

第1章 Python与财务概述

Python简介

Python的发展历程

- 🕒 早期阶段：Python的诞生
- 📈 Python2.x阶段：广泛应用但存遗憾
- 🔥 Python3.x阶段：重大革新
- 🌟 现代阶段：Python 3的全面普及
- 🕒 Python的未来与现状

Python的特点及应用范围

- 🌟 Python的特点
- 🚀 Python的应用范围

Python在财务领域中的应用

- 1 流程自动化：提升效率，减少人为错误
- 2 数据分析：智能建模，辅助决策

◆ 总结

Python环境配置及使用

Anaconda的下载与安装

Jupyter Notebook的使用

快捷键大全

✅ 命令模式快捷键（按 Esc 进入）

✅ 编辑模式快捷键（按 Enter 进入）

Python简介

Python的发展历程

Python是一种高级编程语言，由吉多·范罗苏姆（Guido van Rossum）在1989年底至1990年初期间开发，最初的目的是打造一种简洁、易读且易用的编程语言。

Python的发展大致可以分为以下几个阶段：

🎯 早期阶段：Python的诞生

- 1989年：吉多在研究ABC语言时，受到其清晰和模块化的思想启发，开始开发Python。
- 1991年：Python的第一个公开版本 **Python 0.9.0** 诞生，其中已经包含了今天常见的特性，如异常处理、模块、函数等。
- 1994年：发布了 **Python 1.0**，这是第一个正式版本，引入了诸如lambda、map、filter等函数式编程特性。

📈 Python2.x阶段：广泛应用但存遗憾

- 2000年：Python 2.0 发布，新增了垃圾回收机制和列表推导式等功能。
- Python 2.x 系列成为了主流版本，并在Web开发、数据处理等领域广泛使用。
- 遗憾之处：Python 2.x 存在一些设计缺陷，比如字符串处理上的不一致性、Unicode支持不足等问题。

🔥 Python3.x阶段：重大革新

- 2008年：Python 3.0 发布，引入了许多**不兼容** Python 2.x 的改进，如：
 - `print` 语句变为函数：`print "hello"` → `print("hello")`
 - 默认使用 Unicode 编码处理字符串
 - 统一了整数类型
- 由于 Python 2.x 和 3.x **互不兼容**，最初很多项目难以迁移，导致3.x的普及速度较慢。

🌟 现代阶段：Python 3的全面普及

- 2018年以后：随着Python 2的官方维护即将结束，Python 3逐渐成为标准。
- 2020年1月1日：Python 2.7 停止维护，标志着Python正式迈入Python 3时代。
- Python 3.x 的重要更新：
 - Python 3.6：引入了f-string格式化字符串，如 `f"Hello {name}!"`
 - Python 3.8：新增“海象运算符”（`:`），可在表达式中赋值
 - Python 3.9：增加字典合并运算符（`|`）等功能
 - Python 3.12（最新）：进一步提升性能和开发体验

🎯 Python的未来与现状

Python如今已经成为数据科学、人工智能、Web开发、自动化等领域的主流语言。凭借简洁易读的语法、强大的第三方库生态和活跃的社区支持，Python正不断扩展它的影响力！

—— 总结一句话：从1991年吉多开发的简洁脚本语言，到今天覆盖数据科学、人工智能等领域的主流语言，Python用它的简洁、灵活和强大征服了世界！🐍

Python的特点及应用范围

🌟 Python的特点

1. **简洁易读**：语法清晰，代码结构简明，非常适合初学者和大型项目开发。
2. **语法灵活**：支持面向对象、函数式、过程式编程，动态类型语言。
3. **丰富库支持**：强大的标准库和第三方库，涵盖数据分析、Web开发、科学计算等。
4. **跨平台性**：可在Windows、MacOS、Linux等多平台运行。
5. **高效稳定**：拥有自动内存管理机制（垃圾回收），性能稳定可靠。
6. **大数据生态**：内置NumPy、Pandas、Matplotlib等库，广泛应用于数据科学和机器学习。
7. **社区与开源**：拥有活跃的开发社区，开源生态助力持续发展。

🚀 Python的应用范围

1. **Web开发**：使用Django、Flask等框架，快速搭建Web应用。
2. **数据科学与机器学习**：基于NumPy、Pandas、TensorFlow等库，进行数据分析与AI开发。
3. **自动化与脚本编程**：编写自动化脚本、系统管理、批处理任务。
4. **科学计算与工程应用**：借助SciPy、Matplotlib等库进行数值计算与可视化。
5. **游戏开发与图形处理**：使用Pygame等库进行游戏开发及图像处理。

Python在财务领域中的应用

财政部发布的《会计信息化发展规划（2021-2025 年）》指出，数据已成为五大生产要素之一，会计数据要素是企业经营管理的重要资源。如何将零散、非结构化的会计数据转化为聚合、结构化的数据，使其更好地服务于企业价值创造，是会计数字化转型的核心挑战。

Python 作为先进的财务数字化工具，凭借简洁的语法和强大的第三方库，在会计流程自动化、数据分析及管理会计价值创造方面具有广泛应用，能够有效提升企业财务管理水平。本文将从**流程自动化**和**数据分析**两个方面探讨 Python 在会计领域的应用。

1 流程自动化：提升效率，减少人为错误

Python 可自动执行会计工作流程，减少重复性劳动，提高工作效率，释放人力资源，使会计人员专注于更有价值的工作。

✓ 自动化数据采集

- 网络爬虫技术：自动爬取上市公司财务数据、电子税务局税费申报信息、最新财税政策等。
- OCR 文字识别：批量识别纸质发票、PDF 电子发票、全电发票等票面信息，并自动核验增值税发票。

✓ 自动化财务软件操作

- 流程自动化（RPA）：模拟鼠标键盘操作，实现批量录入财务数据，如自动登录自然人电子税务局客户端批量申报个税、批量录入单据至财务软件。

✓ 办公自动化（Excel、Word、PDF）

- Excel 自动化：批量合并多期间会计表格，自动填充 Excel 报表。
- Word 自动化：根据 Excel 数据自动生成 Word 财务分析报告。
- PDF 处理：从 PDF 文档中批量提取关键信息，生成 Excel 统计表格。

2 数据分析：智能建模，辅助决策

Python 在财务分析中的应用不仅限于数据处理，更能帮助会计人员构建智能化的财务管理模型，提升决策效率。

✓ 财务管理建模

- 现金流分析：计算净现值（NPV）、内含报酬率（IRR）、投资回收期等投资决策指标。
- 财经数据接口：快速获取股票历史收盘价、上市公司财务报表数据等。

✓ 数据处理与统计分析

- Pandas 库：提供高效的数据处理能力，包括数据清洗、统计分析、数据聚合等。
- Matplotlib 库：可视化财务数据，生成折线图、柱状图、饼图、热力图等直观报表。

✓ 高级财务分析

- 线性回归分析：预测未来财务趋势，分析业务增长情况。

- **投资组合优化**：基于数据分析，优化投资组合，降低风险，提高收益。
- **求解方程式、排列组合**：解决财务计算中的复杂数学问题。

◆ 总结

Python 在会计领域的应用正在深入发展，通过自动化流程减少人为操作，提高工作效率，同时利用强大的数据分析能力，助力财务管理决策。未来，Python 将成为会计人员的重要工具，推动企业财务数字化转型，实现更高效、更智能的财务管理。🚀

Python环境配置及使用

Anaconda的下载与安装

详见《[软件安装与账号注册](#)》

Jupyter Notebook的使用

Jupyter Notebook是一个**网页交互式**的编程环境，适用于**文档编写、代码编写、数据分析和报告生成**。

Jupyter Notebook具有以下特点：

- (1) **交互性**。Jupyter Notebook提供了一个交互式的编程环境，在一个Web浏览器中以单元的形式展示代码和输出结果。我们可以在单元格中编写代码，并即时执行并查看结果。
- (2) **支持多种编程语言**。Jupyter Notebook初始时支持Python，但现在已经可以支持多种编程语言，包括R、Julia、Scala等。每一种编程语言都有对应的内核(kernel)，我们可以选择所需的内核来运行代码。
- (3) **可视化功能**。Jupyter Notebook提供了丰富的可视化功能，我们可以通过各种库和工具创建图表、绘制图像等，以便更好地理解 and 展示数据。
- (4) **文档编写和展示**。Jupyter Notebook不仅可以作为代码编写和运行的工具，还可以用于编写和展示文档。我们可以在Markdown单元格中编写文本，使用Markdown语法来排版和添加链接、图片等。

(5) **共享和协作**。Jupyter Notebook支持将Notebook文件导出为各种格式（如HTML、PDF等），方便与他人共享。此外，我们还可以使用版本控制工具（如Git）对Notebook文件进行管理和协作编辑。

(6) **插件扩展和高级定制**。Jupyter Notebook支持插件扩展，我们可以根据需要添加各种功能和工具。此外，我们还可以进行高级定制，改变界面、主题等。

总之，Jupyter Notebook是一个功能强大、灵活且易于使用的编程环境，适用于数据分析、机器学习、科学计算等各种领域的工作和学习。

快捷键大全

Jupyter Notebook 中有两种模式：

- **命令模式（蓝色边框）**：对单元格进行操作，如创建、删除等。
- **编辑模式（绿色边框）**：在单元格内输入代码或文本。

👉 常用快捷键汇总：

✅ **命令模式快捷键（按 `Esc` 进入）**

快捷键	功能
<code>A</code>	在当前单元格上方插入新单元格
<code>B</code>	在当前单元格下方插入新单元格
<code>D D</code>	删除当前单元格
<code>Z</code>	撤销删除操作
<code>M</code>	将单元格转为Markdown格式
<code>Y</code>	将单元格转为代码格式
<code>X</code>	剪切当前单元格
<code>C</code>	复制当前单元格
<code>V</code>	粘贴单元格
<code>Shift + M</code>	合并选中的单元格
<code>Ctrl + Shift + H</code>	显示快捷键帮助

✔ 编辑模式快捷键（按 **Enter** 进入）

快捷键	功能
Ctrl + Enter	运行当前单元格
Shift + Enter	运行当前单元格并跳到下一个单元格
Alt + Enter	运行当前单元格并插入新单元格
Tab	代码补全
Shift + Tab	显示函数签名或提示信息
Ctrl + /	注释/取消注释选中代码