一、核心流程控制语句

1. 循环结构

python

复制

# for循环（固定次数迭代）for i in range(3): print(i)# while循环（条件驱动）count = 0while count < 3: print(count) count += 1# 循环控制语句for i in range(5): if i == 2: continue # 跳过本次迭代 if i == 4: break # 提前终止循环

2. 分支结构

python

复制

# 多条件分支score = 85if score >= 90: grade = 'A'elif score >= 80: grade = 'B' # 本例执行此分支else: grade = 'C'# 异常处理结构try: result = 10 / 0except ZeroDivisionError: print("除零错误")except Exception as e: print(f"未知错误: {e}")else: print("无异常时执行")finally: print("总会执行的清理代码")

二、模块与包

1. 模块化开发

python

复制

# mylib.pydef func8(\*args): """接收任意数量位置参数""" print(f"接收参数: {args}")# myjob.pyfrom mypkg.mylib import func8func8(1, 2, 3) # 输出: 接收参数: (1, 2, 3)

2. 参数类型

| 参数类型 | 示例 | 说明 |
| --- | --- | --- |
| 位置参数 | def func3(a): | 必须传入且顺序固定 |
| 命名参数 | def func4(b=10): | 可省略（使用默认值） |
| 仅位置参数 | def func6(a, /): | 必须用位置方式传递 |
| 仅命名参数 | def func7(\*, a): | 必须用命名方式传递 |
| 可变位置参数 | def func8(\*args): | 接收元组 |
| 可变命名参数 | def func9(​\*\*​kwargs): | 接收字典 |

3. 参数解包

python

复制

# 元组解包def func10(a, b, c=0): print(a, b, c)func10(\*[1, 2], c=3) # 输出: 1 2 3# 字典解包def func11(a, b=0, c=0): print(a, b, c)func11(​\*\*​{'a': 10, 'c': 30}) # 输出: 10 0 30

三、调试与异常

​​pdb调试器​​

python

复制

import pdbdef demo(): x = 10 pdb.set\_trace() # 设置断点 y = x \* 2 print(y)

常用命令：n(下一行), s(进入函数), l(查看代码), p(打印变量)

​​异常处理实践​​

python

复制

def safe\_division(dividend, divisor): try: return dividend / divisor except ZeroDivisionError: print("警告：除数不能为零") return float('inf') except TypeError: raise ValueError("输入必须为数值类型")

四、最佳实践

​​类型注解​​

python

复制

def func12(a: int, b: str) -> str: """文档字符串示例 Args: a: 整型参数 b: 字符串参数 Returns: 拼接后的字符串 """ return f"{a}-{b}"

​​包管理原则​​

使用 \_\_init\_\_.py 控制模块导入

通过 pyproject.toml 定义包元数据

toml

复制

[build-system]requires = ["setuptools>=42"]build-backend = "setuptools.build\_meta"

五、常见问题

​​ImportError解决方案​​

检查包目录结构是否完整

验证 \_\_init\_\_.py 文件存在

确认已通过 pip install -e . 安装

​​参数传递错误​​

位置参数必须在命名参数之前

仅位置参数不能使用命名方式传递

可变参数后的参数必须为命名参数