《Python程序设计基础》（精简版）

第一单元 打开编程的大门

1.1 编程世界初探

- **程序**：执行特定任务的**指令集合**，描述了计算机解决问题的工作步骤。
- **程序设计语言**：用于编写指令集合的**形式化语言**。
  - **低级语言**：
    - **机器语言**：由二进制代码（0/1）组成，硬件直接执行。
    - **汇编语言**：使用助记符（如`MOV`）代替二进制，需**汇编器**转换。
  - **高级语言**：接近自然语言，易读易维护，需通过编译或解释转换。

语言	特点	应用
`C`	高效、控制力强	系统/嵌入式开发
`C++`	面向对象、高性能	游戏/高性能应用
`Python`	易读、库丰富	Web/数据/AI
`Java`	跨平台、生态强	企业级/安卓开发
`JavaScript`	Web核心	前端/部分后端

1.2 初识Python语言

1. **发展简史**
   - 1989年：**吉多·范罗苏姆**开始创建。
   - 1991年：`0.9.0`发布。2000年：`2.0`发布（引入垃圾回收）。
   - 2008年：`3.0`发布（不兼容2.x）。2020年：`2.7`终结。
2. **特点**
   - **简洁清晰**：强调可读性。
   - **免费开源**：自由使用分发。
   - **跨平台**：Write Once, Run Anywhere。
   - **库丰富**：`NumPy`(数据), `Matplotlib`(可视化), `Flask`(Web), `PyTorch`(AI)。
   - **解释型**：逐行解释执行，无需预编译。
3. **应用**：Web开发、人工智能、网络爬虫、游戏开发（`Pygame`）。
4. **面向对象**：以**对象**为核心。
   - **类(`Class`)**：对象的模板。**对象(`Object`)**：类的实例。
   - **属性**：特征信息。**行为**：执行动作。

1.3 迈出Python编程的第一步

1. **环境**：选择版本 -> 安装配置PATH -> `python --version`检查。
2. **文件**：保存为 `.py` 扩展名。
3. **编辑器对比**：

功能	专业版	社区版
价格	付费	**免费**
Web/数据库	支持	有限/不支持

第二单元 与Python语言熟悉起来

2.1 走进算法：流程图

1. **算法**：解决问题的计算步骤。
2. **流程图符号**：

符号	名称	作用
圆角矩形	起止框	开始/结束
平行四边形	输入输出框	数据I/O
矩形	处理框	计算/赋值
菱形	判断框	条件判断
箭头	流程线	执行方向

符号	名称	作用
圆圈	连接点	流程连接

3. **三种基本结构**
    - **顺序结构**：自上而下逐条执行。
    - **选择结构**：根据条件判断执行不同分支。
    - **循环结构**：满足条件时重复执行。

2.2 触摸生活中的标志：`turtle`

1. **概述**：`turtle`（海龟）是Python内置绘图模块。
    - **起始**：坐标`(0, 0)`，方向**向右。**
    - **导入**：`import turtle`
2. **常用语句**

语句	简写	描述
`forward(n)`	`fd(n)`	前进`n`像素
`backward(n)`	`bk(n)`	后退`n`像素
`left(n)`	`lt(n)`	左转`n`度
`right(n)`	`rt(n)`	右转`n`度
`goto(x, y)`	-	移至`(x, y)`
`color(m, n)`	-	`m`画笔色，`n`填充色
`begin/end_fill()`	-	填充闭合图形（成对使用）
`penup/pendown()`	`pu/pd`	提笔/落笔（成对使用）
`circle(r, extent)`	-	画圆：`r`半径，`extent`弧度(正逆负顺)
`pensize(n)`	-	画笔粗细
`done()`	-	结束绘画（不关闭窗口）

2.3 初探Python基础知识：常见数据标识与语句

1. **常见标识符**
    - **常量**：运行过程中值**保持不变**的量。
    - **变量**：运行过程中值**会发生变化**的量。
        - **命名规则**：只能包含字母、数字、下划线；**不能以数字开头**；**区分大小写**；建议**见名知意。**
        - **命名规范**：
            - **大驼峰**：所有单词首字母大写（UserClass）。
            - **小驼峰**：首单词首字母小写，其余大写（userName）。

        - **下划线**：单词间用下划线连接（user_name）。
    - **注意事项**：
        - 禁止使用**系统关键字**（可通过`import keyword; print(keyword.kwlist)`查看）。
        - 不建议使用**内置函数/类型/模块名**（可通过`dir(__builtins__)`查看）。
2. **基本语句**
    - **赋值语句**：将等号右边的值输送到等号左边的变量中。
    - **输入语句**：`variable = input("提示文字")`，接收内容均为**字符串**。
    - **输出语句**：`print(内容)`，将结果输出到控制台。

2.4 体会程序编写规范：命名与注释

1. **注释**：被解释器忽略，用于说明代码。
    - **单行注释**：以`#`开头。
    - **多行注释**：使用三个单引号`'''`或双引号`"""`包裹。
2. **代码缩进**：Python利用**缩进**（通常4个空格）和**冒号**区分代码块层次，同一代码块缩进必须一致。
3. **空行**：用于分隔不同功能代码，便于维护。
4. **多行语句**：
    - 使用反斜杠`\`续行。
    - 在`[]`、`{}`、`()`内可直接换行。
5. **同行多语句**：使用分号`;`分隔。

第三单元 程序世界中的数据奥秘

3.1 数据类型的概念

- **常用数据类型**

类型	名称	描述	特点
数值	`Numeric`	表示数字	用于数学运算
字符串	`String`	文本/字符序列	**有序、不可变**
列表	`List`	有序集合	**有序、可变**
元组	`Tuple`	不可变序列	**有序、不可变**
集合	`Set`	无序不重复集合	**无序、可变**、去重
字典	`Dict`	键值对集合	**无序、可变**、键唯一

- **分类方式**
    - **有序**：字符串、列表、元组
    - **无序**：字典、集合
    - **可变**：列表、字典、集合
    - **不可变**：数值、字符串、元组

3.2 跳出数字的舞蹈：数值类型

1. **分类**
   - **整数 (`int`)**：无小数，范围仅受内存限制。
   - **浮点数 (`float`)**：含小数，支持科学计数法（`1.2e3`）。
   - **复数 (`complex`)**：含实部和虚部（`j`），如 `1+2j`。
2. **运算**
   - **算术运算**：`+`, `-`, `*`, `/` (浮点除), `//` (整除), `%` (取余), `**` (幂)。优先级：`**` > `+`/`-` (正负) > `*`/`/` > `+`/`-` (加减)。
   - **比较运算**：`==`, `!=`, `>`, `<`, `>=`, `<=`。结果为 `True` (1) 或 `False` (0)。
   - **逻辑运算**：`and` (与), `or` (或), `not` (非)。优先级：`not` > `and` > `or`。结果不限于布尔值。
   - **优先级**：算术 > 比较 > 逻辑。
3. **常用函数**

函数	描述	示例
`abs(x)`	绝对值	`abs(-1.5)` → `1.5`
`max/min()`	最值	`max(1,9)` → `9`
`pow(x, y)`	x的y次幂	`pow(2, 3)` → `8`
`math.sqrt(x)`	平方根	`sqrt(100)` → `10.0`

4. **类型转换**
   - `int(x)`：转整数。
   - `float(x)`：转浮点数。
   - `eval(str)`：执行字符串表达式并返回数值。

3.3 奏响文本的旋律：字符串类型

1. **索引与切片**
   - 格式：`str[start:stop:step]`。
   - 规则：索引从0开始；**左闭右开**（含start不含stop）；支持负数索引（倒数）；`step`为步长（负数表示逆向）。
2. **格式化输出**

方法	示例
`format`	`"{} {}".format(a, b)`
`f-string`	`f"{a} {b}"`
`%`操作符	`"%s %d" % (a, b)`

   - **对齐与填充**：`{:<10}`(左对齐), `{:>10}`(右对齐), `{:^10}`(居中), `{:*^10}`(居中填充*)。
   - **精度**：`{:.2f}` 保留2位小数。

3. **运算**

运算符	描述
`+`	连接
`*`	重复
`>`	字典序比较 (ASCII)
`in / not in`	包含判断

   - **注意**：中文比较不能直接用比较运算符。

4. **常用函数**

函数	描述	示例
`len(s)`	长度	`len("Hi")` → `2`
`max/min(s)`	最值字符	`max("abc")` → `'c'`
`s.replace(o, n)`	替换	`s.replace('a','b')`
`s.find(x)`	查找索引 (无则-1)	`s.find('a')`
`s.index(x)`	查找索引 (无则报错)	`s.index('a')`
`j.join(s)`	连接	`','.join("abc")` → `"a,b,c"`
`s.upper/lower()`	大小写转换	`s.upper()`
`s.count(x)`	计数	`s.count('a')`
`s.split(x)`	分割	`s.split(',')`
`s.strip()`	去两端空格	`s.strip()`

5. **类型转换**

   - `str(x)`：将x转换为字符串。

3.4 打开数据的宝箱：列表类型

1. **访问**

   - **索引**：`list[i]` (从0开始)。
   - **切片**：`list[start:stop]` (左闭右开)。

2. **运算**

运算符	描述
`+`	合并列表
`*`	重复列表
`==` / `>`	逐元素比较

运算符	描述
`in` / `not in`	成员判断

3. **常用函数**

函数	描述	示例
`append(x)`	末尾追加	`l.append(1)`
`extend(seq)`	末尾扩展	`l.extend([2,3])`
`insert(i, x)`	插入	`l.insert(0, 1)`
`remove(x)`	移除首个匹配	`l.remove(1)`
`pop(i)`	移除并返回 (默认最后)	`l.pop()`
`index(x)`	查找索引	`l.index(1)`
`count(x)`	计数	`l.count(1)`
`sort()`	排序 (修改原列表)	`l.sort()`
`reverse()`	反转 (修改原列表)	`l.reverse()`

4. **类型转换**
   - `list(x)`：将序列x转换为列表。

第四单元 控制程序的“指挥棒”

4.1 顺序结构与选择结构

1. **顺序结构**：程序默认从上到下、从左到右依次执行。
2. **选择结构**：根据条件选择执行路径。
   - **单分支** (`if`)：条件成立则执行。
   - **双分支** (`if...else`)：成立执行A，否则执行B。
   - **多分支** (`if...elif...else`)：满足不同条件执行对应代码。

4.2 循环结构

1. `for` **循环**
   - **遍历序列**：

```
for item in sequence:  # 遍历字符串、列表等
    pass
```

   - **范围循环** (`range`)：

```
for i in range(start, stop, step):
    pass
```

- `start`：起始值（默认为0）。
- `stop`：结束值（**不含**，不可省略）。
- `step`：步长（默认为1）。

2. `while`**循环**
   - **条件循环**：

     ```
     while condition:  # 条件为真时持续执行
         pass
     ```

   - **无限循环**：`while True:` 配合 `break` 使用。
3. **控制语句**
   - `break`：**终止**整个循环。
   - `continue`：**跳过**本次循环，进入下一次。

4.3 流程嵌套与算法

1. **流程嵌套**：控制结构内部包含另一个控制结构（如循环嵌循环，if嵌if）。
2. **枚举算法**：穷举所有可能，检查是否满足条件（循环+选择）。
3. **冒泡排序**：
   - **原理**：相邻元素比较交换，大值上浮。
   - **作用**：将无序列表转为有序。
4. **程序调试**：
   - **断点**：点击行号设置/取消。
   - **操作**：单步执行，监视变量。

第五单元

5.1 内置函数与模块

1. **模块导入**
   - `import 模块名` -> `模块名.函数()`
   - `from 模块名 import 函数名/*` -> `函数名()`
   - `import 模块名 as 别名` -> `别名.函数()`
2. **常用模块**

模块	函数	描述
`math`	`sqrt(x)`	平方根 (返回浮点)
	`pow(x,y)`	x的y次幂
`random`	`randint(a,b)`	[a,b] 随机整数
	`random()`	[0.0, 1.0) 随机浮点
`time`	`time()`	时间戳

模块	函数	描述
	`sleep(s)`	暂停s秒
`string`	`digits`	数字 0-9
	`ascii_letters`	所有英文字母
`os`	`system(cmd)`	执行系统命令
	`getcwd()`	获取当前目录

5.2 自定义函数

1. **定义**

```
def 函数名(参数):
    函数体
    return 返回值
```

2. **参数**：位置参数、关键字参数、默认参数、不定长参数。
3. **作用域**
    - **局部变量**：函数内定义，仅函数内可用。
    - **全局变量**：函数外定义，全局可用。

5.3 异常处理

1. **异常类型**：语法错误（代码不对）、运行时错误（逻辑/输入问题）、异常（不可预料）。
2. **处理机制**

```
try:
    # 可能出错的代码
except:
    # 出错后执行
else:
    # 未出错执行
finally:
    # 无论是否出错都执行
```

出处：23数媒2班 陆云清