

实验一

王为 2311605

1 引言

Python 是一种广泛使用的高级编程语言，具有简单明了的语法和强大的功能，被广泛应用于数据分析、人工智能、Web 开发等领域。对于初学者来说，搭建 Python 开发环境，编写基本的输入输出程序，了解 Python 程序的结构，以及掌握程序的运行和调试方法，是迈入编程世界的第一步。本文将详细介绍以上内容，帮助读者快速入门 Python 编程。

2 搭建 Python 开发环境

2.1 安装 Python 解释器

要编写和运行 Python 程序，首先需要安装 Python 解释器。目前，Python 有两个主要版本：Python 2.x 和 Python 3.x。建议选择最新的 Python 3 版本。

安装步骤：

1. 下载 Python 安装包：

- 访问 Python 官网下载页面：<https://www.python.org/downloads/>
- 点击 “Download Python 3.x.x” 下载对应操作系统的安装包。

2. 安装 Python：

- 运行下载的安装包。
- 在安装过程中，勾选 “Add Python 3.x to PATH” 选项，方便在命令行中使用 Python。

- 按照提示完成安装。

3. 验证安装是否成功:

- 打开命令提示符 (Windows) 或终端 (macOS/Linux)。
- 输入 `python --version` 或 `python3 --version`, 查看是否显示 Python 版本信息。

2.2 选择代码编辑器或集成开发环境 (IDE)

选择合适的编辑器或 IDE, 可以提高编程效率。

常用的代码编辑器和 IDE:

- **Visual Studio Code:**
 - 由微软开发, 支持多种编程语言, 具有丰富的扩展插件。
 - 官网: <https://code.visualstudio.com/>
- **PyCharm:**
 - 专业的 Python 集成开发环境, 提供代码自动补全、调试等功能。
 - 官网: <https://www.jetbrains.com/pycharm/>
- **Sublime Text:**
 - 轻量级代码编辑器, 界面简洁, 支持插件扩展。
 - 官网: <https://www.sublimetext.com/>

选择并安装编辑器:

- 以 Visual Studio Code (简称 VS Code) 为例:
 - 下载并安装 VS Code。
 - 安装 Python 插件:
 - * 打开 VS Code, 点击左侧扩展图标 (五角星形状)。
 - * 在搜索框中输入 “Python”, 选择由 Microsoft 发布的插件并安装。

3 编写具有输入输出功能的 Python 程序

3.1 程序需求设计

我们将编写一个简单的 Python 程序，实现以下功能：

- **输入：**提示用户输入一个姓名和一个数字。
- **处理：**程序接收输入，并对数字进行简单计算（例如，平方）。
- **输出：**显示处理结果，并向用户问好。

3.2 Python 程序结构

一个基本的 Python 程序由以下部分组成：

- **注释：**用于解释代码，Python 中使用 # 开头的行表示注释。
- **导入模块：**使用 import 语句导入 Python 标准库或第三方库的模块。
- **函数定义：**使用 def 关键字定义函数，包含可重复使用的代码块。
- **主程序：**执行具体功能的代码。

3.3 编写代码

3.3.1 创建 Python 文件

在代码编辑器中新建一个文件，命名为 `greeting.py`。

3.3.2 编写代码

```
1  # greeting.py
2
3  # -*- coding: utf-8 -*-
4  # 程序功能：一个简单的输入输出程序
5
6  def main():
7      # 提示用户输入姓名
8      name = input("请输入您的姓名：")
9
```

```
10 # 提示用户输入一个数字
11 number_str = input("请输入一个数字: ")
12
13 # 检查输入的是否为数字
14 if number_str.isdigit():
15     number = int(number_str)
16     # 对数字进行平方计算
17     result = number ** 2
18     # 输出结果
19     print(f"您好, {name}! 您输入的数字的平方是: {result}")
20 else:
21     print("输入的不是有效的数字, 请重新运行程序。")
22
23 # 判断是否作为主程序运行
24 if __name__ == "__main__":
25     main()
```

3.3.3 代码解析

- 编码声明:

```
1 # -*- coding: utf-8 -*-
```

这行注释指定了源代码文件的编码格式为 UTF-8, 确保支持中文字符。

- 注释:

```
1 # 程序功能: 一个简单的输入输出程序
```

用于说明程序的作用。

- 定义主函数:

```
1 def main():
2     # 函数体
```

使用 `def` 关键字定义了一个名为 `main` 的函数, 包含程序的主要逻辑。

- 输入姓名:

```
1 name = input("请输入您的姓名: ")
```

使用 `input` 函数提示用户输入,接收输入的字符串并赋值给变量 `name`。

- 输入数字并验证:

```
1 number_str = input("请输入一个数字: ")
2
3 if number_str.isdigit():
4     number = int(number_str)
5     # 后续处理
6 else:
7     print("输入的不是有效的数字, 请重新运行程序。")
```

- `number_str.isdigit()` 用于判断输入的字符串是否只包含数字字符。
- 如果是有效的数字, 转换为整数类型进行计算。
- 否则, 提示输入无效。

- 计算与输出:

```
1 result = number ** 2
2 print(f"您好, {name}! 您输入的数字的平方是: {result}")
```

- 计算数字的平方, 使用 `**` 表示幂运算。
- 使用 `print` 函数输出结果, 采用 Python 3 的格式化字符串 `f-string`。

- 主程序入口:

```
1 if __name__ == "__main__":
2     main()
```

- `__name__` 是一个内置变量, 表示模块的名称。
- 当运行脚本文件时, `__name__` 的值为 `"__main__"`。
- 该条件确保只有在直接运行该脚本时才会调用 `main()` 函数, 避免在被导入时自动执行。

4 运行和调试 Python 程序

4.1 运行程序

4.1.1 在命令行中运行

1. 打开命令行：

- Windows：按 Win + R，输入 cmd，按回车。
- macOS/Linux：打开终端。

2. 导航到脚本所在目录：

- 使用 cd 命令切换目录，例如：
`cd /path/to/your/script`

3. 运行脚本：

- 输入以下命令并回车：

```
python greeting.py
```

或者

```
python3 greeting.py
```

- 根据系统配置，可能需要使用 python3 命令。

4.1.2 在编辑器中运行

- 在 VS Code 中运行：

- 打开 greeting.py 文件。
- 运行代码：
 - * 方法一：在编辑器中点击右上角的“运行”按钮（三角形图标）。
 - * 方法二：按快捷键 Ctrl + F5 (Windows) 或 Cmd + F5 (macOS)。

4.2 程序运行示例

请输入您的姓名：张三

请输入一个数字：5

您好，张三！您输入的数字的平方是：25

4.3 调试方法

4.3.1 使用打印语句调试

可以在代码中插入 `print` 语句，输出变量的值，检查程序的执行流程。

```
1 print(f"Debug: 用户输入的数字是 {number}")
```

4.3.2 使用编辑器的调试功能

- VS Code 调试：

- 设置断点：

- * 在代码行号左侧点击，添加断点（红色圆点）。

- 启动调试：

- * 点击调试按钮（左侧边栏的小虫子图标）。
- * 点击“开始调试”（绿色三角形）。

- 调试操作：

- * 程序运行到断点处会暂停，可查看变量值。
- * 使用调试控制台执行单步运行、继续运行。

4.3.3 常见错误及解决方法

- 语法错误：

- 例如，忘记使用冒号 `:`，括号不匹配。
- 解决：根据错误提示，检查代码，确保语法正确。

- 类型错误：

- 例如，将字符串当作数字计算。

- 解决：在需要时进行类型转换，如 `int(number_str)`。
- **逻辑错误：**
 - 程序没有按照预期运行，可能是条件判断或计算有误。
 - 解决：通过调试，检查变量值和程序流程。

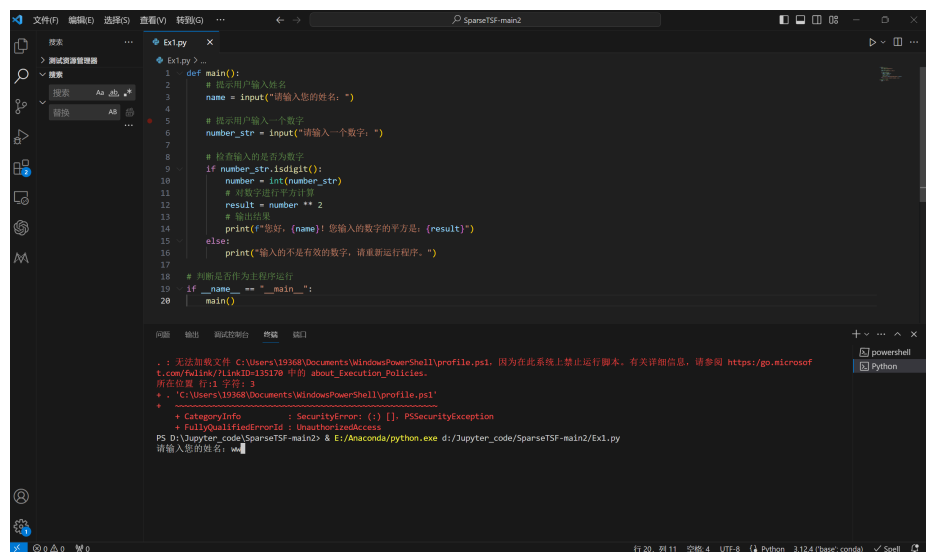
5 总结

通过本文的介绍，我们完成了 Python 开发环境的搭建，编写了一个具有输入输出功能的简单程序，了解了 Python 程序的基本结构，并掌握了运行和调试方法。

6 参考资料

- Python 官方文档: <https://docs.python.org/zh-cn/3/>
- 《Python 编程：从入门到实践》作者：Eric Matthes
- Visual Studio Code 用户指南: <https://code.visualstudio.com/docs>

7 实验图片



The image shows a screenshot of a Visual Studio Code editor window. The editor is open to a file named `Ex1.py`. The code is a Python script that prompts the user for a name and a number, then calculates the square of the number and prints the result. The code is as follows:

```
1. def main():
2.     # 提示用户输入姓名
3.     name = input("请输入您的姓名: ")
4.
5.     # 提示用户输入一个数字
6.     number_str = input("请输入一个数字: ")
7.
8.     # 检查输入的是否为数字
9.     if number_str.isdigit():
10.        number = int(number_str)
11.        # 对数字进行平方计算
12.        result = number ** 2
13.        # 输出结果
14.        print("您好, {name}! 您输入的数字的平方是: {result}")
15.    else:
16.        print("输入的不是有效的数字, 请重新运行程序.")
17.
18. # 判断是否作为主程序运行
19. if __name__ == "__main__":
20.     main()
```

The terminal output shows the execution of the script. It prompts for a name and a number, and then prints the result. The output is as follows:

```
- : 无法加载文件 C:\Users\19368\Documents\WindowsPowerShell\profile.ps1, 因为在此系统上禁止运行脚本。有关详细信息, 请参阅 https://go.microsoft.com/fwlink/?linkID=135170 中的 about_Execution_Policies。
所在位置: 行:1 字符: 3
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS D:\Jupyter_code\SparseTSF-main> & E:/Anaconda/python.exe d:/Jupyter_code/SparseTSF-main/Ex1.py
请输入您的姓名: 123
请输入您的姓名: 123
```

图 1: Enter Caption