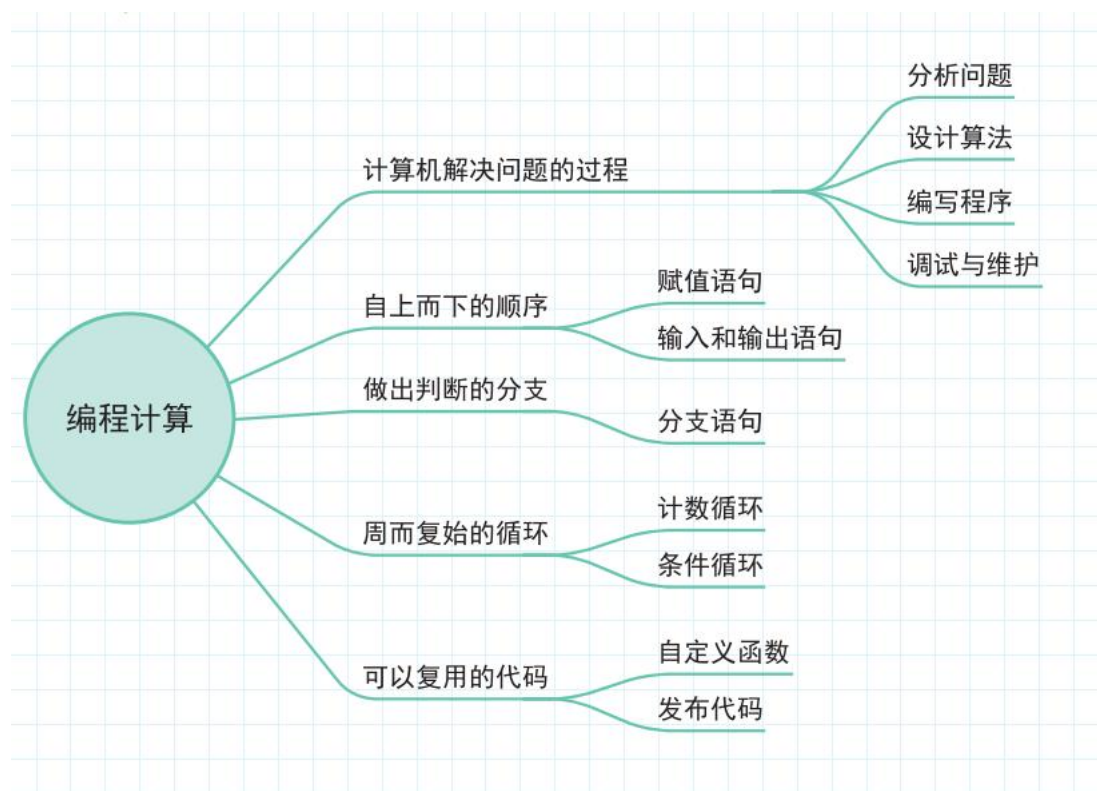


必修 1 第二单元：编程计算（知识梳理）

单元结构如下：



▶▶▶ 程序设计语言的基本知识 ◀◀◀

1.1 Python 语法基础

【知识链接】

1.常量与变量。

常量是指在程序运行过程中值始终保持不变的量。

变量是指在程序运行过程中值允许改变的量。在 Python 中，变量名必须以字母或下划线开头，后面可以是字母、数字或下划线；变量名区分大小写，同时不能使用系统保留字。

2.数据类型。

Python 提供了**整数 (int)**、**浮点数 (float)**、**字符串(str)** **布尔 (bool)**、**列表(list)** 等常见的数据类型。部分数据类型之间可以进行转换，如"int (x) "表示将" x"转换为整型数据。

3.运算符。

(1) 算术运算符： 是进行某种运算的标识符号，它标明对操作数进行的运算。常用的算术运算符有加(+)、减(-)、乘(*)、幂(**)除(/)、整除(//)、取模(%)等。

(2) 关系运算符：用于式子的大小比较。常用的关系运算符有**等于(=)**、**大于(>)**、**大于等于(>=)**、**小于(<)**、**小于等于(<=)**、**不等于(!=)**等。

(3) 逻辑运算符： 可以对关系式或布尔值进行逻辑运算，结果仍为布尔值"True" 或"False" 。 常用的逻辑运算符有**与 (and)**或 **(or)**、**非(not)** 等。

4.表达式。

(1) 算术表达式： 由一个或多个操作数通过运算符组合而成的式子。

(2) 关系表达式： 用关系运算符连接起来的式子，结果为布尔值。如分支语句中的条件表达式，当条件**成立时值为真(True)**，**不成立时值**

为假(False) 。

(3) 逻辑表达式：用 逻辑运算符连接起来的式子，结果为布尔值。

当逻辑表达式成立时值为真(True)，不成立时值为假(False)。如逻辑表达式" (2 < 3) or (3 > 4)" 的值为" True" 。

5.赋值语句。

将赋值号右边表达式的值赋给左边变量的语句。基本格式为： 变量 = 表达式。

6.输入和输出语句。

在 Python 中，使用" input () " 语句从键盘输入获取数据 ； 使用" print () " 语句在屏幕上输出数据。如有多项数据输出，用","分隔需要输出的内容。

1.2 分支结构

【知识链接】

1.分支结构

当程序中需要根据不同的条件判断来决定程序执行的走向时，可以使用 IF 分支语句来实现 。 常见的分支语句有单分支语句 、双分支语句和多分支语句。

(1) 单分支语句的一般格式：

```
if 条件:
    语句或语句组
```

条件成立 (True) 时执行语句或语句组, 不成立 (False) 时不执行。
注意: 条件后面的冒号不能省略。

(2) 双分支语句的一般格式:

```
if 条件:
    语句或语句组 1
else:
    语句或语句组 2
```

条件成立 (True) 时执行语句或语句组 1, 不成立 (False) 时执行语句或语句组 2。注意: else 与 if 要对齐。

(3) 多分支语句的一般格式:

```
if 条件 1:
    语句或语句组 1
elif 条件 2:
    语句或语句组 2
.....
elif 条件 N:
    语句或语句组 N
else:
    语句或语句组 N+1
```

条件 1 成立 (True) 时执行语句或语句组 1; 条件 2 成立时执行语句或语句组 2;; 条件 N 成立时执行语句或语句组 N; 如果以上都不成立, 则执行语句或语句组 N+1。elif 为 else if 的简写, else 及其后面的语句或语句组可以省略。

2. 代码的缩进

Python 用缩进 (在相应的代码前加空格) 来表示代码之间的从属关系, 并且允许多层嵌套。

1.3 循环结构

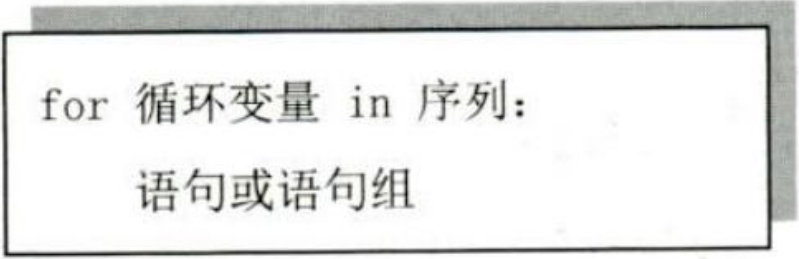
【知识链接】

1.列表

列表可以包含多个数据元素，它是一个有序的容器。列表中的元素允许是不同的数据类型，其索引下标从 "0" 开始。允许对列表中的元素进行赋值、删除（`remove`）、插入（`insert`）、添加（`append`）等操作。用 `list()` 函数可以将集合或字符串类型的数据转换成列表类型的数据。

2.计数循环。

循环次数一定时，常使用计数循环（for 语句）来实现。for 语句的基本格式为：



```
for 循环变量 in 序列:  
    语句或语句组
```

缩进的语句或语句组称循环体。Python 规定循环体必须相对关键词 `for` 有缩进。in 后面也可以是 `range(a, b)` 函数，它可以生成指定范围 `[a, b)` 的数字序列；如果只有一个参数 `a`，则表示生成范围 `[0, a)` 的数字序列。

3.条件循环

条件循环（while 语句）一般用于循环次数未知的情况。

while 语句的基本格式为：

```
while 表达式:  
    语句或语句组
```

当表达式成立时，执行循环体。Python 允许表达式为数字，如果该数字为"0" 则表示表达式为假（即不成立），为非 0 数字则表示为真（即成立）。

4.循环嵌套

一个循环结构的内部也可以包含另一个循环，这样的结构称为循环嵌套（也称为多重循环）。常用的循环嵌套是二重循环，外层循环称为外循环，内层循环称为内循环。内循环是外循环的循环体，外循环每执行一次，内循环则执行一次完整的循环。以计数循环 for 语句为例的二重循环格式如下：

```
for 循环变量1 in 序列1:           #外循环  
    for 循环变量2 in 序列2:       #内循环，注意缩进  
        语句或语句组             #循环体，注意缩进
```

5.break 语句和 continue 语句

break 语句用于终止某个循环，使程序跳到循环体以外的第一个可执行语句；**continue** 语句用于终止本次循环进入下一次循环，即跳过循环体内还未执行的语句，进入下一次循环的判定。

1.4 自定义函数

【知识链接】

1.代码复用。

程序中多处用到功能相似或相同的代码时可以用代码复用的方法来提高程序的效率。通常使用“函数”和“模块”等方法实现。

代码复用可以减少重复编写代码的工作量，也可以使程序的结构更加清晰。

2.函数

函数分为系统函数和用户自定义函数。

系统函数即函数库里的标准函数，是程序设计语言或操作系统提供给用户的一系列已经编写好的程序。如 `print()` 函数和 `input()` 函数等。

用户自定义函数是用户自己编写的程序代码。一般包括函数名、参数、返回值和函数体等四个部分。定义的格式如下：

```
def 函数名(参数):  
    语句或语句组  
    return 返回值
```

需要说明的是：

- (1) 函数名和函数体必不可少；
- (2) 参数是用来向函数传递值的，可以省略(空括号不能省略)；
参数也可以有多个，中间用 " " 分隔；
- (3) 函数执行完毕，由 `return` 语句将值返回给调用者；如果没有返回值，该语句可以省略；
- (4) 采用函数名（参数列表）的方式调用函数。

3.发布代码

将定义的函数发布成模块可以让更多人受益于你的劳动。

4.模块的导入

无论是用户定义的模块还是第三方模块，在使用前都必须先导入程序中才可以使用。常见的模块导入方法有以下两种。

方法 1: `import 模块名 as 别名` `as 别名` 可以省略

方法 2: `from 模块名 import 方法名` 如果方法名为 `*`，表示导入该模块的所有方法
使用方法 1 导入模块时，调用其内置函数的方法是“模块名.方法名(参数)；使“用方法 2 导入模块 时，调用

其内置函数的 方法是“方法名（参数）”。

1.5 程序的调试与运行

【知识链接】

1. 调试程序

程序发生错误后，寻找产生错误的原因、位置，并改正错误的过程称为调试程序。常见的调试方法有以下几种。

(1) 利用 Python 中的调试器 pdb 调试程序。调试的步骤是：

- ① 在程序的开头加入语句" import pdb"，导入 pdb 调试器；
- ② 在程序中需要调试的位置设置断点" pdb. set_trace()"，运行到此处时程序暂停运行，进入 pdb 调试环境；
- ③ 用命令" p 变址 “查看变址的值；用命令" c"继续运行；用命令 " q" 结束调试。

(2) 利用 print () 函数输出变址随程序运行时不断变化的值，找到问题所在。

(3) 在 Python Shell 窗口中逐条输入语句，观察变址的值和最终输出的结果。

2.程序的运行。

在 Python 的 ID LE 窗口中执行 " Run"菜单中的" Run Mod ule"（或按 F5 键）命令即可运行程序。

程序在运行和调试过程中可能会出现**两类错误**： 一类是语法错

误,即程序的书写不符合 Python 语言的语法要求;一类是逻辑错误,即程序可以运行,但结果不符合预期。