PYTHON®
Scripting for ArcGIS

>>> 面向ArcGIS的Python脚本编程

[美] Paul A. Zandbergen 著
李明巨 刘昱君 陶旸 张磊 译

人民邮电出版社
北京

图书在版编目（CIP）数据

面向ArcGIS的Python脚本编程 /（美）赞德伯根(Zandbergen, P. A.）著 ；李明巨等译. -- 北京 ：人民邮电出版社，2014.11
ISBN 978-7-115-37019-8

Ⅰ. ①面… Ⅱ. ①赞… ②李… Ⅲ. ①地理信息系统—应用软件 Ⅳ. ①P208②TP311.56

中国版本图书馆CIP数据核字(2014)第216743号

版权声明

Python Scripting for ArcGIS by Paul A. Zandbergen

Simplified Chinese translation copyright ©2014 by Posts and Telecommunications Press.

Copyright ©2013 Esri.

本书原版由 **Esri** 公司出版。本书中文简体字版由 **Esri** 公司授权人民邮电出版社独家出版发行。未经出版者书面许可，对本书的任何部分不得以任何方式复制或抄袭。

版权所有，侵权必究。

◆ 著 [美] Paul A. Zandbergen
译 李明巨 刘昱君 陶 旸 张 磊
责任编辑 陈冀康
责任印制 彭志环 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：800×1000 1/16
印张：18.5
字数：348 千字 2014 年 11 月第 1 版
印数：1－3 000 册 2014 年 11 月河北第 1 次印刷
著作权合同登记号 图字：01-2013-9295 号

定价：49.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315

目　录

第一部分　Python 和地理处理的相关概念

第 1 章　Python 简介……2

1.1　引言……2
1.2　Python 的特点……2
1.3　脚本语言和程序语言……3
1.4　ArcGIS 的脚本语言……3
1.5　Python 的历史和版本……4
1.6　关于本书……5
1.7　Python 脚本实例……7
1.8　Python 脚本编辑器……10
本章要点……16

第 2 章　ArcGIS 中的地理处理……17

2.1　引言……17
2.2　地理处理……17
2.3　ArcObjects……18
2.4　工具箱和工具……19
2.5　工具的类型……21
2.6　工具对话框……21
2.7　环境设置……25
2.8　批处理……27
2.9　模型和 ModelBuilder……30
2.10　运行脚本……34
2.11　运行脚本工具……37

2.12 模型转为脚本……38
2.13 定时运行脚本……40
本章要点……41
第 3 章 Python 窗口……43
3.1 引言……43
3.2 Python 窗口简介……43
3.3 编写并运行脚本代码……44
3.4 获取帮助……46
3.5 Python 窗口选项……48
3.6 保存脚本……49
3.7 在 Python 窗口中加载代码……50
本章要点……51
第 4 章 Python 基础……52
4.1 引言……52
4.2 Python 文档和资源……52
4.3 数据类型与数据结构……53
4.4 数字……53
4.5 变量及其命名规则……54
4.6 语句和表达式……56
4.7 字符串……57
4.8 列表……58
4.9 Python 对象……59
4.10 函数……60
4.11 方法……62
4.12 处理字符串……62
4.13 处理列表……66
4.14 处理路径……70
4.15 模块……71
4.16 条件控制语句……73

4.17 循环语句……75
4.18 用户输入……77
4.19 注释……78
4.20 PythonWin 编辑器……79
4.21 编码规范……81
本章要点……82

第二部分 编写地理处理脚本

第 5 章 使用 Python 进行地理处理……86

5.1 引言……86
5.2 ArcPy 站点包……86
5.3 导入 ArcPy……86
5.4 使用旧版 ArcGIS……88
5.5 使用地理处理工具……89
5.6 使用地理处理工具箱……94
5.7 ArcPy 函数……96
5.8 ArcPy 类……98
5.9 环境设置……100
5.10 工具消息……101
5.11 查询许可……105
5.12 获取帮助……108
本章要点……110

第 6 章 访问空间数据……112

6.1 引言……112
6.2 检查数据的存在性……112
6.3 描述数据……113
6.4 列出数据……116
6.5 对列表使用 for 循环……119
6.6 操作列表……119

6.7 元组……120
6.8 字典……121
本章要点……124
第 7 章 处理空间数据……126
7.1 引言……126
7.2 使用游标访问数据……126
7.3 在 Python 中使用 SQL……131
7.4 处理表和字段名……132
7.5 解析属性表和字段名……134
7.6 处理文本文件……135
本章要点……142
第 8 章 处理几何图形……143
8.1 引言……143
8.2 处理几何对象……143
8.3 读取几何……144
8.4 处理多部分要素……147
8.5 处理有孔洞的多边形……150
8.6 写入几何……152
8.7 使用游标设置空间参考……154
8.8 使用地理处理工具处理几何对象……156
本章要点……157
第 9 章 使用栅格数据……158
9.1 引言……158
9.2 列出栅格数据……158
9.3 描述栅格属性……159
9.4 处理栅格对象……162
9.5 Spatial Analyst 模块……163
9.6 地图代数……164
9.7 ApplyEnvironment 函数……167

9.8 arcpy.as 模块中的类……168
9.9 NumPy 数组……171
本章要点……172

第三部分 执行地理处理任务

第 10 章 制图脚本……174

10.1 引言……174
10.2 ArcPy 的制图模块……174
10.3 地图文档……175
10.4 地图文档的属性和方法……176
10.5 数据框……177
10.6 图层……179
10.7 修复数据链接……183
10.8 页面布局元素……188
10.9 输出地图……191
10.10 打印地图……192
10.11 处理 PDF 文档……192
10.12 新建地图册……193
10.13 制图脚本样例……196
本章要点……198

第 11 章 程序调试与错误处理……199

11.1 引言……199
11.2 识别语法错误……199
11.3 识别异常……201
11.4 程序调试……202
11.5 程序调试技巧……208
11.6 异常的处理……208
11.7 触发异常……209
11.8 处理异常……211

11.9 处理地理异常……213
11.10 其他错误处理方法……215
11.11 常见错误……216
本章要点……217
第 12 章 创建 Python 类和函数……218
12.1 引言……218
12.2 创建函数……218
12.3 调用函数……221
12.4 添加代码到模块中……224
12.5 使用类……226
12.6 地理处理包……229
本章要点……231

第四部分 创建并使用脚本工具

第 13 章 创建自定义工具……234
13.1 引言……234
13.2 为什么要创建自定义工具……234
13.3 创建自定义工具的步骤……235
13.4 编辑脚本代码……240
13.5 工具参数……241
13.6 设置工具参数……246
13.7 脚本工具示例……251
13.8 自定义工具行为……253
13.9 处理消息……254
13.10 处理独立脚本和工具的消息……256
13.11 自定义进度对话框……256
13.12 在进程中运行脚本……259
本章要点……260
第 14 章 共享脚本工具……261
14.1 引言……261

14.2 工具共享的方法……261
14.3 软件许可……262
14.4 共享工具的文件夹结构……262
14.5 处理路径……263
14.6 查找数据和工作空间……266
14.7 创建地理处理包……268
14.8 内嵌脚本并设置工具密码……269
14.9 编写工具文档……270
14.10 示例工具：市场分析……274
本章要点……278

第一部分

Python 和地理处理的相关概念

第 1 章　Python 简介

第 2 章　ArcGIS 中的地理处理

第 3 章　Python 窗口

第 4 章　Python 基础

第1章 Python简介

1.1 引言

Python 是一门简单而又强大的编程语言。对于那些在编程方面有困难的人来说，Python 的出现或许是一个福音。

本章首先介绍 Python 的主要特点以及它在 ArcGIS 中的用途；其次介绍本书的结构以及书中的练习；随后会通过几个例子介绍 Python 的使用方式；最后介绍 Python 编辑器，它可以让使用者更方便地编写和组织代码。

1.2 Python 的特点

Python 的诸多特点使它可以作为 ArcGIS 的脚本语言，这些特点包括以下几方面。

简单且易学。相对于其他高度结构化的编程语言（C++或 Visual Basic）而言，Python 更容易被掌握。它的语法简单，编程者将有更多的时间来解决实际问题，而不需要在学习 Python 语言上耗费太多精力。

免费且开源。Python 是一款免费并且开源的软件。用户可以自由地分发该软件的复本，查看和修改源代码，或者将其中一部分代码用在其他免费的程序里。Python 语言如此好用的一个重要原因在于它有一个十分活跃的用户社区，社区里的成员都积极地参与 Python 的开发和维护。正是由于 Python 是开源的，所以 Esri 才能够将 Python 部署在 ArcGIS 软件中。

跨平台。Python 支持包括 Windows、Mac、Linux 在内的各种平台。不同平台上的 Python 程序只需要做极小的改动甚至不改动，就能在其他平台上正常运行。由于 ArcGIS for Desktop

只能在 Windows 上运行，所以 Python 的这种特性在 ArcGIS 中似乎没有得到明显的体现，但是需要了解的是，Python 的用户之所以如此庞大，其中一个重要原因就是它跨平台的特性。

解释性。许多程序语言（例如 C++或 Visual Basic）需要将程序源文件转换成计算机可以理解的二进制代码。这就需要有适用于各种程序语言的编译器。而 Python 是一种解释性语言，它不需要编译就可以直接运行。这一特点使 Python 使用起来更加简单，并具有更强的移植性。

面向对象。Python 是一门面向对象的编程语言。面向对象的程序不再是功能的堆砌，而是由一系列相互作用的对象构建起来的。很多现代编程语言都支持面向对象的编程。ArcGIS 也支持面向对象的编程，从这个角度看，将 Python 作为 ArcGIS 的脚本语言是一个不错的选择。

1.3　脚本语言和程序语言

Python 作为一门程序语言，也常被称为脚本语言。那么，两者的区别在哪里呢？一般而言，脚本语言用于控制其他应用程序以实现任务自动化；而程序语言则是用于开发结构复杂、功能完备的应用程序。脚本语言是一种“粘合剂”，它可以将不同的组件组合在一起，从而实现新的功能。而系统语言既可以从头构建组件，也可以将组件组装成不同的应用程序。系统语言（例如 C++和.NET）通过计算机的低级图元和原始资源从头开始创建应用程序。脚本语言（例如 Python 和 Perl）使用计算机内置的高级函数并且回避了系统编程语言必须处理的一些细节。

以 Esri 为例，他们主要使用 C++语言开发 ArcGIS 软件。在 ArcGIS 软件中，所有的组件或对象被称为 ArcObjects。利用 C++既可以新建一个对象，也可以开发一个含有 ArcObjects 对象的应用程序。利用 Python 则既可以访问 ArcGIS 现有的功能，也可以通过组合相关函数来扩展 ArcGIS 的功能。

Python 既是一种脚本语言也是一种程序语言。与 C++相比，Python 并不用于底层开发，而是用来完成一些相对简单的脚本编程或一些高级程序设计项目。本书将重点介绍如何编写脚本来控制 ArcGIS 软件执行任务。虽然 Python 也可以用于应用程序的开发，但是本书并不介绍这方面的内容，而是介绍如何通过 Python 调用 ArcGIS 中现有的函数。

1.4　ArcGIS 的脚本语言

ArcGIS 9 引入了脚本处理技术，并支持多种脚本语言，包括 Python、VBScritp、JavaScritp、

JScritp 和 Perl。由于 ArcGIS 是基于组件对象模型（COM）构建起来的，又因为脚本语言是面向对象的，所以脚本语言可以访问 ArcGIS 中所有获得许可的函数，也包括所有的扩展模块。因此，脚本语言才可以高效地实现任务自动化，受到了程序员的青睐。虽然像 C++和.NET 这样的系统语言也能实现任务的自动化，但是与这些程序语言相比，脚本语言更加简洁。

目前，Python 已经成为 GIS 专业人员的一个基本工具。他们使用 Python 来扩展 ArcGIS 的功能并实现任务的自动化。几年前，或许内置于 ArcGIS 的 VBA 仍是最流行的编程工具。但是，几年后，Python 就作为 VBA 强有力的补充和替代品出现了。从 ArcGIS 10 开始，VBA 开发环境已经不再作为 ArcGIS 的默认安装程序，并且 Esri 公司也在积极劝阻用户使用 VBA。虽然应用程序的开发还是继续使用 C++或者.NET 语言，但是对于那些不从事程序开发的 GIS 专业人员来说，Python 的优势更为明显。

Python 虽然不是唯一一种可以在 ArcGIS 中使用的脚本语言，但是它一定是被最广泛使用的一种。这很大程度上是因为 Python 简单易操作，并且具有完备的程序开发功能。Python 被封装在 ArcGIS 的安装程序中，同时，Python 也被直接嵌入到 ArcGIS 的许多地理处理工具集中。例如，在 ArcGIS 的 Spatial Statistics 工具箱中，几乎全是 Python 的脚本工具（或许一般用户并不一定能注意到或用到）。ArcGIS10 已经将 Python 进一步整合到 ArcGIS 的用户界面里，而且 Esri 已正式将 Python 作为 ArcGIS 首选的脚本工具，并对 ArcGIS10.1 进行了升级以包括 Python2.7 版本。

1.5 Python 的历史和版本

Python 是由 Guido van Rossum 开发的，那时候他还在荷兰的国家数学和计算机科学研究院（CWI）。1991 年，Van Rossum 发布了第一版 Python。目前，虽然已经有很多志愿者参与到 Python 的维护与发展之中，但是 Van Rossum 在该领域仍然十分活跃。不同于其他编程语言，Python 只经历了为数不多的版本更新。

Python 中既有字符串、列表和字典等元素，也有其他更高级的元素，例如元类、生成器和列表推导式。Python 的稳定性和健壮性，反映了所有程序员对 Python 的需求，即 Python 中既需要有一些基本的元素，更需要有其他高级语言中常见的高级元素。

本书推荐用户在 ArcGIS 10.1 中使用 Python 2.7 版。虽然 ArcGIS 10.1 中自带了 Python 2.7.2 版，但是也可以免费下载和安装其他版本的 Python。目前 Python 2.x 版依旧表现十分出色，

而且也得到了广泛的使用，但是在未来新的版本中，还需要解决一些多年积累下来的问题，以使语言更加简洁。Python2.7 版和 Python3.x 版虽然有许多差异，但是两个版本中语言的基本结构没有改变。尽管已经有了 3.x 版，但是目前 ArcGIS10.1 仍使用 2.7 版。如果某一天 ArcGIS 将 3.x 版作为其首选版本，那么 2.7 版的脚本文件可以通过各种工具转换成 3.x 版的脚本文件，同样，3.x 版的脚本文件也可以转换成 2.7 版的脚本文件。

需要注意的是，尽管 ArcGIS 和 Python 被打包在一个安装包中，但是 Python 并不是由 Esri 开发的。Esri 之所以将 Python 作为首选的脚本语言部署在 ArcGIS 中，主要是由于它免费和开源的特性，而且为了更方便地使用 Python，Esri 还在 ArcGIS 中开发了相关的功能模块。然而，与其他 Python 应用相比，目前 Python 在 ArcGIS 中并没有得到广泛应用。因此，本书的另一个优点就是：在学习了本书的知识后，您不仅可以在 ArcGIS 中，也可以在其他项目中使用 Python。

1.6　关于本书

本书分为两个部分：

（1）纸质版，包括了使用 Python 的理论知识。

（2）数字版，包括了与书本对应的一套练习。

纸质版共有 14 章，分别解释了 Python 的结构和语法，介绍了如何为 ArcGIS 编写脚本。整本书中虽然有很多示例代码，但是书中不会一句一句地解释这些代码。随书的练习为实际操作提供了详细的指导。希望读者先学习章节的内容，然后完成对应的练习，再阅读下一章。大多数练习在最后都会有一点挑战，相信它可以提高您的能力。

本书的结构

第 1 章介绍了 Python 的概况，并教读者使用 Python 编辑器和 ArcGIS 中的 Python 窗口。

第 2 章介绍了 ArcGIS 地理处理环境，包括地理处理工具和 ModelBuilder 的使用。熟练的 ArcGIS 用户会比较熟悉 ArcGIS 中的大多数功能，但是回顾一下这些内容也是有好处的。了解 ArcToolBox 中工具的功能将有助于高效地编写脚本。Python 脚本和 ModelBuilder 往往需要结合在一起使用，所以要想充分发挥脚本的功能，就需要掌握如何使用 ModelBuilder。

第 3 章介绍了 ArcGIS 中的 Python 窗口，这是一个交互式的 Python 解释器。在 Python 窗口里，可以直接运行一行或多行代码。这些代码可以保存成一个脚本文件。同时，现有的脚本文件也可以加载到 Python 窗口里。

第 4 章介绍了 Python 语言的基础知识，包括它的语句、表达式、函数、方法、模块、控制语句。除此之外，还要练习编写脚本。本章涵盖了 Python 初学者需要掌握的基本语法，而熟练使用 Python 的用户将会对这一章比较熟悉。

第 5 章介绍了第一次引入 ArcGIS 10 的 ArcPy 站点包。ArcPy 包含了大量的模块、类和函数，它们将 ArcGIS 和 Python 有效地整合在一起。本章还介绍了数据路径、环境设置和使用许可等方面的内容。

第 6 章主要介绍了数据描述的相关方法，既包括数据描述和数据结构，也包括使用字符串、列表、元组和字典抽象化表达数据。

第 7 章介绍了如何处理空间数据，包括使用游标、检索数据、处理属性表和字段。学完本章后就可以通过 Python 执行 SQL 语句，从而查询空间数据。

第 8 章介绍了如何处理空间对象的几何属性，包括如何使用现有要素的属性以及如何创建新的要素。

第 9 章介绍了如何使用 Python 处理栅格数据，包括 ArcPy 的 Spatial Analyst 扩展模块的使用。这个模块专门用于对栅格数据进行一系列代数运算和空间分析。

第 10 章介绍了 ArcPy 的自动化制图模块，包括使用地图文档、数据框架、图层、输出和打印地图。

第 11 章介绍了常见的错误以及错误处理的方法。这些方法能保证脚本的稳定性。这一章还介绍了如何交互使用 PythonWin 的调试器来调试代码。

第 12 章介绍了如何在 Python 中创建自定义的函数和类。通过函数和类可以组织更为复杂的代码并且重用其中部分代码。

第 13 章介绍了如何使用 Python 创建类似于 ArcToolBox 中的工具。这种方法可以更好地共享脚本工具，同时也便于用户在 ModelBuilder 里添加脚本工具。

第 14 章介绍了共享脚本工具的方法，包括如何组织文件，定义路径、提供脚本说明文档等。

1.7　Python 脚本实例

本节将通过两个例子来说明如何使用 Python 编写脚本。这两个例子来源于 Esri 和 ArcGIS 的用户社区。列举脚本实例的原因之一就是为了让读者熟悉 Python 代码。学习编写代码的一个最好的方法就是练习现有的代码。现在不要求读者能够理解这些代码，但是这些例子将会让您了解本书即将讲些什么。

例一：确定错误地址

AddressError 脚本工具是由 Esri 公司的员工 Bruce Harold 开发的。该工具会在几条路段的范围内，检查路段中心线可能存在的错误。AddressError 工具输出的结果是包含了所有相关街道段可能错误的线状要素，属性表将描述错误相关信息。

该脚本工具类似于工具集中的工具。尽管它是用 Python 编写的，但是它的访问方式和其他地理处理工具一样，如图 1.1 所示。



图 1.1

脚本工具的对话框和其他常规地理处理工具的对话框一样，如图 1.2 所示。

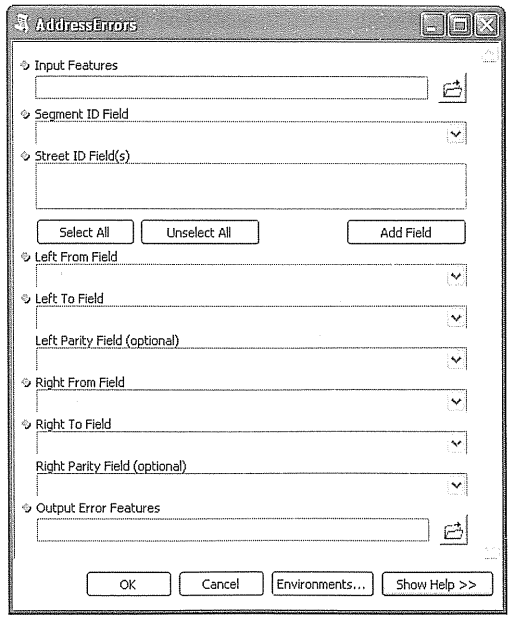


图 1.2

这个工具有 10 个输入参数，其中两个是可选的。输入要素通常是一个表示街道中心线的

线状要素，该要素的属性表存储了一系列代表街道号码范围的属性。输出结果是一个新的要素类。

这个脚本工具只调用了一个 Python 脚本文件，可以打开脚本文件查看这个工具是如何运行的。在 Python 编辑器中打开这个脚本时，会出现如图 1.3 所示的界面。

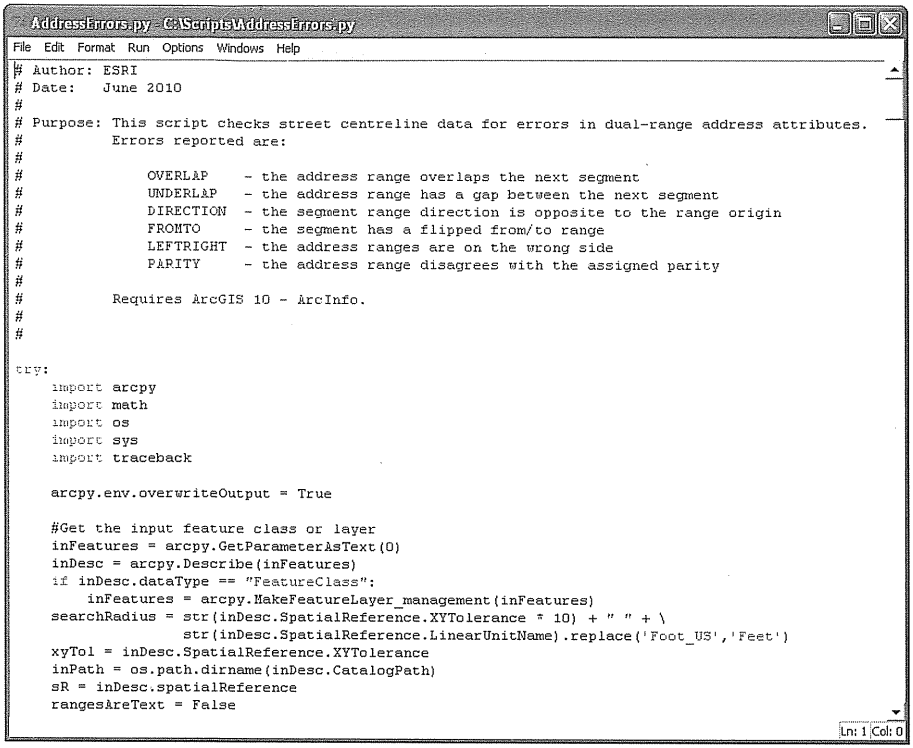


图 1.3

如果以前没有使用过 Python 或者其他编程语言，这些代码看起来可能有些吓人。不管怎样，这本书以及后面的练习主要是为了让读者熟悉 Python 的语法和逻辑，以便更好地使用 ArcGIS 执行相关任务。所以学完本书后，读者不仅能够理解上面的脚本，还能够写出相对复杂的脚本程序。

例二：利用 Huff 模型进行市场分析

Huff 模型的脚本工具是由 Esri 的员工 Drew Flater 开发的。这里有一份对这个工具的简单描述可以作为这个工具的使用说明：

Huff 模型是一个空间引力模型，它可以计算出不同位置的消费者光顾商店数据库中每个商店的概率。通过这些概率以及可支配收入、人口和其他变量就可以计算出不同位置的销售

潜力。每个位置的概率值还可用于生成研究区各个商店的地理范围及其市场范围。

这是一个相对复杂的脚本，但是跟第一个例子一样，它也是完全由 Python 编写的。这个脚本也可以作为工具箱的一个工具使用，如图 1.4 所示。

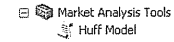


图 1.4

如图 1.5 所示，在该脚本工具对话框中有很多输入参数、输出参数以及分析设置选项，这也反映了 HuffModel 的特性。

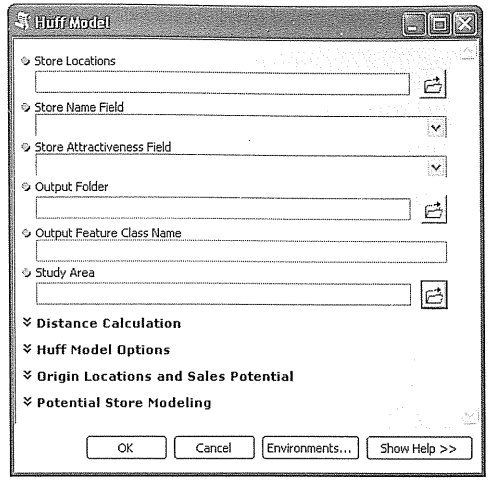


图 1.5

该脚本工具也提供了一个工具帮助文档，它详细地描述了工具是如何运行的。文档中介绍了输入参数、模型公式和输出参数，图 1.6 所示为帮助文档的一部分。

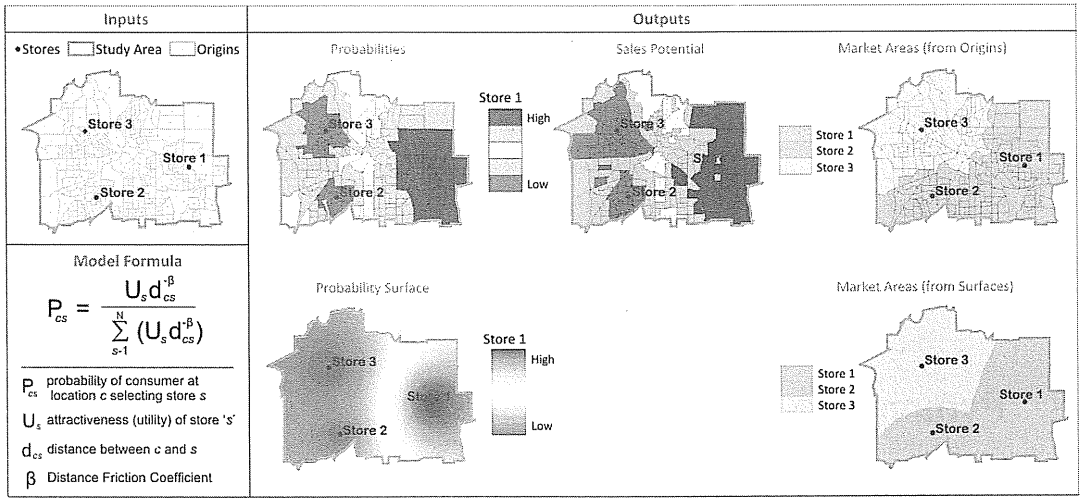


图 1.6

该工具的所有代码都保存在一个 Python 脚本文件中，图 1.7 所示为脚本文件的一部分。

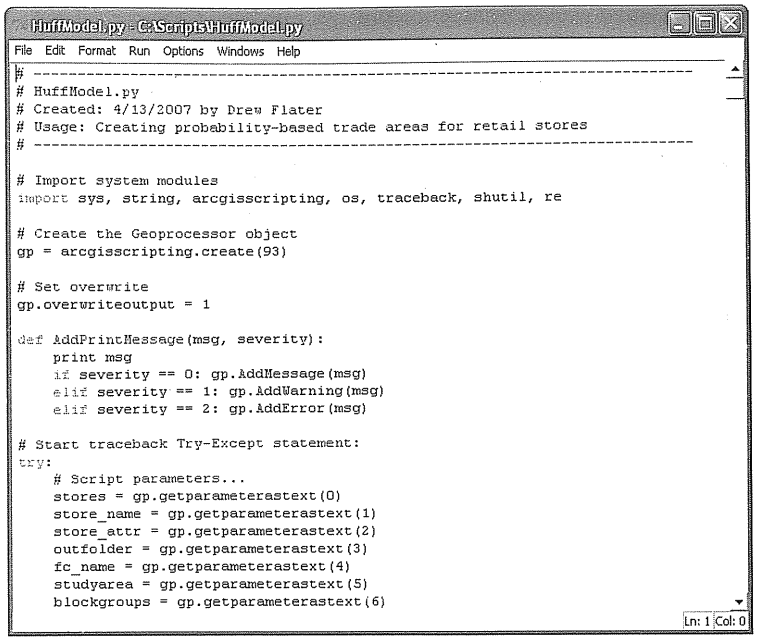


图 1.7

由于 Huff 模型十分复杂，所以这个脚本有 700 多行代码。但是，这些代码所涉及的大部分元素和那些简单脚本中的元素是相同的，所以只要熟悉这些元素，就可以真正读懂一些更复杂的代码。

在学习了如何编写 Python 脚本后，会发现坚持学习 Python 脚本的最好的方法之一就是练习现有的代码。练习示例代码也可以快速提高独立编写代码的能力。

注释：

两个实例来源于 ArcGIS 资源中心（http://resources.ArcGIS.com）的地理处理章节。具体的位置是在 Geoprocessing Model and Script Tool Gallery 中（http://resources.ArcGIS.com/gallery/file/geoprocessing）。

1.8 Python 脚本编辑器

一个脚本文件从本质上来说是一列可以由程序或脚本引擎运行的指令。脚本通常只是简

单的文本文件，它有一个特定的文件扩展名并且使用特定的脚本语法编写指令。一个通用的文本编辑器可以打开和编辑大多数的脚本文件。然而，使用一个专门的脚本编辑器既可以实现更多的编辑功能，也可以直接运行脚本。

使用 Python 的方式很多，最基本的方式就是使用所谓的命令行。如果使用过其他编程语言，那么可能熟悉这种方式。在 Windows 操作系统中，单击 Start 按钮，然后再单击 All Programs>ArcGIS>Python 2.7>Python(Command line)，就可以访问 Python 的命令行界面，如图 1.8 所示。

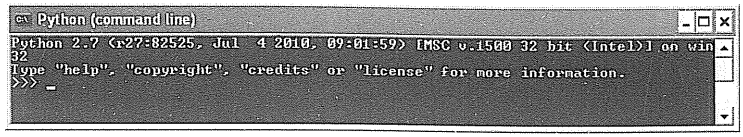


图 1.8

虽然这个界面可以实现 Python 的所有功能，但是它在编写和调试脚本方面的功能有限。所以，与 Python 命令行相比，Python 编辑器会显得更为高效。菜单形式的界面和工具，使得 Python 脚本编辑器在组织和调试脚本等方面更为方便。

Python 编辑器也被称为集成开发环境（IDE）。目前，市场上有许多不同的开发环境，其中既有开源软件，也有商业软件。一些 IDE 是针对特定平台（如 Windows、Mac 和 Linux）设计的，也有一些是针对特定编程语言（如 C++和.NET 语言）设计的。可以在 Python wiki 的页面（http://wiki.python.org/moin/PythonEditors）中了解更多有关 Python 编辑器的内容。使用何种编辑器很大程度上取决于个人习惯，资深的 Python 程序员都有自己最喜欢的编辑器。Python 在不同编辑器中的语法是相同的，这是 Python 的一个优势。

Python 默认的集成开发环境是 IDLE。在 Windows 操作系统中，单击 Start 按钮，然后再单击 All Programs>ArcGIS>Python 2.7>IDLE(Python GUI)，就可以访问 Python IDLE。GUI 表示图形用户界面。IDLE 也被称为 Python Shell，如图 1.9 所示。

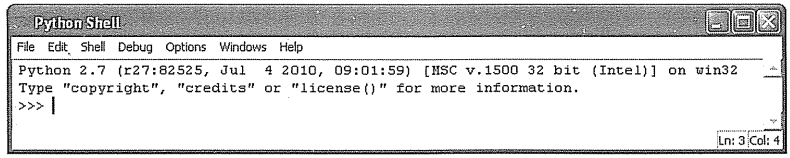


图 1.9

可以从 Help 菜单栏获取所有菜单项的介绍。在 Python Shell 的菜单栏中，可以单击 Help>IDLE Help 查看这些描述。更多有关 IDLE 的信息可以在 http://python.org/idle 上查阅。

注意 Python Shell 的最后一行是以“>>>”开始，它是交互式解释器的命令提示符。在这里可以输入代码并按 ENTER，然后交互式解释器会执行命令。准备好开始编写第一行 Python 代码了吗？

```
>>>print "Hello World"
```

按下 ENTER 后，就会输出如下所示的内容：

```
Hello World
>>>
```

为什么会输出上述内容呢？按下 ENTER 后，交互式解释器将读入输入的命令，在下一行打出字符串"Hello World"，然后再在下一行给出新的提示符，等待下一次的输入。这里的 print 语句是指在屏幕上输出文本。

现在应该知道为什么 Python 被称为一种解释性的编程语言了吧。当用户完成命令行的输入，然后按下 ENTER 时，命令行就会被解释，然后立即执行。

当用户输入一些 Python 解释器不能解释的语句时会出现什么情况呢？例如：

```
>>> I like eggs for breakfast
```

会立刻得到语法无效的提示：

```
StntaxError: invalid syntax
>>>
```

在交互式 Python 解释器中还有一些其他的功能：输入的代码会根据代码的性质显示成不同的颜色。例如，字符 print 会显示成橙色；字符串"Hello World"会显示成绿色。这种方法可以反映交互式解释器如何理解这些代码。在上面的例子中，橙色代表 Python 语句，绿色代表字符串。这种语法高亮显示的方法，对于检查语法错误十分有效。但是要注意，不同的 Python 编辑器会有不同的语法高亮显示规则，因此不要太过习惯于某一种颜色方案。

使用交互式解释器将会有助于学习 Python 语法的基础知识。用户可以立即知道结果，在输入代码的时候不需要再担心要保存代码。但是，如果要编写稍微复杂的代码，最好将它保存在一个脚本文件中。所以，需要牢记在交互式 Python 解释器中编写的代码并不意味着它已经被保存。

现在，您需要了解在交互式编译器中编写代码和编写脚本的不同之处。在 Python Shell 的菜单栏中，单击 File>New windows，就打开了一个名为 Untitled 的新窗口，如图 1.10 所示。这是一个脚本窗口，而且没有任何提示。

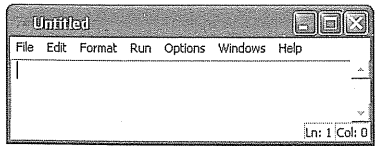


图 1.10

现在输入之前的一行代码：

```
print "Hello World"
```

按下 ENTER 后，任何事情都没发生——这是因为脚本本身并不具有交互性。一个脚本文件需要作为一个程序才可以被运行。脚本必须在保存后才能执行。在菜单栏单击 File>Save As，然后将脚本命名为 hello.py。.py 是脚本文件的后缀名。现在就可以运行脚本了。在菜单栏里，单击 Run>Run Module，字符串"Hello World"就会输出到交互式解释器中。

在编码过程中，最好同时打开交互式解释器和脚本窗口。在交互式解释器里可以立即运行某些代码或者检查某一行语法；而在脚本窗口中，可以保存已经写好的代码，并继续编写代码。有时候，也可以通过交互式解释器输出的结果来测试脚本是否正确。

在 Windows 平台上被广泛使用的一个 Python 编辑器是 PythonWin。在本书的后续部分，将使用 PythonWin 作为默认的脚本编辑器。不管什么编辑器，它们使用的语法是相同的，主要的区别在于编写、组织、测试代码的方式。

注释：

虽然 Python 在安装 ArcGIS 时被默认安装，但是 PythonWin 却没有。在随书的练习 01 中，会有 PythonWin 的安装说明。

PythonWin 界面如图 1.11 所示。在默认情况下，它打开的是一个被称为交互式窗口的交互式解释器，类似于 Python IDLE。

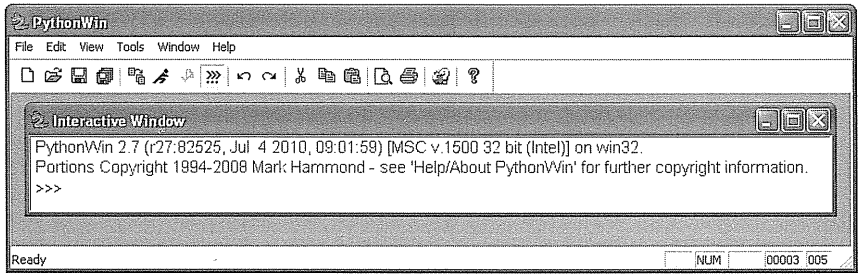


图 1.11

在 PythonWin 菜单栏里，单击 File > New > Python Script，就可以新建一个脚本。调节脚本窗口的大小，以便可以同时看见交互式窗口和脚本窗口。在脚本窗口菜单栏中，单击 File > Save As，就可以保存脚本。在下面的例子中，脚本被保存为 hellowin.py。在脚本窗口中，输入如下代码：

```
print "hello world"
```

在脚本窗口菜单栏中，单击 File > Run 运行脚本，结果会显示在交互式窗口上如图 1.12 所示。PythonWin 中的语法高亮规则与 Python Shell 的略有不同。

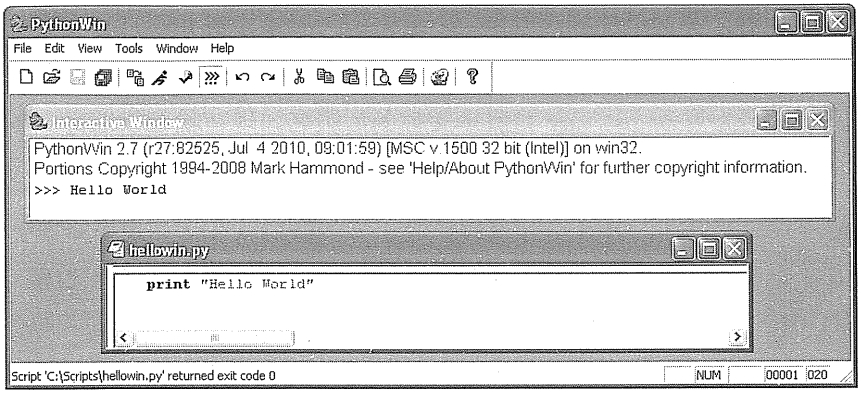


图 1.12

注释：

从现在开始，本书将使用 PythonWin 作为 Python 的编辑器。选择哪种 Python 编辑器很大程度上取决于程序员自己。本书选用 PythonWin 是因为它不仅操作简单，而且在 Windows 平台上表现十分稳定。经验丰富的程序员可以自由地选择编辑器，只要它能兼容 Python。

除了像 IDLE 和 PythonWin 这些专门的 Python 编辑器之外，通用的代码编辑器也可以用于编写 Python 脚本。这些编辑器通常会提供语法高亮、格式设置等代码编辑选项。其中有一些编辑器被广泛使用，例如 Bluefish、Context 和 Notepad++。通常情况下，这些代码编辑器可以处理多种编程语言，所以，经验丰富的程序员可以在一种编辑器上使用多种程序语言。相反，PythonWin 是专门用于编写 Python 代码的编辑器。专门的 Python 编辑器与文本编辑器相比，具有一定的优势，它具有代码提示功能，并可以进行调试，这就是本书使用 PythonWin，而不使用通用代码编辑器的原因之一。

注释：

谨慎使用普通的文本编辑器（例如 Notepad）编写代码，因为它并不是专门用于编写代码

的。这些文本编辑器不能保存脚本的编码风格，也没有语法高亮功能或任何有助于规范脚本格式的工具。所以在选择文本编辑器的时候，要确保它是专门用于编写程序代码的。

从 ArcGIS 10 开始，就可以直接在 ArcGIS for Desktop 中使用 Python，它既方便又高效。ArcGIS 10 中新的 Python 窗口取代了 ArcGIS 9 中的 Command Line 窗口。在新的 Python 窗口中，左半部分是一个 Python 交互式解释器，右半部分是帮助界面，如图 1.13 所示。

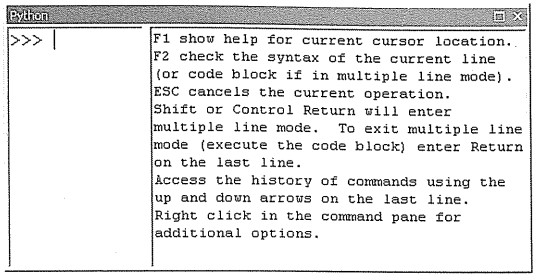


图 1.13

和其他交互式解释器一样，Python 窗口也会立即执行输入完毕的代码，如图 1.14 所示。

Python 窗口可以用于快速测试简单的代码。Python 窗口的一个优点就是它会提供语法提示，即可以自动补全代码。例如，输入字母 p 时，Python 窗口会提示出 pass 和 print 供选择，如图 1.15 所示。这将加快编码的速度，减少出错的次数。

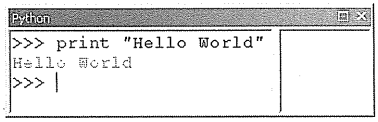


图 1.14

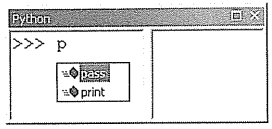


图 1.15

后面的章节会更详细地介绍如何高效使用 Python 窗口。

回顾一下，任何 Python 脚本都是由不同的模块构成，这些模块是 Python 代码最高级的组织形式。Python 脚本文件实际上是简单的文本文件，因此任何文本编辑器都可以编写脚本代码。Python 文件的后缀是.py，这样就可以自动和一个 Python 编辑器进行关联。

注释：

Python 脚本文件的名称必须以字母开头，后面可以跟随任何字母、数字或者下划线。同时，脚本文件的名称中不能出现 Python 的关键字，这些关键字在后面的章节中会提到。

关于示例代码的规定：如果代码前出现>>>提示符，那么该代码是在交互式解释器（例如 ArcGIS for Desktop 中的 Python 窗口或 PythonWin 里的交互式窗口）里编写的。按下 ENTER 键后，代码会被立即执行。如果示例代码前没有提示符，那么该代码是在脚本窗口中编写的，需要运行这个脚本才能执行代码。本书中大多数的示例代码既可以在交互式解释器中运行，也可以直接通过脚本文件运行。

注释：

本书使用 PythonWin 和 ArcGIS 中的 Python 窗口来编写和运行 Python 代码，也可以通过其他的编辑器使用这些代码。

本章要点

- ArcGIS for Desktop 支持使用脚本语言实现任务自动化。Python 是 ArcGIS 首选的脚本语言。
- Python 不是由 Esri 公司开发的。它是一个开放源码的编程语言，因此可以通过第三方发布，包括 Esri。
- Python 相对简单。它有一个庞大的用户社区和许多学习资源。此外，Python 还提供很多函数库以实现更多的功能。
- Python 的一个优点就是它既是一门脚本语言，又是一门程序语言。所以既可以用它编写简单的脚本，也可以用它开发高级应用程序。本书主要介绍如何使用 Python 编写 ArcGIS 的脚本。
- Python 是一门解释性语言，它不需要编译，而是直接从源代码执行。这使得 Python 与 C++和 NET.语言比起来更方便，也更具有移植性。
- Python 脚本可以像常见的地理处理工具一样，变成一个脚本工具，直接整合到 ArcGIS 中。
- 使用 Python 时需要一个编辑器，可以使用通用代码编辑器，也可以用专门的 Python 编辑器。Python 默认安装的编辑器是 IDLE。本书中则使用 PythonWin，因为它在 Windows 平台上使用起来相对简单。ArcGIS for Desktop 也有一个用于编写 Python 代码的窗口，它是一种交互式的 Python 解释器。
- Python 的安装程序被封装在了 ArcGIS for Desktop 的安装程序中。其中，默认安装的是 IDLE 编辑器，而不是 PythonWin。

第2章 ArcGIS 中的地理处理

2.1 引言

本章将介绍 ArcGIS 地理处理框架，包括 ArcToolbox、ModelBuilder 以及 Python。熟练的 ArcGIS 用户会比较熟悉这部分内容，但是回顾一下还是有好处的，因为理解地理处理框架将有助于高效地编写地理处理脚本。另外，Python 和 ModelBuilder 经常需要结合在一起使用，所以掌握好 ModelBuilder 才能更有效地发挥 Python 脚本的作用。

2.2 地理处理

ArcGIS 中的地理处理功能既能够完成空间分析和建模任务，也能够实现 GIS 任务的自动化。一个典型的地理处理工具可以接收数据（如要素类、栅格或属性表），执行相关地理处理任务，并最终生成一个新的数据，如图 2.1 所示。ArcGIS 中含有数百个地理处理工具，例如创建一个缓冲区，向表中添加一个字段、地址编码等。

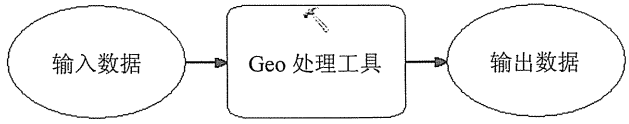


图 2.1

地理处理可以将一系列工具按顺序串联在一起，从而实现工作流的自动化操作。其中，一个工具的输出可以作为另一个工具的输入。在 ArcGIS 中，可以通过模型和脚本来组合各种地理处理工具并实现工作流的自动化操作。

ArcGIS 中地理处理框架是指一组用于管理和执行工具的窗口和对话框。使用地理处理框

架，可以轻松地创建、执行、管理、记录以及共享地理处理工作流。地理处理提供了一系列用于处理数据的工具。基本的地理处理框架由以下内容组成：

- 组织在工具箱和工具集中的一系列工具。
- 查找和执行工具的方法，包括 Search 窗口、Catalog 窗口和 ArcToolbox 窗口。
- 设置工具参数并运行工具的工具对话框。
- 将多个工具按顺序组合在一起的 ModelBuilder。
- 通过 Python 脚本运行工具的 Python 窗口。
- 记录工具相关操作信息的 Results 窗口。
- 创建脚本并将其添加到工具箱的一系列方法。

在接下来的章节中，每个组成部分都会有更详细的介绍。地理处理框架之所以能够以稳定而灵活的方式操作工具，是因为它具备了以下几个特点：

- 所有的工具都可以从对应的工具箱获取，这有利于统一工具、模型以及脚本的访问路径。
- 所有工具的文件格式都相同，这有利于工具的分类和检索。
- 所有工具都有相似的参数设置界面（对话框）。
- 工具可以被共享。

2.3 ArcObjects

您可能还记得第 1 章中的术语“ArcObjects”。本节将简要介绍什么是 ArcObjects 以及它与 ArcGIS 中地理处理的关系。

ArcObjects 库包含了 ESRI 公司开发 ArcGIS 软件用到的所有编程对象。它分别作为.NET 软件开发工具包 (SDK)和 Java 软件开发工具包的一部分提供给程序开发人员。开发人员可使用 ArcObjects 构建新的应用程序或扩展 ArcGIS 应用程序现有的功能。大部分系统工具和整个地理处理框架的构建也是通过 ArcObjects 完成。

ArcObjects 应该和系统编程语言一起使用，此时程序员需要访问低级图元以执行复杂的逻辑和算法。ArcObjects 中包含了数以千计的不同对象，因为这样便可以使程序员能够精细地控制软件的外观和运行机制。系统编程语言是诸如 C++和.NET 之类的语言，它们是最常见