

Python入门教程

1. 实验目的

本教程专为零基础的学习者精心设计，旨在帮助学习者从零开始系统地掌握Python编程。通过17个循序渐进的实验，我们将带您深入探索Python的基础语法、数据结构、控制流、函数、库使用、文件操作、异常处理和面向对象编程等核心知识。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链 v3.07及以上版本。
- 软件要求：推荐安装Visual Studio Code，详细配置可见：[\[RflySim安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\3.PythonConfig](#)
- 硬件要求：笔记本/台式电脑1台。

3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\1.RflySimIntro\0.ApiExps\3.Python_Tutorials](#)

- [./Exp01_HelloWorld.py](#)：基础打印输出，了解Python程序基本结构
- [./Exp02_Variables.py](#)：变量的定义和使用，学习不同类型的变量
- [./Exp03_Operators.py](#)：运算符与表达式的使用，掌握算术、比较、逻辑运算符
- [./Exp04_StringBasics.py](#)：字符串的基本操作，包括创建、索引、切片等
- [./Exp05_StringProcessing.py](#)：字符串处理方法，包括分割、替换、查找等
- [./Exp06_ListBasics.py](#)：列表的创建和基本操作，学习列表的常用方法
- [./Exp07_Conditionals.py](#)：条件判断与选择，掌握if、elif、else语句的使用
- [./Exp08_Loops.py](#)：循环结构，学习for和while循环的使用
- [./Exp09_Libraries.py](#)：Python常用库的使用，学习时间、数学、随机数等库
- [./Exp10_Functions.py](#)：函数的定义和调用，掌握参数和返回值的使用
- [./Exp11_TemperatureConverter.py](#)：温度转换综合案例，实现摄氏温度转华氏温度
- [./Exp12_PrimeNumbers.py](#)：质数求解综合案例，学习质数判断算法

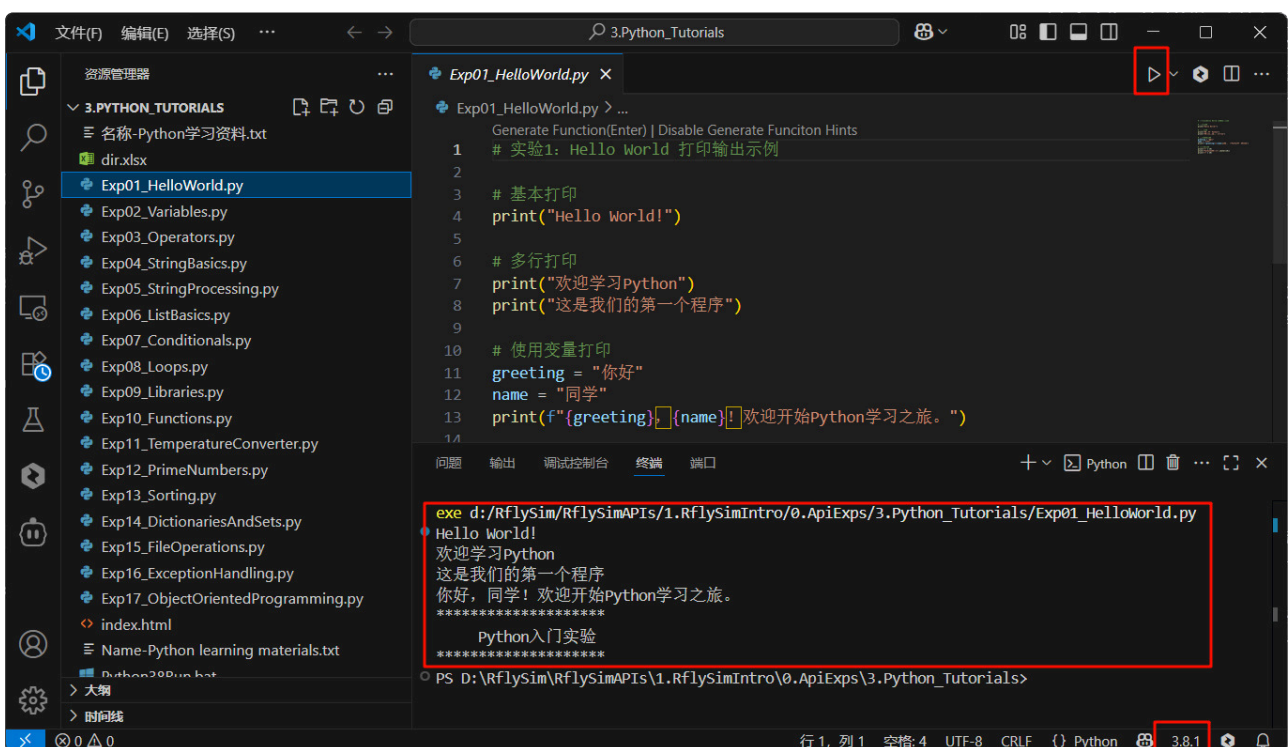
- ./Exp13_Sorting.py：排序算法，学习常见排序方法
- ./Exp14_DictionariesAndSets.py：字典和集合的使用，掌握数据去重和集合运算
- ./Exp15_FileOperations.py：文件操作，学习文件的读写方法
- ./Exp16_ExceptionHandling.py：异常处理机制，掌握try-except语句
- ./Exp17_ObjectOrientedProgramming.py：面向对象编程基础，学习类和对象的概念

4. 实验内容或步骤

本教程提供两种代码运行方式：

方式一：VSCode直接运行

1. 打开VSCode
2. 打开本项目文件夹
3. 右键选择要运行的Python文件
4. 点击"Run Python File in Terminal"



方式二：批处理文件运行

1. 双击 Python38Run.bat
2. 在弹出的命令行窗口中
3. 输入 python Exp**_**.py (将**替换为实验编号)

```
C:\WINDOWS\system32\cmd.exe
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python

D:\RflySim\RflySimAPIs\1.RflySimIntro\0.ApiExps\3.Python_Tutorials>python Exp01>HelloWorld.py
Hello World!
欢迎学习Python
这是我们的第一个程序
你好, 同学! 欢迎开始Python学习之旅。
*****
Python入门实验
*****

D:\RflySim\RflySimAPIs\1.RflySimIntro\0.ApiExps\3.Python_Tutorials>
```

4.1 步骤1: Hello World打印输出

实验目标:

1. 了解Python程序的基本结构
2. 学会使用print()函数
3. 掌握基本的代码编写和运行方法

实验内容:

```
1 | # 基本打印
2 | print("Hello World!")
3 |
4 | # 多行打印
5 | print("欢迎学习Python")
6 | print("这是我们的第一个程序")
7 |
8 | # 使用变量打印
9 | greeting = "你好"
10 | name = "同学"
11 | print(f"{greeting}, {name}! 欢迎开始Python学习之旅。")
```

4.2 步骤2: 认识变量

实验目标:

1. 理解变量的概念
2. 学习不同类型的变量
3. 掌握变量的基本操作

实验内容:

```

1 # 整数变量
2 age = 15
3 print(f"年龄：{age}")
4
5 # 浮点数变量
6 height = 1.75
7 print(f"身高：{height}米")
8
9 # 字符串变量
10 name = "小明"
11 print(f"姓名：{name}")
12
13 # 类型转换
14 str_age = "20"
15 int_age = int(str_age)
16 print(f"字符串转整数：{int_age + 5}")

```

4.3 步骤3：运算符与表达式

实验目标：

1. 学习算术运算符
2. 理解比较运算符
3. 掌握逻辑运算符
4. 了解运算符优先级

实验内容：

```

1 # 算术运算符
2 a, b = 10, 3
3 print(f"加法：{a + b}") # 13
4 print(f"减法：{a - b}") # 7
5 print(f"乘法：{a * b}") # 30
6 print(f"除法：{a / b}") # 3.333
7 print(f"整除：{a // b}") # 3
8 print(f"取余：{a % b}") # 1
9 print(f"幂运算：{a ** b}") # 1000
10
11 # 比较运算符
12 x, y = 5, 7
13 print(f"等于：{x == y}") # False
14 print(f"不等于：{x != y}") # True
15 print(f"大于：{x > y}") # False
16 print(f"小于：{x < y}") # True

```

4.4 步骤4：初始字符串

实验目标：

1. 了解字符串的基本概念
2. 学习字符串的创建方法
3. 掌握基本的字符串操作

实验内容：

```
1 # 字符串的定义
2 name1 = '小明' # 单引号
3 name2 = "小红" # 双引号
4 description = '''这是一个
5 多行字符串的例子'''
6
7 # 字符串索引
8 text = "Python学习"
9 print(f"第一个字符：{text[0]}") # P
10 print(f"最后一个字符：{text[-1]}") # 习
11
12 # 字符串切片
13 print(f"前两个字符：{text[0:2]}") # Py
14 print(f"后两个字符：{text[-2:]}") # 学习
15
16 # 字符串方法
17 greeting = " Hello, World! "
18 print(f"去除空格：'{greeting.strip()}'") # "Hello, World!"
19 print(f"大写：{greeting.upper()}") # " HELLO, WORLD! "
20 print(f"小写：{greeting.lower()}") # " hello, world! "
```

4.5 步骤5：字符串处理

实验目标：

1. 深入学习字符串处理方法
2. 掌握字符串的高级操作
3. 理解字符串处理的实际应用

实验内容：

```

1 # 字符串分割
2 text = "Python,Java,C++,JavaScript"
3 languages = text.split(",")
4 print(f"分割后的语言列表：{languages}")
5
6 # 字符串替换
7 sentence = "I love Python programming"
8 new_sentence = sentence.replace("Python", "Java")
9 print(f"替换后的句子：{new_sentence}")
10
11 # 字符串查找
12 search_text = "Hello, Python is awesome!"
13 index = search_text.find("Python")
14 print(f"'Python'的位置：{index}")

```

4.6 步骤6：初始列表

实验目标：

1. 了解列表的基本概念
2. 学习列表的创建和基本操作
3. 掌握列表的常用方法

实验内容：

```

1 # 创建列表
2 fruits = ["apple", "banana", "cherry"]
3 numbers = [1, 2, 3, 4, 5]
4 mixed_list = [1, "hello", True, 3.14]
5
6 # 访问列表元素
7 print(f"第一个水果：{fruits[0]}")
8 print(f"最后一个数字：{numbers[-1]}")
9
10 # 列表切片
11 print(f"前两个水果：{fruits[:2]}")
12 print(f"后两个数字：{numbers[-2:]}")
13
14 # 列表修改
15 fruits.append("orange")
16 fruits.insert(1, "grape")
17 print(f"添加元素后的水果列表：{fruits}")

```

4.7 步骤7：条件判断与选择

实验目标：

1. 理解条件语句的基本概念
2. 掌握if、elif、else的使用
3. 学习复杂条件的编写

实验内容：

```
1 # 简单的if语句
2 age = 18
3 if age >= 18:
4     print("你已经成年")
5
6 # if-else语句
7 score = 75
8 if score >= 60:
9     print("及格")
10 else:
11     print("未及格")
12
13 # if-elif-else语句
14 grade = 85
15 if grade >= 90:
16     print("优秀")
17 elif grade >= 80:
18     print("良好")
19 elif grade >= 60:
20     print("及格")
21 else:
22     print("不及格")
```

4.8 步骤8：循环 (for、while)

实验目标：

1. 理解循环的基本概念
2. 掌握for循环和while循环
3. 学习循环控制语句

实验内容：

```
1 # for循环遍历列表
2 fruits = ["apple", "banana", "cherry"]
3 for fruit in fruits:
4     print(f"我喜欢吃{fruit}")
5
6 # for循环使用range()
7 for i in range(5):
8     print(f"当前数字: {i}")
9
10 # while循环
11 count = 0
12 while count < 5:
13     print(f"计数: {count}")
14     count += 1
15
16 # break语句
17 for num in range(10):
18     if num == 5:
19         print("找到5, 停止循环")
20         break
21     print(num)
```

4.9 步骤9: 函数

实验目标:

1. 理解函数的基本概念
2. 学习函数的定义和调用
3. 掌握函数参数和返回值的使用

实验内容:

```
1 # 最简单的函数
2 def greet():
3     print("你好, 欢迎学习Python!")
4
5 # 带参数的函数
6 def welcome(name):
7     print(f"欢迎, {name}!")
8
9 # 带返回值的函数
10 def add(a, b):
11     return a + b
12
13 # 默认参数
14 def power(base, exponent=2):
15     return base ** exponent
16
17 # 可变参数
18 def sum_all(*numbers):
19     total = 0
20     for num in numbers:
21         total += num
22     return total
```

4.10 步骤10: Python常用库

实验目标:

1. 了解Python常用标准库
2. 学习时间、数学、随机数等库的使用
3. 掌握库的导入和基本函数调用

实验内容:

```
1 # 导入常用库
2 import time
3 import math
4 import random
5 import datetime
6
7 # 时间操作示例
8 current_timestamp = time.time()
9 formatted_time = time.strftime("%Y-%m-%d %H:%M:%S")
10
11 # 数学运算示例
12 sqrt_value = math.sqrt(16)
13 random_number = random.randint(0, 100)
14
15 # 日期计算示例
16 now = datetime.datetime.now()
17 future_date = now + datetime.timedelta(days=30)
```

4.11 步骤11：综合案例1—摄氏温度转华氏温度

实验目标：

1. 综合运用前面学习的编程知识
2. 实现温度转换的实际应用
3. 学习用户交互和输入处理

实验内容：

```

1  # 温度转换函数
2  def celsius_to_fahrenheit(celsius):
3      """将摄氏温度转换为华氏温度"""
4      fahrenheit = (celsius * 9/5) + 32
5      return fahrenheit
6
7  # 温度转换程序
8  def temperature_converter():
9      """温度转换交互程序"""
10     print("欢迎使用温度转换器!")
11
12     while True:
13         try:
14             # 获取用户输入
15             celsius = float(input("请输入摄氏温度 (输入'q'退出) : "))
16
17             # 转换并显示结果
18             fahrenheit = celsius_to_fahrenheit(celsius)
19             print(f"{celsius}°C = {fahrenheit:.2f}°F")
20
21         except ValueError:
22             # 处理非数字输入
23             user_input = input("输入无效。是否退出? (y/n): ")
24             if user_input.lower() == 'y':
25                 break

```

4.12 步骤12：综合案例2—100以内质数求解

实验目标：

1. 学习质数判断算法
2. 综合运用循环和条件判断
3. 掌握列表操作和函数设计

实验内容：

```
1 # 质数判断函数
2 def is_prime(n):
3     """判断一个数是否为质数"""
4     if n < 2:
5         return False
6     for i in range(2, int(n**0.5) + 1):
7         if n % i == 0:
8             return False
9     return True
10
11 # 查找100以内的质数
12 def find_primes(max_number):
13     """找出指定范围内的所有质数"""
14     primes = []
15     for num in range(2, max_number + 1):
16         if is_prime(num):
17             primes.append(num)
18     return primes
```

4.13 步骤13：综合案例3—排序

实验目标：

1. 学习常见排序算法
2. 理解排序的基本原理
3. 掌握Python中的排序方法

实验内容：

```

1 # 冒泡排序算法
2 def bubble_sort(arr):
3     """冒泡排序：每次比较相邻元素，将最大元素"冒泡"到末尾"""
4     n = len(arr)
5     for i in range(n):
6         # 标记是否发生交换，优化算法
7         swapped = False
8
9         for j in range(0, n - i - 1):
10            # 如果前一个元素大于后一个元素，交换
11            if arr[j] > arr[j + 1]:
12                arr[j], arr[j + 1] = arr[j + 1], arr[j]
13                swapped = True
14
15            # 如果没有发生交换，说明已经排序完成
16            if not swapped:
17                break
18
19        return arr
20
21 # 选择排序算法
22 def selection_sort(arr):
23     """选择排序：每次选择最小元素放到前面"""
24     n = len(arr)
25     for i in range(n):
26         # 假设当前位置是最小值
27         min_idx = i
28
29         # 在未排序区间找最小值
30         for j in range(i + 1, n):
31             if arr[j] < arr[min_idx]:
32                 min_idx = j
33
34         # 交换最小值到正确位置
35         arr[i], arr[min_idx] = arr[min_idx], arr[i]
36
37     return arr

```

4.14 步骤14：字典和集合

实验目标：

1. 了解字典的基本概念和操作
2. 学习集合的特性和常用方法
3. 掌握数据去重和集合运算

实验内容：

```

1 # 字典操作
2 student = {
3     "name": "小明",
4     "age": 18,
5     "subjects": ["数学", "英语", "编程"]
6 }
7
8 # 集合操作
9 fruits1 = {"apple", "banana", "cherry"}
10 fruits2 = {"banana", "date", "elderberry"}
11
12 # 集合运算
13 common_fruits = fruits1.intersection(fruits2)
14 all_fruits = fruits1.union(fruits2)

```

4.15 步骤15：文件操作

实验目标：

1. 学习文件的读写操作
2. 掌握文本和CSV文件处理
3. 了解文件操作的异常处理

实验内容：

```

1 # 写入文本文件
2 with open("sample.txt", "w", encoding="utf-8") as file:
3     file.write("Python文件操作示例")
4
5 # 读取文本文件
6 with open("sample.txt", "r", encoding="utf-8") as file:
7     content = file.read()
8
9 # CSV文件操作
10 import csv
11 students = [
12     ["姓名", "年龄", "班级"],
13     ["小明", 18, "高三1班"]
14 ]
15 with open("students.csv", "w", newline="") as file:
16     csv.writer(file).writerows(students)

```

4.16 步骤16：异常处理

实验目标：

1. 理解异常的基本概念
2. 学习try-except语句
3. 掌握常见异常处理方法

实验内容：

```
1  try:
2      x = int(input("请输入数字："))
3      result = 10 / x
4  except ValueError:
5      print("请输入有效数字")
6  except ZeroDivisionError:
7      print("不能除以零")
8
9  # 自定义异常
10 class AgeError(Exception):
11     def __init__(self, age):
12         self.message = f"年龄{age}不valid"
13         super().__init__(self.message)
14
15     def check_age(age):
16         if age < 0 or age > 120:
17             raise AgeError(age)
```

4.17 步骤17：面向对象编程基础

实验目标：

1. 理解类和对象的概念
2. 学习类的定义和使用
3. 掌握继承和方法重写

实验内容：

```

1 class Student:
2     def __init__(self, name, age):
3         self.name = name
4         self.age = age
5         self.scores = {}
6
7     def add_score(self, subject, score):
8         self.scores[subject] = score
9
10    def get_average_score(self):
11        return sum(self.scores.values()) / len(self.scores)
12
13    # 继承
14    class SportStudent(Student):
15        def __init__(self, name, age, sport):
16            super().__init__(name, age)
17            self.sport = sport

```

5. 关键知识点

关键知识点1：Python基础语法

包括print()函数的使用、变量的定义、基本数据类型(int, float, str, bool)、类型转换方法(int(), float(), str(), bool())、代码编写规范等。

关键知识点2：运算符

包括算术运算符(+, -, *, /, //, %, **)、比较运算符(==, !=, >, <, >=, <=)、逻辑运算符(and, or, not)、运算符优先级和短路运算。

关键知识点3：字符串处理

包括字符串定义(单引号、双引号、三引号)、字符串索引(从0开始的正向和反向索引)、字符串切片(提取子串的方法)、字符串方法(strip(), upper(), lower(), split(), replace(), find())、字符串不可变性和格式化。

关键知识点4：列表操作

包括列表创建(使用方括号)、列表索引(正向和反向索引)、列表切片(提取部分列表)、列表方法(append(), insert(), remove(), pop())等。

| 关键知识点5：条件判断与循环

包括条件语句(if, elif, else)、循环(for, while)、循环控制(break, continue)、range()函数的使用等。

| 关键知识点6：函数定义与使用

包括函数定义(def关键字)、函数参数(普通参数、默认参数、可变参数)、返回值(return语句)、函数调用等。

| 关键知识点7：Python常用库

包括库的导入(import语句)、时间库(time)、数学库(math)、随机数库(random)、日期库(datetime)等常用功能。

| 关键知识点8：文件操作

包括文件的打开与关闭、读写模式(r, w, a)、with语句的使用、CSV文件处理、文件编码指定等。

| 关键知识点9：异常处理

包括异常类型(ValueError, ZeroDivisionError等)、异常捕获(try-except语句)、自定义异常等。

| 关键知识点10：面向对象编程

包括类的定义(class关键字)、构造方法(**init**)、实例方法、属性访问、继承(父类与子类)等。

6. 参考资料

1. [RflySim官方文档](#)
2. [Python官方文档](#)
3. [Python入门教程](#)

7. 常见问题

Q1: 如何在VSCode中运行Python文件?

A1: 在VSCode中打开项目文件夹，右键点击要运行的Python文件，选择"Run Python File in Terminal"即可运行。

Q2: 如何使用批处理文件运行Python代码?

A2: 双击运行 `Python38Run.bat` 文件，在弹出的命令行窗口中输入 `python Exp**_***.py`（将**替换为实验编号）来运行相应的实验文件。

Q3: 为什么我的Python代码运行报错?

A3: 请检查Python环境是否正确安装，代码是否有语法错误，以及是否按要求安装了必要的库。如果遇到特定错误，可以根据错误提示信息搜索解决方案。