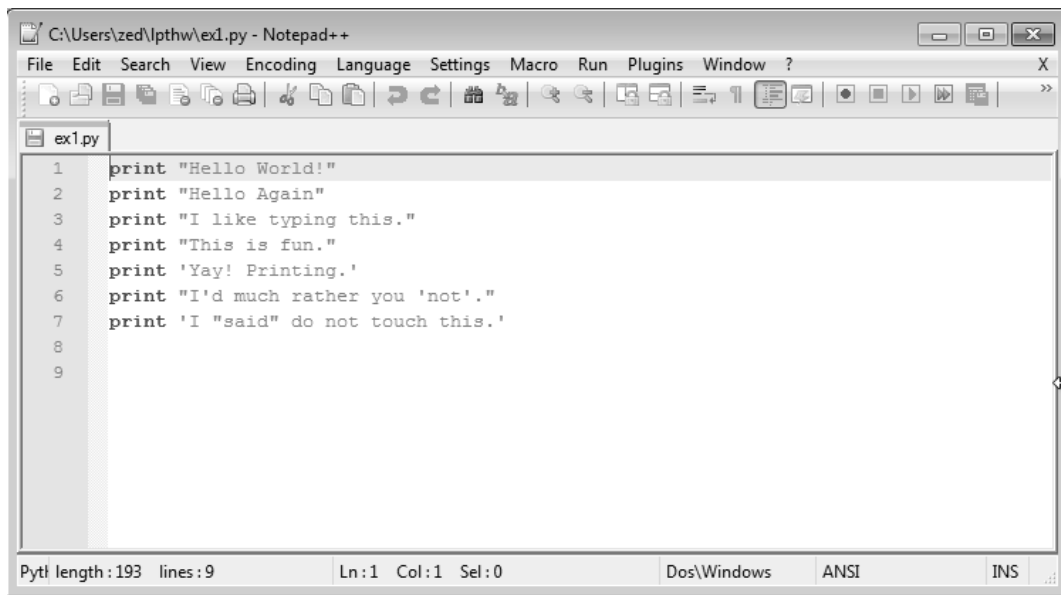


图 1-2

别担心编辑器长得是不是一样，关键是以下几点。

1. 注意我没有键入左边的行号。这些是额外加到书里边的，以便对代码具体的某一行进行讨论。例如“参见第 5 行……”你无需将这些也写进 Python 脚本中去。
2. 注意截图中开始的 print 语句，它和代码范例中是完全一样的。这里要求你做到“完全相同”，仅做到“差不多相同”是不够的。要让这段脚本正常工作，代码中的每个字符都必须完全匹配。当然，你的编辑器显示的颜色可能不一样，这并不重要，只有你键入的字符才是重要的。

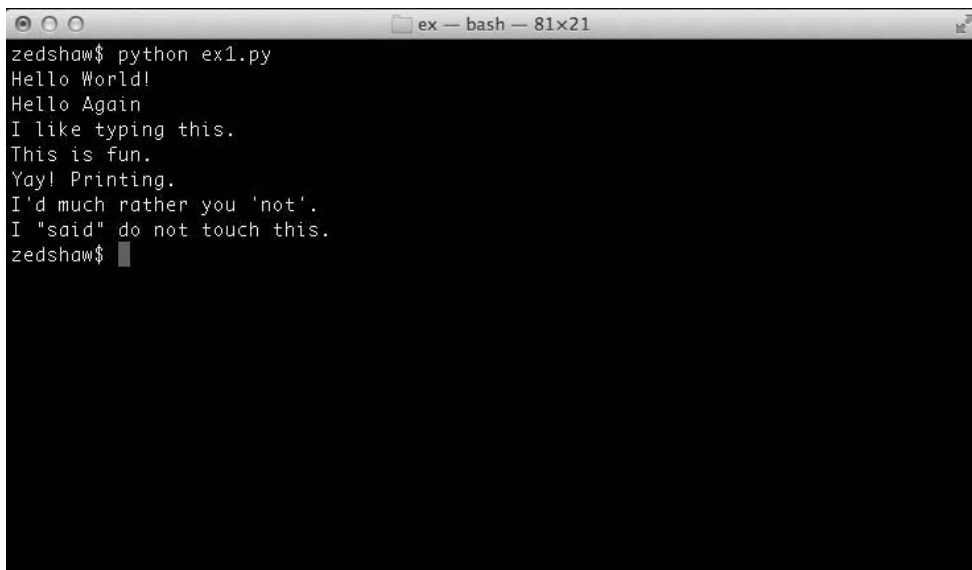
然后需要在终端通过输入以下内容来运行这段代码：

```
python ex1.py
```

如果你写对了，你应该看到和下面一样的内容。如果不一样，就是你哪儿弄错了。不是计算机出错了，计算机不会错。

应该看到的结果


在 Mac OSX 的终端下面应该看到图 1-3 所示的这个样子。

A terminal window titled "ex - bash - 81x21" showing the execution of a Python script. The prompt is "zedshaw\$". The output consists of several lines of text: "Hello World!", "Hello Again", "I like typing this.", "This is fun.", "Yay! Printing.", "I'd much rather you 'not'.", and "I 'said' do not touch this.". The prompt "zedshaw\$" is shown again at the end.

```
zedshaw$ python ex1.py
Hello World!
Hello Again
I like typing this.
This is fun.
Yay! Printing.
I'd much rather you 'not'.
I "said" do not touch this.
zedshaw$
```

图 1-3

在 Windows 的 PowerShell 下应该看到图 1-4 所示的这个样子。

A Windows PowerShell window titled "Windows PowerShell" showing the execution of a Python script. The prompt is "PS C:\Users\zed\lpthw>". The output is identical to Figure 1-3: "Hello World!", "Hello Again", "I like typing this.", "This is fun.", "Yay! Printing.", "I'd much rather you 'not'.", and "I 'said' do not touch this.". The prompt "PS C:\Users\zed\lpthw>" is shown again at the end.

```
PS C:\Users\zed\lpthw> python ex1.py
Hello World!
Hello Again
I like typing this.
This is fun.
Yay! Printing.
I'd much rather you 'not'.
I "said" do not touch this.
PS C:\Users\zed\lpthw>
```

图 1-4

你也许会看到 `python ex1.py` 前面显示的用户名、计算机名及其他一些信息不一样，这不是问题，重要的是你键入了命令，而且看到了相同的输出。

如果有错误，你会看到与下面类似的错误信息：

```
$ python ex/ex1.py
File "ex/ex1.py", line 3
    print "I like typing this."
                                   ^
SyntaxError: EOL while scanning string literal
```

你应该学会看懂这些内容，这是很重要的一点，因为你以后还会犯类似的错误。就是现在的我也会犯这样的错误。让我们一行一行来看。

1. 首先我们在终端输入命令来运行 `ex1.py` 脚本。
2. Python 告诉我们 `ex1.py` 文件的第 3 行有一个错误。
3. 然后这一行的内容被显示出来。
4. 然后 Python 显示一个插入符 (^) 符号，用来指示出错的位置。注意到少了一个双引号 (") 了吗？
5. 最后，它显示一个“语法错误”(SyntaxError)，告诉你究竟是什么样的错误。通常这些错误信息都非常难懂，不过你可以把错误信息的内容复制到搜索引擎里，然后你就能看到别人也遇到过这样的错误，而且你也许能找到如何解决这个问题。

注意 如果你来自另外一个国家，而且你看到关于 ASCII 编码的错误，那就在你的 Python 脚本的最上面加入下面这一行：

```
# -*- coding: utf-8 -*-
```

这样你就在脚本中使用了 Unicode UTF-8 编码，这些错误就不会出现了。

附加练习

每个习题都有附加练习要完成。附加练习里边的内容是供你尝试的。如果你觉得做不出来，可以暂时跳过，过段时间再回来做。

在这个习题中，试试下面几件事儿。

1. 让你的脚本再多打印一行。
2. 让你的脚本只打印一行。
3. 在一行的起始位置放一个“#”字符。它的作用是什么？自己研究一下。

从现在开始，除非特殊情况，我将不再解释每个习题的工作原理了。

注意 # (octothorpe) 有很多的英文名字, 如 pound (英镑符)、hash (电话的#键)、mesh (网) 等。选一个你觉得酷的就行了。

常见问题回答

这些是本书在线的时候收到的真实的学生问题, 其中的一些问题你也有可能遇到, 所以我就把这些问题以及它们的答案都搜集在这里了。

我可不可以使用 IDLE?

不行。你应该使用 OSX 的终端或者 Windows 的 PowerShell, 和我这里演示的一样。如果你不知道如何用它们, 可以去阅读附录的“命令行快速入门”。

怎样让编辑器显示不同颜色?

编辑之前先将文件保存为 .py 格式, 如 ex1.py, 后面编辑时你就可以看到各种颜色了。

运行 ex1.py 时看到 `SyntaxError: invalid syntax`。

你也许已经运行了 Python, 然后又在 Python 环境下运行了一遍 Python。关掉并重启终端, 重来一遍, 只键入 `python ex1.py` 就可以了。

遇到错误信息 `can't open file 'ex1.py': [Errno 2] No such file or directory`。

你需要在自己创建文件的目录下运行命令。确保你事先使用 `cd` 命令进入了这层目录下。假如你的文件存在 `lpthw/ex1.py` 下面, 那你需要先执行 `cd lpthw/`再运行 `python ex1.py`, 如果你不明白该命令的意思, 那就去看看第一个问题中提到的“命令行快速入门”吧。

怎样在代码中输入我们国家的语言文字?

确认在文件开头加入了这行: `# -*- coding: utf-8 -*-`。

我的文件无法运行, 它直接回到了提示符, 没有任何输出。

很有可能是你把代码做了字面理解, 认为 `print "Hello World!"`就是让你在文件中打印 `"Hello World!"`, 于是你没有输入 `print`。你的代码应该和我的一模一样。我的每行里边都有 `print`, 你的也要确保都有, 这样代码才能正常运行。