



“十四五”职业教育河南省规划教材
高等职业教育新型活页式教材

P 人工智能之 Python

活页式

主 编 樊建文 赵 敏
副主编 张麦玲 王 聪



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS



“十四五”职业教育河南省规划教材
高等职业教育新型活页式教材

人工智能之 Python

活页式

主 编 樊建文 赵 敏

副主编 张麦玲 王 聪

上海交通大学出版社

内容提要

本书共6个任务,20个活动,任务一为走进人工智能3.0时代,介绍了人工智能、大数据等新技术,任务二~任务六介绍了Python语言基础、Python结构化程序设计、Python的Turtle绘画、Python的Pygame游戏制作、Python的大数据爬虫。本书取材新颖实用,讲解深入浅出、实例短小精悍、循序渐进,没有编程基础的读者也可轻松阅读。

本书采用活页式设计,集知识点、任务、活动、操作说明为一体,配有活页式评价表和学习笔记,是零基础入门的高职院校Python编程通识课教材,适用于所有专业,用于全面提高学生数字化素养,也可作为编程爱好者选用。

图书在版编目(CIP)数据

人工智能之Python/樊建文,赵敏主编. — 上海:
上海交通大学出版社,2021.2(2024.1重印)
ISBN 978-7-313-24483-3

I. ①人… II. ①樊… ②赵… III. ①软件工具—程
序设计 IV. ①TP311.561

中国版本图书馆CIP数据核字(2021)第028585号

人工智能之Python

RENGONG ZHINENG ZHI Python

主 编:樊建文 赵 敏

出版发行:上海交通大学出版社

邮政编码:200030

印 制:江苏唐邦彩印包装有限公司

开 本:787mm×1092mm 1/16

字 数:273千字

版 次:2021年2月第1版

书 号:ISBN 978-7-313-24483-3

定 价:49.80元

地 址:上海市番禺路951号

电 话:64071208

经 销:全国新华书店

印 张:11.5

印 次:2024年1月第3次印刷

版权所有 侵权必究

告读者:如发现本书有印装质量问题请与印刷厂质量科联系

联系电话:0511-87898558

前 言

2018 年国家印发《高等学校人工智能创新行动计划》，在文件中明确指出人工智能成为国际竞争的新焦点，应逐步开展全民智能教育项目，要将人工智能纳入大学计算机基础教学内容，逐步推广编程教育、建设人工智能学科，培养复合型人才，形成我国人工智能人才高地。因此，为顺应新一代信息技术趋势下的人才培养，在专业升级与数字化改造的现实背景下，要求全国高职院校要紧跟当前经济结构调整及产业发展需要，在以人工智能为技术路径的数字经济时代，使职业教育更好适应数字经济、服务数字经济，为国家经济与社会发展服务。人工智能进入高职教育列入必修通识课程，目的就是培养素质教育，抓手就是 Python 编程实践，Python 语言在人工智能领域有广泛的应用，对于人工智能产品的普及起到积极的推动作用，已成为各行业应用开发的首选编程语言。目前全国高职院校开设人工智能和 Python 通识课程的只是少数，使用的教材和内容基本都是传统模式，学生学习兴趣不高、效果并不理想。

本书的主要特点：

(1) 本书打破常规，采用活页式设计，设计了 6 个任务，共 20 个活动，集知识点、任务、活动、操作说明为一体，具有超强的实用性。本书还配有活页式评价表和学习笔记，每个任务和活动都有评价表，评价表里有具体的评价指标和评价指南，每个活动 5 分，20 个活动总分为 100 分，评价分数可直接作为课程考核成绩，充分体现过程化学习和考核的特点。

(2) 本书作为人工智能和编程的通识课程教材，面向高职零编程基础非计算机专业的学生，为了更好地培养思维模式，淡化专业性较强的内容，将人工智能、大数据、Python 等知识体系细化为知识单元，构建出 6 个任务模块，同时，对应构建出 6 种思维模式，学生一边学习知识，一边完成任务，同时系统递阶式的培养和提高数字化思维模式。

(3) 本书通过任务和活动式的编排、小组学习评价和团队协作项目，让学生在真实的任务情境中，做到“基于任务的学习”，让学生能完成具有智能化知识和技术含量的、创新性的任务，帮助学生顺利进入和适应智能化社会发展下的工作岗位和生活环境。

(4) 本书突出趣味性，制作的活动中都是实用性强、具有现代感的智能工具和应用、中西方简单经典数学问题的解决、可爱软萌的小海龟绘制图形的过程、声图文并茂的游戏项目制作和功能强大的大数据爬虫，这样有色彩、有趣味的案例，能够更高效的塑造学生的数字化思维模式、开发出学生主动创新的潜力。

(5) 本书拥有配套的电子资源，使得学习和教学都变得更加简单、方便。读者可以扫描书中的二维码观看小知识的视频讲解，获取书中任务的源代码、多媒体资源等。

本书由平顶山工业职业技术学院樊建文、赵敏担任主编，张麦玲、王聪担任副主编。任务四和任务五由赵敏编写，任务一和任务六由张麦玲编写，任务二和任务三由王聪编写，全书由樊建文统稿，本书配套的教学视频由赵敏、张麦玲、王聪共同制作完成。

由于本书是对人工智能 Python 编程通识课程的活页式教材的创新设计，从内容和形式上基本没有可以参考的对象和借鉴的范本，难免会有不足之处，敬请专家和读者批评指正。

目 录

任务一 走进人工智能 3.0 时代——创新思维养成	1
活动一 感受人工智能的魅力	2
活动二 拥抱大数据时代	15
任务二 Python 之人人要编程——计算思维养成	22
活动一 编程宣言——hello, Python	23
活动二 神奇计算器	31
任务三 Python 之解决问题——过程化思维养成	39
活动一 结构化思维的认知	40
活动二 顺序结构之鸡兔同笼	48
活动三 选择结构之 BMI 健康测试	56
活动四 循环结构之猴子吃桃	65
活动五 函数之八十天环游地球	76
任务四 Python 之绘画——模块化思维养成	83
活动一 模块化思维与 Turtle 库的认知	84
活动二 安装 Turtle 小乌龟绘图库	88
活动三 绘制基本形状之绿色蟒蛇	95
活动四 绘制复杂图形之迷幻星空	109
活动五 个人创新作业	123
任务五 Python 之游戏——面向对象思维养成	126
活动一 打字母游戏设计与团队组建	127
活动二 开始界面的设计与实现	133
活动三 游戏界面的设计与实现	140
活动四 结束界面的设计与实现	146
任务六 Python 之爬虫大数据——大数据思维养成	153
活动一 爬虫及爬虫工具认知	154
活动二 爬取之遨游“百度”	161
附录	169
附录一 常用 ASCII 码对照表	169
附录二 常用 pygame 模块功能和命令	170
附录三 常用 RGB 颜色对照表	173
参考文献	174

任务一

走进人工智能 3.0 时代 ——创新思维养成



任务目标

- (1) 认知人工智能的定义、发展和应用。
- (2) 体验人工智能的魅力。
- (3) 认知大数据时代思维方式的转变。
- (4) 培养学生思维方式、工作方法和创新能力等综合职业能力。



任务要点

- (1) 人工智能的定义。
- (2) 人工智能的发展。
- (3) 人工智能的应用。
- (4) 大数据的定义。
- (5) 大数据的应用。



信息素养目标

- (1) 职业素养。
 - ① 培养积极、有责任心的工作态度。
 - ② 培养创新思维方式和工作方法。
 - ③ 培养分析问题、解决问题的能力。
 - ④ 培养小组协作互助成长的职业能力。
- (2) 技能素养。

掌握人工智能和大数据的相关软件操作方法。
- (3) 创新能力。

培养日常生活中人工智能和大数据的应用发掘能力。



任务内容

- 活动一：感受人工智能的魅力。
- 活动二：拥抱大数据时代。

活动一 感受人工智能的魅力

📍 活动目标

- (1) 畅想人工智能的未来。
- (2) 认知人工智能的定义和分类。
- (3) 认知人工智能的发展。
- (4) 认知人工智能在行业中的应用。
- (5) 感受人工智能的魅力。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定内容。
- (3) 小组成员互相沟通，服从小组管理。

📍 知识储备

一、科技改变生活

人类与其他动物的本质区别是人类有创造意识，可以制造工具、利用工具改变生存的空间，而其他动物只能根据环境改变自己。人类最重要的进化是不断地学习使用工具，以手机为例，方寸之间，集人类数千年科学和技术成果之大成，数不清的科学家和发明家的努力，才带来今天这种执世界于掌心的智能设备，在人工智能的浪潮下，未来的世界会更加让人向往。

1. 未来已来，人工智能如何改变生活

从历史来看，人类的生活每次都伴随着科技发展而产生巨大的变革。近年来涌现了很多热门的词汇：人工智能、大数据、机器学习、AR、VR 等，人们一时之间还不能理解，这些到底跟我们有什么关系？

不妨从 AlphaGo 轻轻松松赢了排名世界第一的围棋选手柯洁讲起，柯洁从赛前的自信满满到赛后的痛苦落泪，彰显着 AlphaGo 无比强大的人工智能。人们渐渐发现原来人工智能已经悄然来到了我们中间，未来已来，它会怎样改变我们的生活呢？哪些具有未来感的科技产品已经悄然进入我们的生活之中了？

1) 口袋里的翻译官

地球是圆的，我们身在这个全球化的时代里，不管是出差还是旅游，如何与外国人交流是个很重要的问题。不会说英语、出国购物无法砍价、旅游因为语言差异不顺心，那么随身带个私人翻译吧！

翻译机能够快速、准确地实现中英口语间的即时互译。不管是商务还是旅行，拥有了它就可以与世界各地的人无障碍交流了，我们再也不用担心环游世界会遇到语言障碍啦！

2) VR 带你穿越现实与虚幻

还记得小时候看“哆啦 A 梦”，里面有一双神奇的小鞋子，穿上它就可以跳进童话世界里，帮助阿里巴巴战胜四十大盗，去看白雪公主和七个小矮人。相信每个小朋友都很羡慕大雄有“哆啦 A 梦”这个来自未来的小伙伴，而现在的科技让梦想成为现实。

只需戴上虚拟现实的 3D 眼镜，就可以马上进入童话世界了。看电影、玩游戏，它都能给你更加真实的体验。放眼望去，世界就在你的眼前！

3) 智能扫地机器人——解放你的双手

下班回家不想打扫卫生，家里乱糟糟，地板上的污渍不易处理，“懒人症”发作的你该怎么办呢？请个保洁阿姨帮忙打扫，价格太高。自己动手，手脚累得根本不听我指挥啊！难道让自己生活在垃圾堆里吗？

拥有很高吸净率的扫地机器人，可以将家里打扫得干干净净：高效锁水拖把，干湿两用，清除顽固积尘；升级智能芯片，精准计算，全面高效覆盖；自动回充功能，不需要用一次操作一次，高科技更省心。

4) 无人驾驶出租车亮相街头

2020 年 6 月，上海自动驾驶领域发生了一件大事——上海智能网联汽车规模化载人示范应用启动，首批智能网联汽车在滴滴出行 APP 上开放了示范运行，无人驾驶的出租车上路了！自动驾驶与 5G、智能网联汽车、车路协同这些最新技术密不可分。根据目前的技术水平，5G 体验速率可以达到 1Gb/s，通过 5G+自动驾驶，可以精准地获取信息并判断路况，实现快速反应，例如前面有电瓶车或人经过，车就会自动避让。“4G 改变生活，5G 改变社会”，5G 对当今社会的影响力，好比百年前的电力一样。

2. 运用人工智能的未来生活畅想

假如小夏是一位生活在 2030 年的教师，让我们来畅想一下他的日常生活。

角色 1：小夏

角色 2：嘉宝（机器人）

角色 3：小助（机器人）

时间：2030 年 3 月 10 日

6:30，起床时间到了，嘉宝播放了一曲小夏最爱的音乐，并轻声说：“小夏，今天是 2030 年 3 月 10 日，新的一天开始了。今天的气温是 20°C 到 28°C，天气晴朗，加油哦！”

小夏伸了个懒腰，打了一个长长的哈欠，哼着音乐起床。房间感知到了他起床的信号，自动电控的调光玻璃逐渐由漆黑变得透明，金色的阳光洒进卧室。

嘉宝快速扫描小夏全身，根据小夏这段时间的健康智能监测系统数据，为小夏定制了一份营养均衡的早餐——一碗瘦肉粥、蒸玉米、蔬菜和一杯热牛奶。小夏一边享受着嘉宝准备的美味早餐，一边浏览着饭桌上呈现的智能推送的个性化信息，包括新闻热点、空气质量等。

7:30，嘉宝提醒小夏：“上班时间到了，换衣服出发吧。”出门前，小夏对嘉宝说：“开启安防，此时房间内都处于红外保护状态，如果有人恶意闯入，将自动报警。”

8:00，小夏出行上班，智能家居系统检测到小夏要出门，告知仓库中的汽车。自动驾驶汽车来到家门口等待，小夏走上前轻轻对着汽车说：“开门。”识别模块进行声音识别，车锁自动打开，小夏上车后车载精灵问道：“小夏，我是车载精灵，你现在要去哪里？”小夏说：“去上班。”汽车根据当前的交通状况自动选择最优路线并驶向上班地点。小夏下车后，自动驾驶

汽车根据周边停车位空缺，自动停车。

8:50，小夏进入办公室后，小助把今天的工作安排按照紧急程度传送到小夏的桌面，小夏简单地浏览后，开始了高效工作的一天。小夏是一名高校的管理人员，学校注重培养学生的实践动手能力，小夏可以通过显示屏查看学校教学运行的总体情况、学生的考勤情况、课堂的实训场景等。学生通过各种虚拟仿真设备进行实操能力的培养，教师通过各种实训平台进行教学指导，通过大数据平台获取学生的学习效果，并反馈至教学实践中。

11:30，小夏通过桌面订餐系统预定了午餐，并预定了座位。来到餐厅，送餐机器人将烹饪好的午餐送到指定的座位。

17:00，一天充实的工作结束，小夏走出学校，无人驾驶汽车已经在门口等着，上车后家中的冰箱就通过汽车屏幕推送了家里的食物及推荐菜谱，小夏给嘉宝打了个电话，告知晚餐想吃什么。

18:00，小夏回到家，刚一打开门，家里的灯光让人倍感舒畅。妻子早已接孩子放学回来，孩子们都在 AR 环境下复习当天的功课，并完成当天的作业。

20:00，吃完嘉宝准备的美味晚餐，一家人去附近的公园散步。锻炼的人们带着智能手环，手环记录着佩戴者每天锻炼的时间、方式和强度，并且把佩戴者的心跳、血压等信息传到一个公共的医疗数据库中，通过计算，它会准确地提醒主人的身体状况和适合的运动，在紧急情况下还会智能报警。

夜色渐浓，一天即将结束，睡前孩子们躺在床上听嘉宝讲历史故事，慢慢地进入梦乡。

二、探索人工智能

计算机和计算机网络的发展，已经使人们的生活发生了天翻地覆的变化，如今，人工智能已经融入生活的各个方面，但大部分人对人工智能仍然是一头雾水，究竟什么是人工智能？人工智能和人类智能有什么联系，人工智能是怎么发展的，人工智能应用在什么地方？

1. 什么是人工智能

在 1965 年达特茅斯有一群年纪差不多的科学家在一起讨论一个问题。说在数字宇宙里面，生命和智能发展这么快，我们得给这个学科命一个名字，当时有两个备选项，一个叫人工智能，还有一个叫控制论。经过几天的研讨，他们最后决定用人工智能，这个一般大众都能听得懂的名字来命名这一学科，更重要的是在当时开会的这些年轻的科学家中，后来出现了 4 位诺贝尔奖得主，4 位图灵奖得主。

人工智能 (artificial intelligence, AI) 是计算机科学的一个分支。人工智能是对人的意识、思维的信息过程的模拟。人工智能不是人的智能，但能像人那样思考，也可能超过人的智能。

人工智能的定义可以分为两部分，即“人工”和“智能”。“人工”比较好理解，争议性也不大。有时我们会要考虑什么是人力所能制造的，或者人自身的智能程度有没有高到可以创造人工智能的地步，等等。

尼尔逊教授对人工智能下了这样一个定义：“人工智能是关于知识的学科——怎样表示知识以及怎样获得知识并使用知识的学科。”

而另一位美国麻省理工学院的温斯顿教授认为：“人工智能就是研究如何使计算机去做过去只有人才能做的智能工作。”这些说法反映了人工智能学科的基本思想和基本内容，即人工智能是研究人类智能活动的规律，构造具有一定智能的人工系统，研究如何让计算机去完成以往需要人的智力才能胜任的工作，也就是研究如何应用计算机的软硬件来模拟人类某些智能行为的基本理论、方法和技术。

人工智能是计算机学科的一个分支，20 世纪 70 年代以来被称为世界三大尖端技术之一（空间技术、能源技术、人工智能），也被认为是 21 世纪三大尖端技术（基因工程、纳米科学、人工智能）之一。

2. 图灵测试

图灵测试（Turing test）是由艾伦·图灵发明的，指提问者与被测试者（一个人和一台机器）隔开的情况下，通过一些装置（如键盘）向被测试者随意提问。这些问题随便是什么都可以。进行多次测试后，如果有超过 30% 的测试者不能确定被测试者是人还是机器，那么这台机器就通过了测试，并认为其具有人类智能。

图灵测试采用“问”与“答”模式，即提问者通过控制打字机向两个测试对象通话，其中一个是人，另一个是机器。要求提问者不断提出各种问题，从而辨别回答者是人还是机器。

图灵还为这项测试亲自拟定了几个示范性问题：

问：请给我写出有关“第四号桥”主题的十四行诗。

答：不要问我这道题，我从来不会写诗。

问：34957 加 70764 等于多少？

答：（停 30 秒后）105721。

问：你会下国际象棋吗？

答：是的。

问：我在我的 K1 处有棋子 K；你仅在 K6 处有棋子 K，在 R1 处有棋子 R。轮到你走，你应该下哪步棋？

答：（停 15 秒钟后）棋子 R 走到 R8 处，将军！

图灵指出，如果机器在某些现实的条件下，能够非常好地模仿人回答问题，以至提问者在相当长时间里误认为它不是机器。那么机器就可以被认为是具有思维的。

从表面上看，要使机器回答按一定范围提出的问题似乎没有什么困难，可以通过编制特定的程序来实现。然而，如果提问者并不遵循常规标准，则编制回答的程序是极其困难的事情。例如，提问与回答呈现出下列状况：

第一种情况

问：你会下国际象棋吗？

答：是的。

问：你会下国际象棋吗？

答：是的。

问：请再次回答，你会下国际象棋吗？

答：是的。

第二种情况

问：你会下国际象棋吗？

答：是的。

问：你会下国际象棋吗？

答：是的，我不是已经说过了吗？

问：请再次回答，你会下国际象棋吗？

答：你烦不烦，为什么老提同样的问题。

那么，你会认为哪一个机器，哪一个是人呢？

上述两种对话的区别在于：第一种可明显地感到回答者是从知识库中提取简单的答案；第二种则具有分析综合的能力，回答者知道提问者在反复提出同样的问题。

“图灵测试”没有规定问题的范围和提问的标准，如果想要制造出能通过测试的机器，以现有的技术水平，必须在计算机中储存人类所有可以想到的问题，以及储存对这些问题的所有合乎常理的回答，并且还需要理智地做出选择。

如果你是一个图灵测试的提问者，你会提出什么问题来判断机器和人呢？

3. 西尔勒的中文房间

“中文房间”（见图 1-1）最早由美国哲学家约翰·西尔勒（John Searle）于 20 世纪 80 年代初提出。这个实验要求你想象一位只会说英语的人身处一间房间之中，这间房间除了门上有一个小窗口以外，全部都是封闭的。他随身带着一本写有中文翻译程序的书。房间里还有足够多的稿纸、铅笔和橱柜。写着中文的纸片通过小窗口送入房间中。根据西尔勒设计的实验，房间中的人可以使用他带的书来翻译这些文字并用中文回复。虽然他完全不会中文，西尔勒认为通过这个过程，房间里的人可以让任何房间外的人以为他会说流利的中文。



图 1-1 中文房间

西尔勒创造了“中文房间”思想实验来反驳计算机和其他人工智能能够真正思考的观点——房间里的人不会说中文，他不能够用中文思考，但因为他拥有某些特定的工具，他甚至可以让他以中文为母语的人以为他能流利地说中文。根据西尔勒的实验，计算机就是这样工作的。计算机无法真正地理解接收到的信息，但它可以运行一个程序来处理信息，然后给人以“智能”的印象。

4. 人工智能的分类

人工智能的概念在很久以前就提出来了，比较流行的也是该领域较早的定义是由约翰·麦卡锡在 1956 年的达特茅斯会议上提出的：人工智能就是要让机器的行为看起来就像是人所表现出的智能行为一样，但是这个定义似乎忽略了强人工智能的可能性。另一个定义是指人工智能是人造机器所表现出来的智能性。总体来讲，人工智能的概念就是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学，大致分为三种类型，分别是弱人工智能、强人工智能和超人工智能。

1) 弱人工智能

弱人工智能（artificial narrow intelligence, ANI）是指擅长于单个方面的人工智能。例如有能战胜围棋世界冠军的人工智能阿尔法狗，但是它只会下围棋，如果我们问它其他的问题，那么它就不知道怎么回答了。只擅长单方面能力的人工智能就是弱人工智能。

虽然现在还没有达到强人工智能的阶段，但是弱人工智能已经相当普及，成为人们日常生活必不可少的一部分，市场上已经有很多弱人工智能的应用。

汽车的弱人工智能系统：从控制防抱死系统的计算机到控制器，到注入参数的计算机。百度研发的无人驾驶车，就包括了很多弱人工智能，这些弱人工智能能够感知周围环境并做出反应。

手机上的弱人工智能：地图导航软件、音乐电台推荐、查询明天的天气和 Siri 聊天以及其他很多应用其实都是弱人工智能。

谷歌、百度、360 等各大搜索引擎都是巨大的弱人工智能，背后是非常复杂的排序算法和内容检索。除了这些，还有军事、制造、金融、医疗等很多领域都广泛应用了各种复杂的弱人工智能。

2) 强人工智能

强人工智能（artificial general intelligence, AGI）是一种类似于人类级别的人工智能。强人工智能是指在各方面都能与人类比肩的人工智能，人类能干的脑力活它都能干。强人工智能就是一种宽泛的心理能力，能够进行思考、计划、解决问题、抽象思维、理解复杂理念、快速学习和从经验中学习等操作，强人工智能在进行这些操作时应该和人类一样得心应手。

3) 超人工智能

科学家把超人工智能（artificial super intelligence, ASI）定义为在几乎所有领域都比最聪明的人类大脑要聪明很多，包括科学创新、通识和社交技能。超人工智能可以是各方面都比人类强一点，也可以是各方面都比人类强万亿倍。超人工智能也正是为什么人工智能这个话题这么火热的缘故，同样也是为什么永生和灭绝这两个词总是出现在人们的口中。所以对于超人工智能的发展还是需要人们好好把控。

就目前而言，人类已经掌握了弱人工智能。弱人工智能无处不在，人工智能革命是从弱人工智能，通过强人工智能，最后会成为超人工智能。其实不管什么人工智能，都需要我们好好地控制，期盼将来人工智能能够给人类带来更大的福音，造福整个地球。

三、人工智能的发展

随着数字化、智能化的发展，“人工智能”已被频繁提及。人工智能的发展历史最早可以追溯到 20 世纪 50 年代，其整个发展历程可划分为五个阶段。

1. 人工智能横空出世

20 世纪 40—50 年代，人工智能横空出世。1950 年，马文·明斯基（见图 1-2），后被人称为“人工智能之父”与他的同学迪恩·埃德蒙兹一起，建造了世界上第一台神经网络计算机。这也可看作是人工智能的起点。巧合的是，同样是在 1950 年，被称为“计算机之父”的艾伦·图灵（见图 1-3）提出了一个举世瞩目的想法——图灵测试。按照图灵的设想：如果一台机器能够与人类开展对话而不能被辨别出机器身份，那么这台机器就具有智能。而就在这一年，图灵还大胆预言了真正具备智能机器的可行性，甚至有很多学者认为：二十年内，人工智能会做到人类能做到的一切。



图 1-2 马文·明斯基

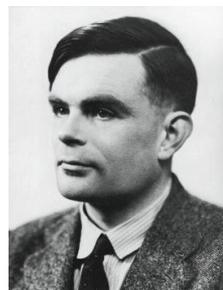


图 1-3 艾伦·图灵

1948 年，早在计算机时代刚刚进入黎明时，诺伯特·维纳就提出了“控制论”的概念。维纳是最先预见到信息技术两面性的人，这把双刃剑可能逃离人类掌控并反过来控制人类。

1950 年，图灵发表了一篇划时代的论文，预言了创造出具有真正智能的机器的可能性。图灵测试是人工智能哲学方面第一个严肃的提案。

1951 年，马文·明斯基与迪恩·埃德蒙兹一起建造了第一台神经网络计算机 SNARC。

1954 年，乔治·戴沃尔设计了世界上第一台可编程机器人。

1955 年，赫伯特·纽尼尔和艾伦·西蒙在 J.C. 肖的协助下开发了“逻辑理论家”程序。这个程序能够证明《数学原理》中前 52 个定理中的 38 个，其中某些证明比原著更加新颖和精巧。这一断言后来被约翰·西尔勒称为“强人工智能”，即机器可以像人一样具有思想。

1956 年，马文·明斯基与约翰·麦卡锡、克劳德·香农等人一起发起并组织“达特茅斯

会议”，并在会议上首度提出“人工智能”概念，这次会议之后被誉为“人工智能的起点”。

2. 人工智能的黄金时代

20 世纪 50—70 年代是人工智能的黄金时代。

1956 年，乔治·戴沃尔与约瑟夫·恩格尔伯格创建了世界上第一家机器人公司——尤尼梅特。

1957 年，赫伯特·细厄尔和艾伦·西蒙等人开始研究一种不依赖具体领域的通用问题求解器，他们称之为通用解题者（general problem solver, GPS），这一时期，搜索式推理是许多人工智能程序共同使用的基本算法。

1958 年，约翰·麦卡锡发明 Lisp 计算机分时编程语言，该语言至今仍在人工智能领域广泛使用。

1958 年，美国国防部高级研究计划局（DARPA）成立，主要负责高新技术的研究、开发和应用。

1962 年，世界上首款工业机器人“尤尼梅特”开始在通用汽车公司的装配线上应用。

1963 年 6 月，麻省理工学院从新建立的 ARPA 即后来的 DARPA 获得了 220 万美元经费，用于资助 MAC 项目，其中包括马文·明斯基和麦卡锡 5 年前建立的 AI 研究组。此后 ARPA 每年提供 300 万美元，直到 20 世纪 70 年代为止。

1966—1972 年，美国斯坦福国际研究所（Stanford Research Institute, SRI）研制了移动式机器人 Shakey。Shakey 是首台采用了人工智能学的移动机器人，引发了人工智能早期工作的大爆炸。

1966 年，麻省理工学院的系统工程师约瑟夫·魏泽堡和精神病学家肯尼斯·科尔比发布了世界上第一个聊天机器人 Eliza。Eliza 的智能之处在于她能通过脚本理解简单的自然语言，并能产生类似人类的互动。而其中最著名的脚本便是模拟罗杰斯心理治疗师。

1968 年 12 月 9 日，加州斯坦福研究所的道格拉斯·恩格尔巴特发明计算机鼠标，构想出了超文本链接概念，它在几十年后成了现代互联网的根基。恩格尔巴特提倡“智能增强”而非取代人类，被誉为“鼠标之父”。

1972 年，维诺格拉德在麻省理工学院建立了一个用自然语言指挥机器人动作的系统 SHIRDLO，它能用普通的英语句子与人交流，还能做出决策并执行操作。

3. 人工智能第一次低谷

20 世纪 70—80 年代，是人工智能的第一次低谷。但低谷也是沉淀，可以探索更多可能。

20 世纪 70 年代，人工智能进入低谷期。由于科研人员在人工智能方面的评估不足，导致美国国防高级研究计划部署的合作计划失败。这一事件的发生，让人们人工智能的可持续性和可行性产生了质疑，在当时，不仅计算机的性能不能够达到人工智能的需求水平，逻辑学、心理学等也远远没有达到人工智能技术的发展需要。因此，人工智能被进一步搁置，美国政府把更多的经费放在了其他的项目上，人工智能进入了发展中的低谷期。

20 世纪 70 年代初，人工智能遭遇瓶颈。当时，计算机有限的内存和处理速度不足以解决任何实际的人工智能问题，多个机构的研究经费开始被削减。

约翰·西尔勒于 1980 年提出“中文房间”实验，试图证明程序并不“理解”它所使用的符号，即所谓的“意向性”问题。西尔勒认为，如果符号对于机器而言没有意义，那么就不能认为机器是在“思考”。

4. 人工智能的繁荣与寒冬

(1) 1980—1987 年，人工智能迎来了大繁荣。20 世纪 80 年代，随着美国一批网络设备的开发，人工智能程序的系统被重视起来。其中智能化计算机软件广泛使用，苹果、微

软等公司相继开发智能操作系统。国际象棋冠军卡斯帕罗夫与“深蓝”计算机决战，“深蓝”获胜，这是人工智能发展的重要里程碑。

1981年，日本经济产业省拨款 8.5 亿美元支持第五代计算机项目。其目标是造出能够与人对话，能翻译语言，能解释图像，并且像人一样推理的机器。其他国家纷纷做出响应，DARPA 也行动起来，组织了战略计算促进会，其 1988 年向人工智能领域的投资是 1984 年的 3 倍。

约翰·霍普菲尔德和大卫·鲁姆哈特重新发展了神经网络理论，人工智能再获新生。

1982 年年初，硅谷著名人工智能公司 Teknowledge 终于能够用两个月的时间处理 100 万美元的业务了。

1986 年，在里根时代“星球大战计划”的推动下，美国与人工智能相关的软硬件销售额高达 4.25 亿美元。

(2) 1987—1993 年，人工智能的寒冬降临。20 世纪 80 年代中期，商业机构对人工智能的追捧与冷落符合经济泡沫的经典模式，泡沫的破裂也是政府机构和投资者考量人工智能时所要考虑的一个因素。

20 世纪 80 年代晚期，战略计算促进会大幅削减对人工智能领域的资助。DARPA 的新任领导认为人工智能并非“下一个浪潮”，拨款将倾向于那些看起来更容易出成果的项目。

5. 人工智能发展的春天

1993 年至今，人工智能蓬勃发展，人工智能技术呈现出井喷的发展现状。人脸识别、指纹识别、语音助手、智能手环等产品越来越多地涌入我们的生活，也使我们的生活变得更为快捷、简便，人工智能不仅可以实现生活的简便化，更可以为我们提供远程医疗、远程法庭等生活服务。相信随着科技、数字、经济文化的不断发展，人工智能会给我们的世界带来更多的色彩。

1997 年 5 月，“深蓝”成为战胜国际象棋世界冠军加里·卡斯帕罗夫的第一个计算机系统。

越来越多的人工智能研究者开始开发和使用复杂的数学工具。人工智能已成为一门更严格的科学分支。这些变化被视为一场“革命”和“简约派的胜利”。

人工智能在产业界开始发挥重要作用。应用了人工智能技术的有数据挖掘、工业机器人、物流语音识别、银行业软件、医疗诊断和搜索引擎等。

2005 年，斯坦福大学研发的一台机器人在一条沙漠小径上成功地自动行驶了约 210 公里，赢得了 DARPA 挑战大赛头奖。

2007 年，亚当·奇耶创立 Siri，苹果公司在 2010 年 4 月 28 日完成了对其的收购。

2010 年，塞巴斯蒂安·特龙领导的谷歌无人驾驶汽车曝光，当时已经创下了行驶超过 16 万公里无事故的纪录。

2013 年年末，Facebook 创始人兼 CEO 马克·扎克伯格前往位于塔霍湖的一家酒店参加神经信息处理系统（NIPS）技术会议。扎克伯格的到来成了一个风向标，人工智能再一次从单纯的学术研究走向商业化。

2016 年 3 月，谷歌公司人工智能程序 AlphaGo 与韩国棋手李世石在围棋上正面交锋。李世石最终以 1:4 的成绩不敌 AlphaGo，投子认输。

2017 年 5 月，AlphaGo 在中国乌镇“围棋峰会”挑战排名世界第一的世界围棋冠军柯洁，并以 3:0 获胜。

四、人工智能的应用

2017年7月，国务院印发了《新一代人工智能发展规划》，人工智能作为新一轮产业变革的核心驱动力，必须与各行各业融合才能发挥作用，形成真正有效的行业智能，以此来助力传统行业转型升级，加快人工智能应用落地。

1. 智能制造

智能制造是在基于互联网的物联网意义上实现的包括企业与社会在内的全过程的制造，把工业 4.0 的“智能工厂”“智能生产”“智能物流”进一步扩展到“智能消费”“智能服务”等全过程的智能化中去，只有在这些意义上，才能真正地认识到我们所面临的前所未有的形势。人工智能在制造业的应用主要有三个方面：

- (1) 智能装备，包括自动识别设备、人机交互系统、工业机器人以及数控机床等具体设备。
- (2) 智能工厂，包括智能设计、智能生产、智能管理以及集成优化等具体内容。
- (3) 智能服务，包括大规模个性化定制、远程运维以及预测性维护等具体服务模式。

虽然目前人工智能的解决方案尚不能完全满足制造业的要求，但作为一项通用性技术，人工智能与制造业融合是大势所趋。

2. 智能家居

智能家居主要是基于物联网技术，通过智能硬件、软件系统和云计算平台构成一套完整的家居生态圈。用户可以进行远程控制设备，设备间可以互联互通，并进行自我学习，来整体优化家居环境的安全性、节能性、便捷性等。

值得一提的是，近两年随着智能语音技术的发展，智能音箱成为一个爆发点。智能音箱不仅是音响产品，同时也是涵盖了内容服务、互联网服务及语音交互功能的智能化产品；不仅具备 WiFi 连接功能，提供音乐、有声读物等内容服务及信息查询、网购等互联网服务，还能与智能家居连接，实现场景化智能家居控制。

3. 智慧金融

人工智能的产生和发展，不仅促进金融机构服务主动性、智慧性，有效提升了金融服务效率，而且提高了金融机构风险管控能力，对金融产业的创新发展带来积极影响。

人工智能在金融领域的应用主要包括智能获客、身份识别、大数据风控、智能投顾、智能客服、金融云等，该行业也是人工智能渗透最早、最全面的行业。未来人工智能将持续带动金融行业的智能应用升级和效率提升。

4. 智能零售

人工智能在零售领域的应用已十分广泛，正在改变人们购物的方式。无人便利店、智慧供应链、客流统计、无人仓/无人车等都是热门方向。通过大数据与业务流程的密切配合，人工智能可以优化整个零售产业链的资源配置，为企业创造更多效益，让消费者体验更好。

在设计环节中，机器可以提供设计方案；在生产制造环节中，机器可以进行全自动制造；在供应链环节中，由计算机管理的无人仓库可以对销量以及库存需求进行预测，合理进行补货、调货；在终端零售环节中，机器可以智能选址，优化商品陈列位置，并分析消费者购物行为。

5. 智慧交通

大数据和人工智能可以让交通更智慧，智能交通系统是通信、信息和控制技术在交通系统中集成应用的产物。通过对交通中的车辆流量、行车速度进行采集和分析，可以对交通实

施监控和调度，有效提高通行能力、简化交通管理、降低环境污染等。

人工智能还可为我们的安全保驾护航。人长时间开车会感觉到疲劳，容易出交通事故，而无人驾驶则很好地解决了这些问题。无人驾驶系统还能对交通信号灯、汽车导航地图和道路汽车数量进行整合分析，规划出最优交通线路，提高道路利用率，减少堵车情况，节约交通出行时间。

6. 智能安防

安防领域涉及的范围较广，小到关系个人、家庭安全，大到与社区、城市、国家安全息息相关。目前智能安防类产品主要有四类：人体分析、车辆分析、行为分析、图像分析；在安防领域的应用主要通过图像识别、大数据及视频结构化等技术进行；从行业角度来看，主要在公安、交通、楼宇、金融、工业、民用等领域应用较广。

7. 智慧医疗

当下人工智能在医疗领域应用广泛，从最开始的药物研发到操刀做手术，利用人工智能都可以做到。眼下，医疗领域人工智能初创公司按领域可划分为八个主要方向，包括医学影像与诊断、医学研究、医疗风险分析、药物挖掘、虚拟护士助理、健康管理监控、精神健康以及营养学。其中，协助诊断及预测患者的疾病已经逐渐成为人工智能技术在医疗领域的主流应用方向。

8. 智慧教育

通过图像识别，可以进行机器批改试卷、识别题目、自动答题等；通过语音识别可以纠正、改进发音；而人机交互可以进行在线答疑解惑等。人工智能与教育的结合一定程度上可以改善教育行业师资分布不均衡、费用高昂等问题，从工具层面给师生提供更有效率的学习方式，但还不能对教育内容产生实质性的影响。

9. 智能物流

物流行业通过利用智能搜索、推理规划、计算机视觉以及智能机器人等技术在运输、仓储、配送装卸等流程上已经进行了自动化改造，能够基本实现无人操作。例如，利用大数据对商品进行智能配送规划，优化配置物流供给、需求匹配、物流资源等。目前物流行业大部分人力分布在“最后一公里”的配送环节。



小知识



1-1 我们身边的人工智能



操作工单

工单一 参与图灵测试

1. 提出问题

假如你参与了图灵测试，你会向计算机提什么问题？

2. 问题描述

你提出一个问题，会得到一个回答，你不知道这个回答是来自计算机还是人类。你要根

据回答来判断这个回答是出自计算机还是出自人。你会提出什么有意思的问题？

3. 图灵测试示意图

图灵测试示意图如图 1-4 所示。



图 1-4 图灵测试示意图

4. 设计思路和判断依据

5. 写出你的问题与预设答案

工单二 感受讯飞输入法的人工智能黑科技

1. 提出问题

怎样在手机上提高输入速度？

2. 问题描述

提到手机输入法，大多数人都不陌生，一款好用的手机输入法不仅能提高沟通的效率，也会大大提升使用手机其他功能的方便程度。你知道在人工智能的参与下怎样提高输入速度吗？普通话不好也可以吗？不会读的字怎么输入？

3. 讯飞输入法的特点

讯飞输入法是一款集智能语音、手写、拼音等输入方式于一体，便捷、好用的人工智能输

入法。

(1) 语音输入：解放双手。普通话识别：轻击键盘上方的麦克风，即可体验 400 字/min 的输入速度，实现说话秒变文字，聊天速度天下无敌。方言识别：将方言变成文字，讯飞输入法能“听懂”23 种方言，包括粤语、四川话、东北话、河南话、闽南语等。随声译：说话随声翻译，包括中译英、英译中、中译韩、中译日四种翻译，逐步满足日益增长的跨语言交流需求。

(2) 特色皮肤：妙趣横生。

(3) 斗图：100 000 张表情资源，“金馆长”“学友哥”“叶良辰”“达康书记”齐聚一堂，让你聊天斗图不输阵。

(4) 键盘手写。连续手写：独家支持键盘直接手写、连续手写，一次可以连续书写多个字（目前点划、双键等部分布局的英文键盘，暂不支持连续手写功能）。

4. 输入法实测

(1) 使用语音输入，你一分钟最多能输入多少个字？

(2) 试一下手写输入你不知道读音的字？

(3) 与同学们“斗图”。

(4) 设置你的专属输入法皮肤。

(5) 试试语音输入将你的方言转换为文字。

(6) 试试语音输入将英语转换为汉字。

5. 写出你最喜欢的输入方式

工单三 百度 AI 体验中心

1. 提出问题

你在上学的路上有没有看到不认识的植物呢？

出差在外，怎样查询发票的真伪呢？

你想检测一下自己的五官和颜值吗？

2. 分析问题

百度 AI 体验中心提供了图像技术、人脸与人体识别、语音技术、知识与语义四个功能，其中在图像技术中包含了各类证件的文字识别和动物、植物、车型、货币、菜品等图像识别以及图像处理功能，在人脸与人体识别中可以检测人像的五官和颜值。

3. 使用方法

(1) 打开微信搜索“百度 AI 体验中心”小程序。

(2) 从网上下载一张植物图片，测试植物识别功能。

(3) 从网上下载一张动物图片，测试动物识别功能。

(4) 从网上下载一张汽车图片，测试车型识别功能。

(5) 使用自己或朋友的照片，测试一下五官和颜值。

(6) 输入一段文字，转换为语音。

上述使用过程参考图 1-5~图 1-7。



图 1-5 百度 AI 体验中心

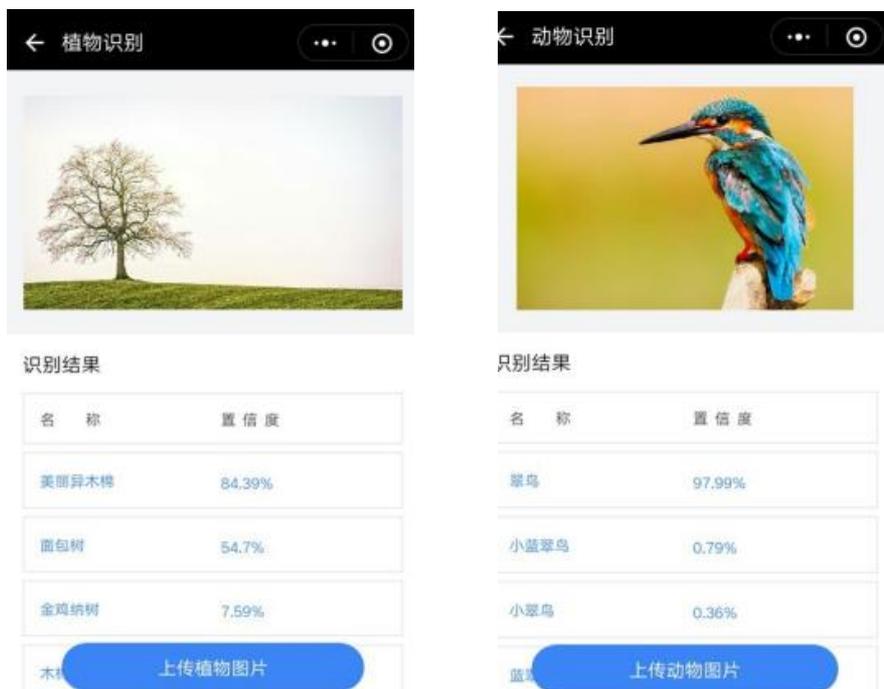


图 1-6 百度 AI 体验中心的植物识别和动物识别

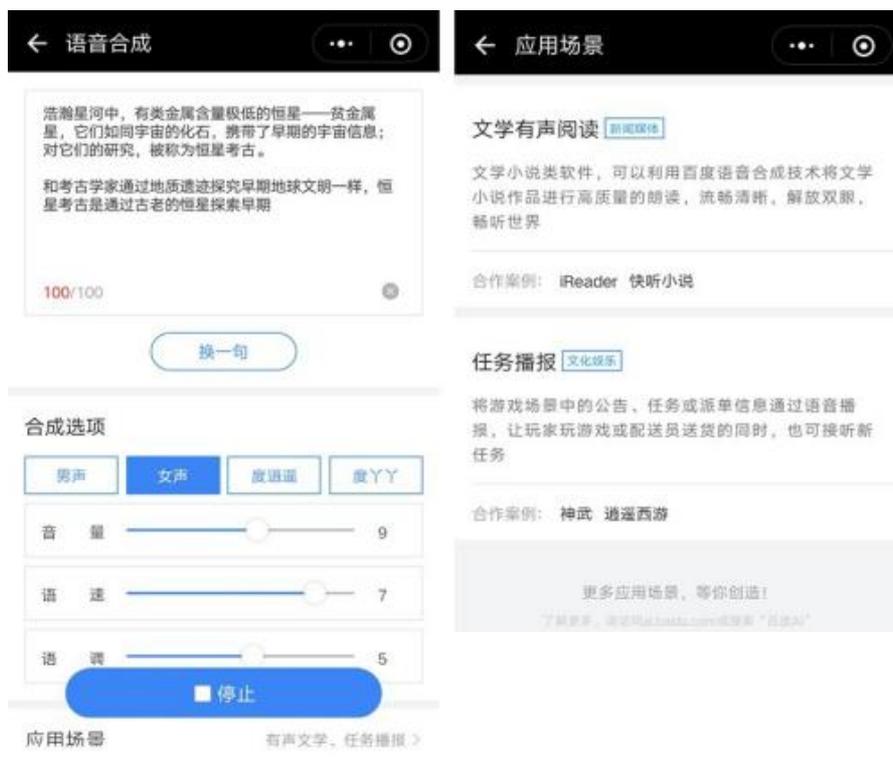


图 1-7 百度 AI 体验中心的语音合成和应用场景

4. 记录自己的测试过程

活动二 拥抱大数据时代

📍 活动目标

- (1) 认知大数据的概念。
- (2) 认知大数据技术的概念。
- (3) 认知大数据技术的发展。
- (4) 认知大数据技术的应用。
- (5) 认知大数据技术与云计算、物联网的联系和区别。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的内容。
- (3) 小组成员互相沟通，服从小组管理。

📍 知识储备

一、什么是大数据

大数据从字面上看就是很“大”的数据。早在很多年前百度首页导航每天需要提供的数据就已经超过 1.5 PB，这些数据如果打印出来，将超过 5 000 亿张 A4 纸。

何为大？——数据度量。

1Byte = 8 bit

1KB = 1 024 Bytes

1MB = 1 024 KB = 1 048 576 Bytes

1GB = 1 024 MB = 1 048 576 KB = 1 073 741 824 Bytes

1TB = 1 024 GB = 1 048 576 MB = 1 099 511 627 776 Bytes

1PB = 1 024 TB = 1 048 576 GB = 1 125 899 906 842 624 Bytes

1EB = 1 024 PB = 1 048 576 TB = 1 152 921 504 606 846 976 Bytes

1ZB = 1 024 EB = 1 180 591 620 717 411 303 424 Bytes

1YB = 1 024 ZB = 1 208 925 819 614 629 174 706 176 Bytes

《红楼梦》含标点约 87 万字（不含标点 853509 字）

每个汉字占两个字节：1 汉字 = 16 bit = 2×8 bit = 2 Bytes

1GB 约等于 671 部《红楼梦》

1TB 约等于 631 903 部《红楼梦》

1PB 约等于 647 068 911 部《红楼梦》

大数据说白了就是巨量数据集合，来源于海量用户一次次的行为数据。但大数据的战略意义不在于掌握庞大的数据信息，而在于对这些含有意义的数据进行专业化的处理。

麦肯锡全球研究所曾给出大数据的定义：一种规模大到在获取、存储、管理、分析方面大大超出了传统数据库软件工具能力范围的数据集合，具有海量的数据规模、快速的数据流转、多样的数据类型和价值密度低四大特征，也就是人们常说的大数据的 4V（即 volume、variety、value、velocity）特征，即大量、多样性、价值和及时性。

二、什么是大数据技术

大数据技术是指大数据的应用技术，涵盖各类大数据平台、大数据指数体系等大数据应用技术。大数据分析常与云计算联系在一起，因为实时的大型数据集分析需要像 MapReduce 一样的框架来向数十、数百，甚至数千台的计算机分配工作。

目前我们生活的很多方面都需要大数据处理技术来分析大量实时数据，并提出结论和预测，以减少未来的风险。

大数据技术主要分为运营大数据技术和分析性大数据技术两种类型。

1. 运营大数据技术

运营大数据技术与我们生活中的日常数据有关。这可能是在线交易、社交媒体或特定组织的数据等。简单地可以认为这是一种原始数据，用于提供分析性大数据技术。

运营大数据技术的一些示例：

- (1) 网上订票，包括火车票、机票、电影票等。
- (2) 在线购物，淘宝、京东、拼多多、支付宝交易等。
- (3) 来自社交媒体网站（如抖音、快手、QQ、微信等应用程序）的数据。
- (4) 医保、车牌号、身份证等与我们息息相关的个人详细信息。

2. 分析性大数据技术

分析性大数据就像大数据技术的高级版本一样，它运营大数据要复杂一些。简言之分析性大数据是实际绩效的组成部分，而关键的实时业务决策是通过分析运营大数据来制订的。

分析性大数据技术的一些示例：

- (1) 股票、基金等金融投资业。
- (2) 分析运行航海、船舶、飞行和太空任务等领域，其中结点的每一个信息都是至关重要的。
- (3) 天气预报信息。
- (4) 监视特定患者健康状况的医学领域。

三、大数据时代的发展

第一次信息化浪潮使计算机开始普及，第二次信息化浪潮人类全面进入互联网时代，第三次信息化浪潮解决了信息爆炸问题，大数据时代到来。存储设备容量不断增加，CPU 处理能力大幅度提升，网络带宽不断增加为大数据时代提供技术支持。数据产生方式从运营式阶段到用户原创内容阶段再到感知式阶段的变革促成了大数据时代的到来。

大数据时代的发展经历了三个阶段：

- (1) 萌芽阶段：20 世纪 90 年代至 21 世纪初，随着数据挖掘理论和数据库技术的逐步成熟，一批商业智能工具和知识管理技术开始应用，如数据仓库、专家系统、知识管理系统等。
- (2) 成熟阶段：21 世纪前 10 年，Web 2.0 应用发展，非结构化数据大量产生，传统处理方法难以应对，带动了大数据技术的快速突破，大数据解决方案逐渐走向成熟，形成了并行计算与分布式系统两大核心技术，谷歌的 GFS 和 MapReduce 等大数据技术受到追捧，Hadoop 平台开始大行其道。
- (3) 大规模应用期：2010 年以后，大数据应用渗透各行业，数据驱动决策，信息社会智能化程度大幅度提高。

四、大数据技术的应用

大数据无处不在，包括金融、汽车、餐饮、电信、能源、体育和娱乐等在内的社会各行各业都已经融入了大数据的印记。

制造业：利用工业大数据提升制造业水平，包括产品故障诊断与预测、分析工艺流程、改进生产工艺，优化生产过程能耗、工业供应链分析与优化、生产计划与排程。

金融业：大数据在高频交易、社交情绪分析和信贷风险分析三大金融创新领域发挥着重大作用。

汽车行业：利用大数据和物联网技术的无人驾驶汽车在不远的未来将走入我们的日常生活。

互联网行业：借助于大数据技术分析用户行为，进行商品推荐和针对性广告投放。

餐饮行业：利用大数据实现餐饮 O2O 模式，彻底改变传统餐饮经营方式。

电信行业：利用大数据技术实现客户离网分析，及时掌握客户离网倾向，出台客户挽留措施。

能源行业：随着智能电网的发展，电力公司可以掌握海量的用户用电信息，利用大数据技术分析用户用电模式，可以改进电网运行，合理设计电力需求响应系统，确保电网运行安全。

物流行业：利用大数据优化物流网络，提高物流效率，降低物流成本。

城市管理：利用大数据实现智能交通、环保监测、城市规划和智能安防。

生物医学：大数据可以帮助我们实现流行病预测、智慧医疗、健康管理，同时还可以帮助我们解读 DNA，了解更多的生命奥秘。

公共安全领域：政府利用大数据技术构建强大的国家安全保障体系，公共安全领域的大数据分析应用，可以有效预防犯罪。

个人生活：大数据还可以应用于个人生活，利用与每个人相关联的“个人大数据”，分析个人生活行为轨迹，为其提供更加周到的个性化服务。

大数据的价值远不止于此，大数据对各行各业的渗透，是推动社会生产和生活的核心要素。

五、大数据与云计算

为解决互联网应用对大规模计算能力、数据存储能力的迫切需求，提出了云计算的概念。云计算是一种分布式计算平台，通过虚拟技术将海量的硬件资源和软件资源组合成虚拟资源池，并根据需求任务的大小，向虚拟资源池获取相应的计算和存储资源。

在大数据处理的需求下，出现了许多优秀的云计算平台，如 Apache 开源的 Hadoop、Google 的 MapReduce、微软的 Dryad 等。

在处理格式多样的大数据时，云计算能协调组织众多节点，提供廉价的资源和服务，具有较可靠的可扩展性和容错性。然而，对于大规模复杂的应用系统来说，云计算还有诸多的技术问题有待深入研究。

为应对数据密集型服务，云计算提供分布式并行编程技术、分布式并行数据库技术，可通过开源的编程接口和工具来调用服务，其优势是能高效处理结构简单的大数据，但对关系复杂的大数据的处理，在效率和准确率方面还不能令人满意。

大数据技术的目标是解决应用中多源、异构、海量数据的管理和使用问题，但其本身不具备处理大规模数据的存储资源和计算资源的能力，因此必须在已有成熟的技术基础上，引入新的与之相对应的大数据存储和计算平台。

云计算以数据为中心，对大数据集进行处理，并向用户返回高效服务，具有并行化、虚拟化、按需服务等特点。从数据管理角度来说，大数据技术是对数据组织结构的描述，研究重点是数据的查询、更新、索引等操作技术；而云计算则是一种分而治之、按需索取的大数据分布式服务模式。

这两个概念提出的背景都是为满足海量异构数据的组织和管理要求。从相互之间的影响来看，前者为后者提供了广阔的应用背景，后者为前者数据管理提供了存储和计算资源，两者相互促进，相互依存。

六、大数据与物联网

随着智能交通、智能家居、智能物流、智慧景区等应用的兴起，物联网已成为未来经济的

新增长点。美国、德国、英国、意大利和丹麦等国家争先推出物联网相关发展策略，使物联网的规模不断扩大。互联网到物联网的跨越，极大地推动了大数据的发展。

物联网是指把所有物品通过信息传感设备与互联网连接起来，实现智能化识别和管理。它从结构上分为四层，即实体层、感知层、网络层和应用层。每层都与数据的产生或者处理息息相关。大数据与物联网的结合使机遇与挑战并存。

首先，产生数据的平台多样化。从原来的个人计算机扩展为传感器、智能手机、各种业务系统、平板计算机、监控录像等，这使得感知层需要感知的数据呈现多样化。目前主流的感知技术有视频文字采集技术、红外线技术、传感器技术和蓝牙技术等，但随着感知的数据数量级的不断增加，相应的感知技术也要不断地改进和完善。

其次，物联网技术的局限性。事物的发展需要一个过程，处于发展初始阶段的物联网还受到一些技术的约束。在大数据的传输和处理方面，物联网技术还存在通信距离短、外部环境适应力不强、异构网络兼容性差等问题。传感器链接的距离范围是 100 m 到 1 000 m，不适合长距离的通信；当外部的环境发生变化，传感器的稳定性大幅度下降，对具有高性能计算存储系统的安全带来风险；物联网的标准建立在广电网、通信网和互联网等异构网的基础上，还没有统一完善的标准体系。

七、大数据与数据空间

大数据来自不同组织，它的跨域、分布、异构性以及海量的特点给传统的数据库管理系统带来巨大挑战，目前，管理着世界上最大数据的谷歌、雅虎和微软等公司，都不使用传统的数据管理系统，而是另辟蹊径去寻找可以满足大数据管理需要的技术。

数据空间是 M. Franklin 等人为应对信息量不断增长以及数据信息管理需求而引入的一种信息管理新概念。目前关于数据空间技术的研究主要集中在个人数据空间方面，并取得了一定成果。国外的研究工作主要以 iMeMex 和 SEMEx 两个个人数据管理系统为代表。iMeMex 由瑞士联邦理工学院开发，它推动了信息抽取和查询技术的进步，但缺点是不支持语义查询；SEMEx 由华盛顿大学开发，成功地把语义关联应用到实例中来高效提取信息。同时，麻省理工学院计算机科学系的 David R. Karger 等人研发了个人数据管理系统 Haystack，该系统采用了 URF (uniform resource identifier) 半结构化数据模型统一表示用户数据，体现了数据空间“pay as you go”的数据集成思想。美国华盛顿大学数据库研究组的 sharedviews 项目实现了名为 Homeview 的原型系统，该系统能够支持个人动态数据的共享，但数据的类型和共享方式有限。

在国内，数据空间技术已经开始受到广泛关注。中国人民大学孟小峰教授等人对数据空间的概念、实现数据空间支撑平台所需的关键技术进行了详细的阐述与分析，并带领中国人民大学网络与移动数据管理实验室研究团队研发了具有代表性的个人数据空间原型系统 OrientsPac。

综上所述，以物联网、云计算技术作为数据收集、数据管理手段，用数据空间技术来组织大数据，实现多层次、多粒度的大数据挖掘，是处理大规模数据行之有效的途径，也符合大数据管理和需求。



小知识



1-2 认识数据的单位

操作工单

工单 探寻生活中的大数据

1. 提出问题

在你平时的生活中，有哪些应用用到了大数据呢？

2. 问题描述

大数据应用在日常生活中非常普遍，大到国家事务，小的柴米油盐，并影响着我们的生活。其中的大数据案例也有很多，在你的身边，都有哪些大数据影响到你了呢？

3. 大数据的应用

1) 大数据正在改善我们的生活

大数据不单单只服务于企业和政府，同样也适用于我们生活当中的每个人。我们可以利用穿戴的装备（如智能手表或者智能手环）生成最新的数据，这让我们可以根据热量的消耗以及睡眠情况来追踪健康状况。

2) 业务流程优化

大数据也更多地帮助业务流程的优化。可以通过利用社交媒体数据、网络搜索以及天气预报挖掘出有价值的信息，其中大数据的应用最广泛的就是供应链以及配送路线的优化。

3) 理解客户、满足客户服务需求

大数据的应用目前在领域是最广为人知的，其重点是如何应用大数据更好地了解客户以及他们的爱好和行为。

4) 金融交易

大数据在金融行业主要是应用于金融交易。高频交易（HFT）是大数据应用比较多的领域。现在很多股权的交易都是利用大数据算法进行，这些算法越来越多地考虑了社交媒体和网站新闻来决定在未来几秒内是买入还是卖出。

5) 改善我们的城市

大数据还被应用于改善我们日常生活的城市，如基于城市实时交通信息、利用社交网络和天气数据来优化最新的交通情况。目前很多城市都在进行大数据的分析和试点。

4. 你身边的大数据应用

按照示例在下面的横线上写出我们身边的大数据应用。

如：淘宝 APP 的猜你喜欢。

如：外出旅游时用的导航。

5. 身处大数据时代，应该怎么保护自己的隐私
在下面的横线上写出保护自己隐私的方法。

任务二

Python 之人人要编程 ——计算思维养成



任务目标

- (1) 认知计算思维及其特征。
- (2) 认知 Python 语言及其特征。
- (3) 熟练掌握 Python 程序开发环境的安装，并完成一个简单程序架构设计。
- (4) 熟练掌握 Python 语言基本语法、变量、标识符和关键字、数值类型、常见运算符和表达式。
- (5) 培养计算思维解决问题的能力。
- (6) 培养学生思维方式、工作方法和创新能力等综合职业能力。



任务要点

- (1) Python 语言和其开发环境安装。
- (2) Python 语言基本语法。
- (3) Python 语言基本数值类型。
- (4) Python 语言常见运算符和表达式。
- (5) 简单 Python 程序架构设计。



信息素养目标

- (1) 职业素养。
 - ① 培养积极、有责任心的工作态度。
 - ② 培养模块化思维方式和工作方法。
 - ③ 培养分析问题、解决问题的能力。
 - ④ 培养小组协作互助成长的职业能力。
- (2) 技能素养。

培养规范化的程序设计、编辑和调试技能。
- (3) 创新能力。

培养运用信息技术独立解决问题的能力。



任务内容

- 活动一：编程宣言——hello, Python。
- 活动二：神奇计算器。

活动一 编程宣言——hello, Python

活动目标

- (1) 认知计算思维。
- (2) 认知 Python 语言。
- (3) 认知 Python 程序开发环境安装及应用。
- (4) 认知 Python 程序架构。
- (5) 认知 Python 程序编译与执行步骤。

活动要求

- (1) 学习态度积极，按时出勤；遵守课堂纪律，完成规定内容。
- (2) 小组成员互相沟通，服从小组管理。

知识储备

一、计算思维

计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。美国卡内基·梅隆大学的周以真教授于 2006 年提出了计算思维的概念。

1. 计算思维的特征

(1) 是人的，不是计算机的思维方式。计算思维是人类求解问题的一条途径，但绝非要使人像计算机那样思考。

(2) 根本的，不是刻板的技能。根本技能是每一个人为了在现代社会中发挥职能所必须掌握的。刻板技能意味着机械重复。具有讽刺意味的是，当计算机像人类一样思考之后，思维可就真的变成机械的了。

(3) 概念化，不是程序化。计算机科学不是计算机编程。像计算机科学家那样去思维意味着远不止能为计算机编程，还要求能够在抽象的多个层次上思维。

(4) 数学和工程思维的互补与融合。计算机科学在本质上源自数学思维，因为像所有的科学一样，其形式化基础建筑于数学之上。计算机科学又从本质上源自工程思维，因为我们建造的是能够与实际世界互动的系统，基本计算设备的限制迫使计算机科学家必须计算性地思考，不能只是数学性地思考。

2. 计算思维解决问题的过程

- (1) 分析要解决的问题。
- (2) 利用计算机和其他工具来解决该问题。
- (3) 要符合逻辑地组织和分析数据。
- (4) 通过抽象（如模型、仿真等）再现数据。

- (5) 通过算法思想一系列有序的步骤来制订问题的多个解决方案。
- (6) 分析有可行性的解决方案，找到最有效的方案。

二、认识 Python

Python 是一种面向对象、解释型语言，也是一种功能强大的通用型语言，已经具有近 20 年的发展历史，成熟且稳定。它包含了一组完善而且容易理解的标准库，能够轻松完成很多常见的任务。

1. Python 的发展历程



图 2-1 吉多·范·罗苏姆

Python 的创始人为吉多·范·罗苏姆（见图 2-1）。1989 年的圣诞节期间，吉多·范·罗苏姆（Guido van Rossum）为了在阿姆斯特丹打发时间，决心开发一个新的脚本解释程序，作为 ABC 语言的一种继承。

1991 年，第一个 Python 编译器诞生，它是一种面向对象的解释型计算机程序设计语言，它是用 C 语言实现的，并能够调用 C 库（.so 文件）。从一出生，Python 已经具有了：类（class）、函数（function）、异常处理（exception），包括表（list）和词典（dictionary）在内的核心数据类型，以及以模块（module）为基础的拓展系统。

Python 诞生在一个幸运的时间。20 世纪 90 年代初，个人计算机开始进入普通家庭，计算机的性能已经有了质的提升，计算机对软件性能的要求放宽，硬件厂商甚至渴望高需求软件的产生，以促进硬件的更新。此时 Internet 悄然渗入人们的生活，许多程序员以及资深计算机用户频繁使用 Internet 进行交流，这使得 Python 没有了硬件上的束缚与传播上的困难，再加上 Python 易于使用的特点，使 Python 得到了一定程度上的传播。2000 年 10 月，Python 2.0 发布，Python 从基于 maillist 的开发方式转为完全开源的开发方式，Python 社区已然成熟，Python 发展速度再度提高。

2. Python 的特点

Python 具有以下特点：

(1) 简洁易学。Python 程序简洁明了，它编写的程序就像是在读英语段落一样流畅。程序员在开发程序时，只需要专注解决问题，而不用搞明白语言本身的语法。

(2) 面向对象。Python 既支持面向过程编程，也支持面向对象编程。在“面向过程”的语言中，程序是由封装了可重用代码的函数构建成的。与其他语言如 C++ 和 Java 相比，Python 以一种非常强大且简单的方式实现了面向对象编程。

(3) 免费、开源。这对 Python 用户群的扩大起了至关重要的作用。而使用者的增加又丰富了 Python 的功能，使用者可以自由地发布这个软件的拷贝，阅读它的源代码，对它做改动，把它的一部分用于新的自由软件中，这实际上是一种良性循环。

(4) 丰富的库。Python 解释器拥有丰富的内置类和函数库，它可以帮助处理各种工作。除了这些标准库之外，世界各地的程序员通过开源社区又贡献了十几万个几乎涵盖各个应用领域的第三方函数库，使开发人员能够借助函数库实现某些复杂的功能。

(5) 可移植性强。Python 作为解释型语言，可以在任何安装有 Python 解释器的平台中执行，因此 Python 具有良好的可移植性，使用 Python 语言编写的程序可以不加修改地在任何平台中运行。

3. Python 的应用领域

Python 有丰富的库，其应用领域非常广。

(1) Web 开发。Python 拥有很多免费数据函数库、免费 Web 网页模板系统以及与 Web 服务器进行交互的库，可以实现 Web 开发，搭建 Web 框架，目前比较有名气的 Python Web 框架为 Django。从事该领域应从数据、组件、安全等多领域进行学习，从底层了解其工作原理并可驾驭任何业内主流的 Web 框架。

(2) 网络编程。网络编程是 Python 应用的另一方向，网络编程在生活和开发中无处不在，哪里有通信哪里就有网络，它可以称为是一切开发的“基石”。对于所有编程开发人员必须要知其然并知其所以然，所以网络部分将从协议、封包、解包等底层进行深入剖析。

(3) 爬虫开发。在爬虫领域，Python 几乎是霸主地位，将网络一切数据作为资源，通过自动化程序进行有针对性的数据采集以及处理。从事该领域应学习爬虫策略、高性能异步 IO、分布式爬虫等，并针对 Scrapy 框架源码进行深入剖析，从而理解其原理并实现自定义爬虫框架。

(4) 云计算开发。Python 是从事云计算工作需要掌握的一门编程语言，目前很火的云计算框架 OpenStack 就是由 Python 开发的，如果想要深入学习并进行二次开发，就需要具备 Python 的技能。

(5) 人工智能。美国国家航空航天局 (NASA) 和谷歌公司早期大量使用 Python，为 Python 积累了丰富的科学运算库，当 AI 时代来临后，Python 从众多编程语言中脱颖而出，各种人工智能算法都基于 Python 编写，尤其在 PyTorch 出现后，Python 作为 AI 时代头牌语言的位置基本确定。

(6) 游戏开发。在网络游戏开发中，Python 也有很多应用，相比于 Lua 或 C++ 语言，Python 比它们有更高阶的抽象能力，可以用更少的代码描述游戏业务逻辑，Python 非常适合编写 1 万行以上的项目，而且能够很好地把网络游戏项目的规模控制在 10 万行代码以内。

(7) 科学运算。Python 是一门很适合做科学计算的编程语言，从 1997 年开始，NASA 就大量使用 Python 进行各种复杂的科学运算，随着 NumPy、SciPy、Matplotlib、Enthought libraris 等众多程序库的开发，Python 越来越适合用于科学计算以及绘制高质量的 2D 和 3D 图像。

三、Python 开发环境的下载及安装

Python IDE 工具是每个 Python 开发者需要使用的，常见的 Python 编辑器有很多。本节将介绍如何安装和配置 Python 开发环境。

1. 安装 Python 解释器

(1) Python 官网提供了 Python 解释器的下载，下载页面如图 2-2 所示。

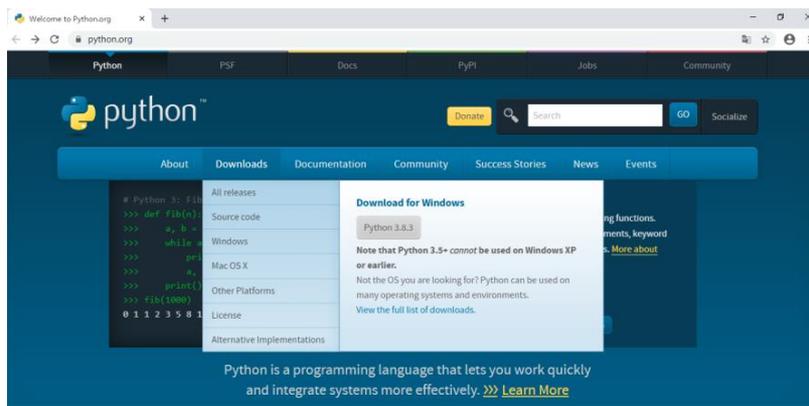


图 2-2 Python 官网的 Python 解释器下载页面

(2) 单击“Downloads”页面，选择 Windows 平台下的安装包，如图 2-3 所示。



图 2-3 选择 Windows 平台安装包

(3) 根据操作系统版本选择相应的软件包，如图 2-4 所示。

版	操作系统	描述	MD5总和	文件大小	GP6
压缩后的源tarball	源代码发布		a7c10a2ac3d62de75a0ca5204e2e7d07	24067487	SIG
XZ压缩源tarball	源代码发布		3000cf50aaa413052aef82fd2122ca78	17912964	SIG
macOS 64位安装程序	Mac OS X	适用于OS X 10.9及更高版本	dd5e7f64e255d21f9d407f39a7a11ba9	30119781	SIG
Windows帮助文件	视图		4aeeebd7cc8dd90d61e7cfd9a9cb9422	8568303	SIG
Windows x86-64可嵌入的zip文件	视图	适用于AMD64 / EM64T / x64	c12ffe7f4c1b447241d5d2aedc9b5d01	8175801	SIG
Windows x86-64可执行安装程序	视图	适用于AMD64 / EM64T / x64	fd2458fa0e9ead1dd9fbc2370a42853b	27805800	SIG
Windows x86-64基于Web的安装程序	视图	适用于AMD64 / EM64T / x64	17e989d2fec77f9f13cf987825b695c4	1364136	SIG
Windows x86可嵌入的zip文件	视图		8ee09403ec0cc2e89d43b4a4f6d1521e	7330315	SIG
Windows x86可执行安装程序	视图		452373e2c467c14220efeb10f40c231f	26744744	SIG
Windows x86基于Web的安装程序	视图		fe72582bbca3dbe07451fd05ece1d752	1325800	SIG

图 2-4 根据操作系统版本选择相应的软件包

(4) 下载完成后，双击安装包，启动安装程序，选择安装方式，Python 解释器的安装方式有两种：“Install Now”将采用默认安装方式，“Customize installation”可自定义安装路径。选择窗口下方的“Add Python 3.8 to PATH”复选框，然后单击“Install Now”即可开始自动安装 Python 解释器，如图 2-5、图 2-6 所示。



图 2-5 选择安装方式

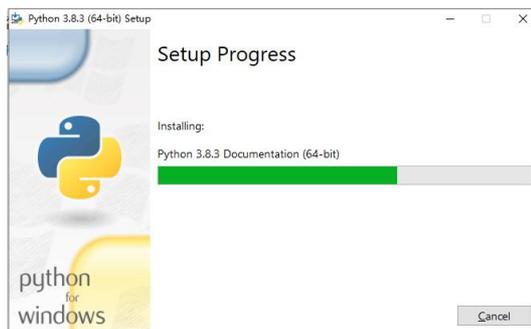


图 2-6 Python 编辑器安装

(5) Python 解释器的安装进度很快，安装成功的界面如图 2-7 所示。

(6) 运行 Python。在“开始”菜单中打开“Python 3.8”文件夹（见图 2-8），找到“IDLE (Python 3.8 64-bit)”并单击打开，打开后的窗口如图 2-9 所示。

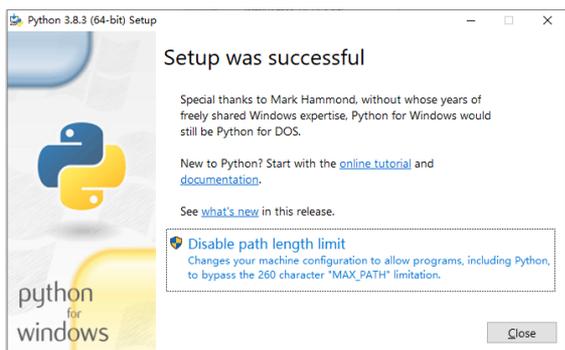


图 2-7 Python 安装成功的界面

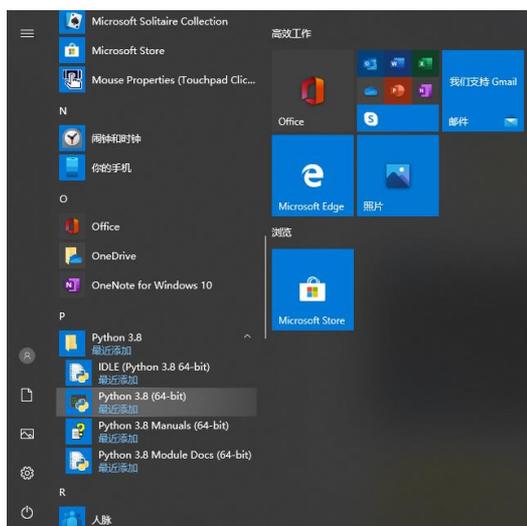


图 2-8 在“开始”菜单中打开“Python 3.8”文件夹

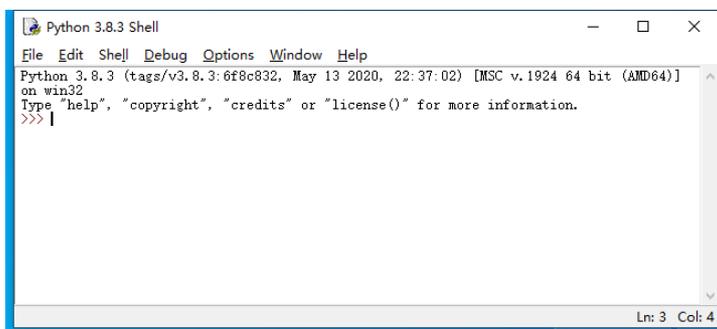


图 2-9 Python 解释器窗口

四、编写“hello, Python”程序

1. Python 编辑器

IDLE 环境中集成有一个 Python 编辑器，可以进行 Python 程序的编辑和执行。在 Python 3.8.3 Shell 窗口中，执行“File”菜单下的“New File”命令（见图 2-10），将会打开一个 Python 编辑器窗口。Python 编辑器除了能够编辑文本之外，还提供了许多辅助编写 Python 代码的功能特性，这些特性包括撤销和恢复、代码着色、智能缩进、语法提示、自动完成等。此外，Python 编辑器还支持多窗口，能够同时编辑多个 Python 源文件。

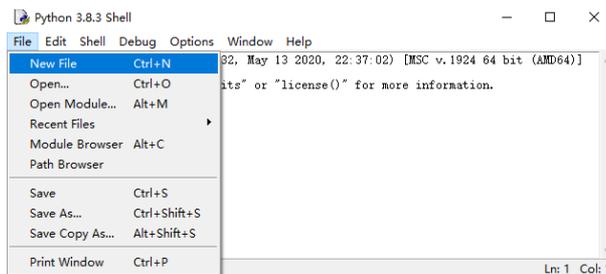


图 2-10 执行“File”菜单下的“New File”命令

2. 编写“hello, Python”程序

(1) 找到“IDLE (Python 3.8 64-bit)”并单击打开。在 Python 编辑器窗口的文本区域输入 Python 代码，如图 2-11 所示。

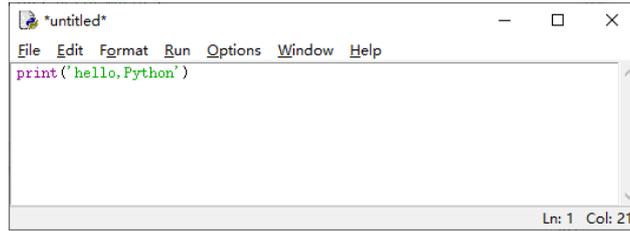


图 2-11 输入“hello, Python”程序

(2) 在 Python 编辑器窗口中，执行“Run”菜单下的“Run Module”命令，如图 2-12 所示。

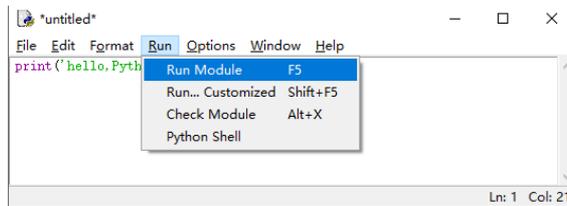


图 2-12 运行程序

(3) 这时会弹出“Save Before Run or Check”对话框，如图 2-13 所示。

(4) 单击“确定”按钮会弹出保存对话框，要求保存程序代码，将源代码保存为 test2-1.py，如图 2-14 所示。

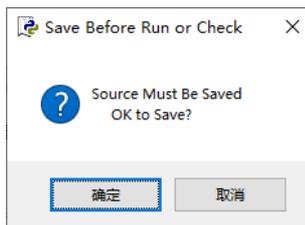


图 2-13 “Save Before Run or Check”对话框

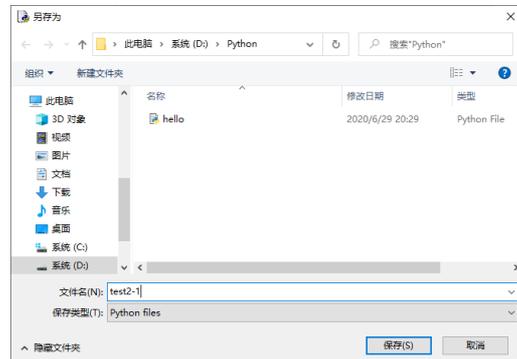


图 2-14 源文件保存对话框

(5) 单击“保存”按钮程序就会被执行，执行结果显示在“>>>”提示符之后，输出内容如图 2-15 所示。

```

>>>
===== RESTART: D:/Python/test2-1.py =====
hello, Python
>>>
    
```

图 2-15 程序执行结果



小知识



2-1 Python 之简易的集成开发工具——IDLE 编辑器



操作工单

工单一 Python 解释器的下载及安装

1. 操作步骤

- (1) 到 Python 官网，找到“Downloads”页面，选择 Windows 平台下的安装包。
- (2) 根据操作系统版本选择相应软件包下载。
- (3) 下载完成后，双击安装包，启动安装程序，选择安装方式，Python 解释器安装方式有两种：“Install Now”将采用默认安装方式，“Customize installation”可自定义安装路径。选择窗口下方的“Add Python 3.8 to PATH”复选框，然后单击“Install Now”即可开始自动安装 Python 解释器。

(4) 运行 Python。在“开始”菜单中的“Python 3.8”文件夹中找到“IDLE (Python 3.8 64-bit)”并单击打开。

2. 操作结果

预期结果如图 2-16 所示，请将实际结果简单绘制在右边的方框内。



图 2-16 预期结果与实际结果

3. 学习记录

工单二 编程宣言

1. 操作步骤

(1) 打开 IDLE，执行“File”菜单下的“New File”命令，新建一个.py 文件。

(2) 输入程序：

```
print('Hello, world!')
```

(3) 保存程序。

(4) 调试程序，并运行程序，按“F5”键或者执行“Run”菜单下的“Module”命令运行。

2. 操作结果

预期结果如图 2-17 所示，请将实际结果简单绘制在右边的方框内。

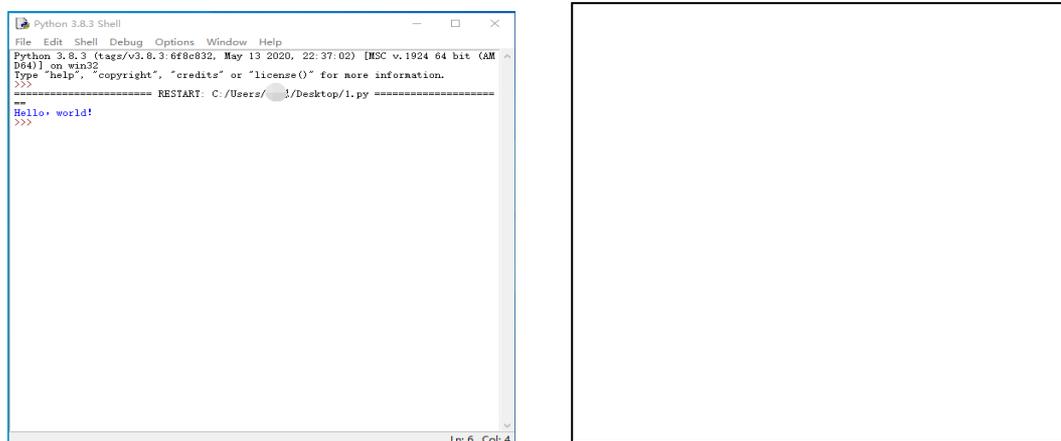


图 2-17 预期结果与实际结果

3. 学习记录

活动二 神奇计算器

📍 活动目标

- (1) 认知 Python 基本语法。
- (2) 认知 Python 变量和数据类型。
- (3) 认知 Python 运算符和表达式。
- (4) 综合应用 Python 数据类型和运算符。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的任务。
- (3) 程序设计、编辑、调试规范，代码风格良好。
- (4) 小组成员互相协作，服从小组管理。

📍 知识储备

一、基本语法

Python 的语法和其他编程语言的语法有所不同，下面对 Python 基本语法进行详细讲解。

1. Python 语句的缩进

Python 语言与 Java、C#等编程语言最大的不同点是，Python 代码块使用缩进对齐表示代码逻辑，而不是使用大括号。Python 每段代码块缩进的空格数量可以任意的，但要确保同段代码块语句必须包含相同的缩进空格数量。在 Python 中使用缩进时，不能使用“Tab”键，也不要“Tab”键和“空格”键混合。

例如：根据一个人的年龄判断其是否成年。

```
age=30
if age>=18:
    print("已经成年")
else:
    print("未成年")
```

程序执行结果：已经成年。

Python 代码中一般条件语句、循环语句、函数、类等都可以通过缩进体现代码的逻辑关系。Python 代码的缩进有严格的规定，不允许出现无意义的缩进，否则运行时会产生错误。

2. 注释

Python 程序中的注释分为单行注释和多行注释。

1) 单行注释

单行注释以“#”标识，单行注释可以单独占据一行，亦可放在一行语句之后。“#”与注释

文本之间由一个空格分隔。

例如：

```
print("hello,Python") # 输出 hello,Python
```

2) 多行注释

多行注释包含在 3 对英文半角单引号 (""") 或 3 对英文半角双引号 (""" """) 之间。程序在执行时，注释将被解释器忽略，不会执行。

二、变量和数据类型

变量是在程序执行过程中其值可以改变的量。Python 使用不同的变量名标识不同的内存区域，通过变量名访问和修改这些内存中的数据。

1. 标识符和关键字

Python 语言规定，标识符由英文字母、数字和下划线组成，其命名方式需要遵循一定的规则，具体如下：

- (1) 标识符由英文字母、数字和下划线组成，并且不能以数字开头。
- (2) 标识符区分大小写。
- (3) 标识符不能使用关键字。

在 Python 中，具有特殊功能的标识符称为关键字。关键字是 Python 语言已经使用的标识符，不允许开发者定义与关键字相同的标识符。Python 常用关键字如表 2-1 所示。

表 2-1 Python 语言中常用关键字

关 键 字	关 键 字	关 键 字	关 键 字	关 键 字
False	def	if	raise	class
None	del	import	return	form
True	elif	in	try	or
and	else	is	while	global
as	except	lambda	with	continue
assert	finally	nonlocal	yield	
break	for	not	pass	

2. 数据类型

Python 语言中，有以下几种基本数据类型。

- (1) 整数类型 (int)：包括正整数、零和负整数，如 1、0、-18。
- (2) 浮点数类型 (float)：也就是带小数位的数字，如 1.2、3.1、-0.2。
- (3) 字符串类型 (str)：指用单引号或双引号括起来的任意文本，如'boy'、'123'。
- (4) 布尔类型 (bool)：指用 True 和 False 表示逻辑真和假的一种数据类型。
- (5) 列表 (list)：由 0 个或多个不同元素组成的集合，使用方括号表示，其中的元素使用逗号分隔，如[3,'a','boy']、['张三','男',18]。
- (6) 元组 (tuple)：与列表相似，由 0 个或多个不同元素组成的集合，使用圆括号标识，如(3,'a','boy')、(张三,'男',18)。
- (7) 字典 (dict)：由 0 个或多个键值对组成的集合，使用大括号标识，其中键值对由标识数

据名称的键和数据的值组成，并用“:”分隔。例如{'中国': '北京','英国': '伦敦'}。

三、Python 语言中常见的运算符

Python 语言具有更加丰富的运算符，而且功能更强大。现介绍 Python 中常见的运算符如下。

1. 算术运算符

Python 算术运算符和相应的使用示例如表 2-2 所示。

表 2-2 Python 算术运算符和相应的使用示例

运算符	说明	表达式	当 a=5, b=2 时, 表达式的值
+	加法运算符	a+b	7
-	减法运算符	a-b	3
*	乘法运算符	a*b	10
/	除法运算符	a/b	2.5
%	取模运算符	a%b	1
//	整除运算符	a//b	2
**	幂运算符	a**2	25

2. 赋值运算符

赋值运算符的作用是将一个表达式或对象赋给一个变量。Python 赋值运算符为“=”，如 a=10，作用是将 10 赋值给变量 a。Python 中赋值运算符还可以结合算术运算符组成复合赋值运算符进行计算。

如：a=12，b=2。

a+=b 等同于 a=a+b a=14

a-=b 等同于 a=a-b a=10

a*=b 等同于 a=a*b a=24

a/=b 等同于 a=a/b a=6

a//=b 等同于 a=a//b a=6

a%=b 等同于 a=a%b a=0

a**=b 等同于 a=a**b a=144

3. 关系运算符

关系运算是两个操作数之间进行比较的运算符。关系运算符的种类、描述及示例见表 2-3。

表 2-3 关系运算符的种类、描述及示例

运算符	描述	示例
>	大于	如 a=3,b=5,a>b, 结果为 False
>=	大于等于	如 a=3,b=5,a>=b, 结果为 False
<	小于	如 a=3,b=5,a<b, 结果为 True
<=	小于等于	如 a=3,b=5,a<=b, 结果为 True
==	等于	如 a=3,b=5,a==b, 结果为 False
!=	不等于	如 a=3,b=5,a!=b, 结果为 True

用关系运算符将两个表达式(算术、关系、逻辑、赋值表达式等)连接起来所构成的表达式，

称为关系表达式。关系表达式的值为“True”或“False”。

例如：

```
>>>3>5
False
>>>3>=5
False
>>>3<5
True
>>>3<=5
True
>>>3==5
False
>>>3!=5
True
```

4. 逻辑运算符

逻辑运算符可以表示运算对象的逻辑关系，逻辑运算符的种类、描述及示例如表 2-4 所示。

表 2-4 逻辑运算符的种类、描述及示例

运 算 符	描 述	示 例
and	逻辑与	如 a=3,b=5,a and b, 结果为 True。左右两边的表达式都为 True, 结果为 True, 否则为 False
or	逻辑或	如 a=3,b=5,a or b, 结果为 True。左右两边的表达式有一个为 True, 结果为 True, 否则为 False
not	逻辑非	如 a=3,not a, 结果为 False。表达式为 True, 结果为 False。表达式为 False, 结果为 True

用逻辑运算符把两个表达式连接起来的式子称为逻辑表达式。逻辑表达式的值为“True”或“False”。

例如：

```
>>>3 and 5
True
>>>3 or 5
True
>>>not 3
False
```

5. 成员运算符

成员运算符用于判断序列中是否包含某个值，如果包含，则为“True”，否则为“False”，如表 2-5 所示。

表 2-5 成员运算符

运 算 符	描 述	示 例
in	存在	如 x 在序列 y 中, 则 x in y 值为 True
not in	不存在	如 x 不在序列 y 中, 则 x not in y 值为 True

用成员运算符把两个表达式连接起来的式子称为成员表达式。成员表达式的值为“True”或

“False”。

例如：

```
>>>1 in [1,2,3,4,5]
True
>>>5 not in [1,2,3,4,5]
False
```

四、运算符优先级和结合性

Python 在进行表达式运算的时候，可以使用多个不同的运算符，为了避免含有多个运算符的表达式出现歧义，Python 为每种运算符设定了优先级。当同一优先级的运算符一起进行运算的时候看结合性，是先从左开始运算还是从右开始运算。Python 运算符优先级和结合性一览表如表 2-6 所示。

表 2-6 Python 运算符优先级和结合性一览表

Python 运算符	运算符说明	优 先 级	结 合 性
()	小括号	1	无
x[i] 或 x[i1:i2[:i3]]	索引运算符	2	左
x.attribute	属性访问	3	左
**	乘方	4	左
~	按位取反	5	右
+ (正号)、- (负号)	符号运算符	6	右
*///%	乘除	7	左
+、-	加减	8	左
>>、<<	位移	9	左
&	按位与	10	右
^	按位异或	11	左
	按位或	12	左
==、!=、>、>=、<、<=	比较运算符	13	左
is、is not	is 运算符	14	左
in、not in	in 运算符	15	左
not	逻辑非	16	右
and	逻辑与	17	左
or	逻辑或	18	左
exp1, exp2	逗号运算符	19	左



小知识



2-2 Python 算术运算符那点事儿

🔦 操作工单

工单一 神奇计算器

1. 操作步骤

- (1) 打开 IDLE。
- (2) 输入以下代码，并在右侧的横线上写出代码注释：

```
>>> 2+3
>>> 2*3
>>> 2/3
>>> 2%3
>>> 2>3
>>> 2>=3
>>> 2<3
>>> 2<=3
>>> 2==3
>>> 2!=3
>>> 2>3 and 2>1
>>> 2>3 or 2>1
>>> not 2
```

2. 操作结果

预期结果如图 2-18 所示，请将实际结果简单绘制在右侧的方框内。

```
Python 2.7.11 Shell
Type "help()", "copyright()", "credits()" or "license()":
>>> 2+3
5
>>> 2*3
6
>>> 2/3
0.6666666666666666
>>> 2%3
2
>>> 2>3
False
>>> 2>=3
False
>>> 2<3
True
>>> 2<=3
True
>>> 2==3
False
>>> 2!=3
True
>>> 2>3 and 2>1
False
>>> 2>3 or 2>1
True
>>> not 2
False
>>>
```

图 2-18 预期结果与实际结果

3. 学习记录

工单二 种类多样的数据

1. 操作步骤

- (1) 打开 IDLE，执行“File”菜单下的“New File”命令，新建一个.py 文件。
- (2) 输入以下代码，并在右侧的横线上写出代码注释：

```

a=1
print(a)
b=1.1
print(b)
c='a'
print(c)
d='girl'
print(d)

```

- (3) 保存程序。
- (4) 调试程序，并运行程序，按“F5”键或者执行“Run”菜单中的“Module”命令运行。

2. 操作结果

预期结果如图 2-19 所示。请将实际结果简单绘制在右侧的方框内。

```

===== RESTART: C:.,
==
1
1.1
a
girl
>>>

```



图 2-19 预期结果与实际结果

3. 学习记录

工单三 酱油和醋的交换

1. 操作步骤

- (1) 打开 IDLE，执行“File”菜单下的“New File”命令，新建一个.py 文件。
- (2) 输入以下代码，并在右侧的横线上写出代码注释：

```

a='醋'
b='酱油'
print('交换前：')
print(a)
print(b)
t=a
a=b
b=t
print('交换后：')
print(a)
print(b)

```

- (3) 保存程序。
- (4) 调试程序，并运行程序，按“F5”键或者执行“Run”菜单下的“Module”命令运行。

2. 操作结果

预期结果如图 2-20 所示，请将实际结果绘制在右侧的方框内。

```

===== RESTART: C:/
==
交换前：
醋
酱油
交换后：
酱油
醋
>>>

```

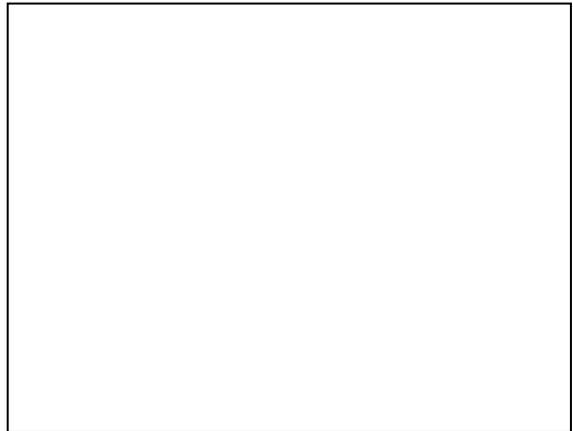


图 2-20 预期结果与实际结果

3. 学习记录

任务三

Python 之解决问题 ——过程化思维养成



任务目标

- (1) 认知结构化程序设计思维。
- (2) 认知三种数据结构。
- (3) 熟练掌握顺序结构、选择结构和循环结构三种结构原理并能运用。
- (4) 培养过程化、结构化解决问题的能力。
- (5) 培养学生思维方式、工作方法和创新能力等综合职业能力。



任务要点

- (1) 结构化思维与程序基本结构的认知。
- (2) 顺序结构的程序设计与问题解决。
- (3) 选择结构的程序设计与问题解决。
- (4) 循环结构的程序设计与问题解决。
- (5) 综合性问题的过程化程序解决。



信息素养目标

- (1) 职业素养。
 - ① 培养积极、有责任心的工作态度。
 - ② 培养结构化思维方式和工作方法。
 - ③ 培养分析问题、解决问题的能力。
 - ④ 培养小组协作互助成长的职业能力。
- (2) 技能素养。

培养规范化的程序设计、编辑和调试技能。
- (3) 创新能力。

培养运用信息技术多种算法、结构解决问题的能力。



任务内容

- 活动一：结构化思维的认知。
- 活动二：顺序结构之鸡兔同笼。
- 活动三：选择结构之 BMI 健康测试。
- 活动四：循环结构之猴子吃桃。
- 活动五：函数之八十天环游地球。

活动一 结构化思维的认知

📍 活动目标

- (1) 认知结构化思维方式和程序设计方法。
- (2) 认知程序的三种基本结构。
- (3) 认知解决问题的算法分析。
- (4) 认知流程图的功能和绘制方法。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的内容。
- (3) 小组成员互相沟通，服从小组管理。

📍 知识储备

一、结构化思维

1. 过程化程序设计方法

计算机产生初期，程序都是使用机器语言“0”和“1”二进制数字序列来编写的，这样的程序可以被机器直接识别和执行，执行速度很快，但是对于编写程序的人来说，用二进制写程序实在是太难了，程序编写效率十分低下，于是就发展出了汇编语言。汇编语言采用了助记符号来编写程序，在一定程度上简化了编程的难度，相比机器语言提升了可读性，但其本质上还是一种面向机器的语言，编写依然非常困难。为了解决“低级语言”存在的问题，计算机科学的前辈们创造了面向过程的语言，相比面向机器的低级语言来说，面向过程的语言被认为是一种“高级语言”，它关注如何一步一步地解决具体的问题，也就是解决问题的过程，这也是面向过程说法的由来，相比面向机器的思想来说，面向过程是一次程序设计思想上的飞跃。

但是在一些面向过程语言中的 `goto` 语句导致的面条式代码，极大地限制了程序的规模和质量，并引发了第一次软件危机，因为软件而导致的重大事故时有发生，例如 1963 年美国的“水手一号”火箭发射失败事故，就是因为一行 Fortran 代码错误导致的。

2. 结构化程序设计方法

第一次软件危机让过程化程序设计方法又向前改进了一步，由此产生了结构化程序设计方法。结构化程序设计本质上还是一种过程化的设计思想，它采取了“自顶向下、逐步细化”的指导思想，从问题本身开始，经过问题分析、逐步细化，将解决问题的步骤分解，使用顺序、选择、循环三种基本控制结构构造程序，降低了软件开发的复杂度，提高了程序的易读性。

1) 顺序结构

在程序设计三种基本结构里，顺序结构是最简单的一种结构，它的执行顺序是自上而下依次执行，流程如图 3-1 所示。事实上，不论程序中包含了什么样的结构，程序的总流程都是按照

自顶向下的顺序执行的。

2) 选择结构

选择结构通过判断某些特定条件是否满足来决定下一步的执行流程，是非常重要的控制结构。常见的有单分支选择结构、双分支选择结构、多分支选择结构以及嵌套的分支结构，形式比较灵活多变，具体使用哪一种取决于要实现的业务逻辑，流程如图 3-2 所示。

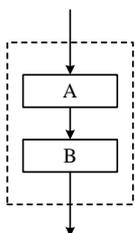


图 3-1 顺序结构

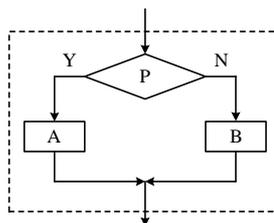
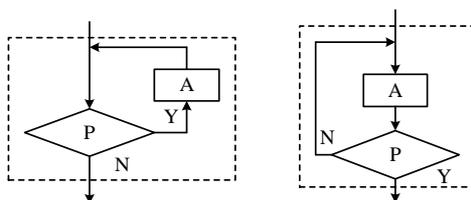


图 3-2 选择结构

3) 循环结构

循环结构是指在程序中需要反复执行某个功能而设置的一种程序结构，它由循环体中的条件判断是继续执行某个功能还是退出循环。根据判断条件，循环结构又可细分为以下两种形式：先判断后执行的当型循环结构和先执行后判断的直到型循环结构。循环结构可以减少源程序重复书写的工作量，用来描述重复执行某段算法的问题，这是程序设计中最能发挥计算机特长的程序结构，如图 3-3 所示。



当型循环结构

直到型循环结构

图 3-3 当型循环结构与直到型循环结构

程序其核心是要解决我们的实际问题，结构化的思维方式其实就是把计算机世界的程序等同于人类生活中的“程序”，例如我们生活中常说的“走流程”“按程序办”，意思就是处理一件事情按照它的流程完成所需的步骤，而程序的目的同样是为了完成一项功能、工作、任务或事务，只不过这些流程和步骤是通过程序来控制计算机代替人类来完成。所以，在结构化的程序设计和开发的过程中，需要大家首先要能分析问题，分解出完成这件事情所需的步骤，并将整个处理流程归纳成一个清晰的流程图，最后通过程序解决问题。在这样的一系列的学习和实践过程中，可以锻炼和培养大家分析问题的能力、逻辑思维能力、抽象归纳能力和解决问题的能力，这些过程化、结构化的思维能力在各个领域的问题处理上都是通用的。

二、算法

在程序世界里，算法是解题的方法和步骤，并非只有“计算”的问题才有算法，广义地说，为解决一个问题而采取的方法和步骤，都称为“算法”。所以说，算法是程序的灵魂，设计算法是程序设计的核心。解决同一个问题的算法可以有多个，如求 $1+2+3+\dots+100$ ，可以先求 $1+2$ ，再加 3、

加 4，一直加到 100，也可 $100 + (1 + 99) + (2 + 98) + \dots + (49 + 51) + 50 = 100 + 50 + 49 \times 100 = 5050$ ，还可以有其他的算法。当然，算法也有优劣之分，有的算法清晰简洁、步骤紧凑，而有些算法则步骤繁杂。一般来讲，我们希望设计出简单、运算步骤少的算法，因此，为了有效地进行解题，不仅需要保证算法正确，还要考虑算法的质量，选择合适、优良的算法。

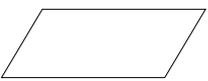
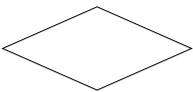
用程序解决一个问题的过程通常包含下面几步：①分析问题；②设计算法；③画流程图；④编写程序；⑤调试程序；⑥运行程序，得到结果。

三、流程图

设计好算法，我们就需要撰写解题步骤，画流程图了，用图表示的算法就是流程图。流程图是用一些图框来表示各种类型的操作，在框内写出各个步骤，然后用带箭头的线把它们连接起来，以表示执行的先后顺序。用图形表示算法，直观形象，易于理解。

美国国家标准化协会（ANSI）规定了一些常用的流程图符号，为世界各国程序工作者普遍采用。在表 3-1 中介绍了用于绘制流程图的基本图形符号及功能说明。

表 3-1 流程图的基本图形符号及功能说明

程序框	符号名称	功能说明
	终端框（起止框）	表示一个算法的开始或结束
	输入输出框	表示数据的输入或结果的输出
	处理框（执行框）	表示一个处理步骤
	判断框（选择框）	表示判断给定条件是否成立，决定如何执行其后的操作
	流程线（指向线）	表示流程的方向

程序框图表示程序内各步骤的内容以及它们的关系和执行的顺序，它说明了程序的逻辑结构。框图应该足够详细，以便可以按照它顺利地写出程序，而不必在编写时临时构思，甚至出现逻辑错误。流程图不仅可以指导编写程序，而且可以在调试程序中用来检查程序的正确性，如果框图是正确的而结果不对，则按照框图逐步检查程序是很容易发现其中错误的。流程图还能作为程序说明书的一部分提供给别人，以便帮助别人理解你编写程序的思路 and 结构。



小知识



3-1 请把大象装进冰箱里

📌 操作工单

工单一 将大象装进冰箱里

1. 提出问题

大家都听说过一个经典的脑筋急转弯问题：将大象装进冰箱里，总共分几步？很多人都知道这道脑筋急转弯题的答案：分三步，①打开冰箱门；②将大象装进冰箱；③关上冰箱门。那么这个问题放到程序中怎样来解决呢？

2. 分析问题

将大象装进冰箱里分三步：①打开冰箱门；②将大象装进冰箱；③关上冰箱门。

3. 算法描述

根据上述分析，将大象装进冰箱里，问题处理步骤如下：

- (1) 程序开始；
- (2) 开冰箱门；
- (3) 将大象装进冰箱；
- (4) 关上冰箱门；
- (5) 程序结束。

4. 画流程图



5. 操作结果

预期结果如图 3-4 所示。

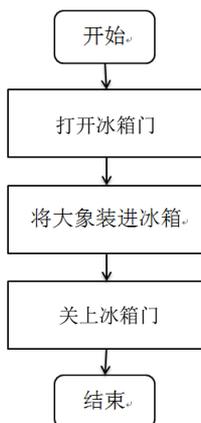


图 3-4 预期结果

6. 学习记录

工单二 判断成绩是否及格

1. 提出问题

本学期 Python 程序设计课程的成绩下发了，同学们想通过自己所学的 Python 编写一个小程序来判断自己的成绩是否及格。

2. 分析问题

成绩考到 60 分为及格，判断是否及格需要将自己的成绩和 60 分进行比较，如果大于等于 60 分则及格，否则不及格。

算法如下：

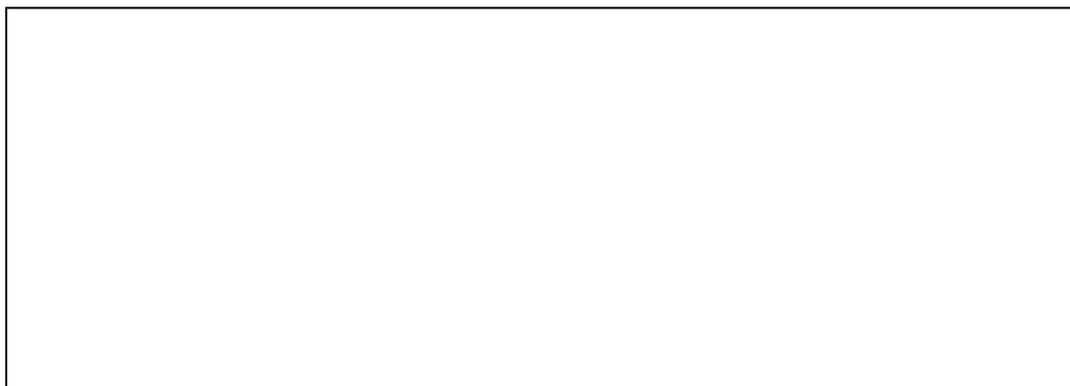
$score \geq 60$ 是否成立，如成立，则及格；如不成立，则不及格。

3. 算法描述

根据上述分析，要判断 Python 课程成绩是否合格，问题处理步骤如下：

- (1) 程序开始；
- (2) 输入成绩 $score$ ；
- (3) 选择判断 $score \geq 60?$ ；
- (4) 如果 $score \geq 60$ 成立，输出“及格”；
- (5) 如果 $score \geq 60$ 不成立，输出“不及格”；
- (6) 程序结束。

4. 画流程图



5. 操作结果

预期结果如图 3-5 所示。

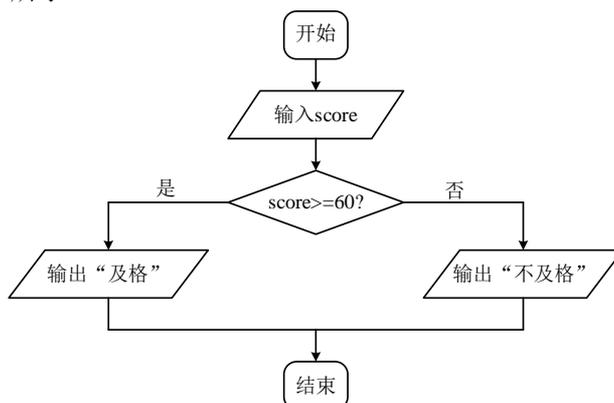


图 3-5 预期结果

6. 学习记录

工单三 1+2+3+...+100=?

1. 提出问题

小学数学竞赛中经常会出现求“ $1+2+3+\dots+100=?$ ”的问题，这道题难倒了很多小学生，那么怎样使用程序来求这题的答案呢？我们先来绘制流程图。

2. 分析问题

本题涉及累积求和操作，将 1~100 进行累积求和，需要用到循环结构进行程序设计。

用 sum 来存放部分和，i 作为被加数，并用循环条件判断；先将 sum 和 i 初值分别设置为 0 和 1。再设置循环条件 $i \leq 100$ 是否成立，如果成立将 $sum+i$ 赋值给 sum， $i=i+1$ ，并重复进行条件判断这一步；如果不成立这时 sum 的值就是 1~100 的值的和，输出 sum。

3. 算法描述

根据上述分析，要计算两数求和，问题处理步骤如下：

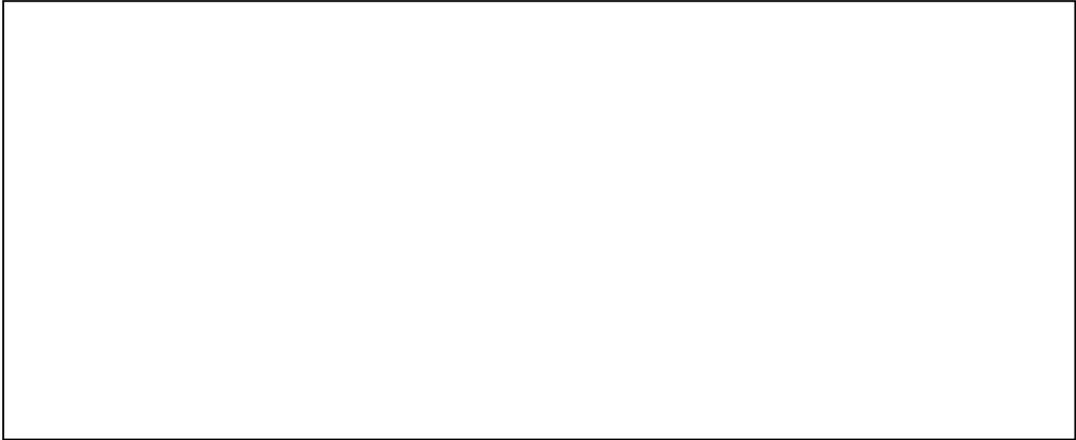
(1) 程序开始；

(2) 设置 $sum=0$ ， $i=1$ ；

(3) 判断循环条件 $i \leq 100$ 是否成立，如果成立执行 $sum=sum+i$ ， $i=i+1$ ，并继续判断循环条件是否成立，如果不成立则跳出循环；

- (4) 输出 sum 的值;
- (5) 程序结束。

4. 画流程图



5. 操作结果

预期结果如图 3-6 所示。

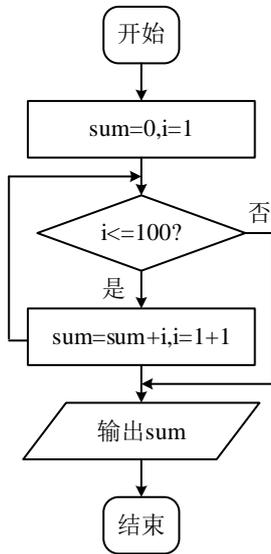


图 3-6 预期结果

6. 学习记录

工单四 独立作业

1. 提出问题

根据输入的数据，判断该数是否是偶数。

2. 分析问题

3. 算法描述

- (1) _____
- (2) _____
- (3) _____
- (4) _____
- (5) _____
- (6) _____

4. 画流程图

5. 学习记录

活动二 顺序结构之鸡兔同笼

活动目标

- (1) 认知顺序结构。
- (2) 认知顺序结构程序设计。

活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的内容。
- (3) 小组成员互相沟通，服从小组管理。

知识储备

一、数据的输入/输出

Python 中的输入/输出操作是通过几个函数来实现，如 `input()` 函数用来实现数据的输入操作，`print()` 函数用来实现数据的输出。

二、数据的输入

Python 中数据的输入操作，常使用 `input()` 函数。`input()` 函数是 Python 标准函数库中的函数，其功能是获取用户从控制台输入的内容，并以字符串形式将其返回。

格式：变量=`input` (“提示信息”)。

如：

```
number=input("请输入一个数字")          # 输入一个数字，将数值赋值给变量 number
```

三、数据的输出

Python 中数据的输入操作，常使用 `print()` 函数。`print()` 函数是 Python 程序中最基本的函数，它的作用是将信息输出到控制台，即在控制台窗口中打印信息。

字符串格式化符号如表 3-2 所示。

表 3-2 字符串格式化符号

符 号	描 述	符 号	描 述
<code>%c</code>	格式化字符及其 ASCII 码	<code>%X</code>	格式化无符号十六进制数（大写）
<code>%s</code>	格式化字符串	<code>%f</code>	格式化浮点数字，可指定小数点后的精度
<code>%d</code>	格式化整数	<code>%e</code>	用科学计数法格式化浮点数
<code>%u</code>	格式化无符号整型	<code>%E</code>	作用同 <code>%e</code> ，用科学计数法格式化浮点数
<code>%o</code>	格式化无符号八进制数	<code>%g</code>	<code>%f</code> 和 <code>%e</code> 的简写
<code>%x</code>	格式化无符号十六进制数	<code>%G</code>	<code>%f</code> 和 <code>%E</code> 的简写

字符串格式化操作符辅助指令如表 3-3 所示。

表 3-3 字符串格式化操作符辅助指令

符 号	功 能	符 号	功 能
*	定义宽度或者小数点精度	0	显示的数字前面填充'0'而不是默认的空格
-	用作左对齐	%	'%%'输出一个单一的'%'
+	在正数前面显示加号(+)	(var)	映射变量(字典参数)
<sp>	在正数前面显示空格	m.n.	m 是显示的最小总宽度, n 是小数点后的位数(如果可用的话)
#	在八进制数前面显示零('0'), 在十六进制前面显示'0x'或者'0X'(取决于用的是'x'还是'X')	—	—

1. 输出字符串和数字

例如:

```
>>> print(560) # 输出数字
```

输出结果: 560。

```
>>> print("root") # 输出字符串
```

输出结果: root。

```
>>> str = 'hello'
>>> print(str) # 输出变量
```

输出结果: hello。

```
>>> d = {'c':3, 'd':5} # 字典
>>> print(d)
```

输出结果: {'c': 3, 'd': 5}。

```
>>> L = [7, 'c'] # 列表
>>> print(L)
```

输出结果: [7, 'c']。

```
>>> t = (3,4,'a','b','c') # 元组
>>> print(t)
```

输出结果: (3,4,'a','b','c')。

2. 格式化输出整数

```
>>> str = "the char: (%s) is %d" %('hello',len('hello'))
>>> print(str)
```

输出结果: the char: (hello) is 5。

```
>>> ch1= 0xFF
>>> print("ch1 = %x,ch1 = %d,ch1 = %o" %(ch1,ch1,ch1))
```

输出结果: ch1= ff,ch1= 255,ch1= 377。

其中:

```
# %x --- hex 十六进制
# %d --- dec 十进制
# %o --- oct 八进制
```

3. 格式化输出浮点数(float)

```
>>> a =5.1234567
>>> print('%10.3f' % a)           # 字段宽 10, 精度
```

输出结果: 5.123。

```
>>> print("a = %.*f" % (3,a))    # 用*从后面的元组中读取字段宽度或精度
```

输出结果: a= 5.123。

```
>>> print('%010.3f' %a)          # 用 0 填充空白
```

输出结果: 000005.123。

```
>>> print('%-10.3f' % a)         # 左对齐
```

输出结果: 5.123。

```
>>> print('%+f' % a)             # 显示正负号
```

输出结果: +5.123456。

4. 自动换行

`print()`函数会自动在行末加上换行符, 如果不需要换行, 只需在 `print()`函数的参数后添加一个逗号“,”, 并设置分隔符参数 `end`, 就可以改变它的行为。

```
>>> for i in range(11,16):
...     print(i)
...
10
11
12
13
14
15
>>> for i in range(11,16):
...     print(i, end=" ")
...

```

输出结果: 10 11 12 13 14 15。

5. print 不换行

在 Python 中 `print` 默认是换行的:

```
>>> for i in range(21,24):
...     print (i)
...
21
22
```

```
23
>>>
    要想不换行你应该写成 print(i, end = ' ')
>>>for i in range(21,24):
... print(i, end = ' ')
...
212223
```



小知识



3-2 Python 格式化输出

📌 操作工单

工单一 鸡兔同笼

1. 提出问题

张老师给班里的同学出了一道古典数学题——鸡兔同笼问题，让同学们来进行解答，题目是这样的鸡兔在一个笼子里，笼子里共有头 30 个，共有脚 90 只，请问笼子里的鸡和兔各有几只？

2. 分析问题

每只鸡和兔的脚分别是 2 只和 4 只，如果头 30 个，脚 90 只，假设 head 表示头的总个数，foot 代表脚的总个数，x 代表鸡，y 代表兔。则有公式如下：

```
y=foot/2-head
x=30-y
```

3. 算法描述

根据上述分析，要计算笼子里的鸡和兔各有几只，问题处理步骤如下：

- (1) 程序开始；
- (2) foot 和 head 分别赋初始值为：foot=90，head=30；
- (3) 计算 $y=foot/2-head$ ， $x=30-y$ ；
- (4) 输出 x 和 y 的值；
- (5) 程序结束。

4. 编程解题

```
head=30
foot=90
y=foot/2-head
x=head-y
print("鸡的个数为:%d"%int(x))
print("兔的个数为:%d"%int(y))
```

5. 学习记录

工单二 求解平均成绩

1. 提出问题

已知小明同学的语文、数学、英语三科成绩，求解平均成绩是多少？

2. 分析问题

根据数学知识，已知三科成绩，可以由总成绩除以门数求得平均成绩，公式如下：

```
sum=a+b+c  
aver=sum/3
```

3. 算法描述

根据上述分析，要计算平均成绩需要先输入三科成绩，然后由总成绩除以门数计算平均成绩，问题处理步骤如下：

- (1) 程序开始；
- (2) 输入三科成绩 a, b, c 的值；
- (3) 计算三科总成绩 $sum=a+b+c$ ；
- (4) 计算三科平均成绩 $aver = sum/3$ ；
- (5) 输出结果 $aver$ ；
- (6) 程序结束。

4. 编程解题

```
a=float(input("输入语文成绩: "))  
b=float(input("输入数学成绩: "))  
c=float(input("输入英语: "))  
sum=a+b+c  
aver= sum/3  
print("三科平均成绩是",aver)
```

5. 学习记录

工单三 每天 0.01 的变化

1. 提出问题

最近网上有两个公式又突然火了起来，那就是 1 的 365 次方是 1，而 1.01 的 365 次方约等于 37.8，0.99 的 365 次方约等于 0.03。这个公式代表着我们每天的努力程度，也可以是看书、学习等的努力程度。用看书来说，每天看一页，一年看 365 页，每天多看百分之一，那一年就多看 37 页。你一年就可以看 402 页，你看是不是多学习了 37 页知识呢？

2. 分析问题

验证“每天 0.01 的变化”的公式如下所示：

$$(1+0.01)^{365} = 37.78$$

$$(1-0.01)^{365} = 0.025$$

3. 算法描述

根据上述分析，要验证“每天 0.01 的变化”问题处理步骤如下：

- (1) 程序开始；
- (2) 计算每天努力 0.01 的值；
- (3) 计算每天懈怠 0.01 的值；
- (4) 输出结果；
- (5) 程序结束。

4. 编程解题

```
work=(1+0.01)**365
lazy=(1-0.01)**365
print('每天努力一点点，365 天后你将收获',work)
print('每天偷懒一点点，365 天后你将收获',lazy)
```

5. 学习记录

工单四 高烧 100 度

1. 提出问题

小华同学发烧了，测量体温的数值显示竟然是 100 度，高烧 100 度让小华吓了一跳，原来电子体温计被设置成了华氏温度，小华的体温究竟是多少摄氏度？

2. 分析问题

目前，世界上包括我国在内的绝大多数国家使用摄氏温度 (c)，只有美国等几个国家使用华

氏温度 (f), 华氏温度 f 和摄氏温度 c 之间转化的公式为 $c = (f - 32) / 1.8$ 。

3. 算法描述

根据上述分析, 要计算小华的提问到底是多少摄氏度, 需要将华氏温度转化为摄氏温度, 问题处理步骤如下:

- (1) 程序开始;
- (2) 输入一个华氏温度 f ;
- (3) 计算摄氏温度 $c = (f - 32) / 1.8$;
- (4) 输出摄氏温度结果 c ;
- (5) 程序结束。

4. 编程解题

```
f=input("请输入华氏温度: ")
f=int(f)
c=(f-32)/1.8
c=round(c,1)
print("摄氏温度为",c)
```

5. 学习记录

工单五 提取位数

1. 提出问题

张明的爸爸问他如果要输出随机录入到计算机中的百位数整数的每个数位上的数是多少, 怎么编写程序实现?

2. 分析问题

求三位数的每个位数, 先假设 n 表示任意一个三位数整数, i 表示个位数, j 表示十位数, k 表示百位数。那么求这三个数字的算法公式如下:

```
i=n%10
j=n/10%10
k=n/100
```

3. 算法描述

根据上述分析, 要提取百位数每位上的数字, 问题处理步骤如下:

- (1) 程序开始;
- (2) 输入 n 的值;
- (3) 计算 i, j, k ;
- (4) 输出 i, j, k 的值;

(5) 程序结束。

4. 编程解题

```
n=int(input('输入一个任意三位数整数: '))
i= int(n%10 )
j= int(n/10%10)
k= int(n/100)
print('个位数:',i)
print('十位数:',j)
print('百位数:',k)
```

5. 学习记录

工单六 独立作业

1. 提出问题

已知圆的半径为 r ，求解圆的周长和面积是多少？

2. 分析问题

3. 算法描述

- (1) _____
- (2) _____
- (3) _____
- (4) _____
- (5) _____
- (6) _____

4. 画流程图

5. 编程解题

6. 执行结果

活动三 选择结构之 BMI 健康测试

📍 活动目标

- (1) 认知选择结构。
- (2) 认知选择结构程序设计。
- (3) 认知 if 语句及其三种形式。
- (4) 认知 if 语句的嵌套。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的内容。
- (3) 小组成员互相沟通，服从小组管理。

📍 知识储备

一、选择结构

在程序设计中，并不是简单地、按顺序地完成人们事先安排好的一些指令，经常需要进行逻辑判断，即给出一个条件，让计算机判断是否满足该条件，并按不同的判断结果执行不同的语句。如比较两个数的大小，输出最大数；判断任意输入的整数是奇数还是偶数等。这些问题在处理的时候都需要使用选择结构来进行程序设计。

二、if 语句

在 Python 中使用 if 语句来实现选择结构程序设计。if 语句包括 if 语句、if-else 语句和 if-elif-else 语句三种形式。

1. 单分支结构 if 语句

Python 中单分支 if 语句的流程图如图 3-7 所示。

格式如下：

```
if <判断条件>:
    <代码段>
```

功能：当判断条件成立时，执行后面的代码段，否则跳过 if 语句，执行后面的代码段。

例如：判断输入的数字是否是 1~7 之间的数。

```
num=int(input("请输入任意一个整数："))
if num in [1,2,3,4,5,6,7]:
    print("是1~7之间的数字")
```

程序执行结果如下：

```
请输入任意一个整数：3
是1~7之间的数字
```

2. 双分支结构 if-else 语句

Python 中双分支 if 语句流程图如图 3-8 所示。

格式如下：

```
if <判断条件>:
    <代码段 1>
else:
    <代码段 2>
```

功能：当判断条件成立时，执行后面的代码段 1，否则执行 else 后面的代码段 2。

例如：输出任意输入的两个数字中最大的数。

```
num1=input("请输入任意第一个数：")
num2=input("请输入任意第二个数：")
if num1>=num2:
    print("最大的数是 num1")
else:
    print("最大的数是 num2")
```

程序执行结果如下：

```
请输入任意第一个数：2
请输入任意第二个数：5
最大的数是 num2
```

3. 多分支结构 if-elif-else 语句

Python 中多分支 if 语句流程图如图 3-9 所示。

格式如下：

```
if <判断条件 1>:
    <代码段 1>
elif <判断条件 2>:
    <代码段 2>
.....
elif <判断条件 n >:
    <代码段 n >
```

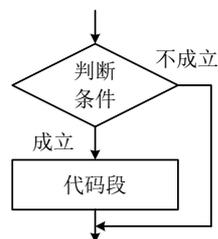


图 3-7 单分支 if 语句流程图

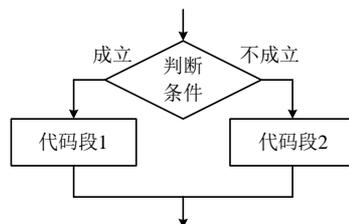


图 3-8 双分支 if 语句流程图

```
else:
    <代码段 n+1>
```

功能：先判断条件 1 是否成立，如果成立则执行代码段 1，并跳出 if 语句，否则判断条件 2 是否成立，如果成立则执行代码段 2，跳出 if 语句，如不成立则继续向下执行，如果所有条件都不成立，则执行 else 语句之后的代码段。

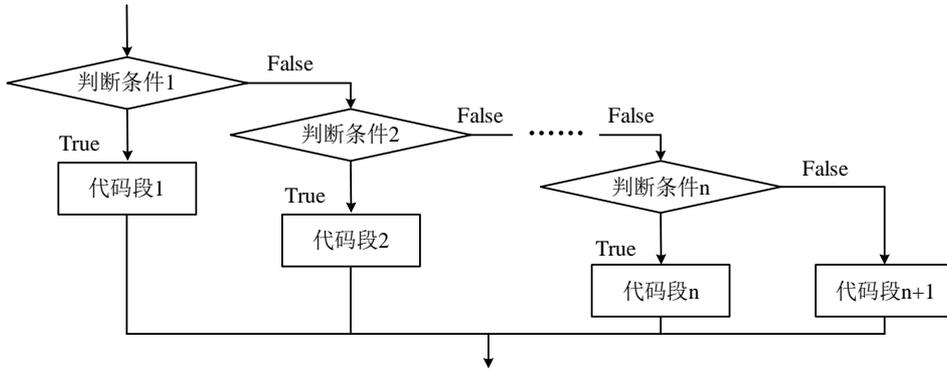


图 3-9 多分支 if 语句流程图

例如：根据输入的百分制成绩输出成绩等级。

```
score=float(input("请输入学生成绩: "))
if score>=90:
    print("A")
elif score>=80:
    print("B")
elif score>=70:
    print("C")
elif score>=60:
    print("D")
else:
    print("E")
```

程序执行结果如下：

```
请输入学生成绩: 56
E
```

三、if 语句的嵌套

if 语句的嵌套是指在 if 语句中嵌套另外一个或多个完整的 if 语句。if 语句的嵌套常用来解决多分支选择问题。

格式 1：在 if 中嵌套。

```
if <判断条件 1>:
    if <判断条件 2>:
        <代码段 1>
    else:
        <代码段 2>
else:
    <代码段 3>
```

功能：判断条件 1 是否成立，如果成立则继续判断条件 2 是否成立，如果成立则执行代码段 2，判断条件 2 不成立则执行代码段 3，如果判断条件 1 不成立，则执行最后的 else 后面的代码段 1。

格式 2：在 else 中嵌套。

```
if <判断条件 1>:
    <代码段 1>
else:
    if <判断条件 2>:
        <代码段 2>
    else:
        <代码段 3>
```

功能：判断条件 1 是否成立，如果成立则执行代码段 1；如果不成立则继续判断 else 里面的判断条件 2 是否成立，如果成立则执行代码段 2，如果不成立则执行代码段 3。

例如：从键盘中输入一个自变量 x ，求分段函数 $y=f(x)$ 的值， $f(x)$ 的表达式如下：

$$y = f(x) = \begin{cases} x + 5, & x > 0 \\ 0, & x = 0 \\ x + 30, & x < 0 \end{cases}$$

```
x=int(input("请输入 x: "))
if x>=0:
    if x>0:
        y=x+5
    else:
        y=0
else:
    y=x+30
print("y=%d"%y)
```

程序执行结果如下：

```
请输入 x:0
y=0
```



小知识



3-3 判断是否酒后驾车



操作工单

工单一 BMI 健康测试

1. 提出问题

身体质量指数是（简称体质指数，BMI）是目前国际上常用的衡量人体胖瘦程度以及是否健

康的一个标准。计算公式为： $BMI = \text{体重} \div \text{身高}^2$ 。一般情况下 $BMI < 18.5$ 为偏瘦； $18.5 \leq BMI < 24$ 为正常， $24 \leq BMI < 28$ 为偏胖， $BMI \geq 28$ 为肥胖。编写程序计算 BMI 的值并判断胖瘦。

2. 分析问题

假设一个人体重为 k ，身高为 h ，则计算 BMI 公式为： $BMI = k/h^2$ ，然后使用多分支选择结构进行程序设计，根据 $BMI < 18.5$ 为偏瘦， $18.5 \leq BMI < 24$ 为正常， $24 \leq BMI < 28$ 为偏胖， $BMI \geq 28$ 为肥胖，将 BMI 的值和胖瘦判断条件进行对比，并输出对应的情况。

3. 算法描述

根据上述分析要实现根据输入的身高和体重算出 BMI 指数，通过 BMI 指数来进行身体健康测试，问题处理步骤如下：

- ① 程序开始；
- ② 输入体重 k 和身高 h ；
- ③ 计算 BMI， $BMI = k/(h \times h)$ ；
- ④ 判断 BMI 的值，输出对应结果；
- ⑤ 程序结束。

4. 编程解题

输入以下代码，并在右侧的横线上写出代码注释：

```

k=float(input("请输入您的体重(kg): "))
h=float(input("请输入您的身高(m): "))
bmi= k/(h**2)
bmi=float(bmi)
print(bmi)
if bmi <18.5:
    print("您的体重偏瘦, 请多加强营养呀!")
elif bmi>=18 and bmi<24:
    print("您的体重正常, 请继续保持!")
elif bmi>=24 and bmi<28:
    print("您的体重偏胖, 需要控制了!")
else:
    print("您的体重肥胖, 要加强锻炼了!")
    
```

5. 学习记录

工单二 排排座

1. 提出问题

小明的老师要求小明编写一个程序来实现对任意输入的三个整数由小到大的输出。

2. 分析问题

对任意输入的三个整数怎样由小到大的输出，最简便的办法就是按顺序进行输出，假设这三个整数分别为 a, b, c 。那么 a 存放最小的数， b 存放第二小的数， c 存放最大的数。然后按顺序进行输出。为了让 a 存放最小的数，就需要进行变量间的数据交换，这时我们需要借助一个中间变量 t 。具体的算法如下：

```
t=a, a=b, b=t
```

3. 算法描述

根据上述分析，要实现任意输入的三个整数由小到大的输出，问题处理步骤如下：

- (1) 程序开始；
- (2) 输入 3 个数 a, b, c ；
- (3) 判断 $a > b$ ，如果成立，那么 a 和 b 进行数据交换，具体代码： $t=a \ a=b \ b=t$ ；
- (4) 判断 $a > c$ ，如果成立，那么 a 和 c 进行数据交换，具体代码： $t=a \ a=c \ c=t$ ；
- (5) 判断 $b > c$ ，如果成立，那么 b 和 c 进行数据交换，具体代码： $t=b \ b=c \ c=t$ ；
- (6) 输出 a, b, c 的值；
- (7) 程序结束。

4. 编程解题

输入以下代码，并在右侧的横线上写出代码注释：

```
a=int(input("输入整数 a:"))
b=int(input("输入整数 b:"))
c=int(input("输入整数 c:"))
if a>b:
    t=a
    a=b
    b=t
if a>c:
    t=a
    a=c
    c=t
if b>c:
    t=b
    b=c
    c=t
print("由小到大为: ", a, b, c)
```

5. 学习记录

工单三 十二生肖

1. 提出问题

小明想快速、准确地知道每个人的生肖，他想到可以通过 Python 编写一个简单的十二生肖小程序来实现。

2. 分析问题

十二生肖分别为申猴、酉鸡、戌狗、亥猪、子鼠、丑牛、寅虎、卯兔、辰龙、巳蛇、午马、未羊。判断生肖的公式为 $year \% 12$ 。可以根据该公式求余的结果输出对应的生肖。

3. 算法描述

根据上述分析，要实现十二生肖小程序，问题处理步骤如下：

- (1) 程序开始；
- (2) 输入年份；
- (3) 计算 $year \% 12$ ；
- (4) 对得到的余数进行判断；
- (5) 输出判断结果；
- (6) 程序结束。

4. 编程解题

输入以下代码，并在右侧的横线上写出代码注释：

```

year=int(input('请输入年份：'))
shengxiao=year % 12
if shengxiao == 0:
    print('申猴')
elif shengxiao == 1:
    print('酉鸡')
elif shengxiao == 2:
    print('戌狗')
elif shengxiao == 3:
    print('亥猪')
elif shengxiao == 4:
    print('子鼠')
elif shengxiao == 5:
    print('丑牛')
elif shengxiao == 6:
    print('寅虎')
elif shengxiao == 7:
    print('卯兔')
elif shengxiao == 8:
    print('辰龙')
elif shengxiao == 9:
    print('巳蛇')
elif shengxiao == 10:

```

```

    print('午马')
elif shengxiao == 11:
    print('未羊')
else:
    print('计算错误, 年份不对')

```

5. 学习记录

工单四 猜拳游戏

1. 提出问题

猜拳游戏是大家经常玩的一种简单的手势游戏，它的手势有剪刀、石头和布三种，编写一个程序实现使用计算机进行猜拳游戏。

2. 分析问题

假设使用 0 代表剪刀, 1 代表石头, 2 代表布, 用户随机录入对应的数字和计算机进行比赛, 计算机这是随机输出 0~2 之间的数字也代表对应的手势, 这时进行比较, 并输出比赛结果。

3. 算法描述

根据上述分析, 要实现人机猜拳的程序设计, 问题处理步骤如下:

- (1) 程序开始。
- (2) 输入 0, 1, 2 中的一个数。
- (3) 将输入的数字和计算机随机输出的数字进行比较。
- (4) 输出比较的结果。
- (5) 程序结束。

4. 编程解题

输入以下代码, 并在右侧的横线上写出代码注释:

```

import random
user=int(input("请输入 (0 剪刀、1 石头、2 布: )"))

siri=random.randint(0,2)
if(user==0 and siri==2) or (user==1 and siri==0) or(user==2 and siri==1):

    print("电脑出的是%s,你赢了! "%siri)

elif (user==0 and siri==0) or (user==1 and siri==1) or(user==2 and siri==2):

```

```
print("电脑出的是%s,打成平手! "%siri)

else :
    print("电脑出的是%s,你输了,继续努力呀! "%siri)
```

5. 学习记录

工单五 独立作业

1. 提出问题

2000 多年前，古希腊天文学家希巴克斯为标示太阳在黄道上观行的位置，就将黄道带分成 12 个区段，以春分点为 0°，自春分点（即黄道零度）算起，每隔 30°为一宫。并以各宫内所包含的主要星座来命名，依次为白羊、金牛、双子、巨蟹、狮子、室女、天秤、天蝎、人马、摩羯、宝瓶、双鱼等宫，称之为黄道十二宫，这就是现代十二星座的来源。现代根据一年中地球运转黄道的位置将日期和对应的星座联系起来（见表 3-4），于是有了十二星座。你知道你是什么星座么？请编写程序试着找出自己的星座并输出。

表 3-4 十二星座与日期对照表

星 座	日 期	星 座	日 期
白羊座	3 月 21 日—4 月 19 日	天秤座	9 月 23 日—10 月 23 日
金牛座	4 月 20 日—5 月 20 日	天蝎座	10 月 24 日—11 月 21 日
双子座	5 月 21 日—6 月 21 日	射手座	11 月 22 日—12 月 21 日
巨蟹座	6 月 22 日—7 月 22 日	摩羯座	12 月 22 日—1 月 19 日
狮子座	7 月 23 日—8 月 22 日	水瓶座	1 月 20 日—2 月 18 日
处女座	8 月 23 日—9 月 22 日	双鱼座	2 月 19 日—3 月 20 日

2. 分析问题

3. 算法描述

- (1) _____
- (2) _____
- (3) _____

(4) _____

(5) _____

(6) _____

4. 画流程图

5. 编程解题

6. 执行结果

活动四 循环结构之猴子吃桃

活动目标

- (1) 认知循环结构。
- (2) 认知循环结构程序设计。
- (3) 认知 while 语句和 for 语句。
- (4) 认知循环的嵌套。
- (5) 认知 continue 语句和 break 语句。

活动要求

- (1) 学习态度积极，按时出勤。

(2) 小组成员互相沟通，服从小组管理。

知识储备

一、循环结构

在程序设计中，使用选择结构解决逻辑判断的问题，但在实际问题中往往有很多重复执行的操作，仅仅使用选择结构不容易解决，这时可以使用循环结构来完成这样的重复性操作。

如累计求和的问题就需要使用重复性的操作来完成，这时候使用循环结构就可以减少冗余的代码。在 Python 中用来实现循环结构程序设计的控制语句有 while 语句和 for 语句两种。

二、while 语句

while 语句的格式如下：

```
while <循环条件>:  
    <代码段>
```

功能：当程序执行到 while 语句时，首先判断循环条件的真假。若循环条件的值为真，则执行代码段，之后返回表达式继续判断；若循环条件的值为假，则跳过代码段来执行。

例如：1~100 累计求和。

```
n= 100  
sum = 0  
i = 1  
while i <= n:  
    sum = sum + i  
    i+= 1  
print("1 到 %d 之和为: %d" % (n,sum))
```

执行结果：

1 到 100 之和为：5050

三、for 语句

Python 中的 for 循环，常用于遍历字符串、列表、元组、字典、集合等序列类型，逐个获取序列中的各个元素。

for 语句的格式如下：

```
for <迭代变量> in <字符串|列表|元组|字典|集合>:  
    <循环语句>
```

功能：判断 in 后面的字符串|列表|元组|字典|集合里面是否有迭代变量，如果有则执行循环语句，然后继续判断是否有迭代变量，如果没有则退出 for 循环，执行后续语句。

1. for 循环进行数值循环

例如：1~100 累计求和。

```
print("计算 1+2+...+100 的结果为: ")  
result = 0  
for i in range(101):  
    result += i
```

```
print(result)
```

执行结果:

计算 $1+2+\dots+100$ 的结果为: 5050

2. for 循环遍历列表和元组

例如: for 循环对列表进行遍历。

```
the_count=[1,2,3,4,5]
for number in the_count:
    print(f"This is count {number}")
```

执行结果:

```
This is count 1
This is count 2
This is count 3
This is count 4
This is count 5
```

3. 遍历字符串

例如: 遍历字符串并输出。

```
list_python='PYTHON'
for j in list_python:
    print(f"{j}")
```

执行结果:

```
P
Y
T
H
O
N
```

四、循环结构中 else 的用法

Python 中, 无论是 while 循环还是 for 循环, 其后都可以紧跟着一个 else 代码块, 它的作用是在当循环条件为 False 跳出循环时, 程序会最先执行 else 代码块中的代码。

1. while 循环中 else 的用法

格式如下:

```
while <循环条件>:
    <代码段 1>
else:
    <代码段 2>
```

功能: 当 while 循环中的循环条件为 False 时, 跳出循环结构, 执行 else 后面的代码段 2。

例如: 计算 $5!$, 计算完毕后输出提示语句。

```
fact=1
i=1
```

```
while i<=5:
    fact=fact*i
    i=i+1
else:
    print("计算完毕, 循环结束!")
print("fact=",fact)
```

执行结果:

```
计算完毕, 循环结束!
fact=120
```

2. for 循环中 else 的用法

格式如下:

```
for <循环变量> in <遍历结构>
    <代码段 1>
else:
    <代码段 2>
```

功能: 当 for 循环中的循环条件为 False 时, 跳出循环结构, 执行 else 后面的代码段 2。

例如: 1~100 累计求和, 计算完毕后输出提示语句。

```
print("计算 1+2+...+100 的结果为: ")
result = 0
for i in range(101):
    result += i
else:
    print("计算完毕, 循环结束!")
print(result)
```

执行结果:

```
计算完毕, 循环结束!
5050
```

五、循环的嵌套

Python 中在一个循环体中嵌套一个完整的循环结构, 称之为循环的嵌套。while 语句和 for 语句都可以进行循环嵌套, 具体格式有如下几种。

格式 1:

```
while <循环条件 1>:
    while <循环条件 2>:
        <代码段 2>
    <代码段 1>
```

功能: 当循环条件 1 成立时候, 判断循环条件 2 是否成立, 如果成立则执行代码段 1, 直到循环条件 2 不成立, 跳出内循环, 后面执行代码段 2, 接着继续判断循环条件 2 是否成立, 如果成立则继续判断循环条件 2 是否成立。这样循环执行, 直到循环条件 1 不成立, 结束循环。

例如: 找出 2~20 中的素数。

```
i = 2
s=0
```

```
print ("2~20 内素数是: ")
while(i < 20):
    j = 2
    while(j <= (i/j)):
        if not(i%j): break
        j = j + 1
    if (j > i/j) : print (i," 是素数")
    i = i + 1
```

执行结果:

2~20 内素数是:

```
2 是素数
3 是素数
5 是素数
7 是素数
11 是素数
13 是素数
17 是素数
19 是素数
```

格式 2:

```
for<循环变量> in <遍历结构>
    for<循环变量> in <遍历结构>
        <代码段 2>
    <代码段 1>
```

功能: 当 for 循环变量在变量结构里面, 这样执行内嵌的 for 循环, 直到内嵌 for 循环结束。

当循环条件 1 成立的时候, 执行内嵌的 for 循环语句, 当内嵌 for 循环语句执行完毕后, 执行代码段 1。接着继续判断 for 循环是否成立, 如果成立则继续执行内嵌的 for 循环, 当内嵌 for 循环结束后, 继续执行代码段 1。这样循环执行, 直到循环条件 1 不成立, 结束循环。

六、continue 语句和 break 语句

在 Python 中使用 continue 语句和 break 语句来跳出循环结构。continue 语句跳出本次循环, 而 break 跳出整个循环。它们都用在 while 和 for 循环中。

1. continue 语句

格式如下:

```
continue
```

功能: 跳出当前循环的剩余语句, 然后继续进行下一轮循环。

例如: 求 100 以内所有偶数之和。

```
sum=0
i=0
while i<=100:
    i=i+1
    if i%2==0:
        sum=sum+i
    else:
        continue
```

```
print('100 以内所有偶数之和为 :', sum)
```

执行结果:

```
2550
```

2. break 语句

格式如下:

```
break
```

功能: **break** 语句用来终止循环语句, 即循环条件没有 **False** 条件或者序列还没被完全递归完, 也会停止执行循环语句。

例如: 输出字符串中设置好的字母位置。

```
for letter in 'hello':  
    if letter == 'l':  
        break  
    print('当前字母:', letter)
```

执行结果:

```
h  
e
```



小知识



3-4 从前有座山, 山里有座庙

💡 操作工单

工单一 猴子吃桃

1. 提出问题

猴子吃桃问题: 猴子第一天摘下若干个桃子, 当即吃了一半, 还不过瘾, 又多吃了一个。第二天早上又将第一天剩下的桃子吃掉一半, 又多吃了一个。以后每天早上都吃了前一天剩下的一半加一个。到第 10 天早上想再吃时, 发现只剩下一个桃子了。编写程序求猴子第一天摘了多少个桃子。

2. 分析问题

解决这个问题, 先定义 **day**、**peach** 为基本整型, 并为 **day** 和 **peach** 赋初值 9 和 1。使用 **while** 语句由后向前推出第一天摘的桃子数。具体公式为 $peach = (1 + peach) * 2$ 。

3. 算法描述

根据上述分析, 要计算猴子第一天摘了多少个桃子, 问题处理步骤如下:

- (1) 程序开始;
- (2) day 和 peach 赋初值 9 和 1;
- (3) 设置循环条件 day>0, 循环体语句 peach = (1 + peach) * 2;
- (4) 判断循环条件是否成立, 如果成立则执行循环体语句, 如果不成立则跳出循环;
- (5) 输出 peach 的值;
- (6) 程序结束。

4. 编程解题

输入以下代码, 并在右侧的横线上写出代码注释:

```
peach = 1 _____  
day= 9 _____  
while day > 0: _____  
    peach = (1 +peach) * 2 _____  
    day -= 1 _____  
print("桃子个数为: %d" % peach ) _____
```

5. 学习记录

工单二 数麦芒

1. 提出问题

如果要数清楚地里的麦芒有多少, 能不能设计一个小程序来实现呢?

2. 分析问题

数麦芒是一个累计求和的问题, 在这里面要使用循环结构来实现累计求和, 循环代码如下:

```
sum=sum+1。
```

3. 算法描述

根据上述分析, 要数清楚麦芒有多少个, 问题处理步骤如下:

- (1) 程序开始;
- (2) 累计求和;
- (3) 输出计数的结果;
- (4) 程序结束。

4. 编程解题

输入以下代码, 并在右侧的横线上写出代码注释:

```

sum=0
for i in range(1,101):
    sum=sum+i
    i+=1
print("麦芒数为: %d"%sum)

```

5. 学习记录

工单三 水仙花数

1. 提出问题

水仙花数 (narcissistic number) 是指一个三位数，它的各个位数上的数字的立方和等于该数字本身，例如： $1^3 + 5^3 + 3^3 = 153$ 。

那么为什么这样的数字叫水仙花数呢？这其中还有个小故事，据说它的故事来源于古希腊神话中的美少年那喀索斯，他在水塘边被自己水中的美丽倒影吸引，久久不愿离开，最后抑郁而死，化作一朵水仙花，水仙花的英文是 narcissus，narcissistic 的意思是“自我陶醉的、自恋的、自我崇拜的”，所以水仙花数又被称为“自恋数”。

2. 分析问题

根据“水仙花数”的定义，判断一个数是否为“水仙花数”，最重要的是要把给出的三位数的个位、十位、百位分别拆分，并求其立方和，若立方和与原数相等，则为“水仙花数”，反之，则不是。“水仙花数”是指满足某一条件的三位数，根据这一信息可以确定整数的取值范围是 100~999。设 a 、 b 、 c 分别代表个位、十位、百位上的数。公式为

```

a = int(i%10)
b = int(i/10%10)
c = int(i/100)

```

3. 算法描述

根据上述分析，要实现将所有的“水仙花数”输出，问题处理步骤如下：

- (1) 程序开始；
- (2) 设置遍历范围 100~999，并依次求取个位、十位、百位上的数，并判断其立方和是否等于这个数本身；
- (3) 输出判断结果；
- (4) 程序结束。

4. 编程解题

输入以下代码，并在右侧的横线上写出代码注释：

```

for i in range(100,1000):
    a = int(i%10)
    b = int(i/10%10)
    c = int(i/100)
    if i ==(pow(a,3)+pow(b,3)+pow(c,3)):
        print(i)
    
```

5. 学习记录

工单四 用户系统登录

1. 提出问题

编写一个程序，实现用户系统登录功能。

2. 分析问题

在系统中，要实现用户的登录，具体功能需求如下：

- (1) 输入用户名和密码；
- (2) 判断用户名和密码是否正确 (name= 'admin', passwd= '123456');
- (3) 登录仅有 3 次机会，超过 3 次会报错。

3. 算法描述

根据上述分析，要实现用户系统登录，问题处理步骤如下：

- (1) 程序开始；
- (2) 输入用户名和密码；
- (3) 判断用户名和密码是否正确；
- (4) 输出提示；
- (5) 程序结束。

4. 编程解题

输入以下代码，并在右侧的横线上写出代码注释：

```

for i in range(3):
    user = input('请输入用户名: ')
    passwd = input('请输入密码: ')
    if user=='admin' and passwd == '123456':
    
```

```

        print('%s 用户登录成功' %user)
        break
    else:
        print('密码错误, 请重新输入密码, 您还剩%d 次机会' %(2-i))

else:
    print('超过三次, 登录失败')

```

5. 学习记录

工单五 百元百鸡

1. 提出问题

我国古代数学家曾提出过著名的“百钱买百鸡”问题，该问题叙述如下：鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一；百钱买百鸡，则翁、母、雏各几何？这题就是说：公鸡一只五元钱，母鸡一只三元钱，小鸡三只一元钱，现在要用一百元钱买一百只鸡，问公鸡、母鸡、小鸡各多少只？

2. 分析问题

如果假设公鸡、母鸡和小鸡各有 a、b、c 只，公鸡最多可以买 20 只，母鸡最多买 33 只，小鸡的只数为 $c=100-a-b$ 。

3. 算法描述

根据上述分析，要实现求公鸡、母鸡、小鸡各多少只，问题处理步骤如下：

- (1) 程序开始；
- (2) 设置循环；
- (3) 输出结果；
- (4) 程序结束。

4. 编程解题

输入以下代码，并在右侧的横线上写出代码注释：

```

for a in range(21):
    for b in range(34):
        c=100-a-b
        if a + b + c ==100 and a*5 + b*3 + (c//3)*1 == 100:
            print('公鸡:%3d 母鸡:%3d 小鸡:%3d' %(a, b, c))

```

5. 学习记录

工单六 独立作业

1. 提出问题

古时候印度有个国王很爱下棋，从来没有人赢过他。时间久了，国王觉得很无聊，就下令谁能赢他，他就会满足这个人一个愿望，有一天一个聪明的大臣提出了要和国王下棋，骄傲的国王根本没有把这个大臣放在眼里，结果输了。

国王决定信守承诺，他对大臣说：“我要重赏你，你想要什么金银珠宝，我都可以给你。”大臣回答说：“我只想要一些麦粒，陛下，请用这个棋盘的格子来计数，数到第一个格子时，给我 1 颗麦粒，第 2 个格子给我 2 颗麦粒，第 3 个格子给我 4 颗麦粒，第 4 个格子给我 8 颗麦粒，照此规律，数完全部 64 个格子，就是我要的麦粒数。”

国王听了大臣的要求，哈哈大笑。立刻嘱咐管粮食的大臣说：“你去拿几袋麦子给他吧。”管粮食的大臣在计算之后大惊失色，忙向国王报告说：“陛下，就算把全国的粮食都给他，也远远不够啊！”国王知道计算结果后，感到进退两难，这时管粮食的大臣对国王说：“陛下，请您下令，让他自己到粮仓取麦子，让他一粒一粒的数出那些麦粒。”

故事的结局是输了棋的国王没有失信于人，而赢了棋的大臣也没有办法取走自己想要的麦粒。因为按照赢棋大臣的算法计算出来的麦粒数量是一个非常巨大的天文数字，无论是给麦粒还是取麦粒，都是一个不可能完成的任务，那么赢棋大臣想要的麦粒数量到底是多少呢？请编写一个程序计算麦粒数。

2. 分析问题

3. 算法描述

- (1) _____
- (2) _____
- (3) _____
- (4) _____
- (5) _____
- (6) _____

4. 画流程图

5. 编程解题

6. 执行结果

活动五 函数之八十天环游地球

活动目标

- (1) 认知函数。
- (2) 认知函数的定义。
- (3) 认知函数的调用。
- (4) 认知函数的值传递。

活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的内容。
- (3) 小组成员互相沟通，服从小组管理。

💡 知识储备

一、函数

在结构化程序设计中，常将一些常用的功能模块编写成函数，这些函数在程序设计中可以任意调用。函数能提高程序的模块性和代码的重复利用率，而且可维护性高。

Python 中的函数常常分为内建函数和用户自定义函数。内建函数有 `print()` 函数和 `input()` 函数等。用户自定义函数指用户可以自定义函数的名称和功能。

二、函数的定义

Python 中用户自定义函数在使用前需要对函数进行定义。

1. 无参数函数

格式如下：

```
def 函数名 ( ):
    函数体
```

函数代码块以 `def` 关键词开头，后接函数标识符名称和圆括号 `()`。任何传入参数和自变量必须放在圆括号中间。圆括号之间可以用于定义参数。函数的第一行语句可以选择性地使用文档字符串——用于存放函数说明。函数内容以冒号起始，并且缩进。

例如：在屏幕上打印字符。

```
def printstar( ):
    print("*****")
def printmsg( ):
    print("  hello")
```

2. 带参数函数

格式如下：

```
def 函数名 (参数列表):
    函数体
    return [表达式]
```

`return [表达式]` 表示结束函数，并有选择性地返回一个值给调用方。不带表达式的 `return` 相当于返回 `None`。

例如：求两数之和。

```
def add(a,b):
    sum=a+b
    print(sum)
```

3. 函数的调用

定义了函数之后，就相当于已经实现设置好能完成一定功能的代码模块，要想让这些代码能够执行，需要调用函数。

1) 无参函数的调用

格式如下：

函数名()

例如：在屏幕上打印字符。

```
def printstar( ):
    print("*****")
def printmsg( ):
    print("  hello")
printstar( )
printmsg( )
printstar( )
```

程序执行结果：

```
*****
  hello
*****
```

2) 有参函数的调用

格式如下：

函数名(参数列表)

例如：输出两数中最大的数。

```
def max(x,y):
    if x>y:
        print("max=",x)
    else:
        print("max=",y)
a=int(input("请输入第一个数: "))
b=int(input("请输入第二个数: "))
max(a,b)
```

程序执行结果：

```
输入: 2 3
max=3
```

函数名后的参数列表是形式参数，这里面可以是一个参数也可以是多个参数。当是多个参数的时候，中间用逗号分隔；在调用这个参数的时候，就需要提供与形式参数数量一致的个数。

三、函数的返回值

返回值就是当函数调用后，返回到调用位置的结果。如执行求两数中最大数的函数，当执行完后，将最大数的结果返回到调用的位置，这个最大数就是函数的返回值。函数中使用 `return` 语句来返回一个值，`return` 语句是可选项，可以出现在函数体的任何位置，它的作用是结束当前函数，将程序返回到函数被调用的位置继续执行，同时将函数中的数据返回给主程序。

例如：求两数之和。

```
def add(x,y)
    sum=x+y
    return sum
```

在这个程序中 `sum` 就是返回值。

四、变量作用域

作用域指的是变量的有效范围。变量并不是在哪个位置都可以访问，访问权限取决于这个变量是在哪里赋值的，也就是在哪个作用域内的。

在 Python 中，根据作用域的不同，变量可以划分为局部变量和全局变量。

1. 局部变量

定义在函数内部的变量拥有一个局部作用域，称为局部变量。所谓的局部变量是相对的。局部变量也有可能是更小范围内的变量的外部变量。

2. 全局变量

定义在函数外的拥有全局作用域的变量，被称为全局变量。全局变量可以在整个程序范围内访问。调用函数时，所有在函数内声明的变量名称都将被加入作用域中。

例如：局部变量和全局变量。

```
num = 0                                # 全局变量
def sum(arg1, arg2):                  # 返回两个参数的和
    num = arg1 + arg2                 # num 在这里是局部变量
    print('局部变量: ', num)
    return num
sum(10, 5)
print('全局变量: ', num)
```

程序执行结果：

```
局部变量: 15
全局变量: 0
```



小知识



3-5 兔子生兔子问题（函数）

💡 操作工单

工单一 八十天环游地球

1. 提出问题

《八十天环游地球》是法国作家儒勒·凡尔纳创作的长篇小说。小说起因于英国绅士福格与朋友打的一个赌：要在 80 天内环游地球一周回到伦敦。随后，他与仆人克服了路途中的艰难险阻，路径地中海、红海、印度洋、太平洋、大西洋，游历印度、新加坡、日本、美国等地，最后返回伦敦，一路上福格机智、勇敢，表现出十足的绅士派头。

2. 分析问题

为了解决这个问题，假设福格环球路线上水路走了 32 000 km，陆路走了 8 000 km，根据输入

的水路速度和陆路速度，得出总小时数，最后根据总小时数求天数。

3. 算法描述

根据上述分析，要计算《八十天环游地球》中环游地球需要多少天，问题处理步骤如下：

- (1) 程序开始；
- (2) 输入水路速度和陆路速度；
- (3) 调用求天数函数；
- (4) 输出天数的值；
- (5) 程序结束。

4. 编程解题

输入以下代码，并在右侧的横线上写出代码注释：

```
def huanyou(wspeed,lspeed):
    wspeed=int(wspeed)
    lspeed=int(lspeed)
    hours=32000/wspeed+8000/lspeed
    days=round(hours/24,1)
    return days
a=input("请输入水路前进时速 km/h:")
b=input("请输入陆路前进时速 km/h:")
day=huanyou(a,b)
print('按福格路线环游地球要'+str(day)+'天')
```

5. 学习记录

工单二 推年龄

1. 提出问题

有 5 个人坐在一起，问第五个人多少岁？他说比第 4 个人大 2 岁。问第 4 个人的岁数，他说比第 3 个人大 2 岁。问第三个人，又说比第 2 人大两岁。问第 2 个人，说比第 1 个人大 2 岁。最后问第 1 个人，他说他 10 岁。请问第五个人多少岁？

2. 分析问题

利用递归的方法，递归分为回推和递推两个阶段。要想知道第五个人的岁数，需知道第四人的岁数，以此类推，推到第一人（10 岁），再往回推。已知第一个人 10 岁，之后每个人比前一个人大 2 岁。设第五个人的岁数为 age。

5: age age = fun(4)

- 4: age-2 fun(4) = fun(3) + 2
- 3: age-2-2 fun(3) = fun(2) + 2
- 2: age-2-2-2 fun(2) = fun(1) + 2
- 1: age-2-2-2-2=10 fun(1) = 10

3. 算法描述

根据上述分析，要推算第 5 个人的年龄，问题处理步骤如下：

- (1) 程序开始；
- (2) 编写一个推年龄的函数；
- (3) 调用函数；
- (4) 推算出年龄；
- (5) 程序结束。

4. 编程解题

输入以下代码，并在右侧的横线上写出代码注释：

```
def fun(i):
    if i==1:
        return 10
    else:
        return fun(i-1)+2
print (fun(5))
```

5. 学习记录

工单三 独立作业

1. 提出问题

编写一个加法函数，实现任意两个数的求和功能。

2. 分析问题

3. 算法描述

- (1) _____
- (2) _____

任务四

Python 之绘画 ——模块化思维养成



任务目标

- (1) 认知模块化思维方式和工作方法。
- (2) 熟练使用 Python 第三方模块 Turtle 绘图。
- (3) 培养学生思维方式、工作方法和创新能力等综合职业能力。



任务要点

- (1) 模块化思维与 Python 模块。
- (2) Turtle 绘图的原理、安装与命令使用。
- (3) 基本形状与复杂图形绘制。
- (4) 创新图形设计与绘制。



信息素养目标

- (1) 职业素养：
 - ① 培养积极、有责任心的工作态度。
 - ② 培养模块化思维方式和工作方法。
 - ③ 培养分析问题、解决问题的能力。
 - ④ 培养小组协作互助成长的职业能力。
- (2) 技能素养：

培养规范化的程序设计、编辑和调试技能。
- (3) 创新能力：

培养运用信息技术独立解决问题的能力。



任务内容

- 活动一：模块化思维与 Turtle 库的认知。
- 活动二：安装 Turtle 小乌龟绘图库。
- 活动三：绘制基本形状之绿色蟒蛇。
- 活动四：绘制复杂图形之迷幻星空。
- 活动五：个人创新作业。

活动一 模块化思维与 Turtle 库的认知

活动目标

- (1) 认知模块化思维方式。
- (2) 认知 Python 第三方库。
- (3) 认知 Turtle 绘图库的基本原理。

活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的内容。
- (3) 小组成员互相沟通，服从小组管理。

知识储备

一、模块化思维

模块化思维是数字化编程中重要的思维模式，是信息化世界用来解决问题的一种常见的思路和方法，用中国传统的描述就是“分而治之”的思想。

简言之，程序设计当中的模块就像是一个黑盒子，我们只需关心模块提供的功能是什么，而不用理会模块内部这个功能是如何实现的。例如，我们买了一部智能手机，当我们用手机拍照的时候，我们只需要按下照相键即可，我们根本不需要知晓手机内部的照相模块是如何照相并形成照片的，这些功能的具体实现过程和步骤对我们用户而言，就像是被封装在了一个黑盒子里面，我们只能看到它的功能，而看不到它内部的实现过程，如图 4-1 所示。



图 4-1 模块示意图

模块化方式的出现让我们在编程时只需找到想要的功能模块，然后学习如何去调用这个模块即可，这种工作方式和思维模式大大降低了人们在数字化编程上的难度，使每个现代人都有可能成为编程高手。模块化工作方式也应用在了大型程序设计上，一个团队在进行程序设计时，将一个程序按照功能划分为若干个模块，每个模块完成一个确定的功能，并在这些模块之间建立必要的联系，通过模块的互相协作完成整个功能的程序设计。

二、第三方模块

Python 模块（库）= 标准库 + 第三方库。

- (1) Python 模块（库）：逻辑上来说 Python 模块（库）就是一组功能的组合，实质上就是一

个包含了 Python 定义和声明的文件，文件名后缀为.py。

- (2) 标准库：随解释器直接安装到操作系统中的功能模块。
- (3) 第三方库：需要经过安装才能使用的第三方功能模块。
- (4) Python 常用的第三方库。
 - ① Turtle：入门级的图形绘制函数库。
 - ② Pygame：2D 游戏的开发库。
 - ③ Pyglet：2D、3D 动画和游戏开发库。
 - ④ Matplotlib：绘制数据图、用于数据分析的库。
 - ⑤ OpenCV：开源的跨平台计算机视觉库，用于图像识别等人工智能。
 - ⑥ Scrapy：高级的 Python 爬虫框架库，用于快速、高层次的屏幕抓取和 Web 抓取。

三、Turtle 库简介

Turtle，中文“海龟”，它是 Python 语言中一个很流行的入门级绘制图像的函数库，想象有一只小海龟，它从画布的正中心开始，在沙滩上爬行，经过的轨迹就形成绘制的图形，这只海龟由程序控制，可以变换颜色、改变宽度等。

四、Turtle 的魅力

使用 Turtle 库，可以轻松绘制如图 4-2 所示的各种图形，在之后的项目中，将对绘制这些图形的方法做详细讲解。



图 4-2 使用 Turtle 库绘制的图形

五、Turtle 原理

1. 沙滩（画布 canvas）

沙滩即小海龟爬行绘画的画布，就是 Turtle 在计算机屏幕上展开的一个用于绘图的区域。

1) 沙滩大小

Turtle 运行后默认打开的是一个 400×300 大小的绘图区域（见图 4-3），即宽为 400 像素、高为 300 像素、背景为白色的一个沙滩。

可以设置沙滩（画布）的大小和背景颜色，如：`turtle.screensize(800,600, "yellow")`，参数分别为画布的宽（单位像素）、高和背景颜色。

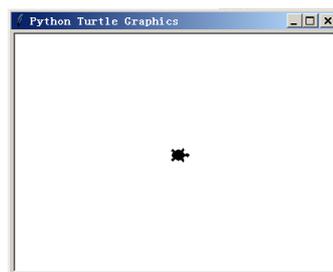


图 4-3 400×300 沙滩（画布）

2) 沙滩坐标体系

在沙滩（画布）上，默认有一个水平的 x 轴和一个垂直的 y 轴， x 轴与 y 轴的交叉点就是沙滩（画布）的坐标原点，即沙滩（画布）的正中心，用 $(0,0)$ 表示，如图 4-4 沙滩坐标图所示，小乌龟的默认起始点就在这里。以 x 轴、 y 轴和中心点 $(0,0)$ 为参照，沙滩（画布）上所有的点都可以用一对 x, y 坐标值表示出来，当小乌龟想去沙滩的任何位置的时候，你只需要告诉计算机这

个位置的坐标值即可。例如，小乌龟从(0,0)起点想爬到 a 点这个位置，你只需要命令计算机 `turtle.goto(100,100)`，小乌龟就准确的跑到了 a 点。当然，我们需要会使用这个沙滩坐标系来定位每个点的绝对坐标，这个很关键。例如， b 点在沙滩（画布）上的绝对坐标为(-100,100)， c 点在沙滩（画布）上的绝对坐标为(-100,-100)， d 点在沙滩（画布）上的绝对坐标为(100,-100)。

2. 海龟（画笔 pen）

Turtle 小海龟，默认在画布的正中心为起始出发点，头朝右为起始方向，它可以根据 Turtle 库里函数命令的控制，按照图 4-5 海龟方向坐标图所示，在这个沙滩（画布）上做前进、后退、向左、向右等平面移动，从而在它爬行的路径上绘制图形。

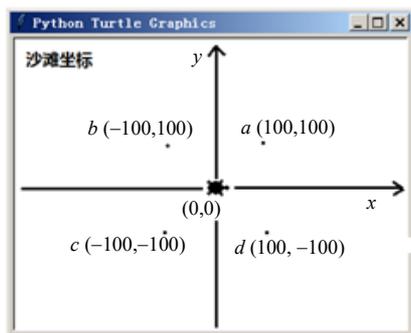


图 4-4 沙滩坐标

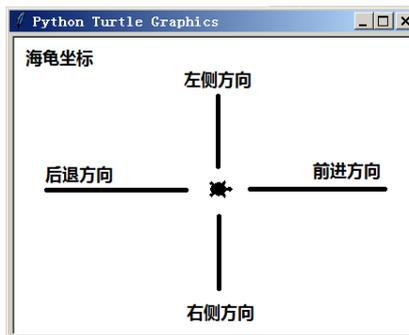


图 4-5 海龟方向坐标

3. Turtle 的颜色

1) RGB 色彩体系

Turtle 的颜色采用的是计算机的 RGB 色彩体系，RGB 色彩是工业界的一种颜色标准。RGB 即 R (red)、G (green)、B (blue)——红、绿、蓝三原色，RGB 色彩体系只使用这三种颜色，并通过这三种颜色的变化，使它们按照不同的比例混合出万物色，RGB 色彩几乎包括了人类视觉所能感知的所有颜色，是目前运用最广的颜色系统之一。

在 Turtle 的 RGB 色彩体系中，红、绿、蓝三种颜色的取值范围是 0~255 之间的整数，或 0~1 中的小数，就像厨师在做菜的时候，用“盐 3 g、糖 1 g、醋 5 g”来混合不同量的调料做出一道又一道不同口味的菜品。按照计算，256 级的 RGB 色彩总共能组合出约 1 678 万种色彩，即 $256 \times 256 \times 256 = 16\,777\,216$ ，通常也被简称为 1 600 万色或千万色。例如，纯白色的 RGB 值是 255, 255, 255，纯红色的 RGB 值是 255, 0, 0，纯绿色的 RGB 值是 0, 255, 0，纯蓝色的 RGB 值是 0, 0, 255。

2) Turtle 的颜色表达方式

- (1) 直接使用颜色字符串，一般为颜色的英文单词。
- (2) 使用 RGB 整数值模式：0~255 之间的整数。
- (3) 使用 RGB 小数值模式：0~1 之间的小数，Turtle 默认采用 RGB 小数值。



小知识



4-1 什么是像素？

📌 操作工单

工单 遨游 Python 第三方库

1. 操作步骤

- (1) 进入 PyPI 官方网站，开始遨游，网址：<https://pypi.org/>。
- (2) 在搜索栏中输入“turtle”搜索小乌龟绘图库。
- (3) 在搜索栏中输入“pygame”搜索 2D 游戏开发库。
- (4) 在搜索栏中输入“pygamelet”搜索 2D、3D 动画和游戏开发库。
- (5) 在搜索栏中输入“matplotlib”搜索分析数据、绘制数据图库。
- (6) 在搜索栏中输入“opencv”搜索开源的跨平台计算机视觉库，用于图像识别等人工智能。
- (7) 在搜索栏中输入“scrapy”搜索高级的 Python 爬虫框架库，用于快速、高层次的屏幕抓取和 Web 抓取。
- (8) 使用百度找到你感兴趣的 Python 第三方库，并在 PyPI 官方网站搜索栏中继续查找。

2. 操作结果

预期结果如图 4-6 所示，请将实际结果简单绘制在右侧的方框内。

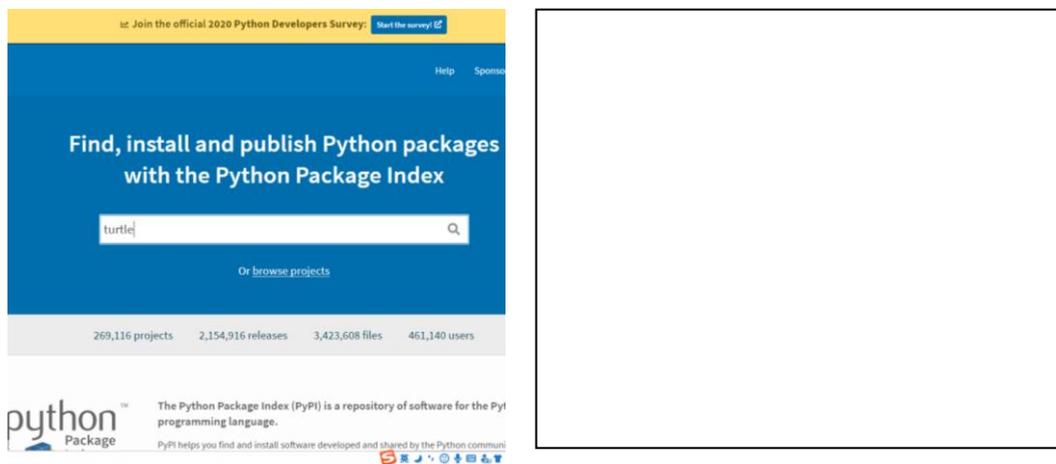


图 4-6 预期结果与实际结果

3. 学习记录

活动二 安装 Turtle 小鸟龟绘图库

📍 活动目标

- (1) 下载 Turtle 图形库安装包。
- (2) 安装 Turtle 图形库。
- (3) 测试 Turtle 库的运行。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的任务。
- (3) 软件安装步骤规范。
- (4) 小组成员互相协作，服从小组管理。

📍 知识储备

一、Turtle 安装包下载

下面要给大家介绍一个可以获得第三方 Python 软件包以补充标准库的站点及系统，它就是 PyPI (Python Package Index)。PyPI 即 Python 包索引，是 Python 官方的第三方库的仓库，是 Python 编程语言的软件存储库，所有人都可以在该站点中下载 Python 第三方库或上传自己开发的库。

PyPI 官方网站的网址：<https://pypi.org/>，打开网站，首页如图 4-7 所示，在网站的搜索栏中输入我们需要寻找的“turtle”，然后单击“搜索”按钮，即出现 Turtle 库搜索结果，如图 4-8 所示。



图 4-7 PyPI 官方网站首页

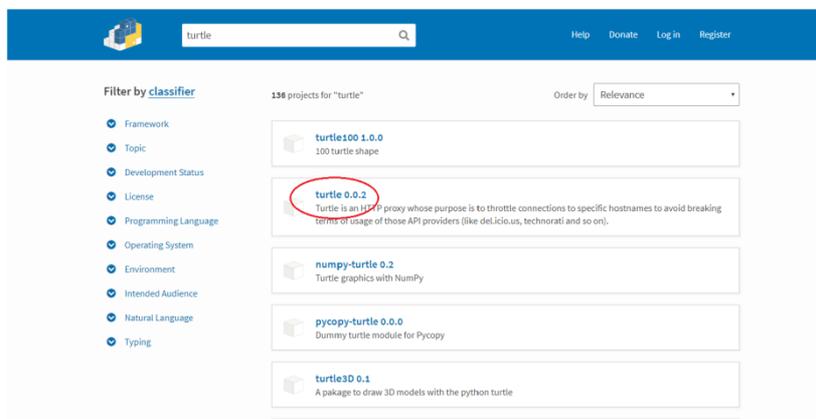


图 4-8 Turtle 库搜索结果

在 Turtle 库搜索结果中选择 Turtle 0.0.2 版本，然后单击，即可进入 Turtle 0.0.2 下载界面，然后单击“Download Files”按钮，在界面右侧出现的“turtle-0.0.2.tar.gz”安装包文件名上单击进行下载，如图 4-9 所示。

选择将“turtle-0.0.2.tar.gz”安装包下载到 D 盘或者其他位置，下载界面如图 4-10 所示。Turtle 0.0.2 下载完成，“turtle-0.0.2.tar.gz”安装包出现在 D 盘中，如图 4-11 所示。将“turtle-0.0.2.tar.gz”安装包解压，等待安装，如图 4-12 所示。

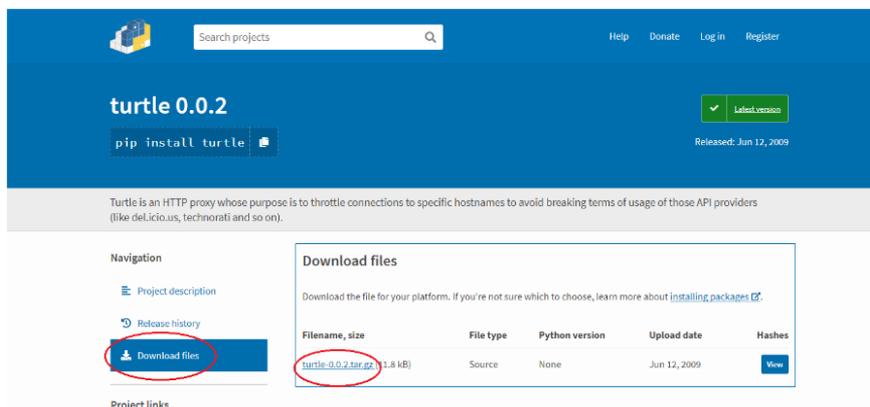


图 4-9 turtle-0.0.2 下载界面



图 4-10 下载 turtle-0.0.2

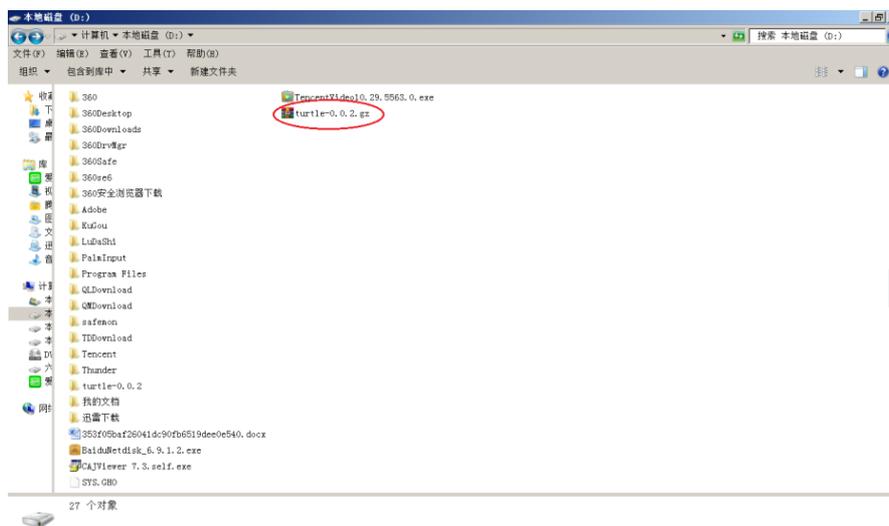


图 4-11 下载的 turtle-0.0.2.tar.gz 安装包

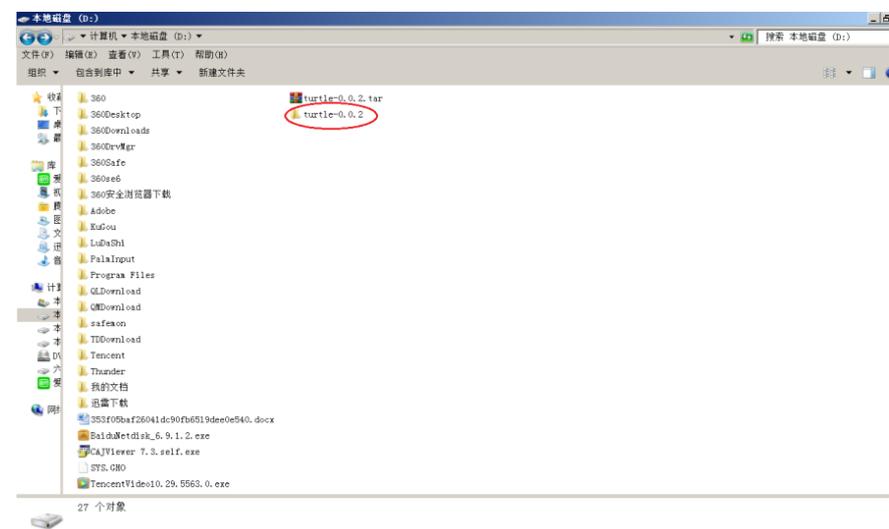


图 4-12 解压安装包

二、Turtle 图形库安装

PyPI 推荐使用 pip 包管理器来下载、安装第三方库，前面章节我们在讲解安装 Python 时，介绍过 pip (python install packages)，它是一个安装和管理 Python 包的工具，Python 的第三方模块（库）的安装一般都可以使用“pip”命令来安装，非常方便。我们安装 Python 模块（库）的时候，通常只用在 cmd 命令行上写上这么一句：pip install xxx，即可完成对应 Python 模块（库）的安装。

我们在前面介绍 Python 版本的时候讲过，Python 有 Python2 和 Python3 之分，因为 Turtle 库原来主要是在 Python2 中使用，Turtle 库安装文件 setup.py 也是基于 Python2 的，在 Python2 中安装 Turtle 库按照 pip 常规安装命令格式即可：

```
pip install turtle
```

但是，众所周知，Python3 对于 Python2 是不完全兼容的，在 Python3 中安装 Turtle 库按照 pip 常规安装命令格式应该是：

```
pip3 install turtle
```

大家按照这样做的结果是出现了提示错误，pip 无法成功安装 Turtle 库。错误出在哪里呢？让我们打开刚才下载的 Turtle 库的安装包，然后在 IDLE 中打开安装文件“setup.py”，如图 4-13 所示。

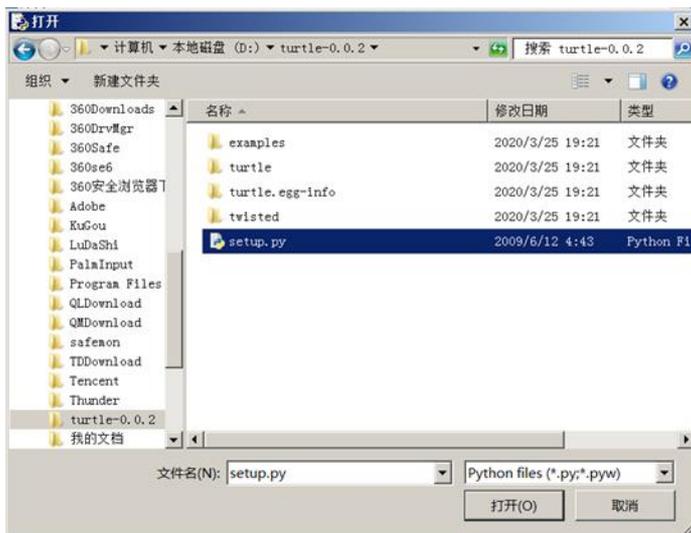


图 4-13 安装文件“setup.py”

在安装文件“setup.py”中，大家可以看到灰色这一行代码为 `except ValueError, ve:`，如图 4-14 所示，这是原来 Python2 的写法，没有括号，在 Python3 需要加括号才能使用，所以我们手动将这行代码修改为 `except (ValueError, ve):`，如图 4-15 所示，然后保存即可。

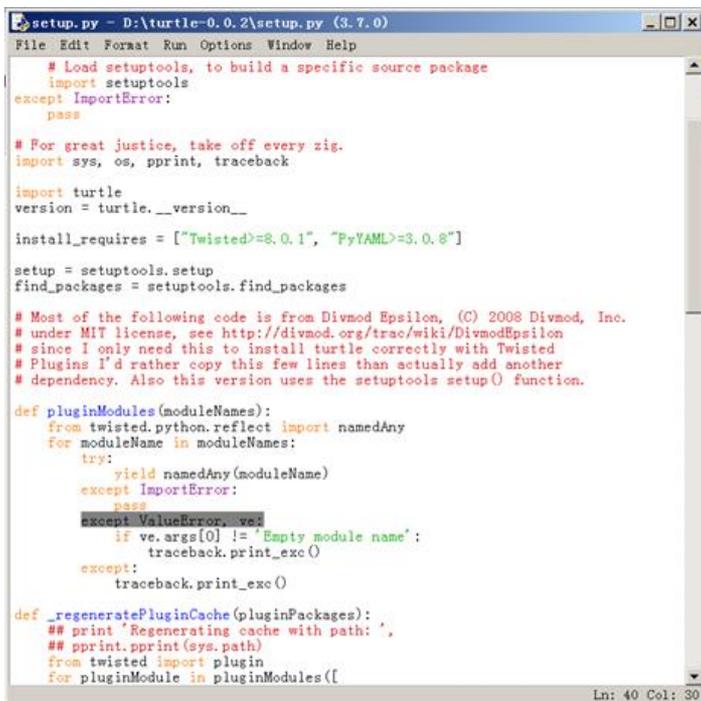


图 4-14 “setup.py”文件修改前

```

+setup.py - D:\turtle-0.0.2\setup.py (3.7.0)+
File Edit Format Run Options Window Help
# Load setuptools, to build a specific source package
import setuptools
except ImportError:
    pass

# For great justice, take off every zig.
import sys, os, pprint, traceback

import turtle
version = turtle.__version__

install_requires = ["Twisted>=8.0.1", "PyYAML>=3.0.8"]

setup = setuptools.setup
find_packages = setuptools.find_packages

# Most of the following code is from Divmod Epsilon, (C) 2008 Divmod, Inc.
# under MIT license, see http://divmod.org/trac/wiki/DivmodEpsilon
# since I only need this to install turtle correctly with Twisted
# Plugins I'd rather copy this few lines than actually add another
# dependency. Also this version uses the setuptools setup() function.

def pluginModules(moduleNames):
    from twisted.python.reflect import namedAny
    for moduleName in moduleNames:
        try:
            yield namedAny(moduleName)
        except ImportError:
            pass
        except (ValueError, ve):
            if ve.args[0] != 'Empty module name':
                traceback.print_exc()
        except:
            traceback.print_exc()

def _regeneratePluginCache(pluginPackages):
    ## print 'Regenerating cache with path: ',
    ## pprint.pprint(sys.path)
    from twisted import plugin
    for pluginModule in pluginModules({

```

图 4-15 “setup.py”文件修改后

然后就可以使用 pip 工具进行 Turtle 库的安装了。注意：pip 工具安装的应该是下载到 D 盘里的 turtle-0.0.2 安装包，因为安装包里面的“setup.py”已经是修改后的安装文件，首先通过输入 cmd 命令进入命令提示符窗口，在提示符后面输入 pip 安装命令“pip install -e d:\turtle-0.0.2”，如图 4-16 所示。

在输入 pip 安装命令“pip install -e d:\turtle-0.0.2”之后按“Enter”键执行命令，进行安装，并提示安装成功“Successfully installed turtle”，如图 4-17 所示。

```

管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation. 保留所有权利。

C:\Users\Administrator>pip install -e d:\turtle-0.0.2

```

图 4-16 pip 安装 turtle-0.0.2

```

管理员: C:\Windows\system32\cmd.exe
local\programs\python\python37-32\lib\site-packages (from Twisted>=8.0.1->turtle
==0.0.2) (20.2.0)
Requirement already satisfied: zope.interface>=4.4.2 in c:\users\administrator\appdata\local\programs\python\python37-32\lib\site-packages (from Twisted>=8.0.1->turtle==0.0.2) (4.7.1)
Requirement already satisfied: constantly>=15.1 in c:\users\administrator\appdata\local\programs\python\python37-32\lib\site-packages (from Twisted>=8.0.1->turtle==0.0.2) (15.1.0)
Requirement already satisfied: attrs>=17.4.0 in c:\users\administrator\appdata\local\programs\python\python37-32\lib\site-packages (from Twisted>=8.0.1->turtle==0.0.2) (19.3.0)
Requirement already satisfied: idna>=2.5 in c:\users\administrator\appdata\local\programs\python\python37-32\lib\site-packages (from hyperlink>=17.1.1->Twisted>=8.0.1->turtle==0.0.2) (2.9)
Requirement already satisfied: six in c:\users\administrator\appdata\local\programs\python\python37-32\lib\site-packages (from Automat>=0.3.0->Twisted>=8.0.1->turtle==0.0.2) (1.14.0)
Requirement already satisfied: setuptools in c:\users\administrator\appdata\local\programs\python\python37-32\lib\site-packages (from zope.interface>=4.4.2->Twisted>=8.0.1->turtle==0.0.2) (39.0.1)
Installing collected packages: turtle
  Running setup.py develop for turtle
Successfully installed turtle

C:\Users\Administrator>

```

图 4-17 安装成功

此时，可以再次使用 pip 工具显示已经安装好的 Turtle 库，只需在命令提示符后输入“pip list”，如图 4-18 所示。

按“Enter”键后，可以看到 Python 下安装的包中已经包括了 Turtle-0.0.2，如图 4-19 所示。



图 4-18 pip list 命令

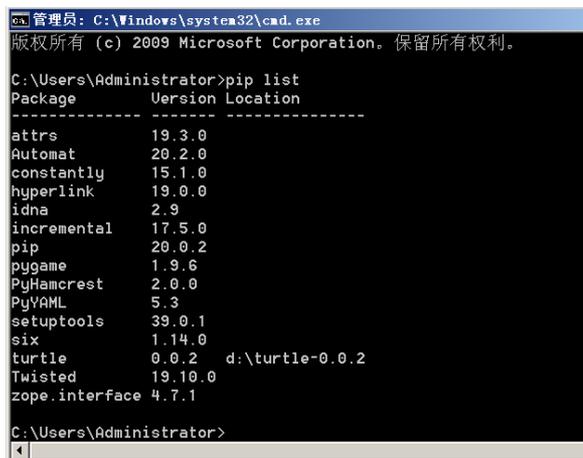


图 4-19 显示已安装的库

三、Turtle 运行调试

成功安装 Turtle 库之后，我们在 Python 的 IDLE 集成开发环境中新建一个 test.py 文件，输入两行代码，对 Turtle 库进行简单的测试，测试 Turtle 是否能正常运行。输入代码如图 4-20 所示。

然后运行程序，Turtle 绘图库的海滩正常打开，中间的箭头代表海龟，趴在沙滩的中心等待用户的命令，如图 4-21 所示。

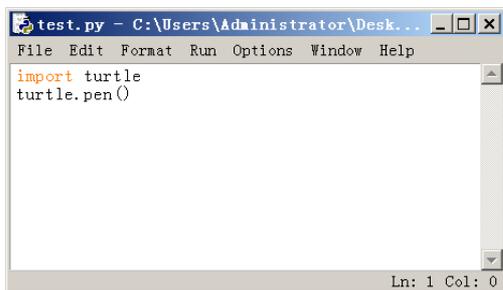


图 4-20 代码调试

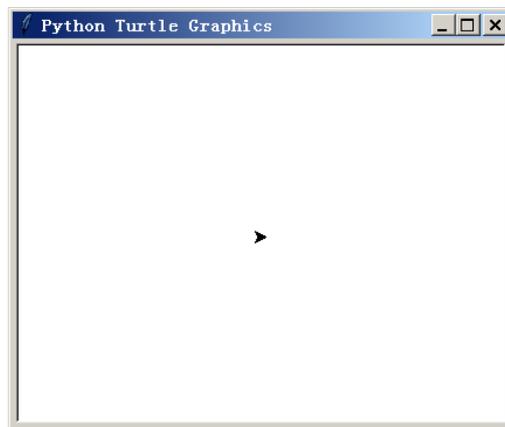


图 4-21 调试成功



小知识



4-2 小海龟不见啦!

🔔 操作工单

工单一 Turtle 库的安装

1. 操作步骤

- (1) 打开 PyPI 官方网站的网址：<https://pypi.org/>，然后搜索 turtle 库。
- (2) 下载 Turtle 0.0.2 安装包，保存到计算机的指定位置。
- (3) 解压缩 Turtle 0.0.2 安装包。
- (4) 打开 Turtle 0.0.2 安装包中的安装文件“setup.py”，找到“except ValueError, ve:”代码行，修改为“except (ValueError, ve):”，并保存。
- (5) 输入 cmd 命令进入命令提示符窗口。
- (6) 使用 pip 工具进行安装，在提示符后面输入并执行“pip install -e d:\turtle-0.0.2”命令。
- (7) 出现“Successfully installed turtle”提示安装成功。
- (8) 在提示符后面输入“pip list”，查看 Python 下安装的包已经包括 turtle-0.0.2。

2. 操作结果

预期结果如图 4-22 所示，请在右边的方框中简单绘制实际效果图。

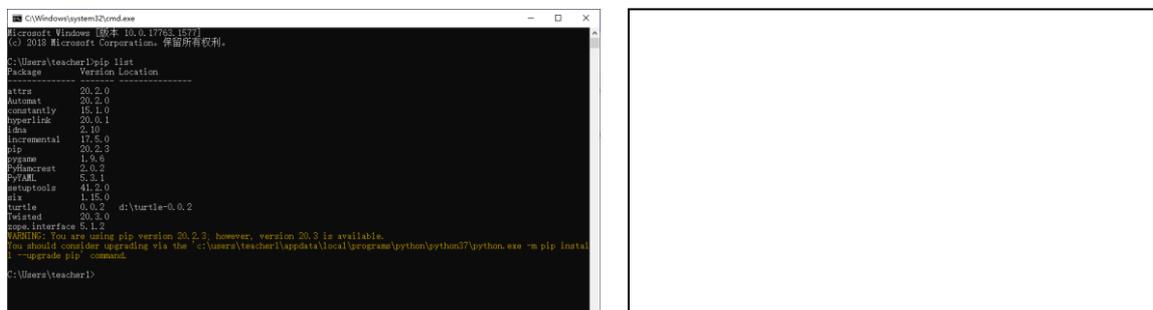


图 4-22 预期结果与实际结果

3. 学习记录

工单二 Turtle 库的调试

1. 操作步骤

- (1) 打开 Python 的 IDEL 集成开发环境，新建一个 test.py 文件，准备对 Turtle 库进行简单的测试。

(2) 在 test.py 文件中输入两行测试代码:

```
import turtle      # 使用之前需要使用 import 导入 Turtle 库  
turtle.pen( )     # 启用画笔
```

(3) 按“F5”键运行程序。

(4) Turtle 的绘图界面(海滩)正常打开,箭头(海龟)在画布的正中央。

(5) 经测试, Turtle 库成功安装, 并正常运行。

2. 操作结果

预期结果如图 4-23 所示, 请在右边的方框中简单绘制实际效果图(加入 pip list 截图)。

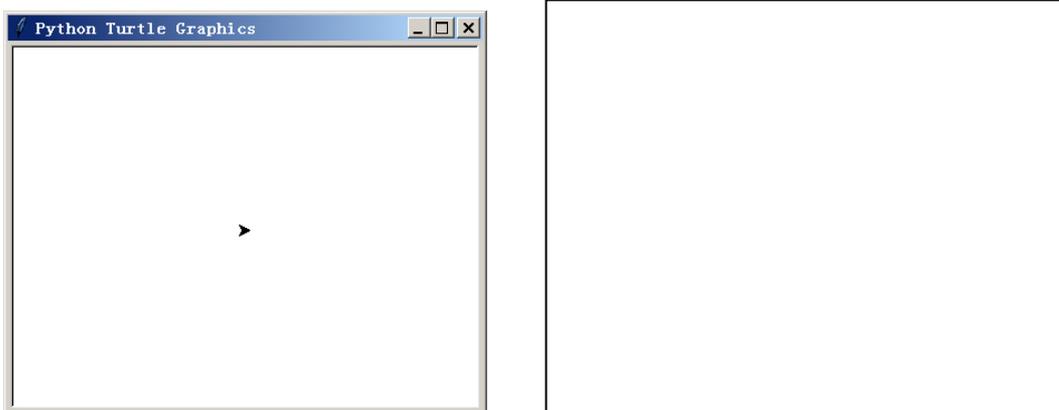


图 4-23 预期结果与实际结果

3. 工作记录

活动三 绘制基本形状之绿色蟒蛇

活动目标

- (1) 认知 Turtle 库中设置画布、画笔和运动的基本命令。
- (2) 绘制长方形、正方形、三角形等多边形。
- (3) 绘制圆形、半圆形。
- (4) 运用基本命令绘制五角星、小蟒蛇。

活动要求

- (1) 遵守课堂纪律，完成规定的任务。
- (2) 程序设计、编辑、调试规范。
- (3) 小组成员互相协作，服从小组管理。

活动储备

在任务一中，我们对 Turtle 绘图库的功能、Turtle 绘图的坐标、运动和色彩原理进行了基本的概要认知，在任务二中，也成功下载并安装好了 Turtle 库，今天的任务三里，我们将开始学习 Turtle 库的各种常用绘图命令，并完成各种基本图形的绘制。

一、import 导入 Turtle 库

在 Python 中，如果我们想要使用内置模块（库）或者第三方模块（库）的功能，那么用 `import` 来导入相应的模块后即可使用。Turtle 库和我们前面使用过的 `math` 库、`random` 库一样，需要先导入库，才可以使用 Turtle 绘图库中的函数，画出各种有趣的图形，这就类似于 C 语言中的 `include` 头文件，在 Python 中用 `import` 导入 Turtle 库的语句需要写在程序的开头，导入语句如下：

```
import turtle
```

二、设置沙滩（画布）

在 Python 中用 `import` 导入 Turtle 库之后就可以开始绘制图形了，Turtle 绘图的原理可以形象地理解成一只由程序控制的小海龟，它从沙滩（画布）的正中心开始，在沙滩上爬行，经过的轨迹就形成了绘制的图形。所以，我们首先要设置一张画布，就是在屏幕上展开一个用于绘图的区域。

Turtle 运行后默认打开的是一个 400×300 大小、白色的绘图区域，如图 4-24 所示。

当然，我们也可以设置画布的大小和背景颜色，如：`turtle.screensize(800,600)`，将画布大小设置为 800×600，在计算机屏幕上的显示如图 4-25 所示。



图 4-24 400×300 画布



图 4-25 800×600 画布

下面学习两个有关画布设置的命令：

1. `turtle.screensize()` 命令

- (1) 功能：设置画布大小和背景颜色。

(2) 格式: `turtle.screensize(canvwidth, canvheight, bgcolor)`。

(3) 参数: 三个参数分别为画布的宽(单位像素)、高、背景颜色, 如果缺省, 分别默认为宽 400、高 300、背景白色。

(4) 例如:

```
turtle.screensize(800,600, "yellow")      # 设置一张 800*600 大小的黄色画布
turtle.screensize()                        # 设置一张 400*300 大小的白色画布
```

2. `turtle.bgcolor()` 命令

(1) 功能: 设置画布背景颜色。

(2) 格式: `turtle.bgcolor(color)`。

(3) 参数: 参数为画布的背景色, 可以是颜色单词字符串, 也可以是 RGB 三元组。

(4) 例如:

```
turtle.bgcolor("blue")                    # 设置画布背景色为蓝色
turtle.bgcolor(255,255,255)              # 设置画布背景色为黑色
```

3. `turtle.title()` 命令

(1) 功能: 设置窗口标题。

(2) 格式: `turtle.title(titlestring)`。

(3) 参数: 参数为标题字符串。

(4) 例如:

```
screen.title("hello turtle")              # 设置 title 窗口标题为“hello turtle”
```

Turtle 窗口默认标题如图 4-26 所示, Turtle 窗口设置“hello turtle”标题如图 4-27 所示。

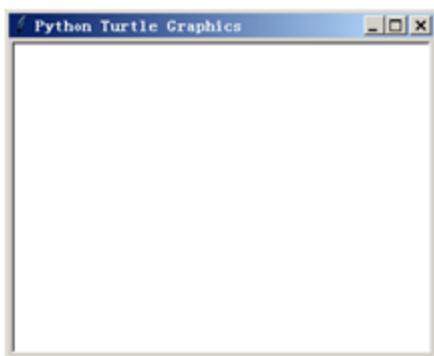


图 4-26 Turtle 窗口默认标题



图 4-27 窗口设置新标题

三、设置海龟(画笔)

设置好画布, 大家会感到很奇怪, 为什么在上面设置画布的效果图里面没有看到那只说好的小海龟呢? 这就是我们马上要讲的设置画笔命令, 要让小海龟现身啦!

1. `turtle.pen()` 命令

(1) 功能: 启用画笔。

(2) 格式: `turtle.pen()`。

(3) 参数: 无参数。

(4) 例如:

```
import turtle
turtle.pen( )
```

程序执行后, 出现了一只默认在画布中心位置面朝 x 轴正方向的小海龟, 这只小海龟就是绘图的画笔, 如图 4-28 所示。

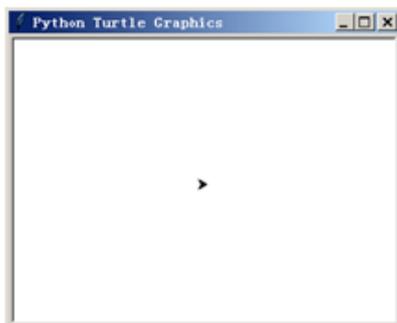


图 4-28 小海龟现身图

下面来学习一些常用的画笔属性设置, 包括设置画笔的颜色、画线的宽度和画笔的速度等。

2. turtle.hideturtle()命令

(1) 功能: 隐藏画笔 (即海龟箭头), 画笔此时只要是按下状态, 虽然隐藏但仍能正常绘制图形, 尤其在复杂的快速绘图时隐藏画笔效果显著。

(2) 格式: `turtle.hideturtle()`。

(3) 参数: 无参数。

3. turtle.showturtle()命令

(1) 功能: 显示隐藏的画笔 (即海龟箭头)。

(2) 格式: `turtle.showturtle()`。

(3) 参数: 无参数。

4. turtle.pensize()命令

(1) 功能: 设置画笔的宽度即绘图线条的粗细。

(2) 格式: `turtle.pensize(width)`。

(3) 参数: 参数为画笔宽度, 取值为正数, 以像素为单位; 若无参数, 返回当前笔粗。

(4) 例如:

```
turtle.pensize(2)      # 设置画笔粗细为 2 个像素宽度
turtle.pensize(25)    # 设置画笔粗细为 25 个像素宽度
```

画笔粗细效果如图 4-29 所示。

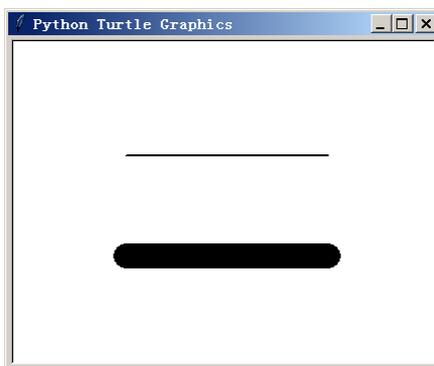


图 4-29 设置画笔宽度

5. turtle.pencolor()命令

- (1) 功能：设置画笔的颜色。
- (2) 格式：turtle.pencolor(color)。
- (3) 参数：参数为画笔颜色，可以是颜色单词字符串，也可以是 RGB 三元组；若无参数，返回当前画笔颜色。
- (4) 例如：

```
turtle.pencolor("red")           # 设置画笔颜色为红色
turtle.pencolor(255,0,0)        # 设置画笔颜色为红色
```

6. turtle.speed()命令

- (1) 功能：设置画笔移动速度，画笔绘制的速度范围是[0,10]之间的整数，数字越大越快。
- (2) 格式：turtle.speed(speed)。
- (3) 参数：参数为画笔移动速度，speed 为[0,10]之间的整数或者是速度字符串，其中 0 表示最快，从 1 到 10 越来越快；若无参数，返回当前画笔速度。
- (4) 例如：

```
turtle.speed(0)                  # 设置画笔速度为数值 0，表示速度最快
turtle.speed("fastest")         # 设置画笔速度为速度字符串 fastest，也表示速度最快
```

速度数值与速度字符串的对应关系如表 4-1 所示。

表 4-1 速度数值与字符串的对应关系表

速度数值	速度字符串
0	"fastest"
1	"slowest"
3	"slow"
6	"normal"
10	"fast"

四、海龟（画笔）运动命令

1. turtle.forward()命令

- (1) 功能：画笔向当前方向画直线，即移动 distance 个像素距离。

- (2) 格式: `turtle.forward(distance)`。
- (3) 参数: 参数为画笔移动的距离, 单位为像素。
- (4) 例如:

```
import turtle  
turtle.forward(200)      # 命令小海龟(画笔)向当前方向移动画 200 像素的直线
```

绘制效果如图 4-30 所示。

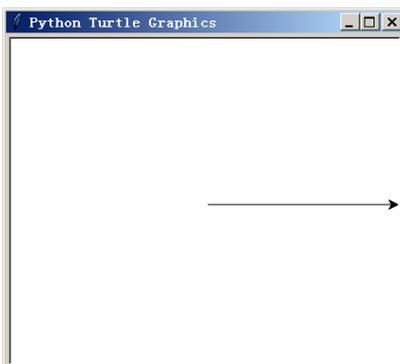


图 4-30 画 直 线

2. `turtle.backward()`命令

- (1) 功能: 画笔向当前方向的反方向画直线, 即反向移动 `distance` 个像素距离。
- (2) 格式: `turtle.backward(distance)`。
- (3) 参数: 参数为数字, 单位默认为像素, 表示画笔移动的距离。
- (4) 例如:

```
turtle.backward(200)    # 命令小海龟(画笔)向当前方向移动画 200 像素的直线
```

注意箭头朝向没有改变, 一直朝向 x 轴正方向, 换句话说讲, `turtle.backward()`绘制反向直线, 不会影响海龟的朝向, 就像人倒着走路一样, 你的面部朝向不会改变, 绘制效果如图 4-31 所示。

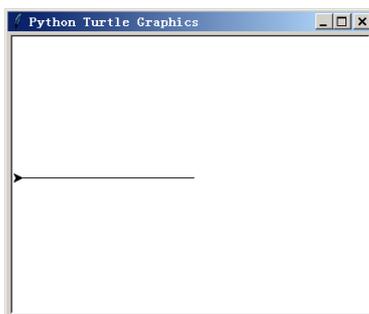


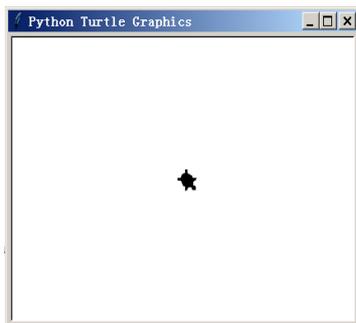
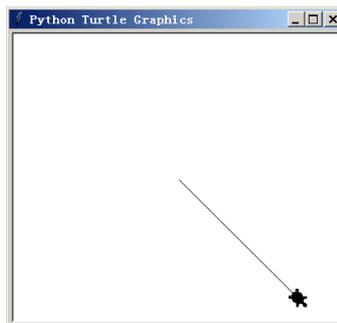
图 4-31 画 直 线

3. `turtle.right()`命令

- (1) 功能: 绘制方向向右旋转 `angle` 度, 即海龟向右转 `angle` 度。
- (2) 格式: `turtle.right(angle)`。
- (3) 参数: 参数为画笔向右旋转的角度数, 单位默认为度数。
- (4) 例如:

```
import turtle
turtle.shape("turtle") # 先将箭头变身为海龟
turtle.right(45) # 命令小海龟（画笔）向右转 45 度
```

为了更清楚地看到海龟的转向，上面程序先将箭头变身为海龟，然后再右转 45° ，绘制效果如图 4-32 所示，如果转向 45° 后画直线，绘制效果如图 4-33 所示。

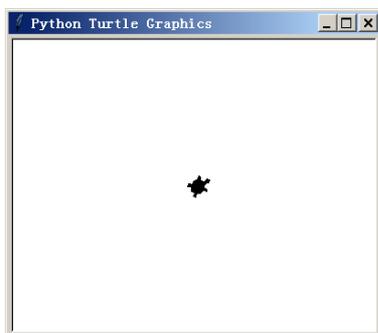
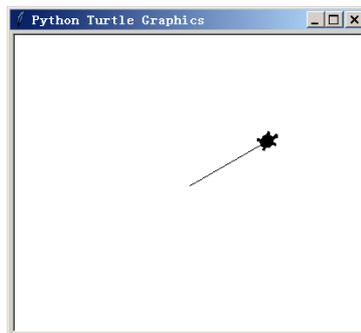
图 4-32 海龟右转 45° 图 4-33 海龟右转 45° 画直线

4. turtle.left()命令

- (1) 功能：绘制方向向左旋转 $angle$ 度，即海龟向左转 $angle$ 度。
- (2) 格式：turtle.left($angle$)。
- (3) 参数：参数为画笔向左旋转的角度数，单位默认为度数。
- (4) 例如：

```
import turtle
turtle.shape("turtle") # 先将箭头变身为海龟
turtle.left(30) # 命令小海龟（画笔）向左转 30 度
```

海龟左转 30° 效果如图 4-34 所示，如果左转 30° 后画直线，绘制效果如图 4-35 所示。

图 4-34 海龟左转 30° 图 4-35 海龟左转 30° 画直线

5. turtle.dot()命令

- (1) 功能：按给定直径 $size$ 画圆点。
- (2) 格式：turtle.dot($size$, $color$)。
- (3) 参数：参数分别为圆点的直径和颜色， $size$ 是一个大于 1 的整数，可以为空，默认用的是 pensize+4 和 2*pensize 的最大值， $color$ 可以是单词字符串或者 RGB 数值。
- (4) 例如：

```
import turtle
```

```
turtle.dot(20,"red") # 画直径为 20 像素大小的红色原点
```

绘制效果如图 4-36 所示。

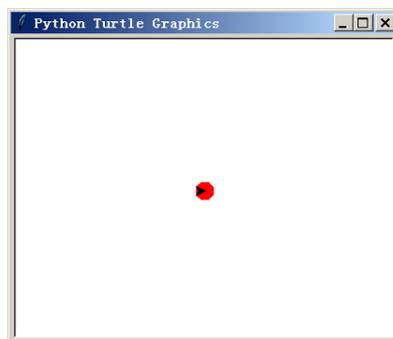


图 4-36 绘制圆点

6. turtle.circle()命令

(1) 功能：按给定的半径画圆，当前位置为圆的初始端点。

(2) 格式：`turtle.circle(radius, extent, steps)`。

(3) 参数：三个参数分别为圆半径、弧度和步数，`radius` 表示圆半径，为正数则逆时针画，为负数则顺时针画；`extent` 是一个角度，决定哪部分圆圈被绘制，缺省时表示画完整的圆；`steps` 表示画完这个圆分几步完成，常用来绘制该圆的内切正多边形。

(4) 例如：

```
import turtle
turtle.circle(60) # 绘制半径为 60 像素的圆，逆时针画圆
turtle.circle(-60) # 绘制半径为-60 像素的圆，顺时针画圆
```

绘制效果如图 4-37 和图 4-38 所示。

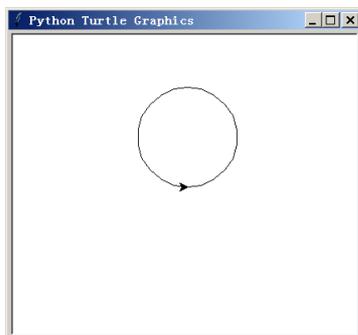


图 4-37 绘制半径为 60 的圆形

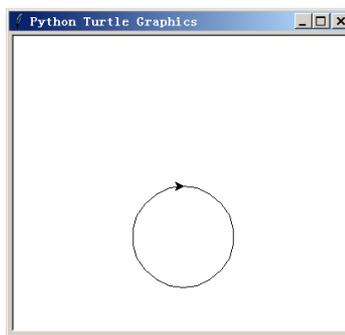


图 4-38 绘制半径为-60 的圆形



小知识



4-3 清屏 reset 与 clear

📌 操作工单

工单一 绘制正方形

1. 操作步骤

- (1) 打开 IDLE，执行“File”菜单下的“New File”命令，新建一个.py 文件。
- (2) 输入程序：

```
import turtle
turtle.pensize(3)
turtle.pencolor("red")
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
```

- (3) 保存程序。
- (4) 调试程序，并运行程序，按“F5”键或者执行“Run”→“Module”命令运行。

2. 操作结果

预期结果如图 4-39 所示，请在右侧的方框中简单绘制实际效果图。

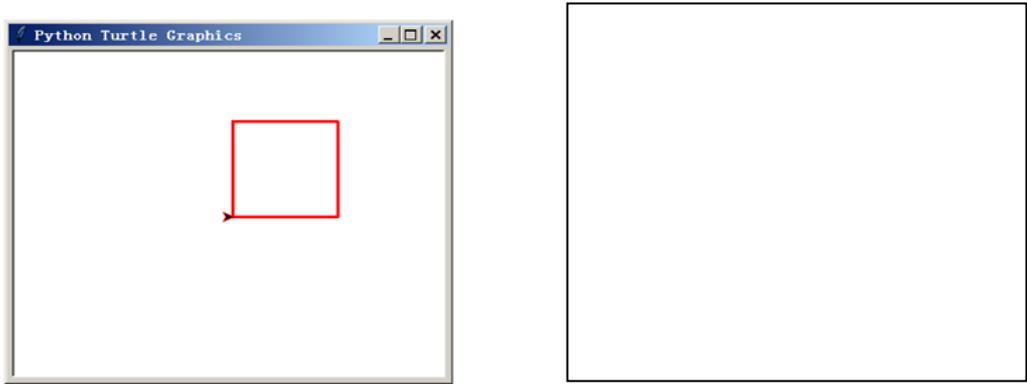


图 4-39 预期结果与实际结果

3. 学习记录

工单二 绘制长方形

1. 操作步骤

(1) 打开 IDLE，执行“File”菜单下的“New File”命令，新建一个.py 文件。

(2) 输入程序：

```
import turtle  
turtle.pensize(3)  
turtle.pencolor("red")  
turtle.forward(100)  
turtle.left(90)  
turtle.forward(150)  
turtle.left(90)  
turtle.forward(100)  
turtle.left(90)  
turtle.forward(150)  
turtle.left(90)
```

(3) 保存程序。

(4) 调试并运行程序。

2. 操作结果

预期结果如图 4-40 所示，请在右侧的方框中简单绘制实际效果图。

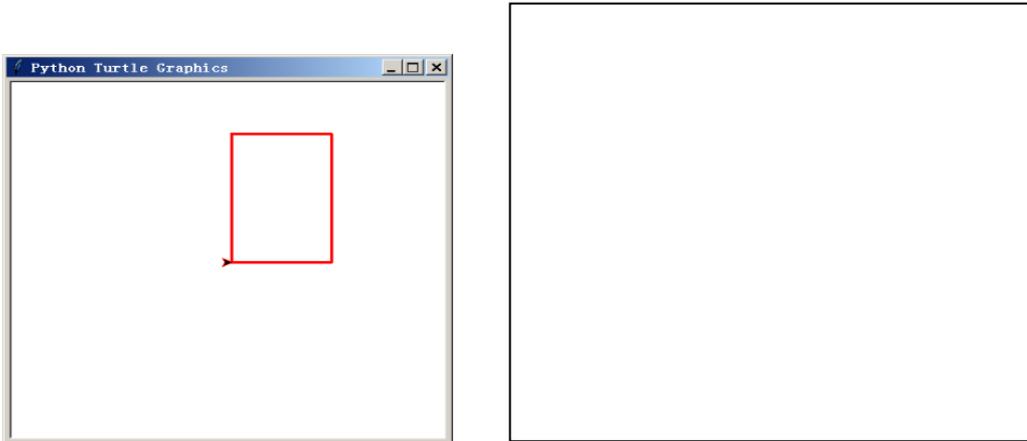


图 4-40 预期结果与实际结果

3. 学习记录

工单三 绘制圆与半圆

1. 操作步骤

(1) 打开 IDLE，执行“File”菜单下的“New File”命令，新建一个.py 文件。

(2) 输入绘制圆程序：

```
import turtle
turtle.pensize(3)
turtle.pencolor("red")
turtle.circle(80)
```

(3) 保存程序。

(4) 调试运行程序。

(5) 打开 IDLE，再新建一个.py 文件。

(6) 输入绘制半圆程序：

```
import turtle
turtle.pensize(3)
turtle.pencolor("red")
turtle.circle(80,180)
```

(7) 保存程序。

(8) 调试并运行程序。

2. 操作结果

预期结果如图 4-41 所示，请在右侧的方框中简单绘制实际效果图。

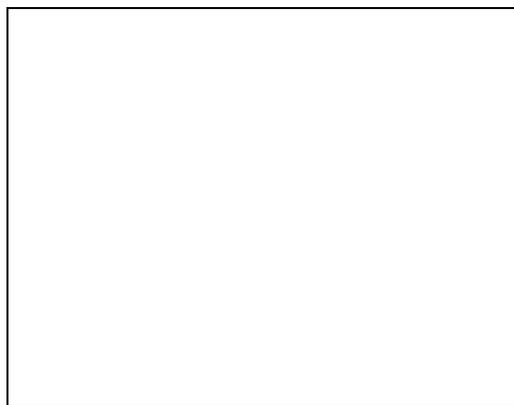
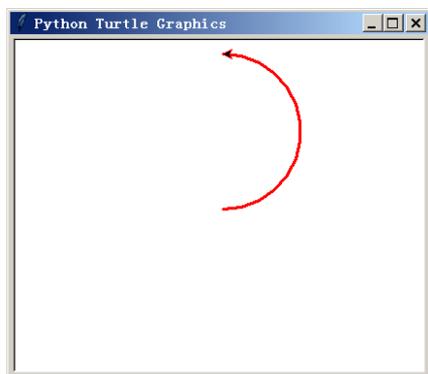


图 4-41 预期结果与实际结果

3. 学习记录

工单四 绘制三角形

1. 操作步骤

(1) 打开 IDLE，执行“File”菜单下的“New File”命令，新建一个.py 文件。

(2) 输入程序：

```
import turtle  
turtle.pensize(3)  
turtle.pencolor("red")  
turtle.forward(100)  
turtle.left(120)  
turtle.forward(100)  
turtle.left(120)  
turtle.forward(100)  
turtle.left(120)
```

(3) 保存程序。

(4) 调试并运行程序。

2. 操作结果

预期结果如图 4-42 所示，请在右侧的方框中简单绘制实际效果图。

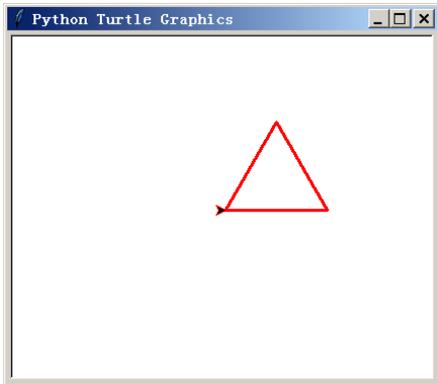


图 4-42 预期结果与实际结果

3. 学习记录

工单五 绘制五角形

1. 操作步骤

(1) 打开 IDLE，新建一个.py 文件。

(2) 输入程序：

```
import turtle
turtle.pensize(3)
turtle.pencolor("red")
turtle.forward(100)
turtle.right(144)
turtle.forward(100)
turtle.right(144)
turtle.forward(100)
turtle.right(144)
turtle.forward(100)
turtle.right(144)
turtle.forward(100)
```

(3) 保存程序。

(4) 调试并运行程序。

2. 操作结果

预期结果如图 4-43 所示，请在右侧的方框中简单绘制实际效果图。

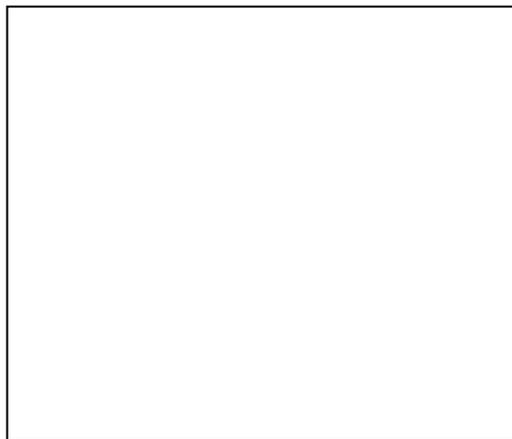
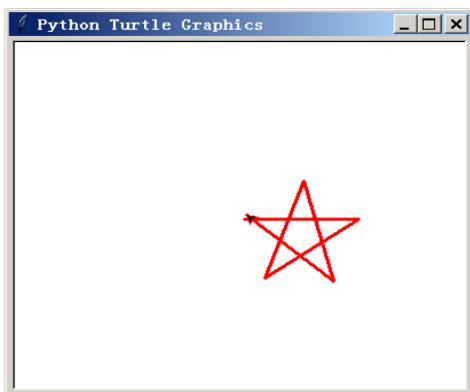


图 4-43 预期结果与实际结果

3. 学习记录

工单六 绿色小蟒蛇

1. 操作步骤

- (1) 打开 IDLE，新建一个.py 文件。
- (2) 输入程序：

```
import turtle
turtle.pensize(25)
turtle.pencolor("green")
turtle.right(90)
turtle.circle(20,180)
turtle.circle(-30,180)
turtle.circle(30,180)
turtle.circle(-30,180)
turtle.circle(30,180)
turtle.circle(-30,180)
turtle.dot(15,"yellow")
```

- (3) 保存程序。
- (4) 调试并运行程序。

2. 操作结果

预期结果如图 4-44 所示，请在右侧的方框中简单绘制实际效果图。

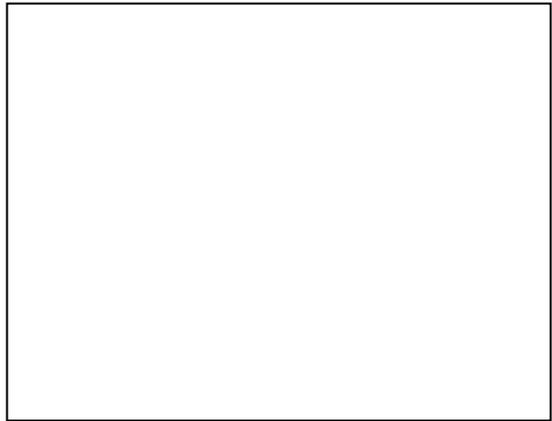
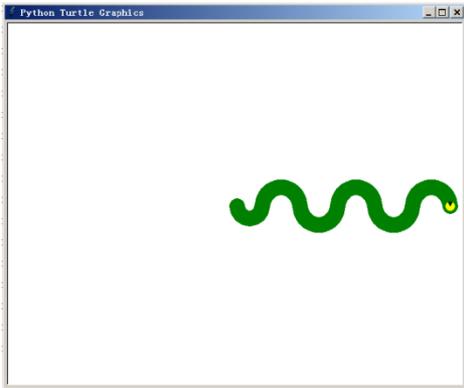


图 4-44 预期结果与实际结果

3. 学习记录

活动四 绘制复杂图形之迷幻星空

活动目标

- (1) 认知 Turtle 库中色彩的运用。
- (2) 认知 Turtle 库中画笔起落、色彩填充、绘制文本等命令。
- (3) 认知 Turtle 的程序优化方法。
- (4) 综合运用循环结构绘制棒棒糖。
- (5) 综合运用随机函数绘制彩色星空。
- (6) 实现动画绘制桃李纷飞。

活动要求

- (1) 程序设计、编辑、调试规范，代码风格良好。
- (2) 小组成员互相协作，服从小组管理。

知识储备

一、Turtle 的丰富色彩

Turtle 的颜色表达方式有使用颜色字符串和 RGB 数值两种方式，前面我们在绘制基本图形时主要使用了比较简单的颜色字符串的方式。例如，设置画笔颜色的语句：`turtle.pencolor("red")`，这种使用颜色字符串的方式需要我们输入准确的各种颜色的英文单词，对于同学们来说，掌握的常见颜色的英文单词大概就是 red、green、blue、yellow、white、black、orange、purple、pink 这些了，这个世界上五彩斑斓的颜色哪里是用这些英语单词就能够表达完全的呢！所以，在今天绘制复杂图形里面，为了表示丰富的色彩，下面我们开始学习 RGB 色彩在 Turtle 中的具体使用。

首先，我们回顾一下计算机的 RGB 色彩体系的原理，RGB 即红、绿、蓝三原色，RGB 色彩体系是将这三种颜色按照不同的比例混合出万物色，256 级的 RGB 色彩总共能组合出约 1 678 万种色彩。在 Turtle 绘图中，红、绿、蓝三种颜色的取值范围都是 0~255 之间的整数，或 0~1 中的小数，在这里我们为大家整理了一些常用颜色的中英文名称和 RGB 整数、小数表示的对照表，如表 4-2 所示，并且在附录三里面，专门为大家制作了更加丰富的色彩使用表，大家在绘图的时候可以参阅。

表 4-2 常用颜色中英文名称及 RGB 数值对照表

英文名称	RGB 整数值	RGB 小数值	中文名称
white	255,255,255	1,1,1	纯白色
black	0,0,0	0,0,0	纯黑色
red	255,0,0	1,0,0	纯红色
green	0,255,0	0,1,0	纯绿色
blue	0,0,255	0,0,1	纯蓝色

(续表)

英文名称	RGB 整数值	RGB 小数值	中文名称
yellow	255,255,0	1,1,0	纯黄色
pink	255,192,203	1,0.75,0.80	粉色
brown	165,42,42	0.65,0.16,0.16	棕色
purple	160,32,240	0.63,0.13,0.94	紫色
gold	255,215,0	1,0.84,0	金色
orange	255,165,0	1,0.64,0	橙色
cyan	0,255,255	0,1,1	青色
tomato	255,99,71	1,0.39,0.28	番茄色
grey	190,190,190	0.74,0.74,0.74	灰色
skyBlue	135,206,235	0.53,0.80,0.92	天蓝色
wheat	245,222,179	0.96,0.87,0.70	小麦色
chocolate	210,105,30	0.82,0.41,0.12	巧克力
violet	238,130,238	0.93,0.51,0.93	紫罗兰

以纯绿色为例，在 Turtle 绘图中将画笔颜色设置为纯绿色有三种方式，分别是：

- (1) 颜色单词字符串表示方式：turtle.pencolor("green")。
- (2) RGB 的小数表示方式：turtle.pencolor(0,1,0)。
- (3) RGB 的整数表示方式：turtle.pencolor(0,255,0)。

Turtle 的 RGB 色彩模式默认采用小数，如果想使用 RGB 整数表示，需要使用 turtle.colormode() 命令来调整表示方式。

1. turtle.colormode() 命令

- (1) 格式：turtle.colormode(mode)。
- (2) 功能：设置色彩模式。
- (3) 参数：mode 为 RGB 色彩表示模式，可取 1.0 或者 255，默认为 1.0，表示 RGB 小数模式，255 表示切换到 RGB 整数模式。
- (4) 例如：

```
import turtle
turtle.colormode(255)          # 将色彩表示模式切换到 RGB 整数模式
turtle.pencolor(0,255,0)      # 用 RGB 整数模式将画笔颜色设置为纯绿色
```

下面我们再来认识一个设置填充色的命令，使用 Turtle 不只可以画出彩色的线条，也可以将画出的封闭线条内的区域进行颜色的填充。

2. turtle.fillcolor() 命令

- (1) 格式：turtle.fillcolor(color)。
- (2) 功能：设置填充色。
- (3) 参数：参数 color 为填充色，取值为颜色单词字符串、RGB 小数或整数值，默认为黑色。
- (4) 例如：

```
import turtle
turtle.colormode(255)          # 将色彩表示模式切换到 RGB 整数模式
turtle.pencolor(0,255,0)      # 画笔颜色设置为纯绿色
turtle.fillcolor(255,255,0)   # 将圆形内填充为黄色
turtle.begin_fill()           # 开始填充颜色
turtle.circle(100)            # 画一个半径为 100 的圆形
turtle.end_fill()             # 结束填充颜色
```

有填充色的圆形效果如图 4-45 所示。

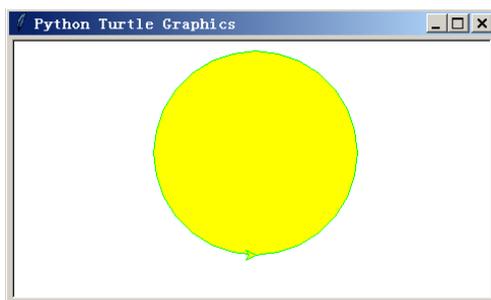


图 4-45 填充圆

注意：在为这个圆填充颜色的程序里，一句 `turtle.fillcolor()` 是不能完成填充任务的，这句代码的作用只是设置填充颜色，在设置好填充颜色后，在画填充图形之前需要使用 `begin_fill()` 以及在绘制之后需要使用 `end_fill()`，这样才能得到一个填充效果。

3. `turtle.begin_fill()` 命令

- (1) 格式：`turtle.begin_fill()`。
- (2) 功能：开始填充颜色。
- (3) 参数：无参数。

注意：这个命令要放在绘制填充图形之前使用。

4. `turtle.end_fill()` 命令

- (1) 格式：`turtle.end_fill()` 命令。
- (2) 功能：结束颜色填充。
- (3) 参数：无参数。

注意：这个命令要放在绘制填充图形之后使用。

二、Turtle 的灵活运笔

学习了怎样使用 RGB 调配出丰富的颜色，但是想要绘制出一副有质量的作品，还需要让你的画笔能够更加灵活地运用。我们在活动三里面绘制的基本图形都是连笔画，就是一幅画一笔画到底，在活动四里面，我们需要绘制更为复杂的图形，有时候需要从一个点直接画到另外一个点，有时候在这个位置需要抬笔，然后在其他位置落笔继续绘制，下面我们来学习这样几个命令。

1. `turtle.goto()` 命令

- (1) 格式：`turtle.goto(x,y)`。
- (2) 功能：画笔直接从当前位置到指定的坐标位置画一条直线。
- (3) 参数：参数 x, y 分别为目标位置的横、纵坐标值，以像素为单位。

(4) 例如:

```
import turtle
turtle.goto(100,100) # 画笔直接从原点到(100,100)坐标位置,画出一条直
turtle.goto(-100,100) # 画笔从(100,100)坐标位置到(-100,100)坐标位置画直线
```

绘制效果如图 4-46 所示。

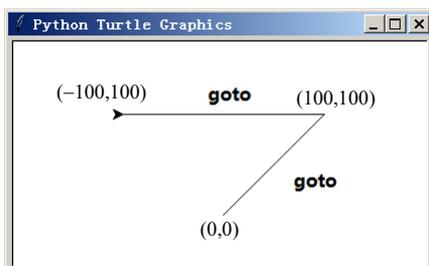


图 4-46 程序执行结果

注意: 小乌龟的朝向还是最初在 origin 位置的默认朝向, 面向 x 轴正方向没有改变。

2. turtle.penup() 命令

(1) 格式: `turtle.penup()`。

(2) 功能: 画笔抬起, 此时画笔移动不会进行绘制操作。

(3) 参数: 无参数。

3. turtle.pendown() 命令

(1) 格式: `turtle.pendown()`。

(2) 功能: 画笔落下, 此时画笔移动会进行绘制操作。

(3) 参数: 无参数。

(4) 例如:

```
import turtle
turtle.penup() # 画笔抬起
turtle.goto(100,100) # 画笔直接从原点到(100,100)坐标位置,但是没有绘制
turtle.pendown() # 画笔落下
turtle.goto(-100,100) # 画笔从(100,100)坐标位置到(-100,100)坐标位置画直线
```

绘制效果如图 4-47 所示。

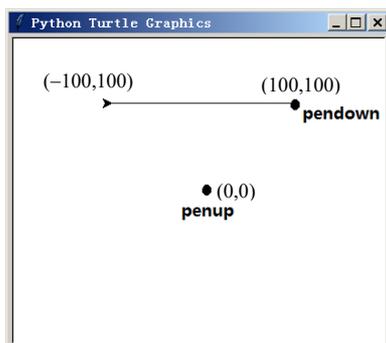


图 4-47 画笔的起落

4. turtle.home()命令

- (1) 格式: `turtle.home()`。
- (2) 功能: 画笔移至初始坐标(0,0)位置, 并设置画笔朝向为初始方向。
- (3) 参数: 无参数。
- (4) 例如:

```
import turtle
turtle.goto(100,100) # 画笔从原点到(100,100)坐标位置画直线
turtle.goto(-100,100) # 画笔从(100,100)坐标位置到(-100,100)坐标位置画直线
turtle.home() # 画笔从当前位置回家即回到初始原点位置, 朝向默认初始方向
```

绘制效果如图 4-48 所示。

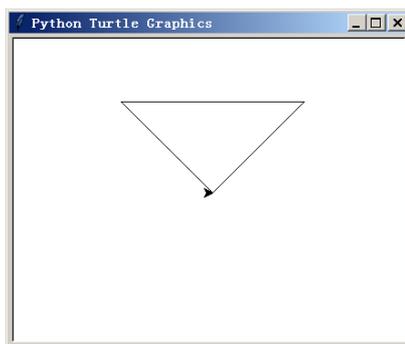


图 4-48 画笔回家

5. turtle.undo()命令

- (1) 格式: `turtle.undo()`。
- (2) 功能: 撤销最近的一个绘制作。
- (3) 参数: 无参数。
- (4) 例如:

```
import turtle
turtle.goto(100,100) # 画笔从原点到(100,100)坐标位置画直线
turtle.undo() # 撤销上一步画的直线, 画笔回到画直线之前的原点
```

绘制效果如图 4-49 所示。

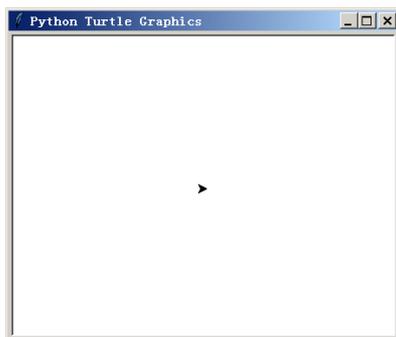


图 4-49 撤销绘制作

6. turtle.clear()命令

- (1) 格式: `turtle.clear()`。

- (2) 功能：清屏，保持画笔位置和方向不变。
- (3) 参数：无参数。

7. turtle.write()命令

- (1) 格式：turtle.write(s,font=("font_name",font_size,"font_type"))。
- (2) 功能：绘制文本。
- (3) 参数：s 为文本字符串，font 为字体的参数，里面分别是字形、字号和类型。
- (4) 例如：

```
import turtle
turtle.write("中国加油",font=("楷体",16,"normal")) # 绘制中国加油四个字
```

绘制效果如图 4-50 所示。

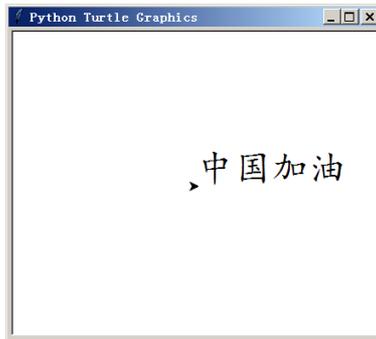


图 4-50 绘制文字

8. turtle.bgpic()命令

- (1) 格式：turtle.bgpic(picname)。
- (2) 功能：设置背景图片，只支持 gif 格式。
- (3) 参数：picname 为背景图片的文件名字符串。
- (4) 例如：

```
import turtle
turtle.bgpic("d:\\beijing.gif") # 将D盘下“beijing.gif”图片设置为画布背景
```

绘制效果如图 4-51 所示。



图 4-51 设置背景图片

三、Turtle 的代码优化

在程序世界里，代码优化是一个很重要的课题，它决定了你的程序是否具有好的代码风格，你是否具有良好的编程修养。代码优化是指对程序代码进行等价（指不改变程序的运行结果）变换。代码优化的目标是为了减小代码体积、提高代码运行效率。可能有些人觉得没用，一些细小的地方有什么好修改的，只要能运行出想要的结果，代码写成什么样又有什么影响呢？其实不然，几行代码的优化可能不能引起你的重视，如果是一千行、一万行、几千万行的程序呢？这样的影响和效率就很可观了。所以大家要在每一个程序里面去培养良好的代码风格和编程素养，下面我们来说一说 Turtle 的代码优化。

1. 使用重命名的方式优化

使用 Python 语言编写程序时，我们最常用的导入 Turtle 模块的方式是使用 `import` 关键字进行导入：

```
import turtle
```

以这种常规方式导入 Turtle 模块后，程序语句的格式为标准的 `turtle.函数名()`，程序在书写时是这样的：

```
import turtle          # 导入 turtle 库
turtle.pencolor("red") # 画笔颜色设置为红色
turtle.circle(100)    # 画一个半径为 100 的圆形
```

我们可以看到，上面这种 `import` 导入方式的程序中，第三方模块的名称“`turtle`”在每一句程序中重复出现，为了简化代码量，下面我们使用第二种导入方式：

```
from turtle import *
```

这种 `from ... import *` 的方式表示导入 Turtle 库中的所有函数，程序在书写时是这样的：

```
from turtle import * # 导入 turtle 库中所有函数
pencolor("red")     # 画笔颜色设置为红色
circle(100)         # 画一个半径为 100 的圆形
```

采用这样的方式导入后，在程序语句中直接书写函数即可，这样的好处是可以减少代码量，但是这种方式更适合只使用单一模块或者库时采用，当程序较为复杂时，可能会引用多个库，这样就不利于代码维护，更要命的是，可能还会有函数命名上的冲突。

那么，怎样才能更好地来精简程序呢？这个办法就是在导入时重命名模块，导入 Turtle 模块语句如下：

```
import turtle as t
```

这样就可以在导入 Turtle 库的时候将 Turtle 重命名为 `t`，在程序中出现 Turtle 的时候只需要输入 `t` 即可，程序书写如下：

```
from turtle as t      # 导入 Turtle 库，将 Turtle 重命名为 t
t.pencolor("red")    # 画笔颜色设置为红色
t.circle(100)        # 画一个半径为 100 的圆形
```

这种导入模块的方式既简化了程序的书写，又不会在复杂程序中引用多个模块时出现冲突。所以，在后面绘制复杂图形的程序中，建议大家在完成程序设计的基础上，能将程序进行合理、

标准的简化，形成良好的程序风格。

另外，除了导入时重命名模块，在程序中对 `turtle.Pen()` 进行重命名也可以完成同样的程序简化效果，程序书写如下：

```
from turtle                # 导入 turtle 库
t= turtle.Pen()           # 生成 turtle 画笔，重命名为 t
t.pencolor("red")        # 画笔颜色设置为红色
t.circle(100)            # 画一个半径为 100 的圆形
```

2. 使用函数简写优化

在 Turtle 中，使用频率非常高的一些函数名有更为简短的缩写形式，在程序中，无论书写原函数名还是规定的简写函数名都是正确的。当然，为了保持代码风格一致，一般在一个程序里统一使用原函数名或者统一使用简写函数名，甚至，每位程序员在自己的工作过程中，应养成其中一种统一的代码书写风格。下面列举一些可以简写的 Turtle 函数，如表 4-3 所示。

表 4-3 可简写的 Turtle 函数举例

函 数	简 写	功 能
<code>pendown()</code>	<code>pd()/down()</code>	落笔
<code>penup()</code>	<code>pu()/up()</code>	提画
<code>forward()</code>	<code>fd()</code>	前进
<code>backward()</code>	<code>bk()/back()</code>	后退
<code>right()</code>	<code>rt()</code>	右转
<code>left()</code>	<code>lt()</code>	左转

3. 使用循环等结构方法优化

循环结构是指在程序中需要反复执行某个功能而设置的一种程序结构，因此循环结构可以很好地减少程序重复书写的工作量，这是程序设计中最能发挥计算机特长的程序结构。

下面我们以绘制正方形为例，来对比优化前后的程序代码。在正方形的绘制中，直走 100 然后左转 90° 这个操作连续重复了 4 次，如果使用循环结构将这个操作循环 4 次即可画出正方形，再运用重命名和函数简写这些代码优化方式，效果如下面的代码所示，图 4-52 为优化前的程序代码，图 4-53 为优化后的程序代码。

```
import turtle
turtle.pensize(3)
turtle.pencolor("red")
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
```

图 4-52 简化前的程序代码

```
import turtle
t=turtle.Pen()
t.pensize(3)
t.pencolor("red")
for i in range(4):
    t.fd(100)
    t.lt(90)
```

图 4-53 简化后的程序代码

四、Turtle 常用函数查询

在任务四 Python 之绘画中，我们学习了 Turtle 库中的常用函数，在前面的具体讲解中，我们称这些函数为命令，是为了配合沙滩上的这只小海龟的各种动作，让大家感受到程序的执行力。因为篇幅的问题，有些函数没有做具体介绍，下面我们把这些常用的函数汇总到表 4-4 中，方便大家在创作和绘制图形时使用。

表 4-4 常用的 Turtle 函数

函 数	描 述
pen()	定义画笔
pendown()	放下画笔
penup()	提起画笔
pensize()	设置画笔粗细
pencolor()	设置画笔的颜色
fillcolor()	设置填充颜色
color()	设置画笔颜色和填充颜色
begin_fill()	开始填充图形
end_fill()	填充图形结束
filling()	返回填充的状态，True 为填充，False 为未填充
bgcolor()	设置画布背景色
colormode()	设置色彩模式
title()	设置画布屏幕标题
clear()	清空当前窗口，但不改变当前画笔的位置
reset()	清空当前窗口，并重置位置等状态为默认值
screensize()	设置画布的长和宽
shape()	设置画笔形状
hideturtle()	隐藏画笔形状
showturtle()	显示画笔形状
isvisible()	如果 turtle 可见，则返回 True
write()	绘制文本
forward()	沿着当前方向前进指定距离
backward()	沿着当前相反方向后退指定距离
right()	向右旋转 angle 角度
left()	向左旋转 angle 角度
goto()	移动到指定位置处
setx()	将当前 x 轴移动到指定位置
sety()	将当前 y 轴移动到指定位置
setheading()	设置当前画笔朝向
home()	设置当前画笔位置为原点，朝向东。
circle()	绘制圆或弧形
dot()	绘制圆点

3. 学习记录

工单二 绘制彩虹棒棒糖

1. 操作步骤

(1) 打开 IDLE，新建一个.py 文件。

(2) 输入程序：

```
import turtle
turtle.pensize(15)
turtle.penup()
turtle.goto(0,100)
turtle.pendown()
colors = ["red","orange", "yellow", "green","cyan", "blue", "purple"]

for i in range(7):
    turtle.pencolor(colors[i])
    turtle.circle((i+1)*15)
    turtle.penup()
    turtle.goto(0,100-(i+1)*15)
    turtle.pendown()
turtle.right(90)
turtle.pencolor("black")
turtle.goto(0,-300)
```

(3) 保存程序。

(4) 调试程序并运行程序。

2. 操作结果

预期结果如图 4-55 所示，请在右侧的方框中简单绘制实际效果图。

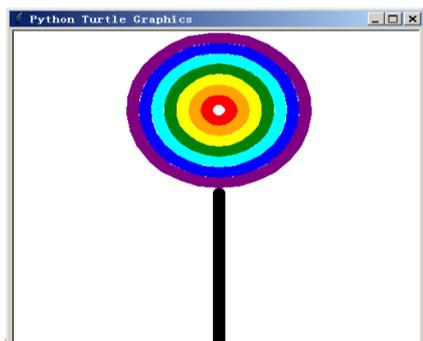


图 4-55 预期结果与实际结果

3. 学习记录

工单三 绘制迷幻星空

1. 操作步骤

(1) 打开 IDLE，新建一个.py 文件。

(2) 输入程序：

```
import turtle
import random
star=int(input("你想要几颗星星: "))
turtle.colormode(255)
turtle.bgcolor(25,25,112)
turtle.hideturtle( )
turtle.speed(0)
for i in range(star):
    r=random.randint(0,255)
    g=random.randint(0,255)
    b=random.randint(0,255)
    turtle.pencolor(r,g,b)
    x=random.randint(-300,300)
    y=random.randint(-200,200)
    turtle.penup()
    turtle.goto(x,y)
    turtle.pendown()
    turtle.dot(2)
```

(3) 保存程序。

(4) 调试程序并运行程序。

2. 操作结果

预期结果如图 4-56 所示，请在右侧的方框中简单绘制实际效果图。

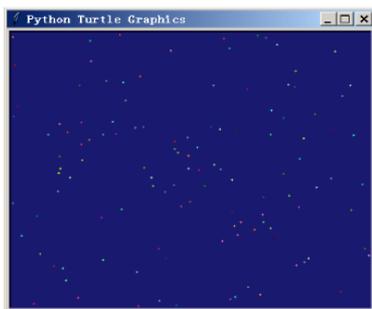


图 4-56 预期结果与实际结果

3. 学习记录

工单四 绘制桃李纷飞

1. 操作步骤

(1) 打开 IDLE，新建一个.py 文件。

(2) 输入程序：

```
import turtle
import random
from time import sleep
turtle.colormode(255)
turtle.bgcolor(0,0,0)
turtle.hideturtle()
turtle.speed(0)
for i in range(15):
    x=random.randint(-292,292)
    for y in range(7):
        turtle.clear()
        turtle.bgpic("f:\\beijing1.gif")
        turtle.penup()
        turtle.goto(x,-y*50)
        turtle.setheading(0)
        turtle.pendown()
        turtle.pencolor(255,192,203)
        turtle.fillcolor(255,192,203)
        turtle.begin_fill()
        turtle.circle(-8,275)
        turtle.forward(8)
        turtle.right(90)
        turtle.forward(8)
        turtle.end_fill()
        sleep(0.2)
```

(3) 保存程序。

(4) 调试程序并运行程序。

2. 操作结果

预期结果如图 4-57 所示，请在右侧的方框中简单绘制实际效果图。

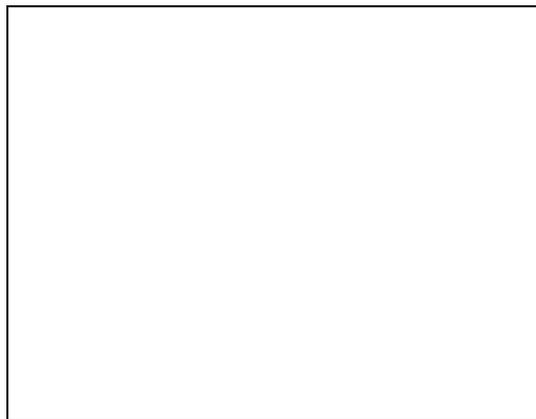
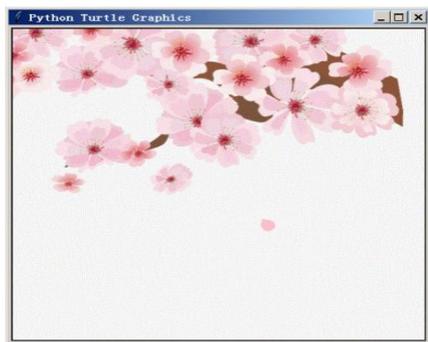


图 4-57 预期结果与实际结果

3. 学习记录

活动五 个人创新作业

📍 活动目标

- (1) 完成个人创新作品的绘制设计。
- (2) 完成个人创新作品的程序设计。
- (3) 调试并成功运行。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的任务。
- (3) 作品设计有创新。
- (4) 程序设计、编辑、调试规范，代码风格良好。
- (5) 小组成员互相协作，服从小组管理。

3. 程序运行及调试记录

(1)

(2)

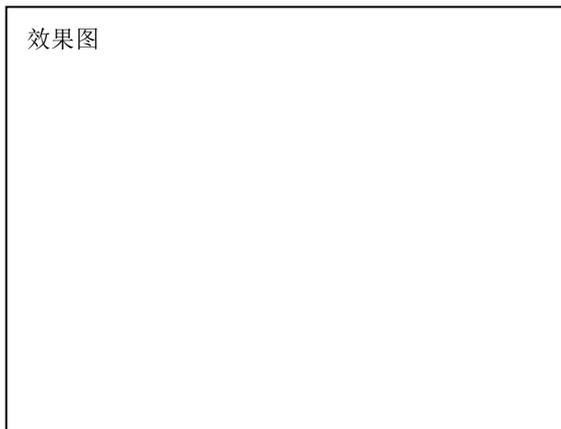
(3)

(4)

4. 作品展示

作品特点：

效果图



5. 心得总结

任务五

Python 之游戏 ——面向对象思维养成



任务目标

- (1) 认知面向对象思维方式和工作方法。
- (2) 熟练使用 Pygame 第三方模块功能。
- (3) 团队游戏项目的设计与实施。
- (4) 培养学生思维方式、工作方法和创新能力等综合职业能力。



任务要点

- (1) 面向对象思维与 Pygame 模块。
- (2) Pygame 的安装，对象和方法的使用。
- (3) 2D 打字母游戏的设计制作。
- (4) 创新游戏设计与制作。



信息素养目标

- (1) 职业素养：
 - ① 培养积极、有责任心的工作态度。
 - ② 培养面向对象思维方式和工作方法。
 - ③ 培养分析问题、解决问题的能力。
 - ④ 培养小组协作互助成长的职业能力。
- (2) 技能素养：

培养规范化的程序设计、编辑和调试技能。
- (3) 创新能力：

培养运用信息技术独立解决问题的能力。



任务内容

- 活动一：打字母游戏设计与团队组建。
- 活动二：开始界面的设计与实现。
- 活动三：游戏界面的设计与实现。
- 活动四：结束界面的设计与实现。

活动一 打字母游戏设计与团队组建

📍 活动目标

- (1) 认知面向对象思维方式。
- (2) 掌握 Pygame 库的安装。
- (3) 打字母游戏功能设计。
- (4) 项目团队组建。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的内容。
- (3) 小组成员互相沟通，服从小组管理。

📍 知识储备

一、面向对象思维

传统的结构化设计方法的基本点是面向过程，系统被分解成若干个过程，通过设计一个算法就可以解决问题。随着计算机技术的不断提高，计算机被用于解决越来越复杂的问题。结构化开发思路逐渐暴露出自己的弱点。面对问题规模的日趋扩大、环境的日趋复杂、需求变化的日趋加快，将利用计算机解决问题的基本方法统一到人类解决问题的习惯方法之上，彻底改变软件设计方法迫在眉睫，因此提出了面向对象的设计方法和思维模式。

面向对象编程（object oriented programming, OOP）的主要思想是把构成问题的各个事务分解成各个对象，建立对象的目的是为了完成一个步骤，而是为了描述一个事物在整个解决问题的步骤中的行为。一切事物皆对象，通过面向对象的方式，将现实世界的事物抽象成对象，将现实世界中的关系抽象成类、继承，帮助人们实现对现实世界的抽象与数字建模。通过面向对象的方法，更利于用人理解的方式对复杂系统进行分析、设计与编程。面向对象程序设计思维是尽可能地在模拟人类的思维。通过这样的方式能够让开发的软件更加符合人类的认知，人们使用起来也能够更加顺手。并且通过应用软件能够切实地解决现实生活中的问题。面向对象程序设计使得描述问题的问题空间和问题的解决方法空间组合在一起，并且尽可能地保持一致。能够将客观世界中的抽象问题转化为具体的问题对象。

面向对象是一种思维方式，也是一门程序设计技术。下面以大家都熟悉的植物大战僵尸游戏为实例介绍面向对象思维，如图 5-1 所示。

在植物大战僵尸游戏中，向日葵就是一个对象，当我们把生产阳光和摇晃这两件事情封装到向日葵这个对象里面之后，生产阳光和摇晃就是向日葵的属性和方法，而我们的主程序只需要把向日葵往地上一摆，它就具有了自己的特点，并可以完成相应的动作了。其他的对象如僵尸、射手也是这个道理，使用面向对象开发，主程序的功能就大大简化了。

面向对象的思维更符合人认知和思考问题的方式，面向对象的概念和应用已经超越了软件开发和程序设计，扩展到了多个领域。



图 5-1 植物大战僵尸游戏

二、Pygame 游戏库

Pygame 是用来开发游戏的 Python 模块集合，它是在优秀的 SDL 库之上开发的功能性包。使用 Python 可以导入 Pygame 来开发具有全部特性的游戏和多媒体程序，Pygame 是极度轻便的，并且可以运行在几乎所有的平台和操作系统上，用它来开发小游戏非常适合。

三、Pygame 库的安装

1. Pygame 安装文件下载

打开 PyPI 官方网站：<https://pypi.org/>，如图 5-2 所示，在网站的搜索栏中输入我们需要寻找的 Pygame，然后单击“搜索”按钮。搜索结果如图 5-3 所示。

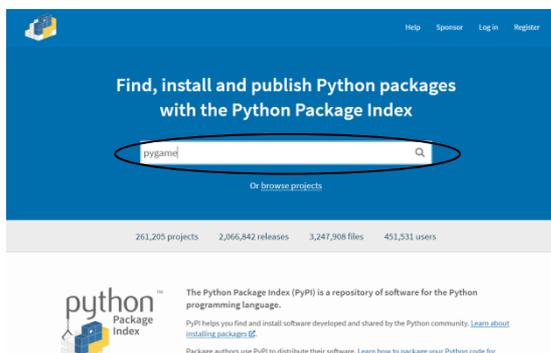


图 5-2 PyPI 官方网站

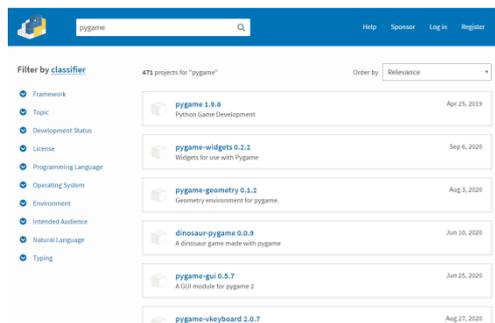


图 5-3 搜索结果

在 Pygame 库搜索结果中选择“pygame 1.9.6”版本，然后单击即可进入“pygame 1.9.6”下载界面，接着单击“Download Files”按钮，在界面右侧出现的各种安装程序中，选择“cp38”，即与我们安装的 python3.8 对应，然后选择 32 位或者 64 位，因此我们选择下载“pygame-1.9.6-cp38-cp38-win_amd64.whl”文件，如图 5-4 所示。

将“pygame-1.9.6-cp38-cp38-win_amd64.whl”文件下载到 D 盘或者计算机中的其他位置。

2. 安装 Pygame 库

Python 的第三方模块（库）的安装一般都可以使用 pip 来安装，非常的方便。首先通过输入 cmd 命令进入命令提示符窗口，在提示符后面输入 pip 安装命令“pip install d:\pygame1.9.6-cp38-

cp38-win_amd64.whl”之后按“Enter”键执行命令，进行安装，并提示安装成功“Successfully installed pygame-1.9.6”，如图 5-5 所示。

pygame-1.9.6-cp36-cp36m-win32.whl (4.0 MB)	Wheel	cp36	Apr 25, 2019	View
pygame-1.9.6-cp36-cp36m-win_amd64.whl (4.3 MB)	Wheel	cp36	Apr 25, 2019	View
pygame-1.9.6-cp37-cp37m-macosx_10_11_intel.whl (4.9 MB)	Wheel	cp37	Apr 25, 2019	View
pygame-1.9.6-cp37-cp37m-manylinux1_i686.whl (11.0 MB)	Wheel	cp37	Apr 25, 2019	View
pygame-1.9.6-cp37-cp37m-manylinux1_x86_64.whl (11.4 MB)	Wheel	cp37	Apr 25, 2019	View
pygame-1.9.6-cp37-cp37m-win32.whl (4.0 MB)	Wheel	cp37	Apr 25, 2019	View
pygame-1.9.6-cp37-cp37m-win_amd64.whl (4.3 MB)	Wheel	cp37	Apr 25, 2019	View
pygame-1.9.6-cp38-cp38-win32.whl (4.4 MB)	Wheel	cp38	Nov 11, 2019	View
pygame-1.9.6-cp38-cp38-win_amd64.whl (4.8 MB)	Wheel	cp38	Nov 11, 2019	View
pygame-1.9.6.tar.gz (3.2 MB)	Source	None	Apr 25, 2019	View

图 5-4 pygame-1.9.6 下载界面

```

Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\>pip install d:\pygame-1.9.6-cp38-cp38-win_amd64.whl
Processing d:\pygame-1.9.6-cp38-cp38-win_amd64.whl
Installing collected packages: pygame
Successfully installed pygame-1.9.6

C:\Users\>
    
```

图 5-5 安装成功



小知识



5-1 随机函数大显神通

📌 操作工单

工单 团队组建与项目计划

1. 项目团队组建

(1) 团队名称: _____

(2) 团队口号: _____

(3) 团队成员: (3~4 人) _____

(4) 团队分工: _____

① 项目组长 (1 人): _____

职责: 负责项目团队管理、项目任务分工、项目制作进度管理。

② 程序员 (3 人): _____

职责: 负责游戏程序设计与调试。

③ 美工与音效 (2 人): _____

职责: 负责游戏项目图片处理与声音效果。

④ 测试 (1 人): _____

职责: 负责游戏试玩, 找出 bug。

2. 游戏项目设计

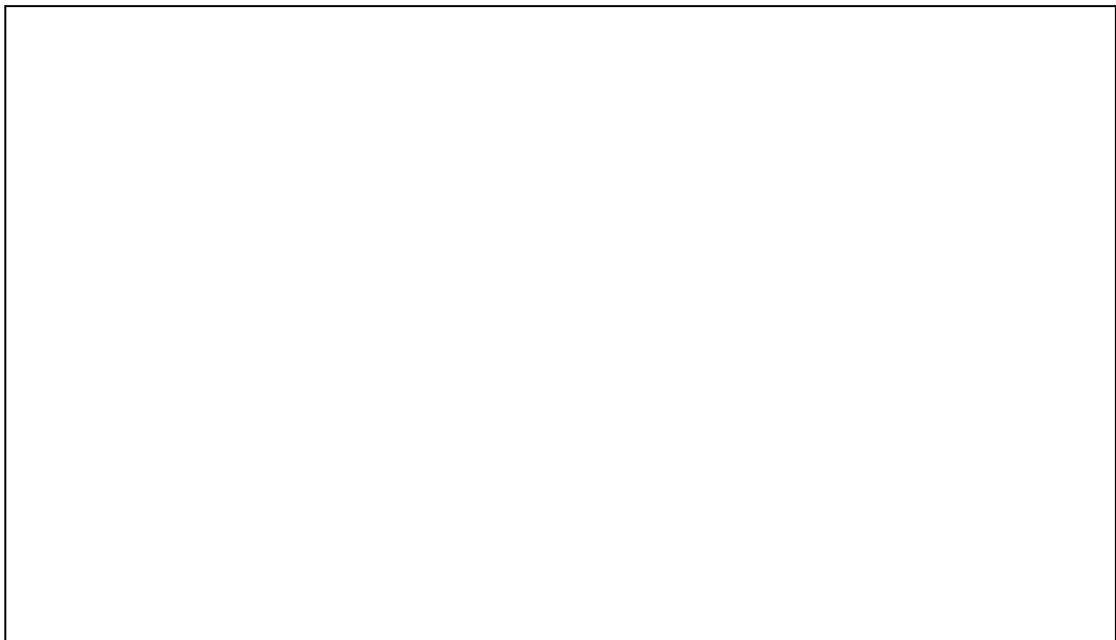
(1) 游戏题目: _____

(2) 设计思路:

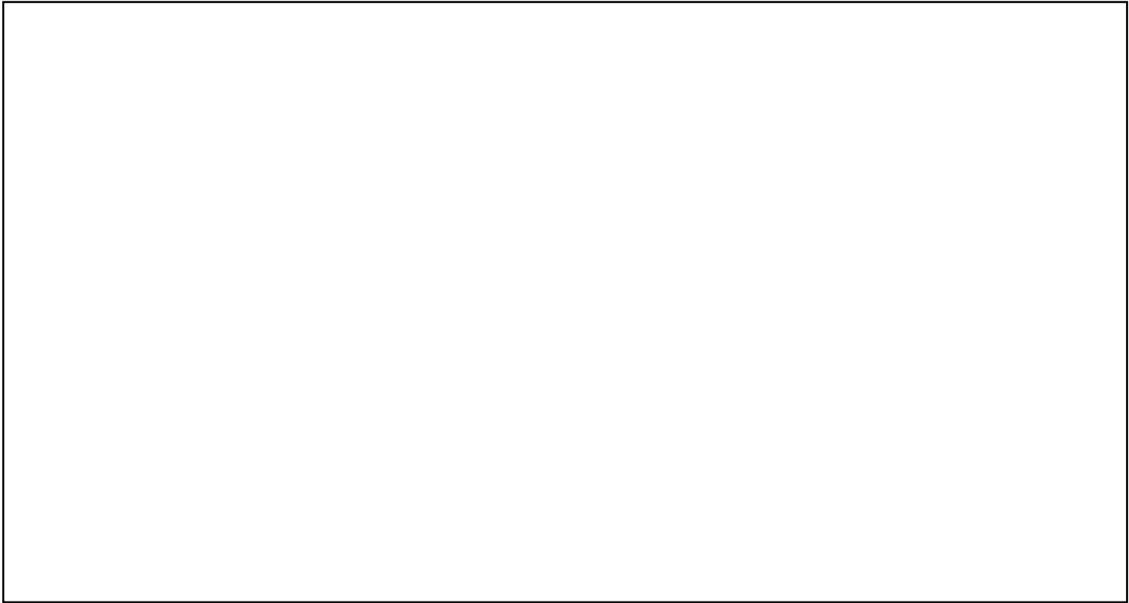
(3) 实现功能:

(4) 界面设计。

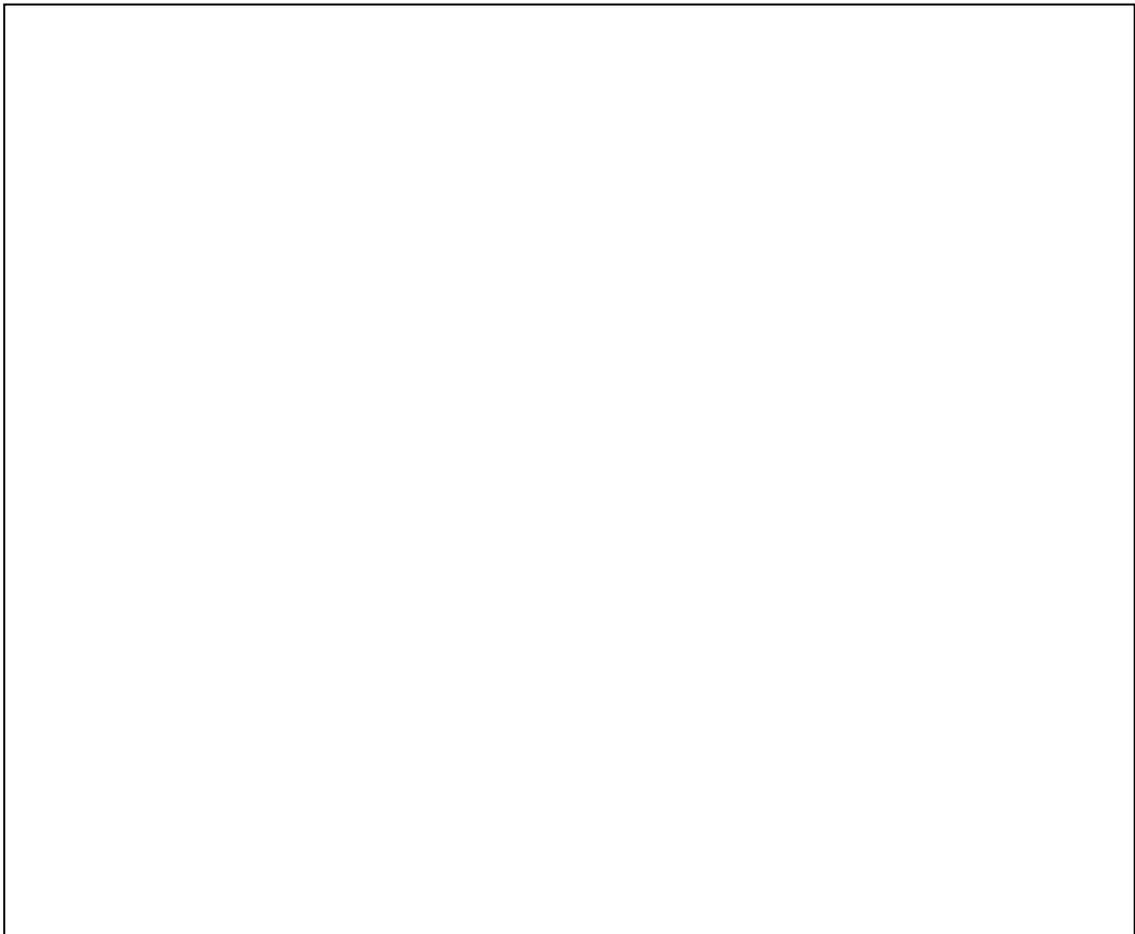
开始界面设计图:



游戏界面设计图：



结束界面设计图：



3. 项目进度安排

(1) 第一周: _____

(2) 第二周: _____

(3) 第三周: _____

(4) 第四周: _____

(5) 第五周: _____

4. 小组成员签名寄语

活动二 开始界面的设计与实现

📍 活动目标

- (1) 熟练导入和初始化 Pygame。
- (2) 运用 `pygame.display` 模块创建窗口。
- (3) 运用 `pygame.image` 模块设置背景。
- (4) 运用 `pygame.font` 模块显示文字。
- (5) 运用 `pygame.music` 模块播放音乐。
- (6) 完成打字母游戏首界面程序设计。

📍 活动要求

- (1) 程序设计、编辑、调试规范。
- (2) 小组成员互相协作，服从小组管理。

📍 知识储备

在活动一中，我们完成了 Pygame 游戏库的安装，今天的活动二中，我们将开始学习 Pygame 游戏库的常用模块和功能，并完成打字母游戏的首界面的制作。我们在开始界面中设计了创建首界面窗口和标题、设置窗口背景颜色或图片、播放背景音乐、显示游戏题目和说明以及“开始游戏”提示等几项功能，下面我们先来学习相关的技术知识。

一、import 导入 Pygame 库

1. 导入 Pygame 库

在 Python 中，Pygame 游戏库和前面的 Turtle 绘图库一样，需要先将 Pygame 库导入，导入语句如下：

```
import pygame
```

2. 导入 Pygame 所有常量

如果需要引入 Pygame 中的所有常量，则导入语句如下：

```
from pygame.locals import *
```

3. 初始化 Pygame

导入 Pygame 库之后，还需要经过 Pygame 的初始化，才能开始使用 Pygame。初始化语句如下：

```
pygame.init()
```

二、pygame.display 创建窗口

`display` 是 Pygame 中用来控制窗口和屏幕显示的模块，这个模块提供控制 Pygame 显示界面

的各种命令。制作游戏项目，我们首先需要创建一个界面窗口。

1. `pygame.display.set_mode()`命令

- (1) 功能：创建一个准备显示的窗口或屏幕，即 `pygame` 的 `Surface` 对象。
- (2) 格式：`pygame.display.set_mode((width, height), flag, depth)`。
- (3) 参数：参数 `width`、`height` 分别为创建屏幕的宽度和高度，`flag`、`depth` 一般缺省。
- (4) 例如：

```
import pygame
pygame.init()
pygame.display.set_mode((800, 600)) # 创建一个 800*600 大小的屏幕
```

输入上面程序后，大家观察运行结果，会发现并没有出现一个新创建的窗口或屏幕出来，这是为什么呢？原来，`pygame.display.set_mode()`命令只是创建了一个准备显示的窗口或屏幕，并不能把这个窗口或屏幕显示出来，因此，下一步，我们需要使用 `pygame.display.flip()`将创建好的窗口或屏幕完整地显示到屏幕上。

2. `pygame.display.flip()`命令

- (1) 功能：更新整个显示界面的内容。
- (2) 格式：`pygame.display.flip()`。
- (3) 参数：无参数。
- (4) 例如：

```
import pygame
pygame.init()
pygame.display.set_mode((800, 600)) # 创建一个 800*600 大小的屏幕
pygame.display.flip() # 更新整个显示界面的内容
```

运行上面这个程序，结果如图 5-6 所示。

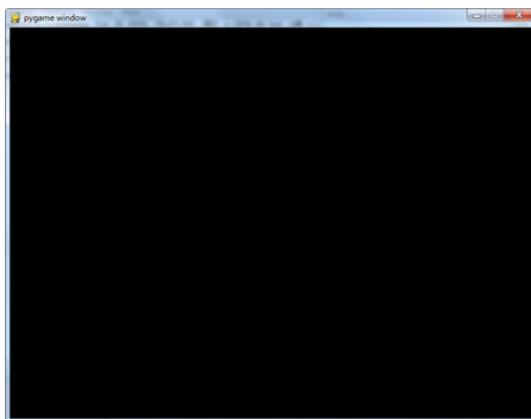


图 5-6 创建 800×600 大小的屏幕

3. `pygame.display.update()`命令

- (1) 功能：更新部分屏幕显示。
- (2) 格式：`pygame.display.update()`。
- (3) 参数：无参数。
- (4) 例如：

```
import pygame
pygame.init()
pygame.display.set_mode((800, 600))    # 创建一个 800*600 大小的屏幕
pygame.display.update()                # 更新屏幕
```

运行上面这个程序，结果如图 5-6 所示一样。`pygame.display.update()` 可以看作是 `pygame.display.flip()` 的优化版，它允许更新屏幕的部分内容，而不必完全更新。`pygame.display.flip()` 是重新更新整个屏幕对应的窗口，`pygame.display.update()` 仅仅更新窗口中有变化的区域。例如，在游戏中如果仅仅是几个物体在移动，那么只需要更新重绘其中移动的部分，没有变化的部分，并不进行重绘，这种情况下 `update` 比 `flip` 速度更快。在一般的游戏中，如果不是场景变化非常频繁的时候，建议使用 `update` 函数，而不是 `flip` 函数。

4. `pygame.display.set_caption()` 命令

- (1) 功能：设置当前窗口标题。
- (2) 格式：`pygame.display.set_caption(titlestring)`。
- (3) 参数：参数 `titlestring` 为标题字符串。
- (4) 例如：

```
import pygame
pygame.init()
pygame.display.set_mode((800, 600))    # 创建一个 800*600 大小的屏幕
pygame.display.set_caption("你好 pygame") # 设置窗口标题
pygame.display.update()                # 更新屏幕
```

运行上面这个程序，窗口标题设置为“你好 pygame”，如图 5-7 所示。

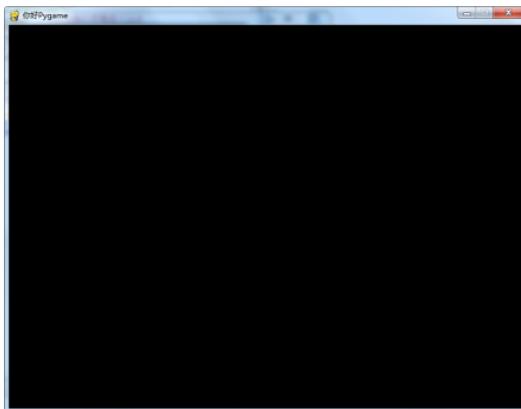


图 5-7 设置窗口标题

三、设置背景颜色或图片

1. `screen.fill()` 命令

- (1) 功能：设置窗口背景色。
- (2) 格式：`screen.fill()`。
- (3) 参数：参数是窗口的背景色，参数值是 RGB 255 三元组。
- (4) 例如：

```
import pygame
pygame.init()
```

```

screen=pygame.display.set_mode((800, 600))    # 创建一个 screen 屏幕对象
screen.fill((0,0,255))                        # 设置窗口背景为蓝色
pygame.display.update()                       # 更新屏幕

```

2. pygame.image.load()命令

- (1) 功能：加载图片。
- (2) 格式：pygame.image.load("filename")。
- (3) 参数：filename 为图片文件名或者图片文件路径，图片格式一般为 JPG、PNG 或 BMP。
- (4) 例如：

```

pygame.image.load("开始背景图.jpg")          # 加载窗口背景图片

```

大家注意：pygame.image.load()命令只是将图片进行了读取和加载，同时创建了一个新的 Surface 对象，但是它并没有绘制和显示这个背景图的功能，我们还需要使用 blit()命令将加载的图片绘制到屏幕上，下面我们继续学习这个命令。

3. screen.blit()命令

- (1) 功能：绘制图像。
- (2) 格式：blit(image,(x,y))。
- (3) 参数：第一个参数 image 为一个 Surface 对象，第二个参数(x,y)为绘制开始左上角位置。
- (4) 例如：

```

import pygame
pygame.init()
screen=pygame.display.set_mode((600, 400))
bg=pygame.image.load("开始背景图.jpg")      # 加载背景图片
screen.blit(bg, (0,0))                       # 从屏幕左上角(0,0)位置开始绘制图像
pygame.display.update()                      # 更新屏幕

```

注意：画完以后一定记得用 update()更新一下，否则画面一片漆黑。另外，背景图片如果和程序放在一起，可以使用相对路径表示，如果不在一起，则需要使用绝对路径表示，如使用路径“D:\打字游戏\开始界面\开始背景图.jpg”下的图片设置的窗口背景图效果如图 5-8 所示。



图 5-8 设置窗口背景图片

四、pygame.mixer.music 音乐播放模块

(1) `pygame.mixer.init()`命令：初始化音频模块。

(2) `pygame.mixer.music.load('xx.mp3')`命令：载入音乐。

(3) `pygame.mixer.music.play(loops, start)`命令：播放载入的音乐。两个参数可选，分别代表重复的次数和开始播放的位置，`loops` 表示播放次数 ($n > 0$)，默认 $n=0$ 表示播放 1 次， $n=-1$ 时循环播放。

(4) `pygame.mixer.music.set_volume(value)`命令：设置播放的音量。音量 `value` 的范围为 $0.0 \sim 1.0$ 。

(5) `pygame.mixer.music.stop()`命令：停止播放。

(6) `pygame.mixer.music.pause()`命令：暂停播放。

例如：

```
import pygame
pygame.init()
screen = pygame.display.set_mode((800, 600))
pygame.mixer.init()           # 初始化混响器
pygame.mixer.music.load("d:\\背景音乐.mp3") # 加载音乐
pygame.mixer.music.set_volume(0.2)         # 设置音量
pygame.mixer.music.play(-1,0)             # 循环播放
```

运行上面这个程序，大家就可以听到美妙的音乐了。有了漂亮的背景图片和优美的背景音乐，如何把游戏的题目文字绘制到屏幕上呢？下面我们接着认识 `pygame` 中加载和绘制字体的模块 `pygame.font`，`Pygame` 没有提供直接的方式在一个现有的 `Surface` 对象上绘制文本，使用 `pygame.font` 模块绘制文本一般需要三步：首先，使用 `font.SysFont()` 命令创建一个 `Font` 对象，然后，使用 `Font.render()` 命令创建一个渲染了文本的图像（`Surface` 对象），最后，使用 `blit()` 命令将这个图像绘制到目标 `Surface` 对象上，最后别忘了使用 `pygame.display.update()` 更新屏幕，大家就可以看到游戏标题了。那么，让我们来一起学习绘制字体模块 `pygame.font` 吧！

五、绘制文本

1. `pygame.font.SysFont()`命令

(1) 功能：从系统字体库中创建一个 `Font` 对象。

(2) 格式：`pygame.font.SysFont(name,size)`。

(3) 参数：第一个参数 `name` 指定字体的名字，第二个参数 `size` 指定字体的大小。

2. `pygame.font.Font.render()`命令

(1) 功能：创建一个新的 `Surface` 对象，并在上边绘制指定的文本。

(2) 格式：`render(text, antialias, color)`。

(3) 参数：第一个参数 `text` 为要绘制的文本，第二个参数 `antialias` 为抗锯齿字体（取值 `True` 或 `Flase`），第三个参数 `color` 为文本的颜色（`RGB` 值）。

(4) 例如：

```
import pygame
pygame.init()
screen = pygame.display.set_mode((800, 600))
font = pygame.font.SysFont("youyuan",100)
```

```

# 创建一个 Font 对象, 字体幼圆, 字号 60
text_surface = font.render("打字游戏", True, (255, 0, 0))
# 创建一个新的 Surface 对象, 并在上边绘制红色打字游戏五个字
screen.blit(text_surface, (150, 100))
# 从屏幕 (100, 100) 位置起将 text_surface 对象绘制到屏幕上
pygame.display.update() # 更新屏幕

```

绘制打字游戏标题效果如图 5-9 所示。



图 5-9 绘制游戏标题



小知识



5-2 查询当前电脑字体

💡 操作工单

工单 打字游戏开始界面

1. 操作步骤

(1) 打开 IDLE, 执行“File”菜单下的“New File”命令, 新建一个“开始界面.py”文件。

(2) 输入程序:

```

import pygame # 导入 pygame
pygame.init() # 初始化 pygame
screen = pygame.display.set_mode((600, 400)) # 设置窗口大小
pygame.display.set_caption("打字游戏") # 设置窗口标题
#背景音效
pygame.mixer.init() # 初始化音频模块

```


(2) 小组成员互相协作，服从小组管理。

知识储备

在活动二中，我们完成了打字母游戏开始界面的制作，今天的活动三中，我们继续制作打字母游戏的核心部分——游戏界面。在这一部分中，除了背景颜色和音乐之外，重点设计实现随机产生字母、字母下落、判断字母击打是否正确、自定义函数和操作程序退出等几项功能，下面我们先来学习相关的技术知识。

一、主循环

在编写游戏程序时，因为考虑到游戏过程中程序需要时刻“监听”键盘或者鼠标事件，例如用户在游戏中单击“窗口关闭”按钮，那么程序就要“监听到”这个事件，并且做出相应的关闭窗口、退出程序的操作，所以我们程序核心部分代码保持持续运行，你知道应该把这些需要在游戏周期中持续运行的代码放在哪里吗？程序员的回答是“放在主循环中”。除非代码在循环中，否则（大多数情况下）它只会运行一次。几乎在所有的 Pygame 游戏程序中都会有这么一段主循环：

```
import pygame
from pygame.locals import *
import sys
while True:                                # 主循环
    for event in pygame.event.get():       # 获取事件
        if event.type == pygame.QUIT:    # 检测退出事件，判断是否执行退出
            pygame.quit()                 # 关闭 pygame
            exit()                         # 退出程序
```

`while True` 这个循环，因为 `True` 为真值，循环会一直执行，为了不形成死循环，在循环体内必定设置有退出循环的出口，例如用户单击了“关闭”按钮、按“Esc”键或者生命值耗尽、规定游戏时间用完等。使用上面的主循环代码后，窗口运行不会卡死，同时也可以通过单击窗口上的“关闭”按钮正常关闭。

首先我们先来认识一下 `pygame.event` 事件处理模块，任何时候，当用户按下按键或者操作了鼠标，Pygame 库就会创建一个 `pygame.event.Event` 对象来记录这个动作，也就是“事件”，这叫作 `Event` 的对象，就存在于 `pygame.event` 事件处理模块中。`pygame` 中的事件处理一般都是在上面我们讲的 `while True` 主循环中来完成的，如果我们想知道发生了什么事件，可以执行 `pygame.event` 模块中的 `pygame.event.get()` 命令来获取事件。

1. `pygame.event.get()` 命令

- (1) 功能：获取事件。
- (2) 格式：`pygame.event.get()`。
- (3) 参数：无参数。

Pygame 库可以处理游戏中的各种事件，在通过 `pygame.event.get()` 命令获取了发生的事件之后，我们还需要进一步地比较，根据具体事件做出相应的程序反应。`event` 对象有一个 `type` 属性，它告诉我们对象表示何种事件，`if event.type == pygame.QUIT` 这句代码就是检查事件的属性类型是否等于常量 `QUIT`，通常需要最先处理 `QUIT` 事件（在用户关闭窗口的时候会产生该事件）。程序中常见的事件是按键按下、按键释放以及鼠标移动等，Pygame 里常用的事件如表 5-1 所示。

表 5-1 Pygame 中的常用事件

序号	事件	产生途径
1	QUIT	用户单击“关闭”按钮
2	KEYDOWN	键盘被按下
3	KEYUP	键盘被放开
4	MOUSEMOTION	鼠标移动
5	MOUSEBUTTONDOWN	鼠标按下
6	MOUSEBUTTONUP	鼠标放开

2. pygame.quit()和 sys.exit()命令

如果 Event 对象是一个停止事件，就会调用 `pygame.quit()` 和 `sys.exit()` 命令。`pygame.quit()` 是 `pygame.init()` 函数的一种相反的函数，它的运行会使得 Pygame 库停止工作。通常情况下，先停止 Pygame 库，然后退出程序，所以先使用 `pygame.quit()` 命令，然后再使用 `exit()` 终止程序。

在这里需要说明的是，`pygame.locals` 模块包含了 Pygame 定义的各种常量，包括上面讲到的事件类型 `event.type` 的各种值，因此，当用到 QUIT、KEYDOWN 等事件时，你可以使用下面的这条语句将所有的 Pygame 常量导入。

```
from pygame.locals import *
```

完成导入后，在程序中我们只需输入 QUIT 就可以了，而不必输入 `pygame.locals.QUIT`。同样地，`sys.exit()` 命令是 `pygame.sys` 模块中的一个函数，事先用 `import sys` 导入 `pygame.sys` 模块，在程序中我们直接输入 `exit()` 即可表示退出程序。

二、用 event.key 判断字母击打是否正确

当我们设计打字母游戏时，通常用 `pygame.event.get()` 获取所有事件，若 `event.type == KEYDOWN`，说明这是键盘事件，再接着判断按键的种类 `event.key`，大写字母“A”至“Z”键和小写字母“a”至“z”键的键码值与字母的 ASCII 码值相同（可参见附录一 ASCII 码表），如大写字母“A”的键码值为 65，小写字母“a”的键码值为 96，如果游戏规则设置为击打字母不区分大小写，那么，判断字母击打是否正确的代码设计如下：

```
for event in pygame.event.get():
    if event.type == KEYDOWN:
        if event.key == zimu or event.key==zimu+32:
            screen.fill((0, 0, 0))
            pygame.display.update()
```

如果键盘有动作
击键是该字母的大写或小写
屏幕填充背景色，让字母消失
更新屏幕

`event.key` 的键码值除了部分可以使用 ASCII 码表示之外，还有一种更为简便的方法表示，即 `k_加键名`，例如：`K_1` 就表示“1”键的键码值，`K_a` 就表示“a”键的键码值，`K_SPACE` 就表示“Space”键的键码值，`K_ESCAPE` 就表示“Esc”键的键码值。

三、用 pygame.time.delay() 命令产生动画效果

在任务四 Turtle 绘图中，我们曾经运用过动画效果，产生动画的原理其实就是绘制→延时（让眼睛看到）→擦掉，然后产生位移，重新绘制→延时→擦掉，循环往复，由此形成了动画效果。在 Pygame 中，绘制、擦掉用前面学习的 `screen.blit()` 和 `screen.fill()` 就可以解决，下面，我们来重点学习 Pygame 的延时。

1. pygame.time.delay()命令

- (1) 功能：暂停程序一段时间。
- (2) 格式：pygame.time.delay(milliseconds)。
- (3) 参数：milliseconds 为延时的时间，单位为毫秒。

四、自定义函数提高编程效率

在前面 Python 的学习中，我们认识了函数这个概念，在很多编程语言里都使用到了函数。其实，Python 中的函数和初中的正弦函数 $\sin 30^\circ$ 的道理是一样的，都是用来封装和实现可重复使用的过程部分，都可以实现特定的功能。在 Python 等程序设计中可以通过调用函数来提高代码的复用性，提高编程效率及程序的可读性。我们在 Pygame 中学习过的很多命令，如 pygame.time.delay()、pygame.event.get() 都是 Pygame 各个模块封装好的函数，我们只需调用即可实现它的功能。当然，我们也可以根据需求进行自定义函数，例如：我们需要在屏幕上绘制 10 个圆，那么就可以将画一个圆的过程定义为函数，然后调用 10 次即可。函数除了可以提高代码的复用性，还可以提高编程效率及程序的可读性，例如，在打字母游戏的程序设计中，我们初步设计了三个界面，分别是开始界面、游戏界面和结束界面，这三个界面项目组可以分工由专人进行单独制作，这样可以使成员在项目进度中实现并行工作，最后进行链接，那么，在实际操作中，将界面封装成函数放在主循环中是一个不错的选择。

```
def 开始界面():
    .....
    return
def 游戏界面():
    .....
    return
def 结束界面():
    .....
    return
while True:
    # 主程序
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            exit()
            # 退出程序
            开始界面()
            # 调用开始界面()函数
            游戏界面()
            # 调用游戏界面()函数
            结束界面()
            # 调用结束界面()函数
```



小知识



5-3 如何做出音乐淡出效果

📌 操作工单

工单 打字母游戏界面设计

1. 操作步骤

(1) 打开 IDLE，执行“File”菜单下的“New File”命令，新建一个“游戏界面.py”文件。

(2) 输入程序：

```
import pygame,sys,time
from pygame.locals import KEYDOWN
import random
pygame.init() # 初始化pygame
screen = pygame.display.set_mode((600,400)) # 设置窗口大小
pygame.display.set_caption("打字母游戏") # 设置窗口标题
myfont = pygame.font.Font(None,60) # 设置字母的字形字号
# 自定义打字母()函数
def 打字母():
    x=random.randint(60,740) # 随机产生字母下落的位置
    zimu=random.randint(65,90) # 随机产生字母
    color=(random.randint(0,255),random.randint(0,255),random.randint(0,255)) # 随机产生字母颜色
# 从屏幕顶端绘制字母
for y in range(0,590):
    zimuImage = myfont.render(chr(zimu),True,color)
    screen.blit(zimuImage,(x,y))
    pygame.display.update()
# 击打正确,字母消失
for event in pygame.event.get():
    if event.type == KEYDOWN: # 如果键盘有动作
        if event.key == zimu or event.key==zimu+32:# 击键是该字母的大写或小写
            screen.fill((0,0,0)) # 屏幕填充背景色,让字母消失
            pygame.display.update() # 更新屏幕
            return # 击打正确,字母消失,结束函数,返回
        screen.fill((0,0,0)) # 擦掉字母
        pygame.display.update() # 更新屏幕
        pygame.time.delay(5) # 延时控制字母掉落速度
# 主程序循环
while True:
    for event in pygame.event.get(): # 获取事件
        if event.type == pygame.QUIT: # 用户关闭窗口
            pygame.quit()
            exit() # 退出程序
        打字母() # 调用打字母()函数
```

(3) 保存程序并运行程序，按“F5”键或者执行“Run”菜单中的“Module”命令运行。

2. 操作结果

预期结果如图 5-11 和图 5-12 所示。

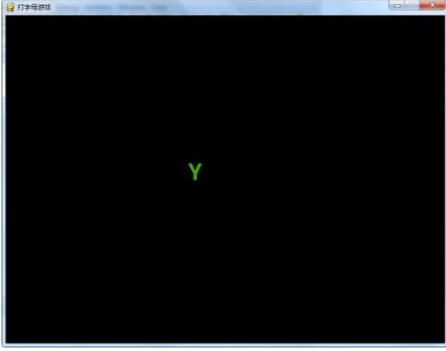
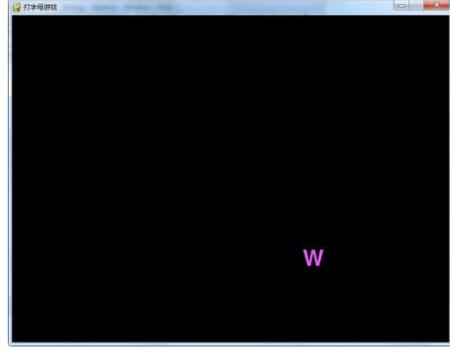


图 5-11 操作结果 (1)



5-12 操作结果 (2)

3. 学习记录

4. 本周进度完成情况

5. 改进与提高

活动四 结束界面的设计与实现

活动目标

- (1) 自定义结束界面()函数。
- (2) 运用计数器 `score` 变量获取得分并显示分数。
- (3) 综合运用主循环和三个界面函数组装打字母程序。
- (4) 综合运用 `for` 循环和字母数量控制游戏结束。
- (5) 测试程序寻找 `bug` 并修复。
- (6) 完成打字母游戏完整程序设计。

活动要求

- (1) 程序设计、编辑、调试规范。
- (2) 小组成员互相协作，服从小组管理。

知识储备

一、结束界面的制作

1. 模仿开始界面制作结束界面

```
import pygame                                # 导入 Pygame
pygame.init()                                # 初始化 Pygame
screen = pygame.display.set_mode((600, 400)) # 设置窗口大小
pygame.display.set_caption("结束界面")      # 设置窗口标题
# 背景音效
pygame.mixer.init()                          # 初始化音频模块
pygame.mixer.music.load("结束背景音乐.mp3") # 加载音乐
pygame.mixer.music.set_volume(0.2)          # 设置音量
pygame.mixer.music.play(-1,0)               # 循环播放
# 绘制背景
bg_img=pygame.image.load("结束背景图.jpg")  # 导入背景图片
screen.blit(bg_img, (0,0))                   # 绘制背景, 从(0,0)位置开始
# 绘制结束标题
font1 = pygame.font.SysFont("仿宋 gb2312", 80)
text1_surface = font1.render("GAME OVER", True, (255, 0, 0))
screen.blit(text1_surface, (120, 50))
pygame.display.update()
# 绘制得分
font2 = pygame.font.SysFont("youyuan", 60)
text2_surface = font2.render("你的得分: ", True, (0,255,0))
screen.blit(text2_surface, (120, 200))
pygame.display.update()                      # 更新屏幕
```

2. 加入得分显示

- (1) 首先，在程序最前面先定义一个得分 `score` 变量，并置初值：`score=0`。
- (2) 然后，在每次判断字母敲击正确时，得分自动增加 1 分，即 `score=score+1`。

```
if event.key == zimu or event.key==zimu+32:    # 如果敲击正确
    global score
# score 在程序最前面定义后为全局变量，在自定义函数内使用时需用 global 声明
    score+=1    # 得分自动加 1
```

(3) 最后，在结束界面中显示“你的得分”后面加入显示分值的部分，只需修改下面这句代码为：

```
text2_surface = font2.render("你的得分: %d"%score, True, (0,255,0))
```

注意：加入的 `%d` 是格式控制符，控制 `score` 的值输出显示的位置，类型应为整数。

3. 自定义结束界面函数()

结束界面的实现总体来说和开始界面基本相同，主要是完成了背景图片、背景音乐、结束标题，比开始界面多了一个显示得分。截至目前，相信大家已经分别把开始界面、打字母界面和结束界面完成了，但是这三个界面是独立的，我们最后需要把小组成员做好的这三个界面组合成为一个完整的打字母游戏程序，所以，我们一般会开始界面、结束界面这样功能相对独立的程序段封装为函数，以便于最后打字母完整游戏的实现。下面，我们将结束界面封装为结束界面()函数。

```
def 结束界面():    # 定义结束界面()函数
    screen = pygame.display.set_mode((600, 400))    # 设置窗口大小
    pygame.display.set_caption("结束界面")    # 设置窗口标题
    # 背景音效(代码略)
    # 绘制背景(代码略)
    # 绘制结束标题
    font1 = pygame.font.SysFont("仿宋 gb2312", 80)
    text1_surface = font1.render("GAME OVER", True, (255, 0, 0))
    screen.blit(text1_surface, (120, 50))
    pygame.display.update()
    # 绘制得分
    font2 = pygame.font.SysFont("youyuan", 60)
    text2_surface = font2.render("你的得分: %d"%score, True, (0,255,0))
    screen.blit(text2_surface, (120, 200))
    pygame.display.update()
    pygame.time.delay(10000)    # 延时
    return    # 函数返回
```

这样，我们就完成了结束界面的程序设计，大家可以试着将开始界面程序段也封装到开始界面()函数中。最后，就可以结构清晰地进行打字母游戏的集成组装了。

二、打字母游戏的代码集成

其实，在任务三中学习函数的时候已经提到了如何将团队成员制作的各个独立的代码段集成为一个完整的游戏程序，下面提供了一种打字母游戏较为具体的集成框架，大家可以参考。当然，根据程序设计的不同，框架也可以不同，所以大家可以继续设计出更清晰、更友好、更完善的打字母游戏代码。

```

# 各种库、模块的引入, 常量、变量和文本对象的定义
import pygame
from pygame.locals import KEYDOWN
import random
pygame.init() # 初始化 pygame
myfont = pygame.font.Font(None, 60) # 创建文本对象
number=5 # 设置字母数
score=0 # 设置用户得分
screen = pygame.display.set_mode((600, 400)) # 设置窗口大小
pygame.display.set_caption("打字母游戏") # 设置窗口标题
# 定义开始界面() 函数
def 开始界面():
    .....
    return
# 定义打字母() 函数
def 游戏界面():
    .....
    return
# 定义结束界面() 函数
def 结束界面():
    .....
    return
# 主程序
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: # 监听退出事件
            pygame.quit()
            exit() # 退出程序
        for i in range(0,number): # 产生 number 个下落字母
            产生字母() # 调用产生字母() 函数
        结束界面() # number 个字母产生完毕, 调用结束界面() 函数

```

按照上面打字母游戏的程序框架, 在活动四的工单中, 我们给大家设计了一款最基本的打字母游戏, 这款游戏只实现了打字母最简单的功能, 我们希望能尽量给大家留有更大的制作和发挥空间, 大家能在基本款的游戏基础上, 运用所学, 开动思维, 发挥想象, 将游戏进行扩展设计。例如, 在游戏中给开始界面、游戏中、结束时设置不同的背景图片和不同节奏的背景音乐, 当字母击打正确时会有成功的声音, 当满分的时候给予玩家奖励, 设置游戏关卡等, 让我们的团队继续加油, 拿出精彩的作品。

小知**识**

5-4 Pygame 中也能画图

📌 操作工单

工单 打字母游戏集成与测试

1. 操作步骤

- (1) 打开 IDLE, 执行“File”菜单下的“New File”命令, 新建一个“打字母游戏.py”文件。
- (2) 输入打字母完整程序:

```
import pygame
from pygame.locals import KEYDOWN
import random

pygame.init()                                # 初始化 Pygame
myfont = pygame.font.Font(None, 60)          # 创建文本对象
number=5                                     # 设置字母数
score=0                                       # 设置用户得分
screen = pygame.display.set_mode((600, 400)) # 设置窗口大小
pygame.display.set_caption("打字母游戏")     # 设置窗口标题
def 开始界面():                               # 定义开始界面()函数
    # 背景音效
    pygame.mixer.init()                       # 初始化音频模块
    pygame.mixer.music.load("开始音乐.mp3")  # 加载音乐
    pygame.mixer.music.set_volume(0.2)       # 设置音量
    pygame.mixer.music.play(-1,0)            # 循环播放
    # 绘制背景
    bg_img=pygame.image.load("开始背景图.jpg") # 导入背景图片
    screen.blit(bg_img, (0,0))               # 绘制背景, 从(0,0)位置开始
    # 绘制游戏标题
    font1 = pygame.font.SysFont("仿宋 gb2312", 80)
    text1_surface = font1.render("打字母游戏", True, (255, 0, 0))
    screen.blit(text1_surface, (100, 50))
    # 绘制游戏说明
    font2 = pygame.font.SysFont("youyuan", 40)
    text2_surface = font2.render("击打正确, 字母会消失!", True, (0,255,0))
    screen.blit(text2_surface, (100, 200))
    # 绘制开始游戏提示
    font3 = pygame.font.SysFont("仿宋 gb2312",30)
    text3_surface = font3.render("5 秒进入游戏", True, (0, 0,255))
    screen.blit(text3_surface, (200, 320))
    pygame.display.update()
    pygame.time.delay(5000)
def 打字母():                                 # 定义打字母()函数
    x=random.randint(60,540)                  # 产生随机字母下落位置
    zimu=random.randint(65,90)               # 产生随机字母
    color=(random.randint(0,255),random.randint(0,255),random.randint(0,255))
    for y in range(0,360):                   # 让字母下落
        zimuImage = myfont.render(chr(zimu), True, color)
        screen.blit(zimuImage, (x,y))        # 显示字母
        pygame.display.update()
    for event in pygame.event.get():
        if event.type == KEYDOWN:
            if event.key == zimu or event.key==zimu+32: # 如果敲击正确
                global score
                score+=1                       # 得分自动加 1
```

```

        screen.fill((0, 0, 0))          # 擦掉字母
        pygame.display.update()
        return

    screen.fill((0,0,0))              # 敲击不正确，字母落地消失
    pygame.display.update()
    pygame.time.delay(5)              # 延时
def 结束界面():                      # 定义结束界面()函数
    # 此处可以设置结束界面的背景图片和背景音乐
    # 绘制结束标题
    font1 = pygame.font.SysFont("仿宋 gb2312", 80)
    text1_surface = font1.render("GAME OVER", True, (255, 0, 0))
    screen.blit(text1_surface, (120, 50))
    pygame.display.update()
    # 绘制得分
    font2 = pygame.font.SysFont("youyuan", 60)
    text2_surface = font2.render("你的得分: %d"%score, True, (0,255,0))
    screen.blit(text2_surface, (120, 200))
    pygame.display.update()
    pygame.time.delay(10000)          # 延时
    pygame.quit()                    # 关闭 Pygame
    exit()                            # 退出程序
    return                             # 函数返回
while True:                           # 主循环
    开始界面()                        # 调用开始界面()函数
    for event in pygame.event.get():   # 监听退出事件
        if event.type == pygame.QUIT :
            pygame.quit()
            exit()
        # 此处可以设置游戏界面的背景图片和背景音乐
        for i in range(0,number):     # 产生 number 个下落字母
            打字母()                 # 调用打字母()函数
        结束界面()                    # number 个字母产生完毕，调用结束界面()函数

```

(3) 测试程序，找出程序中的 bug。

(4) 对程序中出现的 bug 进行定位和修改。

2. 操作结果

预期结果如图 5-13~图 5-16 所示。



图 5-13 操作结果 (1)

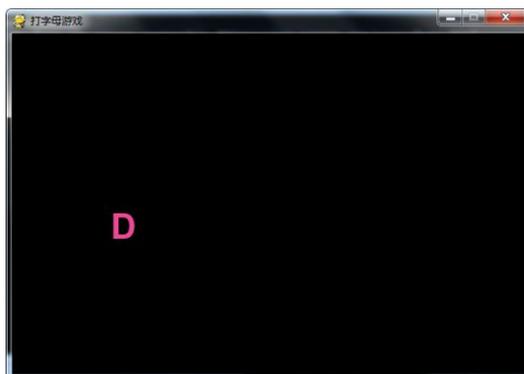


图 5-14 操作结果 (2)

(4) bug4:

原因: _____

修改: _____

(5) bug5:

原因: _____

修改: _____

5. 项目完成情况

6. 项目总结

任务六

Python 之爬虫大数据 ——大数据思维养成



任务目标

- (1) 认知爬虫技术。
- (2) 认知爬虫工具。
- (3) 掌握常见网站数据爬取方法。
- (4) 培养学生思维方式、工作方法和创新能力等综合职业能力。



任务要点

- (1) 掌握网络爬虫的含义。
- (2) 了解网络爬虫的主要类型。
- (3) 了解简单网络爬虫的结构。
- (4) 熟悉网络爬虫的应用场景。
- (5) 大数据的应用。



信息素养目标

- (1) 职业素养：
 - ① 培养积极、有责任心的工作态度。
 - ② 培养创新思维方式和工作方法。
 - ③ 培养分析问题、解决问题的能力。
 - ④ 培养小组协作互助成长的职业能力。
- (2) 技能素养：

培养在网络中爬取数据的能力。
- (3) 创新能力：

培养日常生活中需要爬取大数据的发掘能力。



任务内容

- 活动一：爬虫及爬虫工具认知。
- 活动二：爬取之遨游“百度”。

活动一 爬虫及爬虫工具认知

📍 活动目标

- (1) 认知网络爬虫的含义。
- (2) 认知网络爬虫的主要类型。
- (3) 认知网络爬虫的架构。
- (4) 了解网络爬虫的应用场景。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的内容。
- (3) 小组成员互相沟通，服从小组管理。

📍 知识储备

一、认识网络爬虫

进入信息时代，人类产生的数据爆炸式增长，每过 18 个月，全球几千年所积累的数据总量就会翻一番，这种现象称为“大数据”。大数据的两个主要来源之一就是互联网，互联网有几十亿的用户在不停地拍照、录视频、写文字、做交易、发微信、发微博、发邮件，产生了海量的数据，从互联网上获取数据的最常见方法就是使用“爬虫”程序。

1. 什么是网络爬虫

网络爬虫又称网络蜘蛛、网络蚂蚁、网络机器人等，它可以自动浏览网络中的信息，当然浏览信息的时候需要按照我们制订的规则进行，这些规则我们称之为网络爬虫算法。使用 Python 可以很方便地编写出爬虫程序，进行互联网信息的自动化检索。

搜索引擎离不开爬虫，例如百度搜索引擎的爬虫叫作百度蜘蛛（Baiduspider）。百度蜘蛛每天会在海量的互联网信息中进行爬取，爬取优质信息并收录，当用户在百度搜索引擎上检索对应关键词时，百度将对关键词进行分析处理，从收录的网页中找出相关网页，然后按照制订的排名规则进行排序并将结果展现给用户。在这个过程中，百度蜘蛛起到了至关重要的作用。

那么，如何覆盖互联网中更多的优质网页？又如何筛选这些重复的页面？这些都是由百度蜘蛛爬虫的算法决定的。采用不同的算法，爬虫的运行效率会不同，爬取结果也会有所差异。所以，我们在研究爬虫的时候，不仅要了解爬虫如何实现，还需要知道一些常见的爬虫算法，如果有必要，我们还需要自己去制订相应的算法，这些在后面都会为大家详细地讲解，在此，我们仅需要对爬虫的概念有一个基本的了解。

除了百度搜索引擎离不开爬虫以外，其他搜索引擎也离不开爬虫，它们也拥有自己的爬虫。例如，360 搜索引擎的爬虫称为 360Spider，搜狗搜索引擎的爬虫称为 Sogospider，必应搜索引擎的爬虫称为 Bingbot。

如果想自己实现一款小型的搜索引擎，我们也可以编写出自己的爬虫去实现，当然，虽然可能在性能或者算法上比不上主流的搜索引擎，但是个性化的程度会非常高，并且也有利于我们更深层次地理解搜索引擎内部的工作原理。大数据时代离不开爬虫，在进行大数据分析或数据挖掘时，我们可以去一些比较大型的官方网站下载数据源。但这些数据源比较有限，那么如何才能获取更多、更高质量的数据源呢？此时，我们可以编写自己的爬虫程序，从互联网中进行数据的获取。所以在未来，爬虫的地位会越来越重要。

2. 我们为什么要学网络爬虫

我们已经初步认识了网络爬虫，但是为什么要学习网络爬虫呢？不同的人学习爬虫，可能目的有所不同，在此，我们总结了 4 种常见的学习爬虫的原因。

(1) 学习爬虫，可以私人订制一个搜索引擎，并且可以对搜索引擎的数据采集工作原理进行更深层次地理解。有的朋友希望能够深层次地了解搜索引擎的爬虫工作原理，或者希望自己能够开发出一款私人搜索引擎，那么此时，学习爬虫是非常有必要的。简单来说，我们学会了爬虫之后，就可以利用爬虫自动地采集互联网中的信息，采集回来后进行相应的存储或处理，在需要检索某些信息的时候，只需在采集回来的信息中进行检索，即实现了私人的搜索引擎。当然，信息怎么爬取、怎么存储、怎么进行分词、怎么进行相关性计算等，都是需要我们进行设计的，爬虫技术主要解决信息爬取的问题。

(2) 大数据时代，要进行数据分析，首先要有数据源，而学习爬虫，可以让我们获取更多的数据源，并且这些数据源可以按我们的目的进行采集，能去掉很多无关数据。

在进行大数据分析或者进行数据挖掘的时候，数据源可以从某些提供数据统计的网站获得，也可以从某些文献或内部资料中获得，但是这些获得数据的方式，有时很难满足我们对数据的需求，而手动从互联网中去寻找这些数据，则耗费的精力过大。此时就可以利用爬虫技术，自动地从互联网中获取我们感兴趣的数据内容，并将这些数据内容爬取回来，作为我们的数据源，从而进行更深层次的数据分析，并获得更多有价值的信息。

(3) 对于很多网站优化（SEO）从业者来说，学习爬虫，可以更深层次地理解搜索引擎爬虫的工作原理，从而可以更好地进行搜索引擎优化。既然是搜索引擎优化，那么就必须要对搜索引擎的工作原理非常清楚，同时也需要掌握搜索引擎爬虫的工作原理，这样在进行搜索引擎优化时，才能知己知彼，百战不殆。

(4) 从就业的角度来说，爬虫工程师目前来说属于紧缺人才，并且薪资待遇普遍较高，所以深层次地掌握这门技术，对于就业来说，是非常有利的。

3. 网络爬虫的主要类型

(1) 通用网络爬虫。通用网络爬虫，又称为“全网爬虫”，它从一些种子网站开始爬行，逐步扩展到整个互联网，主要为门户网站、搜索引擎和大型 Web 服务商采集数据。

通用网络爬虫策略：深度优先策略和广度优先策略。

(2) 聚焦网络爬虫。聚焦网络爬虫，又称为“主题网络爬虫”，即预先选择一个（或几个）相关主题，仅爬行并抓取这一类的相关页面，极大地节省了硬件和网络资源，保存的页面因数量少而更新快，可以满足对特定领域信息的需求。

聚焦网络爬虫策略：聚焦网络爬虫增加了链接和内容评价模块，所以其爬虫策略的关键是评价页面的链接和内容后再进行爬虫。

(3) 增量式网络爬虫。增量式网络爬虫是指对已经收录的页面进行更新、爬取新页面和发生

变化的页面，它能够在一定程度上保证所爬取的页面是尽可能新的页面，可有效减少数据下载量，及时更新已经爬取的页面，减小时间和空间上的耗费，但是增加了爬取算法的复杂度。

增量式网络爬虫策略：广度优先策略和 PageRank 优先策略等。

(4) Deep Web 爬虫。搜索引擎蜘蛛可以爬行并抓取的页面称之为“表层网页”，某些不能通过静态链接获得的页面称为“深层网页”，Deep Web 爬虫就是抓取深层网页的爬虫体系，如用户注册后才可见的网页就属于深层页面。

4. 网络爬虫的架构

网络爬虫的两项主要任务是下载目标网页和从网页中解析信息，为了完成这两项任务，一个简单的网络爬虫就要包含爬虫调度端、URL 管理器、网页下载器、网页解析器、价值数据等五个部分，如图 6-1 所示。

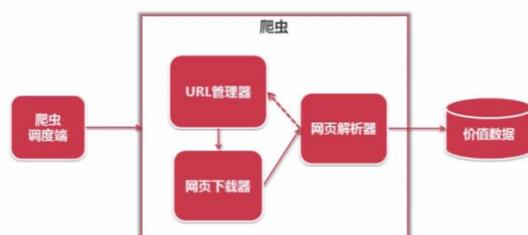


图 6-1 网络爬虫的结构示意图

爬虫调度端：用来启动爬虫、停止爬虫、监视爬虫的运行情况。

URL 管理器：对将要爬取的和已经爬取的 URL 进行管理。

网页下载器：从 URL 管理器中获取一个将要爬取的 URL，将指向的网页下载并存储为一个字符串，是爬虫的核心部分之一。

网页解析器：将网页下载器下载下来的字符串进行解析，提取有价值的信息，是爬虫的另一个核心部分。

价值数据：保存信息组件，用来把解析出来的内容输出到文件或数据库中。

另一方面，每个 URL 中往往还有指向其他页面的 URL，需要补充到 URL 管理器中，上面的逻辑就形成了一个循环，只要有相关联的 URL，就可以一直运行下去，这个简单的爬虫架构的运行，就可以将互联网上相关联的所有网页都爬取下来。

5. 网络爬虫的应用场景

随着互联网信息的爆炸，网络爬虫渐渐为人所熟知。但作为一种自动爬取网页信息的手段，很多人其实都不太清楚它在实际生活中的巨大作用。那么，网络爬虫是干什么的？有哪些应用场景呢？简单来讲，搜索引擎、统计数据、出行类软件、聚合类平台等方面，都离不开网络爬虫的使用，下面让我们来详细看看网络爬虫的应用场景和作用。

如图 6-2 所示，网络爬虫可以代替手工做很多事情，比如可以爬取网站上面的图片，并将某些网站上的图片全部爬取下来，集中进行浏览；同时，网络爬虫也可以用于金融投资领域，比如可以自动爬取一些金融信息，并进行投资分析等。有时，我们比较喜欢的新闻网站可能有几个，每次都要分别打开这些新闻网站进行浏览，比较麻烦。此时可以利用网络爬虫，将多个新闻网站中的新闻信息爬取下来，集中进行阅读。有时，在浏览网页上的信息的时候，会发现有很多广告。此时同样可以利用爬虫将对应网页上的信息爬取过来，这样就可以自动地过滤掉这些广告，

方便对信息的阅读与使用。有时，需要进行营销，那么如何找到目标客户以及目标客户的联系方式是一个关键问题。我们可以手动地在互联网中寻找，但是这样的效率会很低。此时，利用爬虫，可以设置对应的规则，自动地从互联网中采集目标用户的联系方式等数据，供我们进行营销使用。有时，我们想对某个网站的用户信息进行分析，比如分析该网站的用户活跃度、发言数、热门文章等信息，如果我们不是网站管理员，手工统计将是一个非常庞大的工程。此时，可以利用爬虫轻松地将这些数据采集到，以便进行进一步分析，而这一切爬取的操作，都是自动进行的，我们只需要编写好对应的爬虫，并设计好对应的规则即可。

除此之外，爬虫还可以实现很多强大的功能。总之，网络爬虫已经渗透进我们生活的方方面面，可以在一定程度上代替手工访问网页，从而，原先我们需要人工去访问互联网信息的操作，现在都可以用爬虫自动化实现，这样可以更高效地利用好互联网中的有效信息，给我们的生活带来极大的便利。但同时，如果不加以规范，反过来又会侵害我们的利益。因此，网络爬虫是一把双刃剑，就看我们如何使用！



图 6-2 网络爬虫的技能总览

二、反爬虫和反爬虫策略

简单的爬虫在抓取网站数据的时候，因为对网站访问过于频繁，会给服务器造成过大的压力，容易使网站崩溃，因此网站维护者会通过一些手段避免爬虫的访问，以下是几种常见的爬虫和反爬虫策略如表 6-1 所示。

表 6-1 爬虫和反爬虫策略

	爬 虫	反 爬 虫
应对策略	对网站发送请求，获取数据	监控发现某段时间访问陡增，IP 相同，user-agent 都是 Python，限制访问（不能封 IP）
	模拟 user-agent，获取代理 IP	访问量仍然异常，要求登录后才能继续访问
	注册账号，访问时带 cookie 或 token	健全账号体系，即只能访问账号下的好友的信息
	注册多个账号，联合爬取	请求过于频繁，进一步限制 IP 访问频率
	模仿用户操作，限制请求速度	弹出验证码识别
	通过相应的验证码识别手段（如云打码、OpenCV 识别等）	动态加载网站，数据通过 JS 加载，增加网络分析难度



小知识



6-1 利用爬虫能做到哪些有趣的事情呢？

💡 操作工单

工单 使用八爪鱼采集器

1. 提出问题

假如你想在网上购置一台冰箱，你比较关心淘宝网还是京东网的商品，用什么方法可以快速采集这两个网站的冰箱销售数据？

2. 问题描述

京东（JD.com）是中国最大的自营式电商企业，设有京东商城、京东金融、拍拍网、京东智能、O2O 及海外事业部，是电器类网上购物的主要途径之一。

八爪鱼数据采集系统以完全自主研发的分布式云计算平台为核心，可以在很短的时间内，轻松地从各种不同的网站或者网页中获取大量的规范化数据，帮助任何需要从网页中获取信息的用户实现数据自动化采集、编辑、规范化，摆脱对人工搜索及收集数据的依赖，从而降低获取信息的成本，提高效率。

本次任务利用八爪鱼采集器采集京东商城冰箱销售数据作为采购依据。

3. 采集流程

(1) 打开八爪鱼采集器的客户端，登录软件之后新建一个任务，打开要采集的网站地址。

(2) 进入到设计工作流程环节，在界面浏览器中输入要采集的网址，单击打开，就能看到要采集的网站界面，由于这个网址存在多页内容需要采集，我们再设置采集规则的时候可以选择高级选项中的“循环点击下一页”，软件会自动建立一个翻页循环。

(3) 建立好翻页循环后，就是采集当前页上的内容，软件会自动弹出对话框，先建立一个元素循环列表。当前页面的所有元素都被抓取后，循环列表则建立完成。

(4) 设置要抓取的内容，选择元素循环列表中的任意一个元素，在浏览器内找到该元素对应的图片，单击后弹出对话框。

(5) 检查一下，翻页循环框应该将产品循环框嵌套在内，表示先抓取完当前一整页的数据后再翻页。

(6) 设置执行计划后就可以开始采集了，直接单击“完成”步骤下的“检查任务”开始运行任务。采集完毕后可以直接导出成 Excel 文件。

4. 采集京东网冰箱销售数据

(1) 进入八爪鱼采集器主界面，选择热门采集模板中的“京东网”，如图 6-3 所示。

(2) 在京东商品搜索、列表页数据、众筹项目信息、商品评论等栏目中选择商品搜索，如图 6-4 所示。

(3) 在打开的模板界面中单击立即使用，设置任务名称、任务组、搜索关键字“冰箱”，单击“保存并启动”按钮，如图 6-5 所示。



图 6-3 八爪鱼主页图

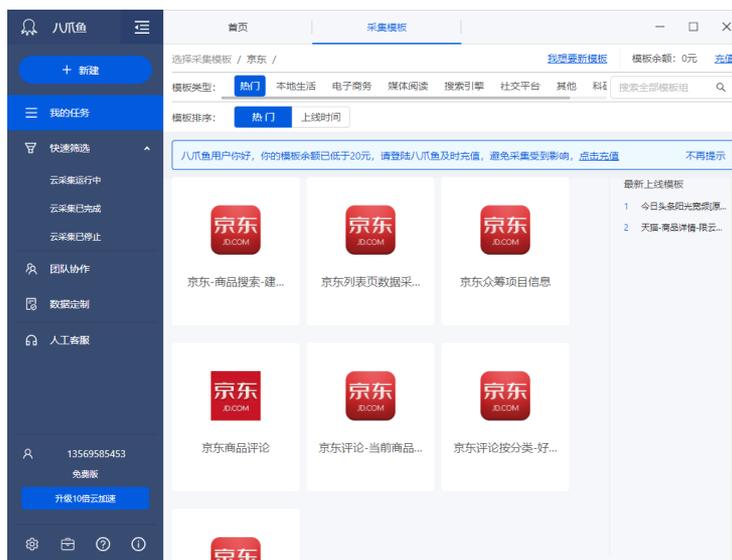


图 6-4 八爪鱼京东搜索

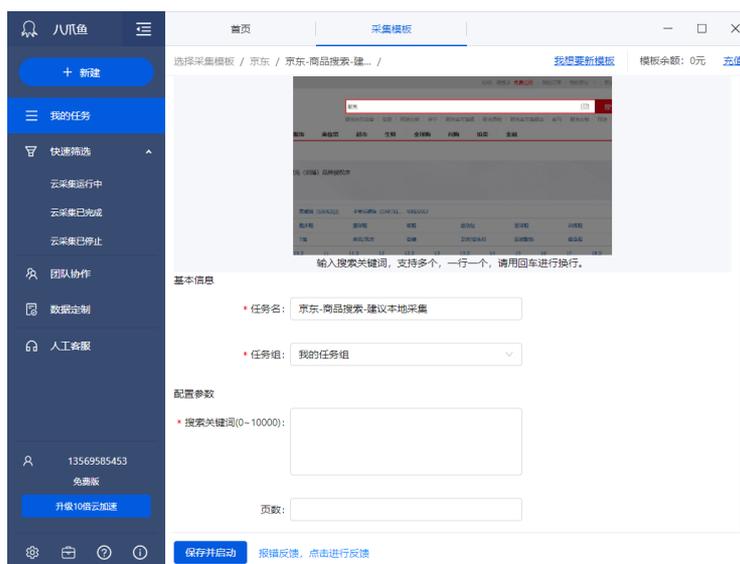


图 6-5 八爪鱼京东商品搜索

(4) 在运行任务对话框中单击“启动本地采集”或“启动云采集”按钮，还可以设置定时云采集，如图 6-6 所示。



图 6-6 八爪鱼采集方式设置

(5) 可以在右下角单击“停止采集”按钮，然后选择导出到 xls 文档中，如图 6-7 所示。

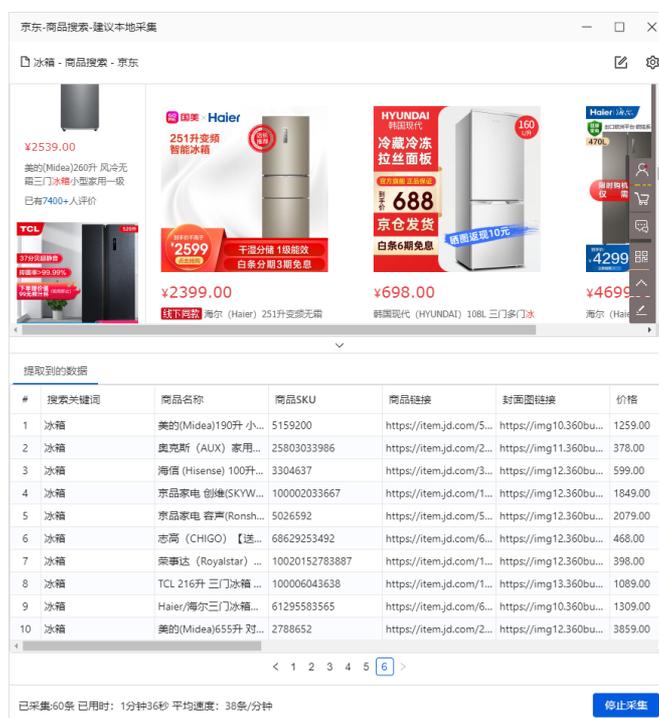


图 6-7 八爪鱼采集结果

5. 任务拓展

- (1) 采集京东网华为 P30 型号手机的销售数据。
- (2) 采集 58 同城平顶山湛河区的租房信息。
- (3) 了解后羿采集器、火车采集器等软件。

活动二 爬取之遨游“百度”

📍 活动目标

- (1) 认知爬取网页的含义。
- (2) 认知爬虫基础 Requests 库。
- (3) 认知爬虫基础 Beautiful Soup 库。
- (4) 认知正则表达式。

📍 活动要求

- (1) 学习态度积极，按时出勤。
- (2) 遵守课堂纪律，完成规定的内容。
- (3) 小组成员互相沟通，服从小组管理。

📍 知识储备

一、爬取网页的含义

1. 浏览网页的过程

抓取网页的过程其实和读者平时使用浏览器浏览网页的道理是一样的。例如，在浏览器的地址栏中输入“www.baidu.com”，打开网页的过程其实就是浏览器作为一个浏览的“客户端”，向服务器端发送了一次请求，把服务器端的文件“抓”到本地，再进行解释、展示。因为每个网页都是使用 HTML 语言标记的，用标签标记内容并加以解析和区分，而浏览器的功能是将获取到的 HTML 代码进行解析，然后将原始的代码转变成我们能直接看到的网站页面。

2. URL 的概念和举例

统一资源定位符（uniform resource locator，URL）是对可以从互联网上得到的资源的位置和访问方法的一种简洁的表示，是互联网上标准资源的地址。互联网上每种可用的资源，如 HTML 文档、图像、视频片段、程序等都由一个通用资源标志符进行定位。简单来说，URL 就是在浏览器端输入的“http://www.baidu.com”这个字符串，也就是人们平时说的“网址”。

URL 的一般格式为（带方括号[]的为可选项）：

```
protocol://hostname[:port]/path/[:parameters][?query]#fragment
```

URL 的格式由三部分组成：

- (1) 第一部分是协议（或称为服务方式）。
- (2) 第二部分是存有该资源的主机 IP 地址（有时也包括端口号）。
- (3) 第三部分是主机资源的具体地址，如目录和文件名等。

小贴士：

第一部分和第二部分用“://”符号隔开。

第二部分和第三部分用“/”符号隔开。

第一部分和第二部分是不可缺少的，第三部分有时可以省略。

下面来看看两个 URL 的小例子。

(1) HTTP 协议的 URL 示例。

使用超级文本传输协议 (hypertext transfer protocol, HTTP) 提供超级文本信息服务的资源。

例: `http://www.peopledaily.com.cn/channel/welcome.htm`。

其计算机域名为 `www.peopledaily.com.cn`。

超文本文件 (文件类型为.html) 是在目录/channel 下的 `welcome.htm`。

这是人民日报社里的一台计算机。

(2) 文件的 URL。用 URL 表示文件时, 服务器方式用 file 表示, 后面要有主机 IP 地址、文件的存取路径 (即目录) 和文件名等信息。

有时可以省略目录和文件名, 但 “/” 符号不能省略。

例: `file://ftp.yoyodyne.com/pub/files/foobar.txt`。

上面这个 URL 代表存放在主机 `ftp.yoyodyne.com` 上的 `pub/files/` 目录下的一个文件, 文件名是 `foobar.txt`。

爬虫最主要的处理对象就是 URL, 它根据 URL 地址取得所需要的文件内容, 然后对它进行进一步的处理。因此, 准确地理解 URL 对理解网络爬虫至关重要。

二、认识 HTTP 请求

1. HTTP 请求的含义

下载网页需要实现 HTTP 请求, 那么如何实现 HTTP 请求呢? 在 Python 中实现 HTTP 请求比较常用的是两个库——Requests 库和 urllib 库。

HTTP 协议定义 Web 客户端如何从 Web 服务器请求 Web 页面, 以及服务器如何把 Web 页面传送给客户端。HTTP 协议采用了请求/响应模型。客户端向服务器发送一个请求报文, 请求报文包含请求的方法、URL、协议版本、请求头部和请求数据。服务器以一个状态行作为响应, 响应的内容包括协议的版本、成功或者错误代码、服务器信息、响应头部和响应数据。

2. HTTP 请求信息

当浏览器向 Web 服务器发出请求时, 它向服务器传递了一个数据块, 也就是请求信息, HTTP 请求信息由请求方法、请求头部和请求正文三个部分组成, 编写爬虫与请求方法和请求头部信息相关。

1) 请求方法

不同的请求方法有不同的作用, HTTP1.0 定义了三种请求方法, 即 Get、Post 和 Head 方法, HTTP1.1 新增了五种请求方法, 即 Options、Put、Delete、Trace 和 Connect 方法, 如表 6-2 所示。

表 6-2 HTTP1.1 的请求方法

序号	方法	描述
1	Get	请求指定的页面信息并返回实体主体
2	Head	类似于 Get 请求, 只不过返回的响应中没有具体的内容, 用于获取报头
3	Post	向指定资源提交数据进行处理请求, 如提交表单或者上传文件。 数据被包含在请求体中, Post 请求可能会导致新的资源的建立和已有资源的修改
4	Put	从客户端向服务器传送的数据取代指定的文档内容

(续表)

序号	方法	描述
5	Delete	请求服务器删除指定的页面
6	Connect	预留给能够将连接改为管道方式的代理服务器
7	Options	允许客户端查看服务器的性能
8	Trace	回显服务器收到的请求，主要用于测试或诊断

“Get”方法提交的数据会直接填充在请求报文的 URL 上,如“https://www.baidu.com/s?ie=utf-8&f=8&rsv_bp=1”,“?”问号划分域名和 Get 提交的参数,A=B 中的 A 是参数名,B 是参数值,多个参数之间用&进行分割,如果参数值是中文,则会转换成诸如“%ab%12”的加密的 16 进制码。一般来说,浏览器处理的 URL 最大限度长度为 1024B(不同浏览器不一样),所以 Get 方法提交参数长度有限制。

“Post”方法提交的数据会附在正文上,一般请求正文的长度是没有限制的,但表单中所能处理的长度一般为 100k(不同协议不同浏览器不一样),而且需要考虑下层报文的传输效率,不推荐过长。

平时打开一个网站,一般使用的是 Get 方法,也就是请求了一个页面,如果涉及向网站提交数据,就用到了 Post 方法,其他的请求方法实际编写爬虫时较少用。

2) 请求头部

请求头部包含许多有关客户端环境和请求正文的信息。例如,请求头部可以声明浏览器所用的语言、请求正文的长度等。一般网站服务器最常见的反爬虫措施就是通过读取请求头部的用户代理信息,来判断这个请求是来自正常的浏览器还是来自爬虫。为了应对服务器的这种反爬虫策略,在编写爬虫时经常需要构造请求头部,来伪装成一个正常的浏览器。

下面我们用谷歌浏览器打开一个网页,实际查看一下浏览器的请求头部。

步骤:打开浏览器→在网页空白处右击→选择检查→单击子菜单中的 Network→输入一个网址(如 www.baidu.com)打开→单击子页面第一条请求记录→就会看到请求头部,如图 6-8 所示。

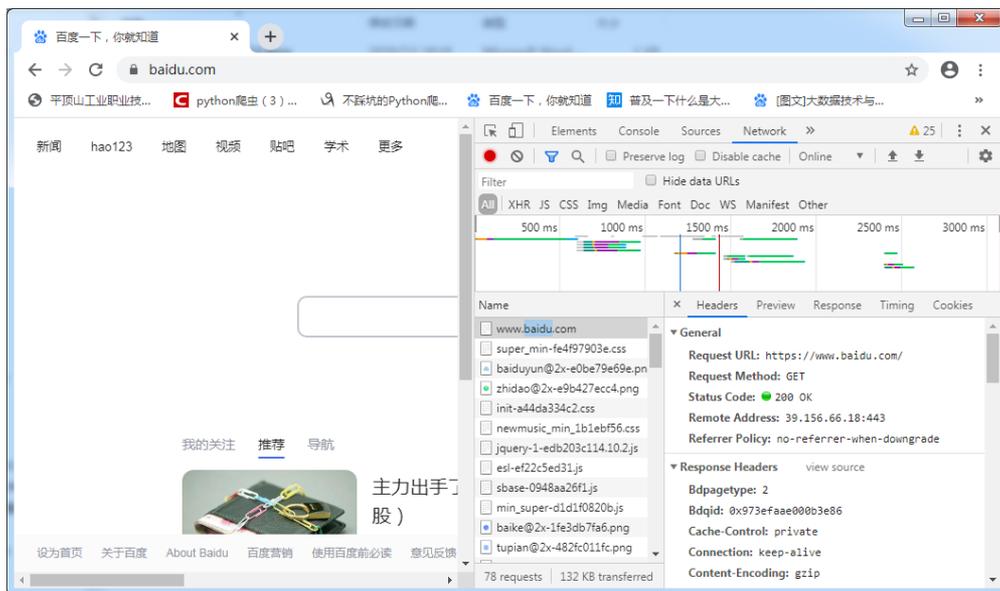


图 6-8 查看请求的详细信息

下面是请求详细信息的前五行。

```
Request URL: https://www.baidu.com/
Request Method: GET
Status Code: 200 OK
Remote Address: 39.156.66.18:443
Referrer Policy: no-referrer-when-downgrade
```

很明显，这里显示了请求的地址、使用的请求方法、相应状态码和远程地址等信息，拉动滚动条，就可以看到 Request Header 的内容，也就是请求头部，如图 6-9 所示。



图 6-9 查看请求的头部

从图 6-9 中可以看出，请求的头部是以字典的形式存在的，这个字典包含了用户代理信息。

三、爬虫基础——Requests 库入门

Requests 库是一个简洁且简单的处理 HTTP 请求的第三方库，其最大的优点是程序编写过程更接近正常 URL 访问过程。

Requests 库支持非常丰富的链接访问功能，包括国际域名和 URL 获取、HTTP 长连接和连接缓存、HTTP 会话和 Cookie 保持、浏览器使用风格的 SSL 验证、基本的摘要认证、有效的键值对 Cookie 记录、自动解压缩、自动内容解码、文件分块上传、HTTP(S)代理功能、连接超时处理、流数据下载等。

1. Requests 库的安装

Requests 库不是标准库，可以在命令中使用 pip 安装：

```
>pip install requests
```

在 Linux 操作系统下，需要输入如下命令安装：

```
> sudo pip install requests
```

安装完成后，需要在 Python 的 shell 中导入 Requests 库：

```
>>>import requests
```

2. Requests 库请求方法

导入 Requests 库后，使用 Requests 发送网络请求非常简单，假如现在要获取百度的首页，可以使用以下命令：

```
>>>response =requests.get("http://www.baidu.com/")
```

这样我们就通过了 Requests 的 Get 方法获取到了百度网站的首页，并把获取到的响应对象赋值给了 response。通过 Requests 的方法名称，很容易判断出使用的请求方法是 get，当然也可以使用 request post 方法。

```
>>> requests.post("http://www.baidu.com/")
```

Requests 库提供了 7 种方法，如表 6-3 所示。

表 6-3 Requests 库的 7 种方法

方 法	说 明
requests.request()	构造一个请求，支撑以下各方法的基础方法
requests.get()	获取 HTML 网页的主要方法，对应于 HTTP 的 Get
requests.post()	向 HTML 网页提交 Post 请求的方法，对应于 HTTP 的 Post
requests.head()	获取 HTML 网页头信息的方法，对应于 HTTP 的 Head
requests.put()	向 HTML 网页提交 Put 请求的方法，对应于 HTTP 的 Put
requests.patch()	向 HTML 网页提交局部修改请求，对应于 HTTP 的 Patch
requests.delete()	向 HTML 页面提交删除请求，对应于 HTTP 的 Delete

四、爬虫基础——Urllib 库入门

1. Urllib 库简介

Urllib 库是 Python 的标准库，提供了一系列用于操作 URL 的功能，使用不需要安装。其大部分功能与 Requests 库类似，它的 API 并不像 Requests 那样直观简洁，也有一些特别的用法。

2. 使用 Urllib 库下载网页

首先来看一下使用 Urllib 库下载一个网页的代码：

```
import urllib2
response = urllib2.urlopen("http://www.baidu.com")
print response.read()
```

如上述代码所示，真正的程序就两行，把它保存成 demo.py，进入该文件的目录，就可以执行了。

```
python demo.py
```

那么我们来分析这两行代码，第一行：

```
response = urllib2.urlopen("http://www.baidu.com")
```

首先我们调用的是 urllib2 库里面的 urlopen 方法，传入一个 URL，这个网址是百度首页，协议是 HTTP 协议，当然你也可以把 HTTP 换作 FTP、FILE、HTTPS 等，只是代表了一种访问控制协议，urlopen 一般接收三个参数，它的参数如下：

```
urlopen(url, data, timeout)
```

第一个参数 `url` 是必须要传送的，在这个例子里面我们传送了百度搜索引擎的 URL，执行 `urlopen` 方法之后，返回一个 `response` 对象，返回信息便保存在这里面。

第二个参数 `data` 是访问 URL 时要传送的数据，第三个参数 `timeout` 用于设置超时时间。

`response` 对象有一个 `read` 方法，可以返回获取到的网页内容，即第二行代码：

```
print response.read()
```

由于 `Requests` 库已经可以满足编写简单爬虫的大部分需求，并且学习更简单、使用更方便，所以大部分程序员在爬取网页的时候都会使用 `Requests` 库。在构造爬虫 URL 的时候有可能用到 `Urllib` 库。

五、Beautiful Soup 库

1. Beautiful Soup 库的简介

简单来说，`Beautiful Soup` 是 Python 的一个库，最主要的功能是从网页中抓取数据。`Beautiful Soup` 提供一些简单的、Python 式的函数用来处理导航、搜索、修改分析树等功能。它是一个工具箱，通过解析文档为用户提供需要抓取的数据，因为简单，所以不需要多少代码就可以写出一个完整的应用程序。`Beautiful Soup` 自动将输入的文档转换为 Unicode 编码，并将输出文档转换为 utf-8 编码。用户不需要考虑编码方式，除非文档没有指定一个编码方式，这时 `Beautiful Soup` 就不能自动识别编码方式了。`Beautiful Soup` 已成为和 `lxml`、`html6lib` 一样出色的 Python 解释器，为用户灵活地提供不同的解析策略或更快的解析速度。

2. Beautiful Soup 库的安装

`Beautiful Soup` 不是标准库，可以在命令行中使用 `pip` 安装：

```
>pip install Beautiful Soup4
```

在 Linux 系统下，需要输入如下命令安装：

```
> sudo pip install requests
```

安装完成后，需要在 Python 的 shell 中导入 `Beautiful Soup` 库：

```
>>>from 4 import BeautifulSoup
```



小知识



6-2 百度 logo 的由来

🔦 操作工单

工单 编程获取百度搜索数据

1. 提出问题

你平时用过哪些搜索引擎呢？你想不想一次性获取到搜索结果中的前 10 条数据呢？

2. 问题描述

从互联网中获取数据的最常见方法就是使用“爬虫”程序，互联网就像一张蜘蛛网一样，爬虫就像在上面爬行的蜘蛛一样，爬虫爬到哪里，就把哪里信息获取过来，只要是网络浏览器能够打开的页面，都可以用爬虫程序来获取数据，例如百度网站，下面我们来编写一个小爬虫吧。

3. 爬取百度数据

1) 安装第三方库

执行“开始”→“运行”命令，在弹出的“运行”窗口中输入“cmd”，并按“Enter”键，在弹出的命令行窗口中输入以下命令：

```
pip install requests      # 安装 requests 库
pip install bs4          # 安装 BeautifulSoup4 库
```

2) 导入第三方库

在 IDLE 中输入下面的代码：

```
import requests          # 导入 requests 库
from bs4 import BeautifulSoup # 导入 BeautifulSoup4 库
import re                # 导入 re 模块
import json              # 导入 json 解释库
```

4. 爬虫程序主体

在 IDLE 中继续输入下面的代码：

```
titles=[]                # 获取到的网页标题
key=input("请输入一个关键词：")
for i in range(10):      # 一次获取 10 条内容
    url='http://www.baidu.com/s?wd='+key+'&pn='+str(i*10)
# 用输入的关键词形成百度的查询网址链接，并指定当前的起始编号
    res=requests.get(url) # 获取网址的返回内容到 res 中
    res.encoding='UTF-8'  # 对返回的内容使用标准的 UTF-8 编码
    soup=BeautifulSoup(res.text,'html.parser')
    # 使用 html.parser 剖析器来解释 res 的文本，结果放入 soup 中
for div in soup.find_all('div',{'data-tools':re.compile('title')}):
    # 在 soup 中寻找指定的标签组合，百度搜索结果的标题都嵌在<div>和</div>中间
    data=div.attrs['data-tools'] # 取得 data-tools 属性的数据 data
    d=json.loads(data)          # 以 json 格式导入数据到 d 中
    titles.append(d['title'])   # 把数据的 title 部分加入 titles 中
for i in titles:
    print(i)                  # 打印所有的 titles
```

5. 运行程序

程序运行结果如图 6-10 所示。

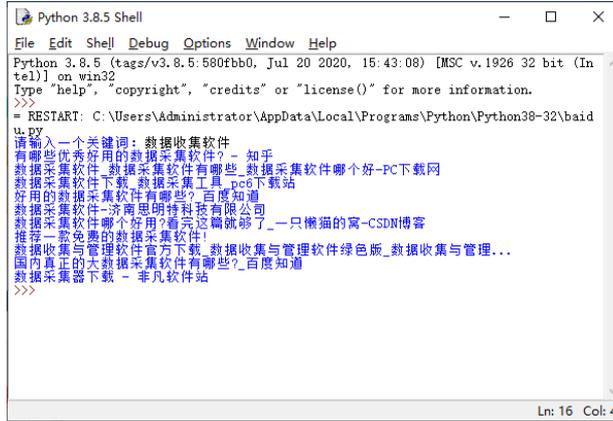


图 6-10 百度搜索结果爬取

6. 编程实现

仿照以上程序，编程实现在百度搜索引擎中获取数据的爬虫程序，要求如下：

- (1) 1次获取15条数据；
- (2) 修改你认为可以修改的变量，并做好标记；
- (3) 在运行界面输入不同的关键字。

7. 在程序实现的过程中，你遇到了哪些问题

附 录

附录一 常用 ASCII 码对照表

ASCII 值	控制字符及含义	ASCII 值	字 符	ASCII 值	字 符	ASCII 值	字 符
0	NUL 空	32	空格	64	@	96	`
1	SOH 标题开始	33	!	65	A	97	a
2	STX 正文开始	34	"	66	B	98	b
3	ETX 正文结束	35	#	67	C	99	c
4	EOT 传输结束	36	\$	68	D	100	d
5	ENQ 询问	37	%	69	E	101	e
6	ACK 响应	38	&	70	F	102	f
7	BEL 响铃	39	'	71	G	103	g
8	BS 退格	40	(72	H	104	h
9	HT 横向制表	41)	73	I	105	i
10	LF 换行	42	*	74	J	106	j
11	VT 纵向制表	43	+	75	K	107	k
12	FF 换页	44	,	76	L	108	l
13	CR 换行	45	-	77	M	109	m
14	SO 移出	46	.	78	N	110	n
15	SI 移入	47	/	79	O	111	o
16	DLE 数据链转义	48	0	80	P	112	p
17	DC1 设备控制 1	49	1	81	Q	113	q
18	DC2 设备控制 2	50	2	82	R	114	r
19	DC3 设备控制 3	51	3	83	S	115	s
20	DC4 设备控制 4	52	4	84	T	116	t
21	NAK 否认	53	5	85	U	117	u
22	SYN 同步空转	54	6	86	V	118	v
23	ETB 组传输结束	55	7	87	W	119	w
24	CAN 作废	56	8	88	X	120	x
25	EM 媒体结束	57	9	89	Y	121	y
26	SUM 取代	58	:	90	Z	122	z
27	ESC 转义	59	;	91	[123	{
28	FS 文卷分隔	60	<	92	\	124	
29	GS 群分隔	61	=	93]	125	}
30	RS 记录分隔	62	>	94	^	126	~
31	US 单元分隔	63	?	95	_	127	DEL

附录二 常用 pygame 模块功能和命令

一、pygame 模块

序号	模块名	功能	序号	模块名	功能
1	pygame.cdrom	访问光驱	12	pygame.music	播放音频
2	pygame.display	访问显示设备	13	pygame.overlay	访问高级视频叠加
3	pygame.draw	绘制形状、线和点	14	pygame.rect	管理矩形区域
4	pygame.cursors	加载光标	15	pygame.scrap	本地剪贴板访问
5	pygame.event	管理事件	16	pygame.sndarray	操作声音数据
6	pygame.font	使用字体	17	pygame.sprite	操作移动图像
7	pygame.image	加载和存储图片	18	pygame.surface	管理图像和屏幕
8	pygame.joystick	使用游戏手柄或者类似的东西	19	pygame.surfarray	管理点阵图像数据
9	pygame.key	读取键盘按键	20	pygame.time	管理时间和帧信息
10	pygame.mixer	声音	21	pygame.transform	缩放和移动图像
11	pygame.mouse	鼠标	22	pygame.movie	播放视频

二、pygame.display 模块的常用方法

序号	方法名	功能
1	pygame.display.init()	初始化 display 模块
2	pygame.display.quit()	结束 display 模块
3	pygame.display.get_init()	如果 display 模块已经被初始化, 则返回 True
4	pygame.display.set_mode()	初始化一个准备显示的界面
5	pygame.display.get_surface()	获取当前的 Surface 对象
6	pygame.display.flip()	更新整个待显示的 Surface 对象到屏幕上
7	pygame.display.update()	更新部分内容显示到屏幕上, 如果没有参数, 则与 flip 功能相同 (上一条)

三、pygame.event 模块的常用方法

序号	方法名	功能
1	pygame.event.get()	获取事件的返回值, 使用 event.type 进行区分
2	pygame.event.wait()	等待发生一个事件才会继续下去
3	pygame.event.poll()	会根据现在的情形返回一个真实的事件
4	pygame.event.set_blocked()	过滤
5	pygame.event.set_allowed()	允许事件

四、pygame. image 模块的常用方法

序号	方法名	功能
1	pygame.image.load()	从文件加载新图片
2	pygame.image.save()	将图像保存到磁盘上
3	pygame.image.get_extended()	检测是否支持载入扩展的图像格式
4	pygame.image.tostring()	将图像转换为字符串描述
5	pygame.image.fromstring()	将字符串描述转换为图像
6	pygame.image.frombuffer()	创建一个与字符串描述共享数据的 Surface 对象

五、pygame. mixer 模块的常用方法

序号	方法名	功能
1	pygame.mixer.music.load()	载入一个音乐文件用于播放
2	pygame.mixer.music.play()	开始播放音乐流
3	pygame.mixer.music.rewind()	重新开始播放音乐
4	pygame.mixer.music.stop()	结束音乐播放
5	pygame.mixer.music.pause()	暂停音乐播放
6	pygame.mixer.music.unpause()	恢复音乐播放
7	pygame.mixer.music.fadeout()	淡出的效果结束音乐播放
8	pygame.mixer.music.set_volume()	设置音量
9	pygame.mixer.music.get_volume()	获取音量
10	pygame.mixer.music.get_busy()	检查是否正在播放音乐
11	pygame.mixer.music.set_pos()	设置播放的位置
12	pygame.mixer.music.get_pos()	获取播放的位置

六、pygame. Surface 模块的常用方法

序号	方法名	功能
1	pygame.Surface.blit()	将一个图像画到另一个图像上
2	pygame.Surface.convert()	转换图像的像素格式
3	pygame.Surface.convert_alpha()	转化图像的像素格式, 包含 alpha 通道的转换
4	pygame.Surface.fill()	使用颜色填充 Surface
5	pygame.Surface.get_rect()	获取 Surface 的矩形区域

七、pygame. time 模块的常用方法

序号	方法名	功能
1	pygame.time.get_ticks()	获取以毫秒为单位的时间
2	pygame.time.wait()	暂停程序一段时间
3	pygame.time.delay()	暂停程序一段时间
4	pygame.time.set_timer()	在事件队列上重复创建一个事件
5	pygame.time.Clock()	创建一个对象来帮助跟踪时间

八、pygame.transform 模块的常用方法

序号	方法名	功能
1	pygame.transform.flip	垂直和水平翻转
2	pygame.transform.scale	调整大小到新的分辨率
3	pygame.transform.rotate	旋转图像
4	pygame.transform.rotozoom	过滤的比例和旋转
5	pygame.transform.scale2x	专业图像倍增器
6	pygame.transform.smoothscale	将表面平滑地缩放到任意大小
7	pygame.transform.chop	获取内部区域已删除的图像副本
8	pygame.transform.laplacian	找到表面中的边缘
9	pygame.transform.average_surfaces	从许多表面找到平均表面
10	pygame.transform.average_color	找到曲面的平均颜色
11	pygame.transform.threshold	查找表面中的哪些像素和多少像素在'search_color'或'search_surf'的阈值内

九、pygame.draw 模块的常用方法

序号	方法名	功能
1	pygame.draw.polygon(surface,color,pointlist,width=0)	绘制一个多边形
2	pygame.draw.circle(surface,color,pos,radius,width=0)	绘制一个圆
3	pygame.draw.ellipse(surface,color,Rect,width=0)	绘制一个椭圆
4	pygame.draw.arc(surface,color,Rect,start_angle,stop_angle,width=1)	绘制一条弧线
5	pygame.draw.line(surface,color,start_pos,end_pos,width=1)	绘制一条线段
6	pygame.draw.lines(surface,color,closed,pointlist,width=1)	绘制一条折线
7	pygame.draw.aaline(surface,color,start_pos,end_pos,width=1)	绘制一根平滑的线
8	pygame.draw.aalines(surface,color,closed,pointlist,width=1)	一系列平滑的线

十、pygame 的常用事件

序号	事件	产生途径
1	QUIT	用户单击“关闭”按钮
2	ACTIVEEVENT	pygame 被激活或者隐藏
3	KEYDOWN	键盘被按下
4	KEYUP	键盘被放开
5	MOUSEMOTION	鼠标移动
6	MOUSEBUTTONDOWN	鼠标按下
7	MOUSEBUTTONUP	鼠标放开
8	VIDEORESIZE	pygame 窗口缩放

附录三 常用 RGB 颜色对照表

颜色样式	RGB 数值	颜色样式	RGB 数值
黑色	0,0,0	白色	255,255,255
象牙黑	88,87,86	天蓝灰	202,235,216
冷灰	128,138,135	灰色	192,192,192
暖灰	128,118,105	象牙灰	251,255,242
石板灰	118,128,105	亚麻灰	250,240,230
白烟灰	245,245,245	杏仁灰	255,235,205
蛋壳灰	252,230,202	贝壳灰	255,245,238
红色	255,0,0	黄色	255,255,0
镉红	227,23,13	镉黄	255,153,18
砖红	156,102,31	香蕉黄	227,207,87
珊瑚红	255,127,80	金黄	255,215,0
番茄红	255,99,71	肉黄	255,125,64
粉红	255,192,203	粉黄	255,227,132
印度红	176,23,31	橘黄	255,128,0
深红	255,0,255	萝卜黄	237,145,33
黑红	116,0,0	黑黄	85,102,0
绿色	0,255,0	棕色	128,42,42
青色	0,255,255	土色	199,97,20
黄绿色	127,255,0	沙棕色	244,164,95
青绿色	64,224,205	棕褐色	210,180,140
靛青色	8,46,84	玫瑰红	188,143,143
森林绿	34,139,34	赫色	160,82,45
草绿色	107,142,35	肖贡土色	199,97,20
蓝色	0,0,255	肖贡土色	160,32,240
锰蓝	3,168,158	淡紫色	218,112,214
深蓝	25,25,112	紫罗兰	138,43,226
土耳其蓝	0,199,140	胡紫色	153,51,250

参 考 文 献

- [1] 董付国. Python 程序设计基础[M]. 2 版. 北京: 清华大学出版社, 2018.
- [2] 嵩天, 礼欣, 黄天羽. Python 语言程序设计基础[M]. 2 版. 北京: 高等教育出版社, 2017.
- [3] 江红, 余青松. Python 程序设计与算法基础教程[M]. 北京: 清华大学出版社, 2017.
- [4] 刘卫国. Python 语言程序设计[M]. 北京: 电子工业出版社, 2016.
- [5] 戴维·施奈德. Python 程序设计[M]. 车万翔, 译. 北京: 机械工业出版社, 2016.
- [6] 伯德, 克莱恩, 洛佩尔. Python 自然语言处理[M]. 陈涛, 张旭, 崔杨, 等, 译. 北京: 人民邮电出版社, 2014.
- [7] 伊德里斯. Python 数据分析基础教程: NumPy 学习指南[M]. 2 版. 张驭宇, 译. 北京: 人民邮电出版社, 2014.
- [8] 潘曲, 尹鲍德. Python 入门经典: 以解决计算问题为导向的 Python 编程实践[M]. 张敏, 译. 北京: 机械工业出版社, 2012.