

全彩
印刷

同步
视频

项目
教学

计算
思维

青少年

Python

(微课版)

创意编程

方其桂 主编

梁祥 王芳 副主编

趣味课堂



一本为青少年量身定制的编程学习书!



案例丰富多彩

75个好玩、有趣的案例，涉及编程的诸多类别，每个案例都有详细的分析和制作指导，降低了学习的难度。



图文精细标注

使用图片替换了大部分的文字说明，图解编程思路与步骤，一目了然，简单易懂。



资源配套全面

提供所有案例的源代码，以及课后练习的所有答案，并且录制了微课视频，可扫描二维码下载或在线学习。



教师同步教学

参与的作者有省级教研人员，以及具备多年教学经验的中小学信息技术教师（包括正高级教师、特级教师）。



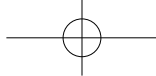
扫码

看视频

源代码与

视频下载

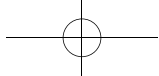
清华大学出版社



青少年Python创意编程 趣味课堂 (微课版)

方其桂 主编 梁祥 王芳 副主编

清华大学出版社
北 京



内 容 简 介

本书以Python为设计工具，共分9章内容，从易到难，从基础到综合实战，详细讲解了Python创意编程知识，通过75个小案例，带领孩子们掌握顺序结构、选择结构、循环结构的基础知识，进入精彩的函数世界，玩转Python字符串，学习列表、元组与字典数据结构，乐当编程小达人，轻轻松松玩转Python创意编程。

本书适合对编程感兴趣的青少年以及不同年龄的初学者阅读，也适合家长和老师作为指导青少年学习计算机程序设计的入门教程。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

青少年Python创意编程趣味课堂：微课版 / 方其桂 主编. —北京：清华大学出版社，2020.1
ISBN 978-7-302-53859-2

I. ①青… II. ①方… III. ①软件工具—程序设计—中小学—教材 IV. ①G634.671

中国版本图书馆CIP数据核字(2019)第207823号

责任编辑：李 磊 焦昭君

封面设计：王 晨

版式设计：孔祥峰

责任校对：牛艳敏

责任印制：丛怀宇

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦A座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：三河市铭诚印务有限公司

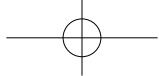
经 销：全国新华书店

开 本：170mm×230mm 印 张：16 字 数：332千字

版 次：2020年1月第1版 印 次：2020年1月第1次印刷

定 价：69.80元

产品编号：084296-01

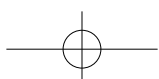
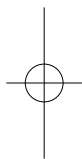
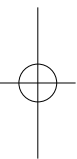


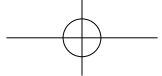
编委会

主 编 方其桂

副主编 梁 祥 王 芳

编写者 梁 祥 王 芳 张 青 宣国庆 范德生





前言

这是一本写给零基础编程读者的学习入门书，也是一本写给家长朋友们转变教育观念的书。这不是一本用来考试的编程书，但编者希望它是一本帮助同学们打开编程世界的引导书。

1. 编程干什么

编写程序简称编程，通俗地讲，编程就是告诉计算机，要帮人做什么、怎么做。但是计算机无法直接听懂人类的语言，所以需要一种计算机和我们人类都能理解的语言，这种语言就是计算机语言。使用计算机语言编写的文件称为程序。

我们身处的这个时代是人类历史上一个发展迅速的伟大时代！互联网、智能手机、各种 App、大数据、机器人等都已经一步步变成现实。5G、物联网、人工智能也很快会大面积实现和普及，就像今天的互联网和手机一样。所有这一切的背后，都离不开人类编写的软件。事实上，编程已经成为中小学教育的重要组成部分，因为编程有以下几方面很显著的作用。

- ♥ **编程是极佳的智力启蒙活动** 编程能提高孩子们的记忆力、想象力、逻辑推理能力，有效促进智力培养。
- ♥ **兴趣是学习编程最好的老师** 本书选取的例子贴近生活，符合孩子的身心特点，容易引起他们的共鸣，激发他们的学习兴趣，让他们感觉学习 Python 是一件非常有趣的事。
- ♥ **编程的过程是一种思维方式** 它教给孩子们如何创造性思考、协同学习和逻辑推理，提高做事的计划性，增强分析问题、解决问题的能力。
- ♥ **编程是处理信息的现代方式** 在信息社会，如何认识信息、理解信息、驾驭信息，最好的途径之一就是学习编程，发挥信息的作用。

在未来世界中，编程能力可以说是一个受过教育的人的基本能力，就像今天一个上过学、读过书的人要具备基本的读写能力一样。

2. Python 是什么

Python 是一门非常优秀的计算机编程语言，功能强大、兼容性好、可移植，有相对较少的关键字、结构简单，有定义明确的语法，简单易学。Python 已经成为三大主流编程语言之一，它适合孩子的编程启蒙。具体而言，Python 有如下优点。

- ♥ **入门容易** 其使用界面简洁，编写程序过程简便、容易上手，非常适合初学编程者学习。
- ♥ **难度值低** Python 虽简单，但其设计却很严谨，使用户可以将全部心思放在程序的设计逻辑之中。
- ♥ **兼容性强** 具有免费开源的特点，可移植、可扩展、可嵌入多平台使用。
- ♥ **众多外挂** 具有丰富的外挂库，例如，使用海龟模块可以绘制各种图案，形象直观。

3. 本书结构

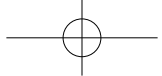
万栋高楼平地起，学习编程也要一步一步来。本书按照由易到难的顺序，将所有的知识点融入一个个好玩、有趣的案例中，让小朋友先模仿案例去动手做一做，边玩边学，在玩的过程中逐渐理解，在完成模仿项目的基础上进行拓展，激发创新思维。全书按照知识顺序、难度分成9章，每章还设计了多个栏目，便于读者学习和教师教学。

- ♥ **章节语** 成语故事引入情景、英语关键词解释代码含义、核心知识点强调重点。
- ♥ **研究室** 思路分析、以算法描述的方式分析程序、对编写过程中的重难点进行解读。
- ♥ **工作坊** 通过编程实现、运行调试、优化程序、答疑解惑等掌握程序编写调试过程。
- ♥ **知识库** 介绍程序中使用到的知识点或程序中的重点、难点知识。
- ♥ **创新园** 通过阅读程序写结果、查找错误代码、编写程序等多种方式，鼓励同学们创新作品。

4. 本书特色

本书适合零基础或已经接触过 Python 编程，且对 Python 感兴趣的青少年阅读，也适合家长和老师作为指导孩子们程序设计的提升教程。为充分调动他们学习的积极性，在编写时努力体现如下特色。

- ♥ **案例丰富** 本书案例丰富，涉及编程的诸多类别，内容编排合理，难度适中。每个案例都有详细的分析和制作指导，降低了学习的难度，使读者对所学知识更加容易理解。
- ♥ **图文并茂** 本书使用图片代替大部分的文字说明，一目了然，让学习者能轻松读懂描述的内容。具体操作步骤图文并茂，图文结合来讲解程序的编写，便于读者边学边练。
- ♥ **资源丰富** 本书配备了所有案例的素材和源文件，提供了相应的微课，从数量到内容上都有着更多的选择。
- ♥ **形式贴心** 如果读者在学习过程中遇到疑问，可以阅读“提示”部分，避免在学习过程中走弯路。



5. 适用对象

本书适合 10 岁以上有阅读能力的读者使用。对于低龄儿童，建议在家长和老师指导下阅读。教师、家长在使用本书教学时，可以让学生先用手机扫描书中二维码，借助微课先行学习，然后再利用本书上机操作实践。

为了使读者在阅读本书时取得最大的价值，获得更好的学习效果，我们提出如下建议。

- ♥ **按顺序阅读** 本书的知识点做了精心设计，建议读者按照顺序，由简到难阅读。
- ♥ **在做中学习** 也就是在计算机旁边阅读本书，一边实践、一边体会书中案例的作用。
- ♥ **多思考尝试** 通过思考，构思可以怎么做，分析为什么这样做。只要有想法，就尝试去实现它。
- ♥ **不怕困难失败** 学习肯定会遇到各种各样的困难，失败也是很正常的，失败了，说明这种方法不可行，也就距离可行的方法近了一步。
- ♥ **多与他人交流** 和朋友一起学习和探讨，分享自己的项目，从而快速学习别人的优点。遇到问题，多向老师请教。也可以和本书作者联系，我们会努力帮助你们解决问题。

6. 本书作者

本书由方其桂主编，梁祥、王芳为副主编。梁祥编写第 1、2、3、4、5 章，张青编写第 6 章，王芳编写第 7 章，宣国庆编写第 8 章，范德生编写第 9 章。随书资料由方其桂整理制作。

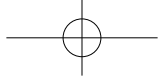
虽然我们有着十多年撰写计算机图书（累计已编写、出版 100 余种）的经验，并尽力认真构思验证和反复审核修改，但仍难免有一些瑕疵。我们深知一本图书的好坏，需要广大读者去检验评说，在此，我们衷心希望读者对本书提出宝贵的意见和建议。读者在学习使用的过程中，对某个问题的解决，可能会有更好的编程方法，也可能对书中某些案例算法的科学性可行性提出质疑，敬请读者批评指正。

本书提供了每个案例的微课，请扫描一下书中案例名称旁边的二维码，即可直接打开视频进行观看，或者推送到自己的邮箱中下载后进行观看。另外，本书提供教学课件和案例源文件，通过扫描下面的二维码，然后将内容推送到自己的邮箱中，即可下载获取相应的资源（注意：请将这几个二维码下的压缩文件全部下载完毕后，再进行解压，即可得到完整的文件内容）。

服务电子邮箱为 wkservice@vip.163.com。



方其桂



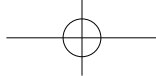
目录

第1章 揭开 Python 神秘面纱

1.1 厉兵秣马——做好学习 Python 语言的准备	2
1.1.1 按图索骥——下载与安装 Python	2
1.1.2 整装待发——设置 Python 环境	4
1.2 发号施令——Python 编程初体验	9
1.2.1 有问必答——与 Python 交流	9
案例 1 Hello, 你好	9
案例 2 首先来画个圆	10
1.2.2 妙笔生花——让 Python 绘画	12
案例 3 画多个圆	12
案例 4 画彩色圆	16
1.3 小试牛刀——体验编写程序的乐趣	19
1.3.1 心中有数——出题考考 Python 计算能力	20
案例 5 测试 Python 计算能力	20
1.3.2 初生牛犊不怕虎——编写 Python 小游戏	22
案例 6 猜猜我多大	22

第2章 编程基础先打好

2.1 知人善任——常量和变量	27
2.1.1 有一得一——常量	27
案例 1 求圆的周长与面积	27
2.1.2 以一当十——变量	30
案例 2 小猴子摘果子	31
2.2 不分伯仲——数据类型	34

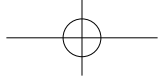


青少年 Python 创意编程趣味课堂 (微课版)

2.2.1	量入为出——数字型	35
	案例 3 比比谁算得快	35
2.2.2	非此即彼——布尔型	38
	案例 4 是真的吗	38
2.2.3	张冠李戴——字符串	40
	案例 5 字符朋友手牵手	41
2.3	变化多端——数据类型转换	43
2.3.1	再接再厉——整数型与浮点型转换	44
	案例 6 小数计算也不难	44
2.3.2	变化莫测——数字型与字符串转换	47
	案例 7 字符加密不神奇	47

第3章 顺序结构直向前

3.1	军令如山——赋值语句	53
3.1.1	登坛拜将——赋值符号	53
	案例 1 长跑健将	53
	案例 2 整理图书	55
3.1.2	发号施令——赋值运算	59
	案例 3 统计人数	59
	案例 4 温度转换器	60
3.2	字字珠玑——运算符和表达式	64
3.2.1	各司其职——运算符	64
	案例 5 简便计算器	64
	案例 6 灯亮不亮	66
3.2.2	意简言咳——表达式	69
	案例 7 分糖果游戏	69
3.3	进退有度——输入输出语句	72
3.3.1	予取予求——数据输入	73
	案例 8 求圆环的面积	73
	案例 9 距节日还有多少天	74
3.3.2	千变万化——数据输出	77



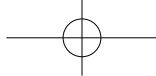
案例 10 农夫的难题	77
-------------------	----

第4章 选择结构左右分

4.1 去伪存真——单分支 if 语句	82
4.1.1 淘沙取金——if 语句	82
案例 1 比较两个数的大小	82
4.1.2 挑挑拣拣——if 语句组合	84
案例 2 判断闰年	84
4.2 一分为二——双分支 if...else 语句	87
4.2.1 分道扬镳——if...else 语句	87
案例 3 判断奇数与偶数	88
4.2.2 何去何从——if...else 条件语句嵌套	90
案例 4 判断函数的值	90
案例 5 出租车计费	92
4.3 分门别类——多分支 if...elif...else 语句	97
4.3.1 分兵把守——if...elif...else 语句	97
案例 6 成绩折算等级	98
案例 7 智能胖瘦分析	100
4.3.2 多种多样——多分支的其他语句组成形式	103
案例 8 根据成绩判断等级 (改良版)	104

第5章 循环结构环环绕

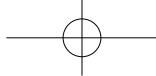
5.1 反反复复——循环基本语句	109
5.1.1 周而复始——for 循环语句	109
案例 1 统计摩天轮旋转次数	109
案例 2 棋盘上的麦粒	111
5.1.2 循环反复——while 循环语句	114
案例 3 $1+2+\dots+100$	115
案例 4 绘制多彩图形	117



5.2	息息相关——循环嵌套	121
5.2.1	九九归一——for 循环语句嵌套	121
	案例 5 打印乘法口诀表	121
5.2.2	照猫画虎——while 语句循环嵌套	124
	案例 6 打印倒序乘法口诀表	124
5.3	指挥若定——循环控制语句	128
5.3.1	令行禁止——continue 语句	128
	案例 7 数字宝宝玩游戏	128
5.3.2	戛然而止——break 语句	130
	案例 8 密码中止程序	130

第6章 使用函数效率高

6.1	没有规矩，不成方圆——定义函数	136
6.1.1	百花齐放——函数类型	136
	案例 1 判断水仙花数	136
	案例 2 判断是否获奖	140
6.1.2	不越雷池——变量的作用域	144
	案例 3 判断素数	144
	案例 4 10 以内加法测验程序	147
6.2	种瓜得瓜，种豆得豆——函数的参数	151
6.2.1	步调一致——必须参数	152
	案例 5 计算三角形的面积	152
6.2.2	心照不宣——默认值参数	155
	案例 6 计算 x^n	155
6.3	空穴来风，必有其因——函数的返回值	158
6.3.1	有来有往——无返回值函数	158
	案例 7 交换两个数	158
6.3.2	投桃报李——有返回值函数	160
	案例 8 计算 $n!$	160

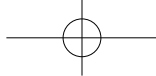


第7章 字符串串滋味香

7.1 收放自如——字符串变形	165
7.1.1 比翼连枝——拼接字符串	165
案例 1 成语接龙	165
案例 2 真心话大冒险	167
7.1.2 断章取义——字符串切片	169
案例 3 秘密约定	170
案例 4 回文字符串	171
7.2 当家做主——字符串操控	174
7.2.1 用兵点将——字符串查询	174
案例 5 评选优秀学生	175
案例 6 成员查询	177
7.2.2 偷梁换柱——字符串替换	179
案例 7 回避敏感词	180
案例 8 排课表	181
7.3 情有独钟——格式化输出	184
7.3.1 当仁不让——我的地盘	184
案例 9 明星档案	184
7.3.2 锦上添花——文本排版	187
案例 10 学习计划表	187

第8章 数据类型能分清

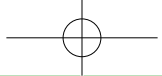
8.1 一目了然——列表	191
8.1.1 井然有序——列表的创建和引用	191
案例 1 创建课程表	191
案例 2 查询课程表	193
8.1.2 志同道合——插入新元素、合并列表	195
案例 3 社团的新成员	196
案例 4 合二为一	198



8.1.3	当机立断——查找删除列表元素	200
	案例 5 百里挑一	200
8.2	包罗万象——元组	203
8.2.1	运筹帷幄——创建和引用元组	203
	案例 6 查询跳远成绩	203
8.2.2	出类拔萃——修改元组	206
	案例 7 修改校运动会记录	206
8.2.3	神机妙算——遍历元组	209
	案例 8 你来自哪颗星	210
8.3	百无一漏——字典	213
8.3.1	有约在先——创建查询字典	213
	案例 9 燃烧我的卡路里	214
8.3.2	未雨绸缪——修改字典	217
	案例 10 小刺猬的烦恼	217

第9章 乐当编程小达人

9.1	迎刃而解——编程解决数学问题	222
9.1.1	精打细算——兑换零钱	222
9.1.2	生生不息——兔子对数	224
9.2	轻而易举——编程解决生活问题	228
9.2.1	积少成多——52 周存钱法	228
9.2.2	居安思危——密码强度	230
9.3	笔走龙蛇——编程绘制有趣图形	234
9.3.1	电闪雷鸣——闪电图形	235
9.3.2	栩栩如生——五角星	237
9.4	乐在其中——编程创作精彩游戏	240
9.4.1	鸿运当头——抽奖游戏	240
9.4.2	一决雌雄——石头剪刀布	241



第 1 章

揭开 Python 神秘面纱

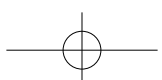
计算机是我们的好伙伴，它本领高强，再难的问题都能轻松解决，而且从不叫苦叫累，命令它干啥就干啥。学会了计算机语言，就能让计算机按我们的命令去做事。计算机语言有许多种，因为 Python 的语法接近人类正常语言，代码容易看懂和编写，它逐渐成为一种被广泛应用的语言。

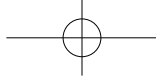
学习 Python 并不是件困难的事情，让家人和我们一起玩编程，在玩中学习。还等什么呢？让我们一起揭开 Python 的神秘面纱吧！



学习内容

- ❏ 做好学习 Python 语言的准备
- ❏ 了解 Python 运行环境
- ❏ 感受 Python 编程乐趣





1.1

厉兵秣马——做好学习 Python 语言的准备

成语“厉兵秣马”出自《左传》，意思是磨好兵器、喂饱战马，比喻事前做好准备工作。学习 Python，首先要在计算机上下载并安装 Python，再设置 Python 的运行环境。

1.1.1 按图索骥——下载与安装 Python

按图索骥，意思是按照画像去寻求好马，比喻按照线索去寻求。在学习下载与安装 Python 之前，我们先来认识一下关键词吧！



扫一扫，看视频

✔ Python(蟒蛇)

✔ Downloads(下载)

✔ Install(安装)

✔ Cancel(取消)

在我们的生活中，程序随处都有，但许多人始终没有接触过编程，原因就是他们的计算机中没有编程软件，所以安装编程软件算是初学者要完成的第一个任务。

通常下载与安装软件的操作步骤如下。

第一步：查找下载网站。

第二步：找到下载网页，下载所需软件。

第三步：安装软件。



你也许认为安装编程软件会非常难，其实只要你愿意动手去做，就没有什么解决不了的问题。因为 Python 的下载与安装过程真的很简单。

工作坊

01 查找 Python 官网 打开浏览器，按图 1.1 所示操作，搜索到 Python 官网网址并打开。

02 下载 Python 软件 在 Python 官网首页双击 Downloads 菜单选项，按图 1.2 所示操作，下载安装程序。

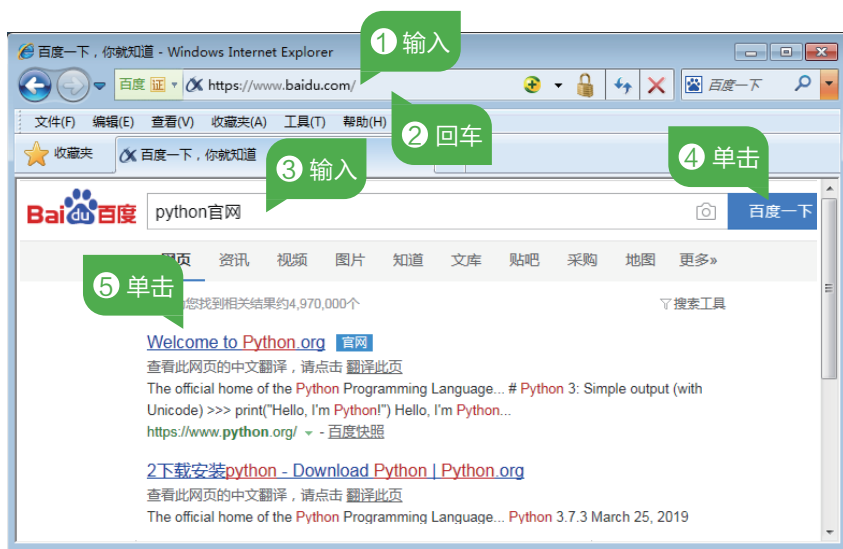


图 1.1 查找 Python 官网

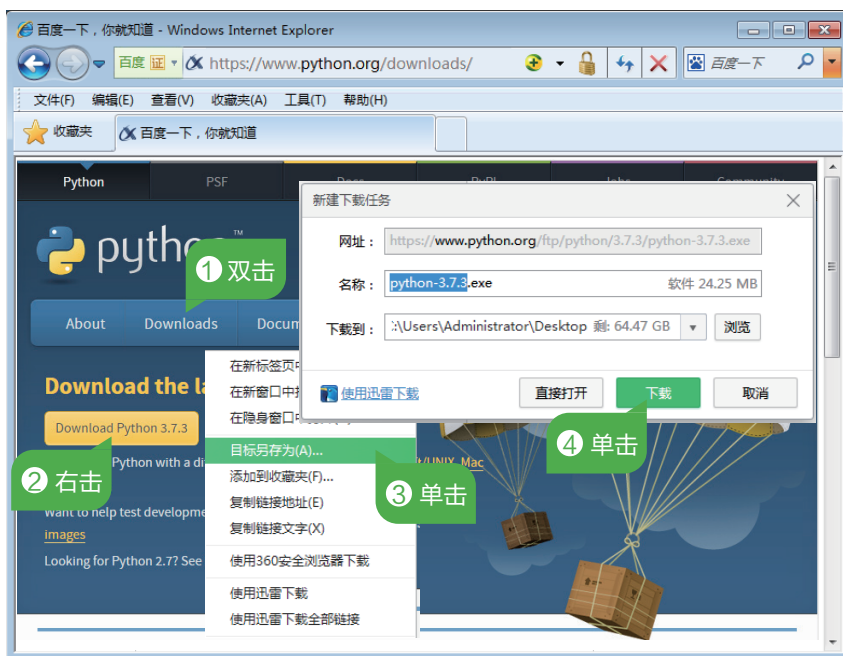


图 1.2 查找下载页面并下载软件

03 安装 Python 软件 双击下载的 Python-3.7.3.exe 文件，按图 1.3 所示操作，按照提示安装 Python 软件。



图 1.3 安装 Python

04 查看选项 安装结束后，按图 1.4 所示操作，在“开始”菜单中多了一项 Python。

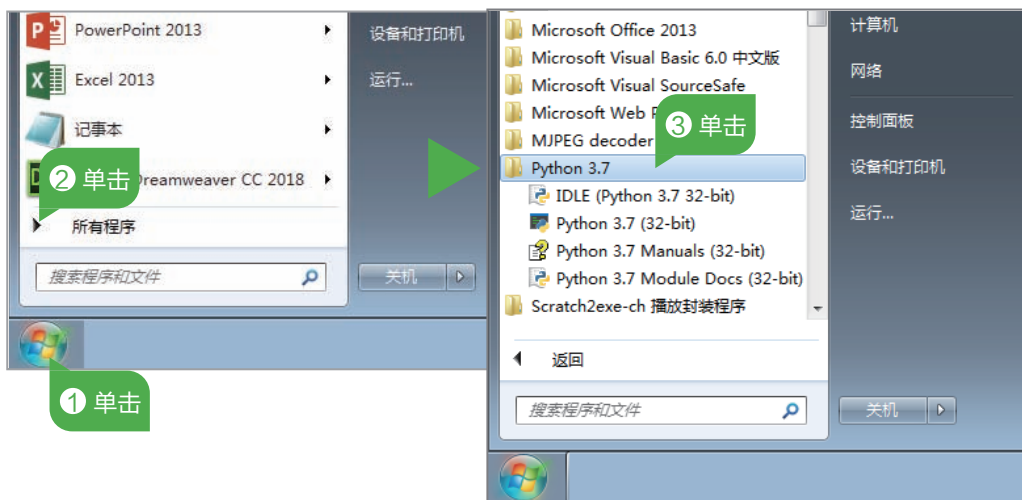


图 1.4 查看选项

1.1.2 整装待发——设置 Python 环境

前面已经安装好 Python，下一步需要设置 Python 环境，才能正常使用。

- ✓ Path(路径)
- ✓ IDLE(空闲)
- ✓ Manuals(手册)
- ✓ Module Docs(模块文档)

在 Windows 下设置 Python 环境变量，就是把 Python 的安装目录添加到系统



扫一扫，看视频

Path 中。具体的操作也不复杂，跟着我一步一步做吧！

🏠 工作坊

01 判断是否要设置环境变量 按图 1.5 所示操作，如果出现图中所示界面，说明安装成功，否则需要配置环境变量。

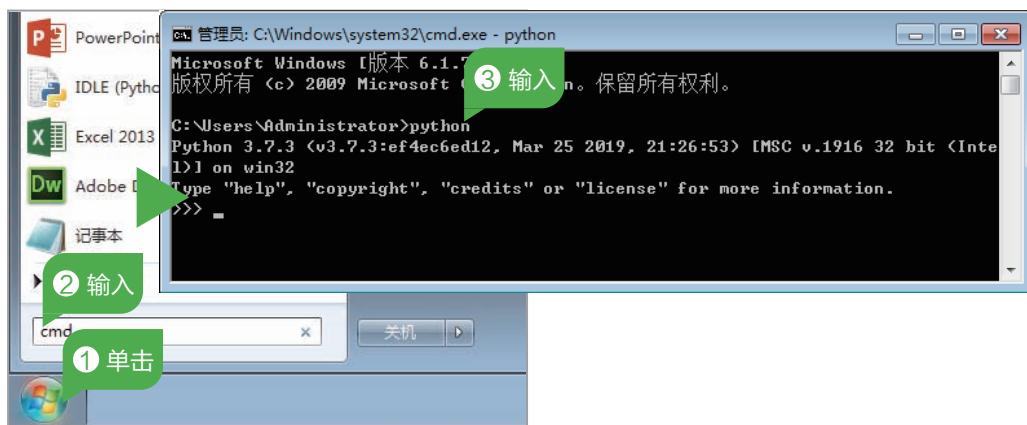


图 1.5 判断 Python 环境变量是否设置

提示

如果在 Python 安装时执行了图 1.3 所示的第 2 步操作，即勾选 Add Python 3.7 to PATH 复选框，那么安装时会自动设置好，否则需手工设置。



02 查找安装目录路径 按图 1.6 所示操作，查找并复制 Python 安装目录路径。

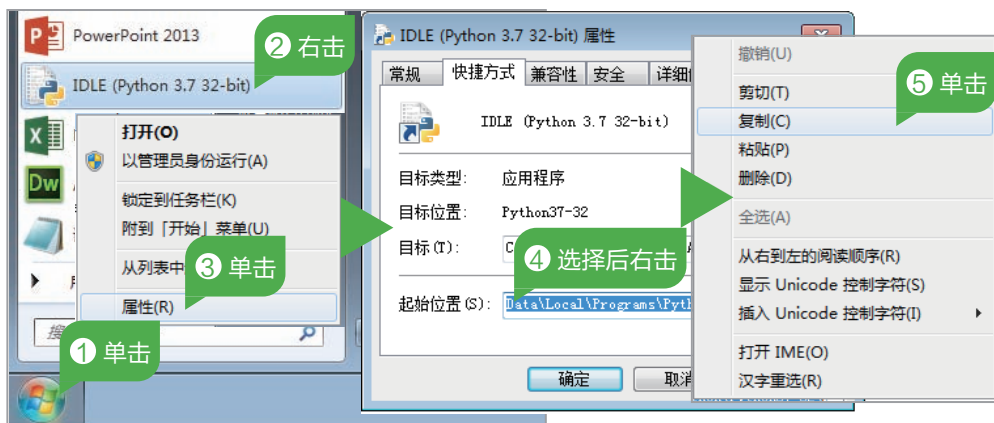


图 1.6 查找 Python 安装目录路径

03 打开高级系统设置 右击桌面上的“计算机”图标，按图 1.7 所示操作，打开计算机的“高级系统设置”窗口。

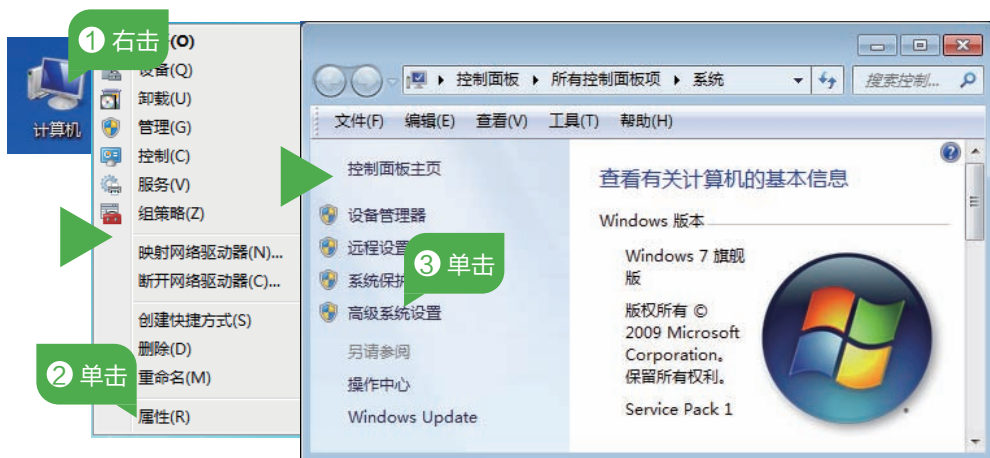


图 1.7 打开计算机的高级系统设置

04 设置环境变量 按图 1.8 所示操作，将图 1.6 操作中获取的 Python 目录路径粘贴到“变量值”文本框中。

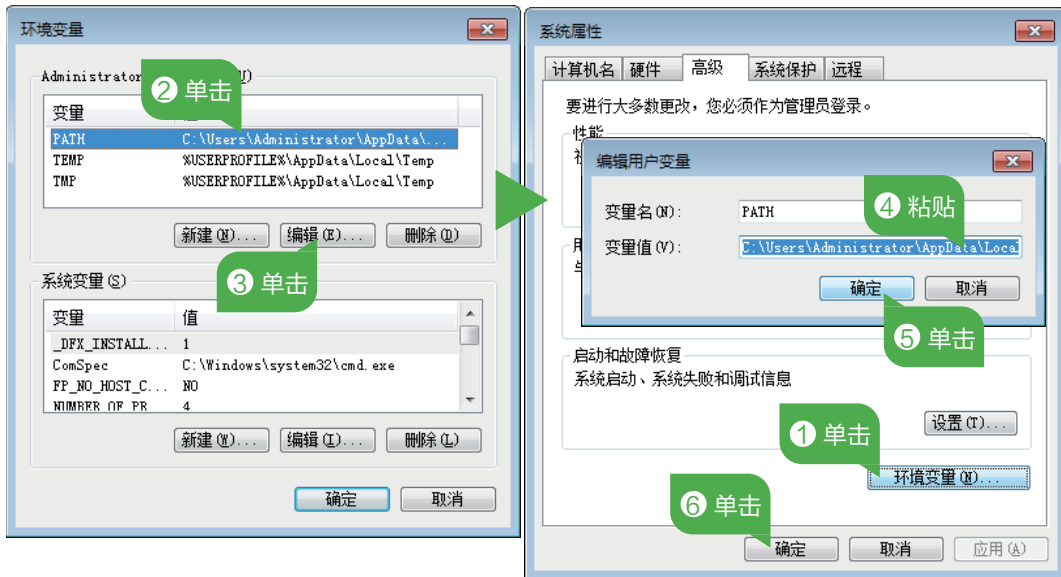


图 1.8 设置环境变量

05 认识 Python Shell 按图 1.9 所示操作，打开 Python，标题栏上的 Python 3.7.3 是版本号。在命令提示符 >>> 处即可输入 Python 指令。

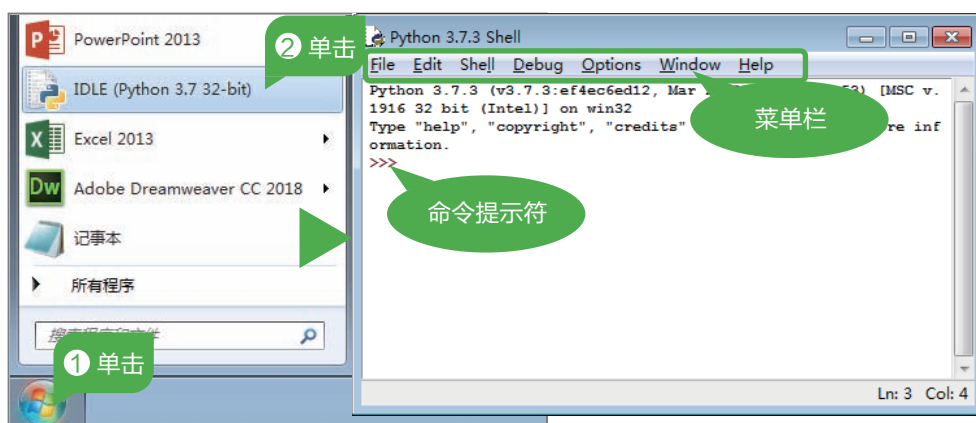


图 1.9 进入 Python Shell 窗口

提示

图 1.9 中的 >>> 是 Python 提示符。它在告诉你，Python 已经准备好了，在等着你键入 Python 指令。



- 06 设置字体** 为了后续编写的程序便于阅读，按图 1.10 所示操作，在 Settings 对话框中设置 Font Face 字体为 Arial Unicode MS，Size 字号为 18，Bold 字型为加粗。

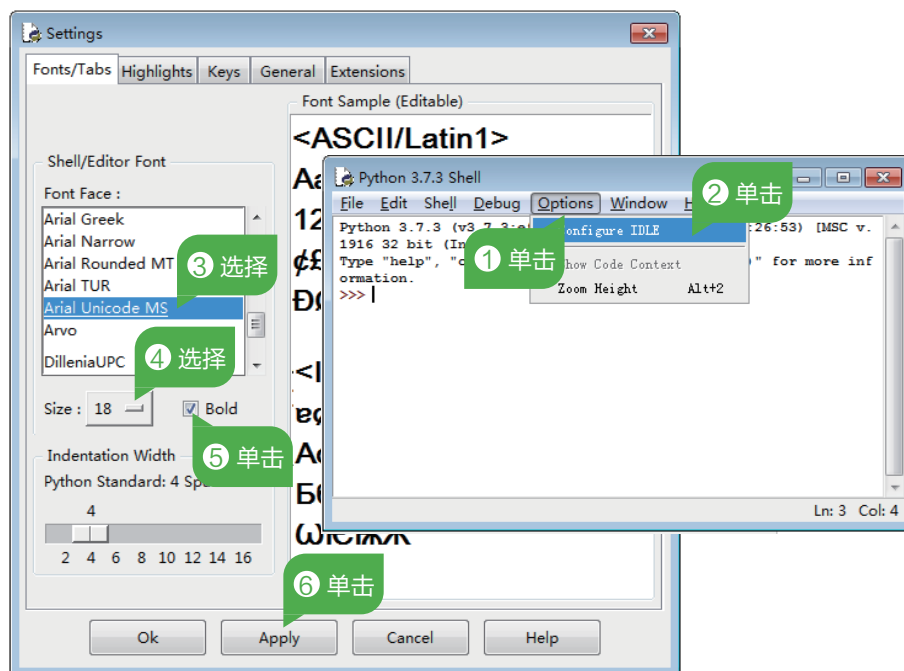


图 1.10 设置 Python 编辑器字体

知识库

1. IDLE

IDLE 是 Python 自带的开发环境，可实现计算机与人之间的互动交流，分为 2 种，如图 1.11 所示，我们习惯在“环境一”中编写程序。

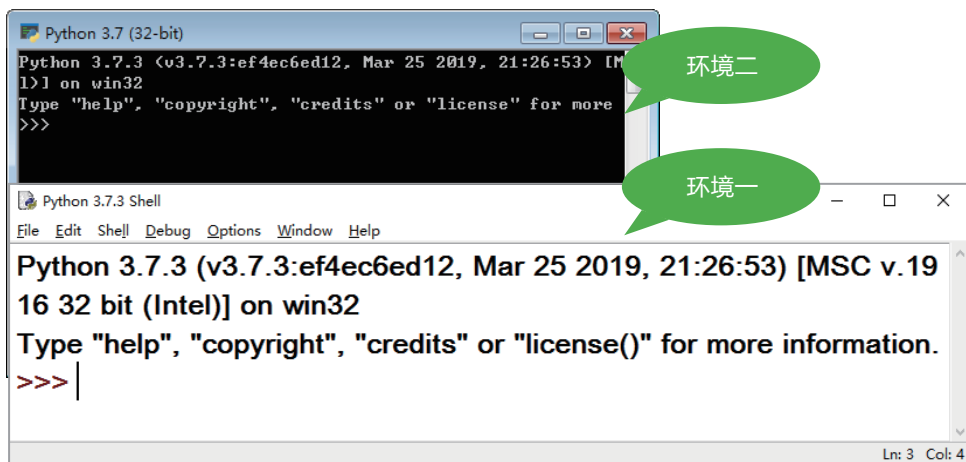


图 1.11 IDLE 的开发环境

2. 编程方式

在 Python 中编写程序有 2 种方式，如图 1.12 所示。第 1 种是逐行执行指令；第 2 种编写多行程序，然后集中执行。



图 1.12 逐行执行与多行程序编写

1.2 发号施令——Python 编程初体验

发号施令是指发命令、下指示。编程就是使用计算机的语言，精确地给计算机发号施令，让它去完成某一个任务。

- ✔ print(打印)
- ✔ turtle(龟)
- ✔ pen(笔)
- ✔ circle(圆)

1.2.1 有问必答——与 Python 交流

Python 如同我们的好朋友一样，你可以与他进行交流对话。如果你说的他能听懂，他就会按你的要求执行。如果你说的他听不懂，Python 还会给出红色的错误提示。

案例 1 Hello, 你好

第一次打开 Python，使用 Python 编程是在逐行执行的窗口中进行。先和它打声招呼吧，在屏幕上输出“Hello”，接着再考一考 Python 的计算能力。



扫一扫，看视频

工作坊

01 打个招呼 按图 1.13 所示操作，运行 Python 软件，在光标闪烁处输入 `print('hello')` 命令后，再按回车键。

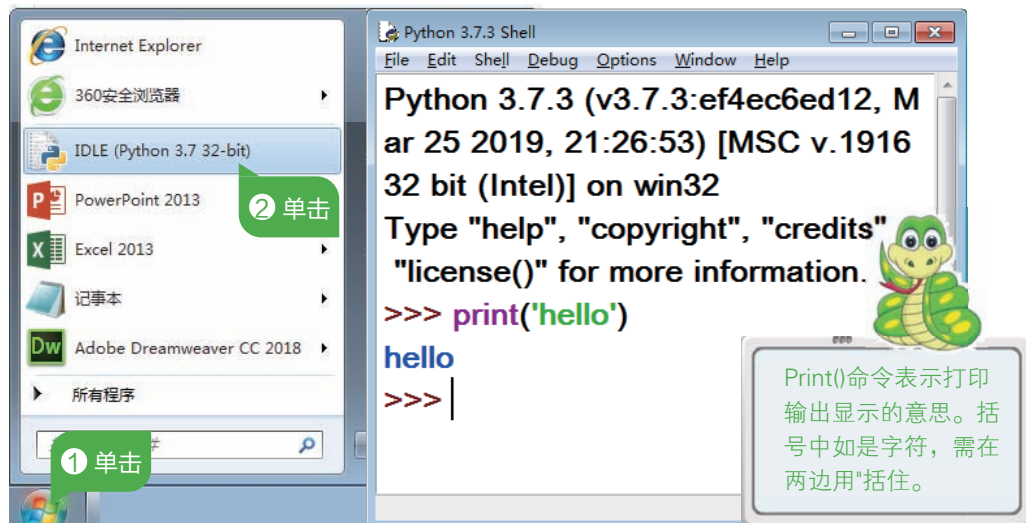
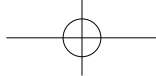


图 1.13 输入命令



02 考考计算能力 按图 1.14 所示操作，在光标闪烁处输入 `print(3+4)` 命令后按回车键，看看出现什么结果。

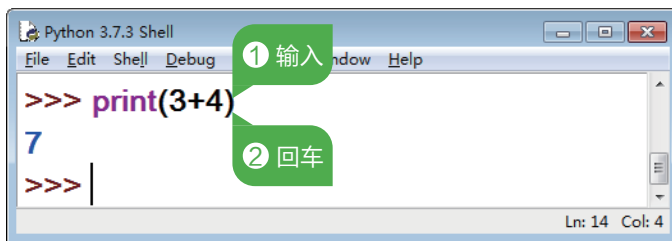


图 1.14 考考计算能力

03 发号施令 按图 1.15 所示操作，在光标闪烁处输入 `print('Python hello!')` 命令后按回车键，接着输入 `print(3*4)` 命令后按回车键，试一试结果是否正确。



图 1.15 发号施令输入指令

04 调试程序 如果你输入命令后，按回车键没有出现上面的操作结果，而是如图 1.16 所示出现一段红色的英文，这就表明你输入出错了。

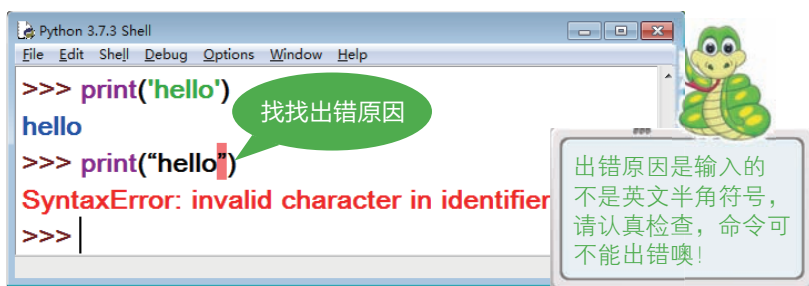


图 1.16 调试程序

案例 2 首先来画个圆

案例 1 的操作是在 Python 提示符 `>>>` 下一条一条地给出命令。我们有时会一次性将多条指令写成一个文件，然后让程序一次性执



扫一扫，看视频

行。如图 1.17 所示，我们让 Python 画个圆，来开始真正的第一次编程之旅吧！

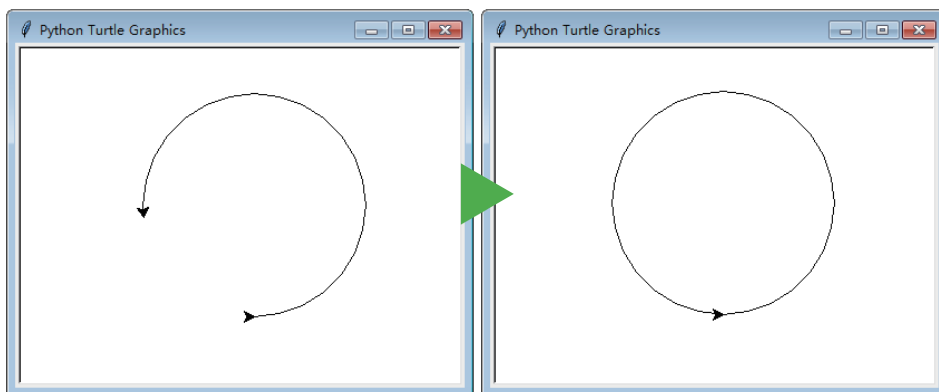


图 1.17 “小海龟画图”程序运行效果

🏠 工作坊

01 新建源文件 选择 File → New File 命令，新建一个源文件，进入 Python 多行编程界面，如图 1.18 所示。

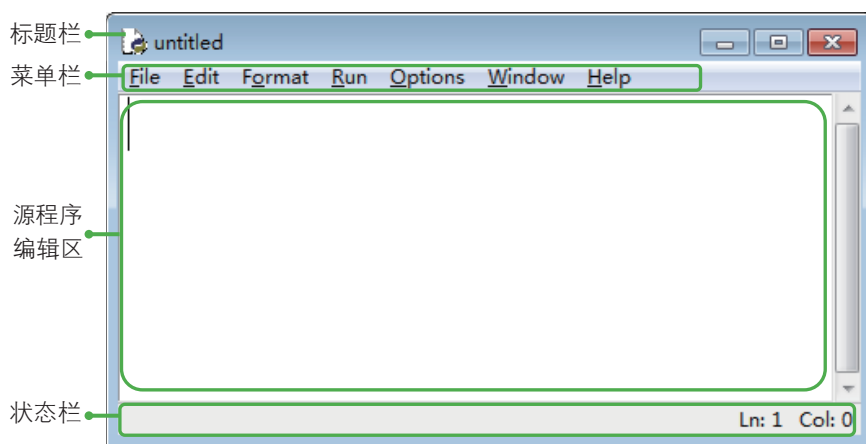


图 1.18 Python 多行编程界面

提示

一个源文件只能编写一个程序，如再编写另一个程序，还需要新建一个源文件。



02 编写程序 在源文件的编辑界面输入以下代码，如图 1.19 所示。输入代码需注意使用英文半角，还要注意英语的大小写。

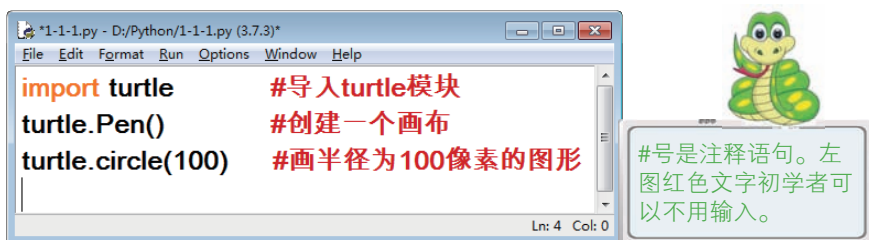


图 1.19 输入程序代码

03 保存程序 按图 1.20 所示操作，以 1-1-1.py 为程序名保存文件。



图 1.20 保存文件

04 运行程序 选择 Run → Run Module 命令，运行程序，运行结果如图 1.17 所示，小海龟在画布上慢慢爬行，画出一个圆。

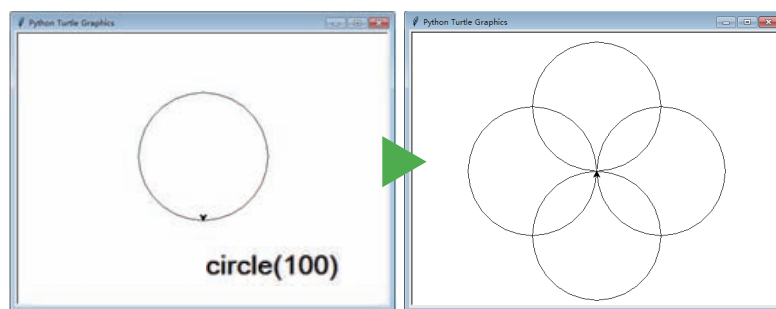
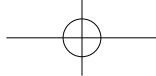
1.2.2 妙笔生花——让 Python 绘画

使用 Python 小海龟库中的画图命令，能很方便地绘制图形。关键是你如何妙笔生花般地编写程序，让我们的小海龟笔走龙蛇般地绘制精美图形。

- ✔ left(向左)
- ✔ right(向右)
- ✔ pencolor(笔的颜色)
- ✔ width(线条粗细)

案例 3 画多个圆

在 Python 中有一个海龟库，又称为海龟模块，库里有一只完全听指挥的小海龟，用于绘制有趣的图形。如图 1.21 所示，只要给它发出指令，它就会跟着指令画出美丽的图形。



画 1 个圆

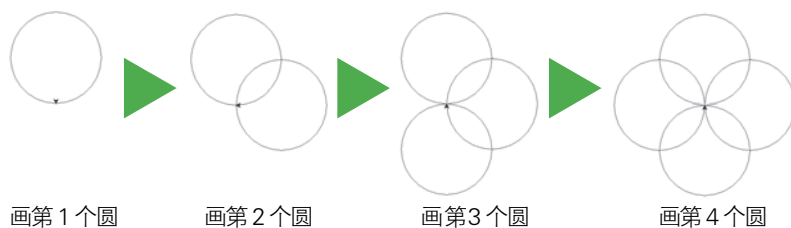
画 1 个圆旋转 90 度，重复 3 次

图 1.21 “画多个圆”程序运行效果

💡 研究室

1. 思路分析

案例 3 是让小海龟画多个圆的图案。如图 1.22 所示，就是在画 1 个圆的基础上，让小海龟右转 90 度，再画第 2 个圆，以此类推，完成 4 个圆的绘制。



画第 1 个圆

画第 2 个圆

画第 3 个圆

画第 4 个圆

图 1.22 “画多个圆”思路分析

2. 算法描述

- 第一步：调用小海龟模块，让小海龟出场。
- 第二步：画第 1 个圆，右转 90 度。
- 第三步：画第 2 个圆，右转 90 度。
- 第四步：画第 3 个圆，右转 90 度。
- 第五步：画第 4 个圆，完成任务。

🏠 工作坊

1. 编程实现

代码清单 1-2-1：画多个圆

<code>import turtle</code>	#导入turtle模块
<code>xa=turtle.Pen()</code>	#创建画布，定义海龟小名为xa
<code>xa.circle(100)</code>	#画第1个半径为100像素的圆形
<code>xa.right(90)</code>	#右转90度
<code>xa.circle(100)</code>	#画第2个半径为100像素的圆形
<code>xa.right(90)</code>	#右转90度
<code>xa.circle(100)</code>	#画第3个半径为100像素的圆形
<code>xa.right(90)</code>	#右转90度
<code>xa.circle(100)</code>	#画第4个半径为100像素的圆形

提示

代码中“#”为注释符，“#”后面的文字表明是对左侧代码的注解。初学程序时，红字是注释部分，可以不用输入。



2. 编写程序

按图 1.23 所示操作，打开 IDLE 编辑器，新建 Python 文件，输入代码后，按 Ctrl+S 键，以文件名“1-2-1.py”保存文件。

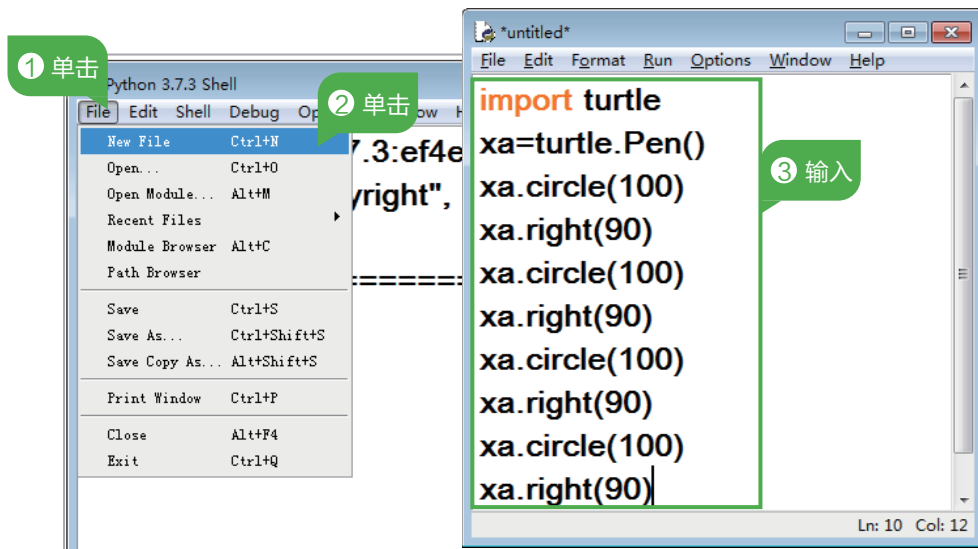


图 1.23 “画多个圆”程序编写过程

3. 运行调试

按图 1.24 所示操作，可执行程序。除此之外，也可以按 F5 键直接运行程序。

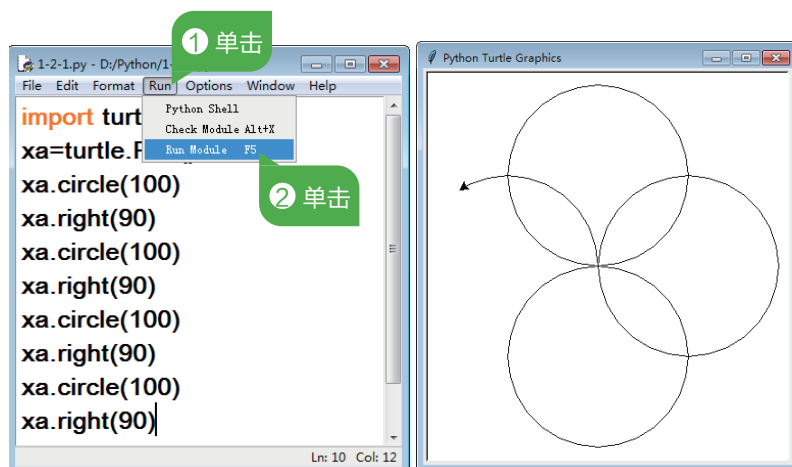


图 1.24 “画多个圆”程序运行效果

4. 答疑解惑

在程序运行调试过程中，如果出现程序代码写错，系统会自动给出提示。如图 1.25 所示，红色 turtle 前的 “=” 是中文符号，不是英文 “=”，所以在执行程序时会弹出报错提示。单击“确定”按钮后，将中文的 “=” 修改为英文的 “=” 后，再按 F5 键执行程序，小海龟就能正常绘画操作了。

对于相同的代码，可以使用复制粘贴的方式，提高代码的输入效率，但输入一定要在英文半角状态下。Python 语言区别大小写，其中 Pen() 中 P 是大写字母。

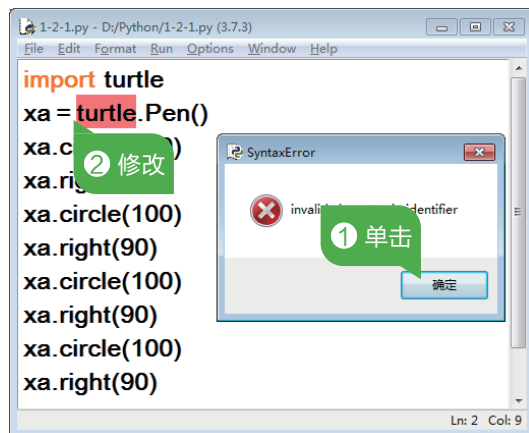
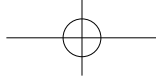


图 1.25 “画多个圆”程序调试运行过程

5. 修改程序

编程最大的乐趣在于，你可以让计算机按自己的意图执行程序。单击 File → Open 命令，打开 “1-2-1.py” 文件。试着修改小海龟画圆的半径数值，如图 1.26 所示操



青少年 Python 创意编程趣味课堂（微课版）

作，将第 1 次半径改为 40，第 2 次半径改为 60，第 3 次半径改为 80，第 4 次半径改为 60，并另存为“1-2-1a.py”。按 F5 键运行程序，画出大小不一样的圆。

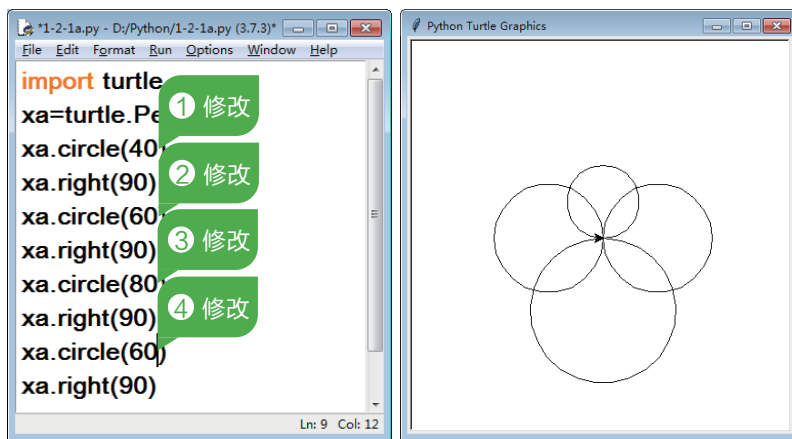


图 1.26 “画多个圆”程序修改运行效果

案例 4 画彩色圆

“1-2-1.py”程序是画多个圆，只需在这个案例的基础上，添加设置小海龟颜色的语句，就能画出不同颜色的圆。如图 1.27 所示就是小海龟绘制出的彩色圆。怎么样，有趣吧？快来学习改变画笔颜色，让小海龟绘制彩色图形吧！

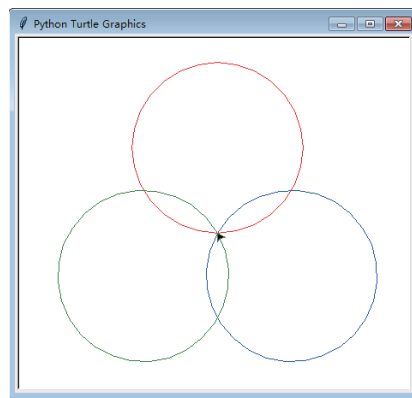


图 1.27 “画彩色圆”程序运行效果

研究室

1. 思路分析

为小海龟定义画笔的颜色，其中 red 表示红色，blue 表示蓝色，green 表示绿色。如图 1.28 所示，分析“画彩色圆”程序可以看出，先需要设置第 1 个圆的颜色，让小海龟先画红色的圆，再旋转 120 度画蓝色的圆，接着旋转 120 度画绿色的圆。

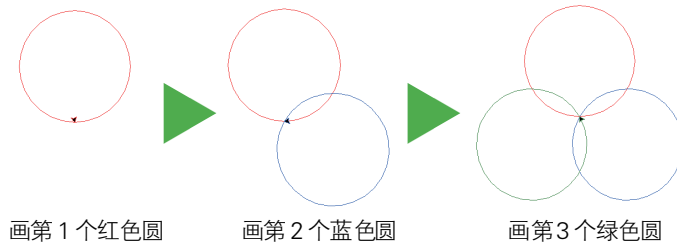


图 1.28 “画彩色圆”思路分析

2. 算法描述

根据以上分析，算法描述如下。

第一步：调用海龟模块，并定义小海龟名称为 xa。

第二步：设置海龟画线的颜色为红色。

第三步：画第 1 个半径为 100 像素的红色圆，并右转 120 度。

第四步：设置海龟画线的颜色为蓝色。

第五步：画第 2 个半径为 100 像素的蓝色圆，并右转 120 度。

第六步：设置海龟画线的颜色为绿色。

第七步：画第 3 个半径为 100 像素的绿色圆，完成任务。

🏠 工作坊

1. 编程实现

代码清单 1-2-2：画彩色圆形

```
import turtle           #导入turtle海龟模块
xa=turtle.Pen()        #定义海龟名称为xa
xa.pencolor('red')     #设置海龟画线的颜色为红色
xa.circle(100)         #画第1个半径为100像素的圆
xa.right(120)          #右转120度
xa.pencolor('blue')   #设置海龟画线的颜色为蓝色
xa.circle(100)         #画第2个半径为100像素的圆
xa.right(120)          #右转120度
xa.pencolor('green')  #设置海龟画线的颜色为绿色
xa.circle(100)         #画第3个半径为100像素的圆
```

2. 运行调试

按快捷键 F5 运行程序，效果如图 1.27 所示。你还可以设置修改变量 xa.pencolor('red') 的颜色为 yellow、orange、purple、black 等。

3. 修改完善

如图 1.29 所示，还可以设置线的粗细，使用 xa.width(3) 让蓝圆的线宽度为 3 个像素，使用 xa.width(6) 让绿圆的线宽度为 6 个像素，使画出的圆更有特色。

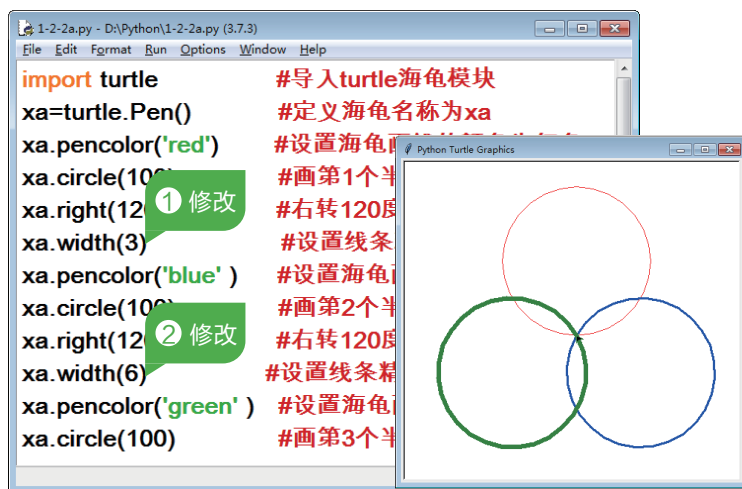
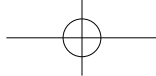


图 1.29 “画彩色圆”程序调试运行效果

知识库

1. Turtle 海龟绘图模块

在 Python 语言中，可以调用 Turtle 模块快速绘制图形。如图 1.30 所示，让一个小海龟在一个横轴为 X、纵轴为 Y 的坐标系原点 (0,0) 位置开始，根据一组函数指令的控制在这个平面坐标系中移动，可以依据它爬行的路径绘图。

2. Turtle 海龟模块指令

Python 海龟模块提供了很多指令，这些指令的意思与其英文的意思基本相同，所以很好理解，常用的 Python 海龟模块指令作用如下。

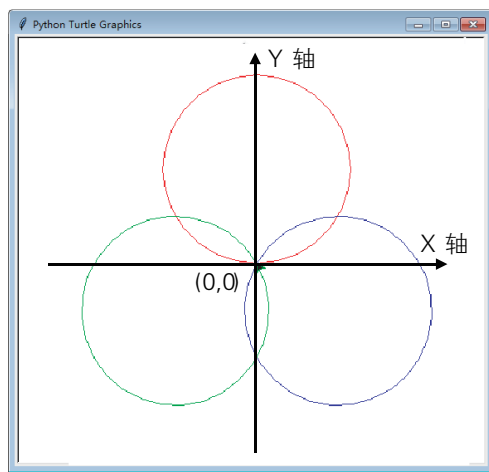


图 1.30 用海龟绘图模块画图

指令	作用
Import turtle	导入 turtle 模块
xa=turtle.Pen()	创建一个画布，给海龟起名为 xa
xa.forward(100)	让 xa 海龟往前移动 100 像素
xa.backward(100)	让 xa 海龟往后移动 100 像素
xa.right(45)	让 xa 海龟右转 45 度
xa.left(45)	让 xa 海龟左转 45 度
xa.circle(100)	让 xa 海龟画半径为 100 像素的圆形

创新园

1. 阅读程序写结果

```
import turtle
xa=turtle.Pen()
xa.circle(100)
xa.right(60)
xa.circle(100)
xa.right(60)
xa.circle(100)
```

画的图形是：_____

2. 修改程序错误

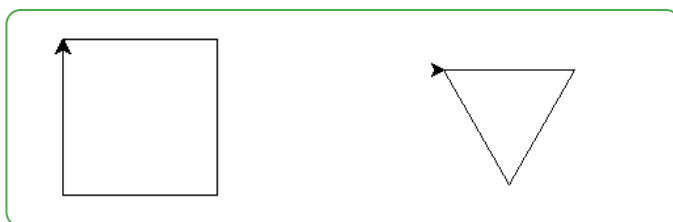
下面程序有两处错误，请修改。

```
import turtle
xa=turtle.Pen()
xa.pencolor(3) ①
xa.width('red') ②
xa.circle(100)
xa.right(120)
```

错误 1：_____ 错误 2：_____

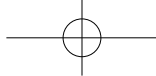
3. 编写程序

`xa.forward(100)` 是让小海龟画一个线段，`xa.right(90)` 是让小海龟转 90 度，请试一试编程，绘制下面的图形。



1.3 小试牛刀——体验编写程序的乐趣

对于零起点的初学者来说，Python 语言学习上手也很容易。首先我们通过模仿，能读懂案例程序，并对部分语句进行修改，从中体验编写程序的乐趣。下面就小试牛刀，开启我们的程序初体验吧！



- ✔ guess(猜)
- ✔ input(输入)
- ✔ age(年龄)
- ✔ print(打印)
- ✔ if(如果)
- ✔ else(其他)

1.3.1 心中有数——出题考考 Python 计算能力

“心中有数”出自《庄子·天道》，意思是对情况和问题有基本的了解，处理事情有一定把握。Python 有着强大的计算能力，只要你能给出让 Python 读懂的命令，Python 都能立即给出正确的计算结果。

案例 5 测试 Python 计算能力

我们首先体验一下交互式编程，比如你可以考一考计算机的计算能力。如图 1.31 所示，在窗口提示符中输入 `print(999*999)` 后按回车键，Python 立即给出答案 998001。



扫一扫，看视频

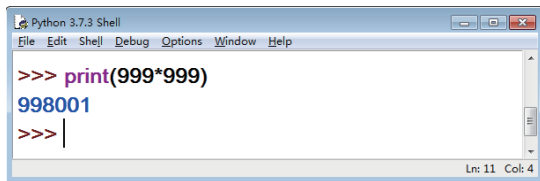


图 1.31 Python 交互式编程体验

研究室

1. 思路分析

如图 1.32 所示，`print` 是紫色，代表是输出打印命令，之后的内容为参数，在 `print` 命令中参数需要用 `()` 括起来。计算结果 998001 的颜色是蓝色。Python 的色彩设置可以方便阅读程序。

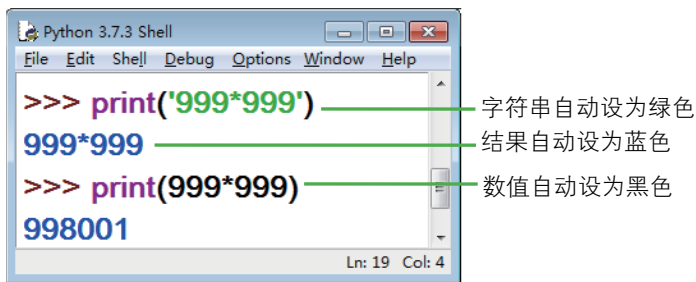


图 1.32 Python 语言自动设置颜色

根据这种方式我们可以测一测 Python 的计算能力，看一看 Python 的计算与普通的计算器有什么区别。

💡 研究室

1. 思路分析

从运行中我们可以看到，此 Python 程序具备一定的判断能力，只有输入 Python 语言正确的诞生年份，才能给出正确的回答，否则给出错误的回答。

这种形式就是以后章节中我们要学习到的条件语句，在这个案例中我们可以先体验一下，感受一下条件语句的使用。

2. 算法描述

根据上述分析，如图 1.36 所示，猜 Python 年龄程序中必有一条非常重要的判断语句，这条语句的功能可实现对输入的数据进行判断。

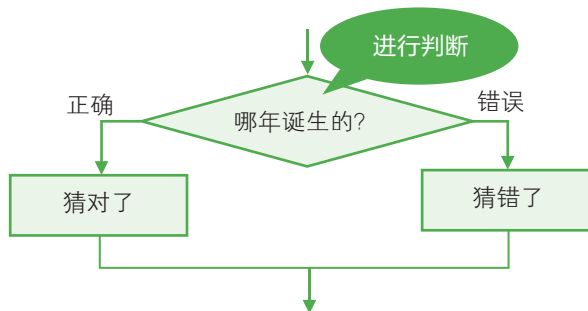


图 1.36 “猜猜我多大”程序问题描述

🏠 工作坊

1. 编程实现

代码清单 1-2-3: 猜猜我多大

```

guess=input('猜猜Python语言哪一年诞生:')
age =int(guess)
if age ==1989:    #如果age等于1989，判断正确后执行
    print('*_*，你的Python语言知识了解得真多！')
    print('可是，猜对了也没有奖励！')
else:            #如果age不等于1989，执行else:以下语句
    print('^_^,猜错啦，Python是1989年诞生的！')
print('游戏结束，不玩啦！')
  
```

2. 答疑解惑

在运行程序过程中，输入的内容应是数字，但是如果不小心输入的是字符，Python 就会立即给出错误的提示，如图 1.37 所示。

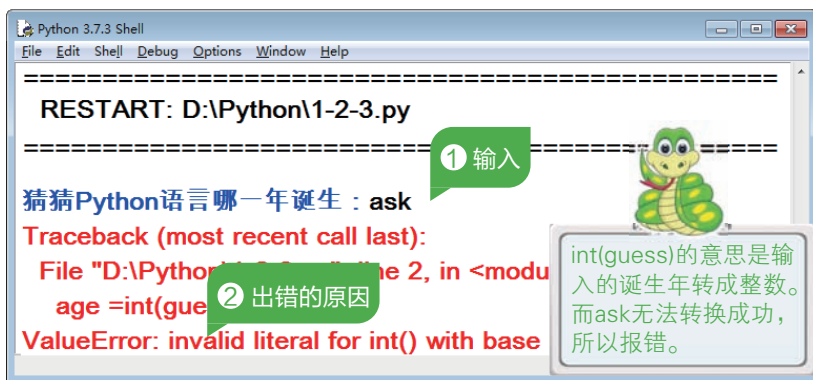


图 1.37 “猜猜我多大” 编程调试过程

3. 修改程序

根据猜 Python 年龄“1-2-3.py”文件，修改为“1-2-3a.py”文件。
代码清单 1-2-3a：猜猜我多大

```
guess=input('猜猜我是哪一年出生的?')
age =int(guess)
if age ==2005:
    print('你真聪明!')
    print('可是，猜对了也没有奖励!')
else:
    print('^.^,猜错啦，我是2005年诞生的!')
print('游戏结束，不玩啦!')
```

知识库

1. Python 行

可以分为物理行和逻辑行。逻辑行中句与句之间用“;”区分。以下两种代码执行时功能是一样的。

物理行代码

```
print('Hello')
print('Python')
print('!')
```

逻辑行代码

```
print('Hello');print('Python');print('!')
```

2. Python 文件

Python 文件的打开、保存等操作，与我们所学过的 Word 等软件相同。需要注意的是 Python 文件扩展名为 .py。在保存文件时，要注意给文件起一个好记的名字。本书为了便于大家学习，所有文件名命名方式统一。例如，“1-2-2.py”是表示第1章第2节的案例2程序，“1-2-2a.py”表示以前面的程序为基础进行修改后的文件。

创新园

1. 阅读程序写结果 1

```
>>> print('欢迎'+ '您！')
```

输出：_____

2. 阅读程序写结果 2

```
>>> print('欢迎'+ '您！'*3)
```

输出：_____

3. 阅读程序写结果 3

```
score=float(input('输入一个成绩：'))
if score>=90:
    print('A')
elif score>=60:
    print('B')
else:
    print('C')
```

输入：85 输出：_____