



Python 科学计算基础

第一章 绪论

主讲：罗奇鸣
luoq@ustc.edu.cn

2025 年 9 月 9 日



目录

利用计算机解决问题

科学计算

程序设计语言的分类

过程式编程范式

面向对象编程范式

学习 Python 语言的理由

Python 语言的发展历史

Python 语言的特点

Python 科学计算环境

选课须知

实验 1：安装和使用 Python 开发环境



利用计算机解决问题

计算机是能够自动对数据进行计算的电子设备。
计算机的优势是运算速度快。以下举例说明。

1. ENIAC(1946) 是世界上第一台通用计算机。人工计算一条弹道需要 20 多分钟时间，ENIAC 仅需 30 秒!
2. 派-曙光 (2018) 是我国自主研发的用于数值天气预报的高性能计算机系统，其峰值运算速度达到每秒 8189.5 万亿次，内存总容量达到 690432GB。近年来，我国天气预报的准确度达国际领先水平。



科学计算

- ▶ 定义：是以数学和计算机科学为基础形成的交叉学科，是利用计算机的计算能力求解科学和工程问题的数学模型所需的理论、技术和工具的集合；
- ▶ 特点：随着计算机的计算能力的不断提升，科学计算也得到了迅速发展；
- ▶ 地位：理论分析、实验和科学计算并称为当今科学发现的三个支柱。



程序设计语言的分类

为了利用计算机解决问题，必须使用某种程序设计语言把解决问题的详细过程编写为程序，即一组计算机能识别和执行的指令。计算机通过运行程序解决问题。程序设计语言可分为三类：

- ▶ 机器语言：用 0 和 1 组成的二进制串表示 CPU(处理器) 指令和数据，适用于早期的计算机；
- ▶ 汇编语言：用易于理解和记忆的符号来代替二进制串，克服了机器语言难以理解的缺点；
- ▶ 高级语言：采用接近自然语言和数学公式的方式表达解决问题的过程，不再依赖于 CPU，实现了可移植性。



高级语言的分类

1956 年投入使用的 Fortran 语言是第一种高级语言。近几十年来，高级语言不断涌现，数量达到几百种。高级语言按照编程范式 (programming paradigm) 可划分为以下几个类别：

1. 命令式 (imperative): 例如 C、C++、Java 和 Python
2. 声明式 (declarative): 例如 SQL
3. 过程式 (procedural): 例如 C、C++、Java 和 Python
4. 函数式 (functional): 例如 Lisp、ML、Haskell 和 Scala
5. 逻辑式 (logic): 例如 Prolog
6. 面向对象 (object-oriented): 例如 C++、Java 和 Python

这些编程范式并非互斥，一种语言可同时支持多种范式。



过程式编程范式的特点

过程式编程范式的特点如下：

1. 基于输入和输出将一个较复杂的问题逐步分解成多个子问题。如果分解得到的某个子问题仍然较复杂，则继续对其分解，直至所有子问题都易于解决为止。
2. 过程式的程序由多个过程构成，每个过程解决一个子问题。这些过程形成一个树状结构。每个过程内部均是由顺序、选择和循环三种基本结构组成。
3. 在软件维护时，如果需要修改软件使用的数据，则处理数据的过程也需要进行修改。

过程式的缺点是软件需求发生变化时的维护代价较高，也不易实现代码复用，因此不适于开发大型软件。



举例说明过程式编程范式

编写程序根据年和月输出日历。程序的运行结果如图所示。

```
1 In [2]: run month_calendar.py --year 2022 --month 10
2 2022 10
3
4 Sun Mon Tue Wed Thu Fri Sat
5                                     1
6 2    3    4    5    6    7    8
7 9    10   11   12   13   14   15
8 16   17   18   19   20   21   22
9 23   24   25   26   27   28   29
10 30   31
```




举例说明过程式编程范式

采用过程式编程范式进行问题分解，得到以下设计方案：

1. 读取用户输入的年和月

2. 输出日历的标题

3. 输出日历的主体

3.1 怎样确定指定的某年某月有多少天？如果是2月，怎样确定指定年是否是闰年？

3.2 怎样确定这个月的第一天是星期几？用 $w(y, m, d)$ 表示计算 y 年 m 月 d 日是星期几的函数，函数值为 $0, 1, \dots, 6$ 依次表示周日、周一、...、周六。已知公式：

$$w(y, 1, 1) = (y + \lfloor (y-1)/4 \rfloor - \lfloor (y-1)/100 \rfloor + \lfloor (y-1)/400 \rfloor) \% 7 \quad (1)$$

用 $v(y, m, 1, y, 1, 1)$ 表示计算从 y 年1月1日到 y 年 m 月1日经历了多少天的函数，则易知

$$w(y, m, 1) = (w(y, 1, 1) + v(y, m, 1, y, 1, 1)) \% 7 \quad (2)$$

剩余的问题就是计算 $v(y, m, 1, y, 1, 1)$ 。

3.2.1 怎样确定任意指定的某年某月有多少天？已由3.1解决。



面向对象编程范式

面向对象编程范式的特点如下：

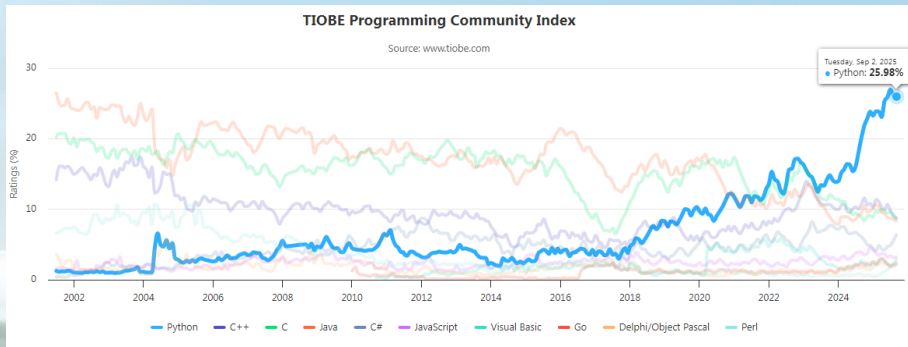
1. 将客观事物直接映射到软件系统的对象。对象是将数据及处理数据的过程封装在一起得到的整体，用以表示客观事物的状态和行为。
2. 从同一类型的对象中抽象出其共同的属性和操作，形成类。类是创建对象的模板，对象是类的实例。
3. 每个类作为一个独立单元进行开发、测试和维护。如果需要修改一个类的实现细节，只要不改变其接口就不会影响使用该类的外部代码，使软件系统更易于维护。
4. 通过继承可以重复利用已有类的代码，并根据需要进行扩展，从而提升了软件系统的开发效率。
5. 程序由多个类构成。程序在运行时由各个类创建一些对象，对象之间通过明确定义的接口进行交互，完成系统功能。

面向对象编程范式的优点是易于开发和维护大型软件，缺点是在程序的运行效率上不如过程式编程范式。



学习 Python 语言的理由

1. TIOBE 指数由荷兰 TIOBE 公司自 2001 年开始每月定期发布，用于评估程序设计语言的流行度。



2. Google 公司的决策: "Python where we can, C++ where we must."
3. 多所国外名校都开设了 Python 相关课程。



Python 语言的设计目标

Python 由荷兰程序员 Guido van Rossum 于 1989 年设计和开发。Guido 希望有一种语言可以兼具 C 和 shell 的优点。Guido 总结的设计目标列举如下：

1. 一种简单直观的语言，并与主要竞争者一样强大；
2. 代码像纯英语那样容易理解；
3. 适用于短期开发的日常任务；
4. 开源，以便任何人都可以为它做贡献。



Python 语言的版本更新

Python 软件基金会 (Python Software Foundation, <https://www.python.org/psf/>) 是 Python 的版权持有者，致力于推动 Python 开源技术和发布 Python 的新版本。

1. 2008 年 12 月，Python3.0 发布，这是一次重大的升级，与 Python2.x 不兼容。
2. 2019 年 10 月，Python3.8 发布。
3. 每年 10 月发布一个新版本。
4. 2024 年 10 月，Python3.13 发布。



Python 语言的特点

1. 简单易学：语法简单清晰，程序容易理解；
2. 动态类型：使用变量之前无需声明其类型，变量的类型由运行时系统推断；
3. 功能强大：标准库和扩展库提供了数据结构、系统管理、网络通信、文本处理、科学计算、图形系统、Web 开发等丰富的功能；
4. 面向对象：适于大规模软件开发；
5. 解释执行：交互式的开发环境；
6. 易于扩展：可以把部分代码用 C 或 C++ 编写并编译成扩展库，然后在 Python 程序中使用它们。



Python 科学计算环境

- ▶ 科学计算使用的程序设计语言分为两大类：低层语言 and 高层语言。前者包括 Fortran、C 和 C++，后者包括 MATLAB 和 Python。
- ▶ 用前者开发的程序比用后者开发的程序的运行效率更高，但开发耗时更长，软件维护代价也更高。
- ▶ Python 语言及其众多的扩展库 (NumPy、SciPy、SymPy 和 Matplotlib 等) 所构成的开发环境十分适合开发科学计算应用程序。
- ▶ 和美国公司开发的付费商业软件 MATLAB 相比，Python 科学计算环境的优势是免费开源并且没有国际政治风险。



选课要求和课程安排

本课程面向满足以下条件的学生：

- ▶ 已经学习过微积分和线性代数；
- ▶ 对于学习计算机科学和数学具有浓厚兴趣；
- ▶ 自备计算机。

大一新生不宜在秋季学期选修。

每次课程包括两节理论课和一节实验课。如有问题，可以在理论课举手提问、在实验课提问或发送电子邮件给老师。

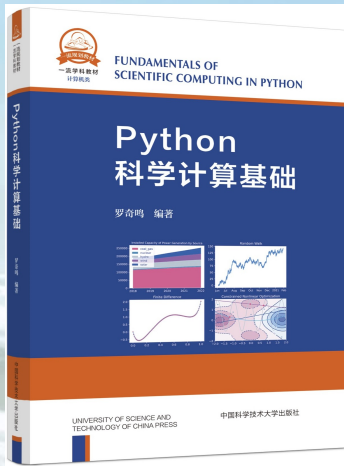
上课使用的课件和源程序等将发送至选课同学的电子信箱。

教材



Python
科学计算基础

书号 9787312058172。 京东 等电商网站有售。





学习方法

1946 年著名学习专家爱德加·戴尔发现不同学习方式的学习效果按从高到低呈金字塔分布。总体上，被动学习(听讲、阅读、视听)的效果低于主动学习(讨论、实践、教他人)。

学习本课程的最有效方法是编写和调试程序，在实践中掌握知识和提升能力。

编程中遇到自己解决不了的问题怎么办？

- ▶ 仔细阅读讲义、书籍和技术文档(Blackboard 系统);
- ▶ 上网搜索：百度, 微软 Bing 国际版, stackoverflow, ...;
- ▶ 在实验课和其他同学讨论，或者在 Blackboard 系统的讨论版发帖；
- ▶ 向老师提问。



考核方式

考核方式是课堂参与、实验课作业和大作业。

- ▶ 课堂参与：上课过程中会随机抽取同学回答问题。无故缺勤会扣分。
- ▶ 实验课作业：在 Blackboard 系统提交一个文本文件 (txt 后缀)，文件中记录每道题的源程序和运行结果。
- ▶ 大作业：设计和实现一个 Python 科学计算程序并在课堂讲解和演示。未在讲解日期 (最后三周) 之前提交符合要求的大作业，或提交了但未在课堂讲解，则成绩为不及格。



大作业的选题建议

选题原则是体现学以致用，例如：

1. 对于某种自然规律和实验现象的模拟，用图示显示模拟结果；
2. 对于某种科学理论或数学模型的模拟，用图示显示模拟结果；
3. 实现某种计算机算法，可视化算法的运行过程和结果；
4. 解决一个课程学习或研究工作中遇到的科学计算问题。

如果程序实现的功能与科学计算无关，如实现一个游戏或实现某种信息管理系统等，则成绩为不及格。

大作业的详细要求会发送至选课同学的电子信箱。

Q & A





实验 1：Python 开发环境

本实验的目的是安装 Python 开发环境并熟悉其基本功能。无需提交作业。



安装 Python 开发环境

Anaconda 是一个开源的 Python 发行版本，包含 Python 解释器、集成开发环境 spyder、包管理器 conda 和多个科学计算扩展库（numpy、scipy 等），可运行在 Windows、Linux 和 Mac OS 系统上。安装过程中可指定安装路径，路径中不可包含中文字符。

安装完成以后，可以使用包管理器 conda 安装其他 Python 扩展库。例如在 Anaconda Powershell Prompt 中运行命令“conda install scikit-learn”可安装 Scikit-learn 库。



运行 spyder

安装完成以后，在 Windows 系统中从已安装程序的列表中可以找到 Anaconda 文件夹下的 spyder 的图标，点击此图标即可运行 spyder。也可以通过在命令行 (控制台) 输入命令“spyder”运行 spyder。

如果需要设置 spyder 开发环境的字体和配色，可以点击 Tools 菜单的 Preference 菜单项，此时出现一个对话框。选中对话框左边列表中的 Appearance，此时对话框中间的 “Syntax highlighting theme” 部分有一个下拉列表，其中的每个选项对应一种背景和语法高亮的颜色方案；“Fonts” 部分可以设置字体类型和大小。



在 spyder 中运行 Python 程序

在 spyder 中运行 Python 程序的方式有两种, 分别适用于简短和较长的程序:

1. 在右下角的 IPython 窗口中输入一条或多条语句, 然后回车;
2. 在左边的编辑窗口中输入一个完整的程序, 点击 Run 菜单的 Run 菜单项执行。运行结果显示在 IPython 窗口中。