

# ***Python Baisc Tutorial***

*Kevin Stephen*



# Preface | 前言

这个笔记是我上课内容的总结，也是我对知识点的梳理



# 目 录

<b>Preface   前言</b>	<b>i</b>
<b>Contents   目录</b>	<b>I</b>
<b>1 begin   入门</b>	<b>1</b>
1.1 Hello, world!   你好, 世界! . . . . .	1
1.1.1 What is <i>Programming</i> ?   什么是编程? . . . . .	1
1.1.2 What is <i>Python</i> ?   什么是 <i>Python</i> ? . . . . .	1
1.1.3 first code   第一行代码 . . . . .	2
<b>Postscript   后记</b>	<b>I</b>
<b>Figures   图片清单</b>	<b>III</b>



# Chapter 1

## begin | 入门

*Python* 作为一门编程语言，考虑到其语法和实用性，很适合作为新手学习编程的第一门语言，不过在开始正式的学习之前，有一些概念需要先说明白。

### 1.1 Hello, world! | 你好，世界！

*Hello World is a very simple program,  
but it contains all the basic elements of  
computer science.*

— **Brian Kernighan**

#### 1.1.1 What is *Programming*? | 什么是编程？

关于“什么是编程？”，到网上去搜索这个问题会可以看到各式各样的答案，但是这么多答案对于新手而言，未免太过繁杂，更别说还有些答案间可能的矛盾，增加了理解的成本，吓跑了许多“初窥门径”的人。

对于这个问题，我的答案非常简单：

编程就是通过编程语言让计算机帮我们完成“计算”

这里的“计算”指的不仅仅是数学上的运算，还包括了各种各样的数据处理与生成。

#### 1.1.2 What is *Python*? | 什么是 *Python*？

那么 *Python* 在这里是什么作用呢？其实 *Python* 就是人与计算机间的一个翻译官！

电脑其实是无法直接听懂我们说话的，我们想要计算机帮我们做事，就必须要用某种方式让电脑听懂我们的要求，这个方式就是编程，其中的媒介就是编程语言，而 *Python* 正是一门编程语言。

关于 *Python* 的正式介绍如下：

Python 是一种解释型、交互式、面向对象的编程语言。它包含了模块、异常、动态类型、高层级动态数据类型以及类等特性。在面向对象编程以外它还支持多种编程范式，例如过程式和函数式编程等。Python 结合了超强的功能和极清晰的语法。它带有许多系统调用和库以及多种窗口系统的接口，并且能用 C 或 C++ 来进行扩展。它还可用作需要可编程接口的应用程序的扩展语言。最后，Python 非常易于移植：它可以在包括 Linux 和 macOS 在内的许多 Unix 变种以及 Windows 上运行

— *Python Docs*

总之，Python 是一门非常厉害的编程语言，我们可以用 Python 做非常多的事情。

### 1.1.3 first code | 第一行代码

我们学习还是要脚踏实地，所以我们的第一个 Python 程序 — 按照传统 — 就是 *Hello World*。

```
1 print("Hello, world")
```

对于这个程序，我们可以使用一个任意一个文本编辑器创建一个文本，要注意文件的后缀要是 `.py`，然后在其中输入这些内容后保存，然后执行 `python3 path/to/file` 就好。



```
hello-world -- zsh -- 80x11
(dev) kevinstephen@ksmac> nano first-code.py
(dev) kevinstephen@ksmac> bat first-code.py

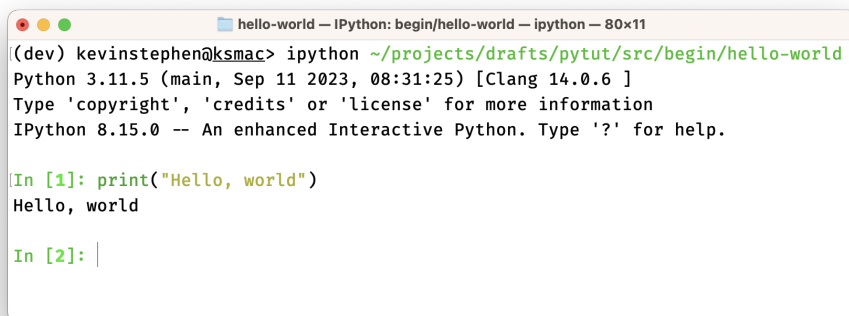
File: first-code.py
1 print("Hello, world")

(dev) kevinstephen@ksmac> python3 first-code.py
Hello, world
(dev) kevinstephen@ksmac> | ~/projects/drafts/pytut/src/begin/hello-world
```

图 1-1 使用文本模式

亦或者可以打开终端程序输入 `python3` 或者 `ipython`，然后输入内容。对于后一种方法，我们一输入内容，就可以看到“输出”，这种就被称为 *REPL*，全称是“*read evaluate print loop*”，也就是所谓的“交互模式”，对应前一种创建一个文本的方式就称为“文本模式”。



A screenshot of a terminal window titled "hello-world — IPython: begin/hello-world — ipython — 80x11". The terminal shows a user running the command `ipython ~/projects/drafts/pytut/src/begin/hello-world`. The output displays the IPython environment version (8.15.0) and the Python version (3.11.5). The user then enters `print("Hello, world")`, and the output is `Hello, world`. The prompt `In [2]:` is visible at the bottom.

```
(dev) kevinstephen@ksmac> ipython ~/projects/drafts/pytut/src/begin/hello-world ]
Python 3.11.5 (main, Sep 11 2023, 08:31:25) [Clang 14.0.6 ]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.15.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print("Hello, world")
Hello, world

In [2]: |
```

图 1-2 使用交互模式



# Postscript | 后记

这是后记



# 插图清单

1-1 使用文本模式 . . . . .	2
1-2 使用交互模式 . . . . .	3

