

Python程序设计07-复习

北京大学 陈斌

2019.04.22

目录

- 程序的基本概念
- 数据及数据类型
- 基本计算过程
- 基本数据类型
- 海龟作图：turtle
- 基本控制流程
- 函数和递归
- 硬件编程基础：microbit



程序的基本概念

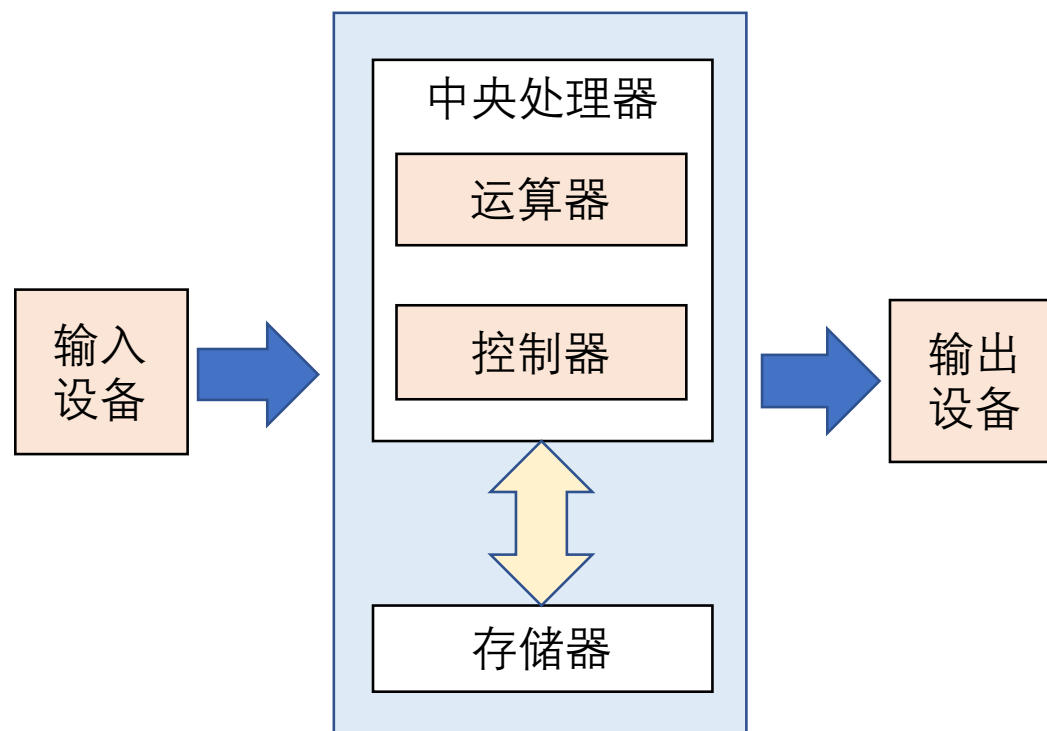
- “程序”就是做一件事情或者解决一个问题所采取的一系列固定步骤
- 程序设计语言经历了机器语言、汇编语言、高级语言等几个发展阶段
- Python语言是高级程序设计语言，应用范围广泛，可以运行在从单片机、PC到大型服务器等各种不同规模的计算机上
- Python语言解释器可以有两种执行方式：单条语句立即执行的Shell和编辑源代码批量执行的Editor
- Python语言程序源代码的主要特征：分行、左对齐、缩进、注释

数据及数据类型

- 数据(data)是信息的表现形式和载体，是对现实世界实体和概念的抽象
- 数据有各种类型，每种数据类型都有自己独特的运算和处理方式
 - 整数、浮点数、字符串、逻辑、日期时间、多媒体数据类型等
 - 组织数据的容器类型：列表、元组、集合、字典
- 对需要处理的信息进行数据类型的分析
 - 区分实体的不同属性
 - 按照属性的取值范围和所要进行的运算处理，来决定数据类型

基本计算过程

- 冯诺伊曼体系结构计算机
- 基本的计算过程
 - 表达式计算
 - 存储和赋值
- 计算控制流程
 - 顺序结构
 - 条件分支结构
 - 循环结构
- 程序流程图



基本数据类型

- 整数int、浮点数float、复数complex
 - 常用的数值运算 (+、-、*、**、/、//、%)
 - 常用的比较 (>、<、>=、<=、==、!=、连续比较)
 - 常用的函数 (math.sqrt, math.sin等)
- 数据对象、命名、变量赋值
 - 名字的规则，名字与变量、几种赋值语句
- 海龟作图的基本程序框架、常用的绘图函数
- 逻辑类型bool (True/False)
 - 比较返回逻辑值；常用的逻辑运算 (and、or、not)

基本数据类型

- 字符串str
 - 引号表示字符串
 - 字符串的常用运算 (+、*、len、[start:end:step])
- 数据类型转换
- 输入函数input
- 输出函数print
- 完整的Python程序结构

```
# 程序功能:  
# 找到不小于用户输入数的最小质数
```

```
# 1, 导入需要的模块  
import math
```

```
# 2, 定义函数  
def isprime(n):  
    for i in range(2, int(math.sqrt(n)) + 1):  
        if n % i == 0:  
            return False  
    else:  
        return True
```

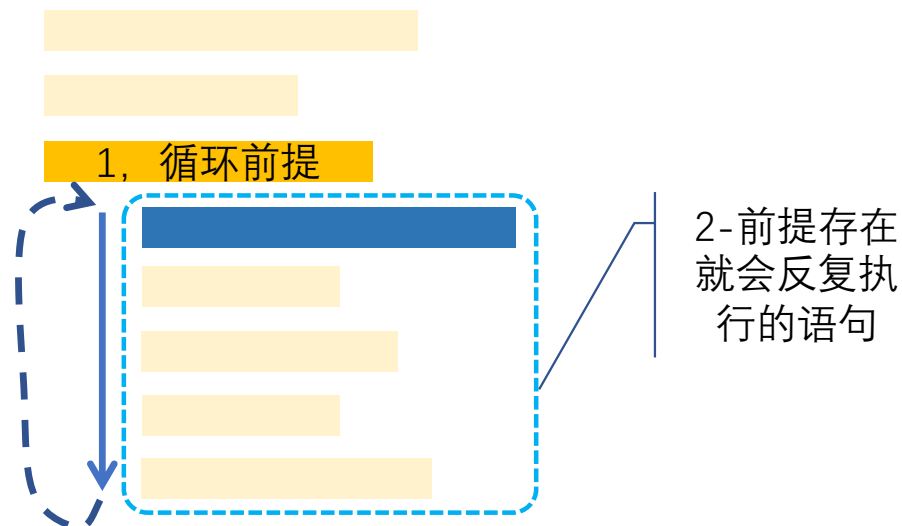
```
# 3, 获取用户输入的数据  
n = int(input("Please input an integer:"))
```

```
# 4, 开始计算搜寻  
temp = n  
while not isprime(temp):  
    temp = temp + 1
```

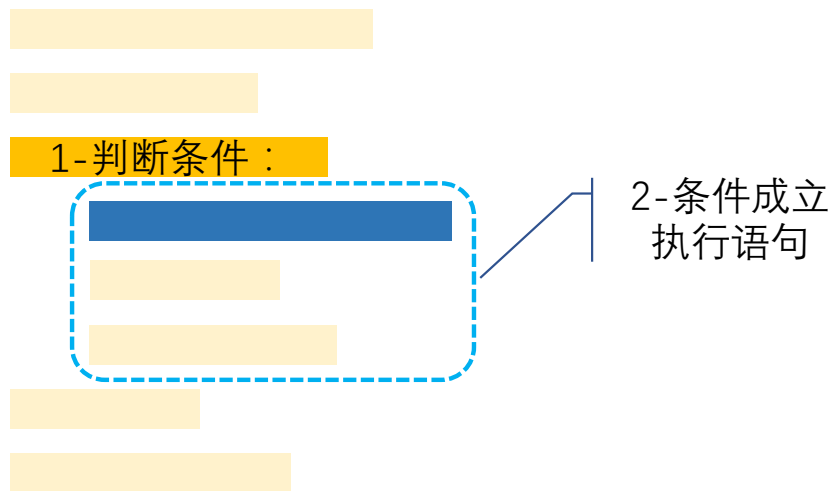
```
# 5, 输出结果  
print("Next prime number is:", temp)
```

基本控制流程

- 循环结构
- 迭代循环for语句
 - range函数
 - list列表
- 条件分支if语句
 - 判断条件
 - if-elif-else
- 条件循环while语句
- 循环语句里的break和continue



循环结构的两个基本要素



条件分支结构两个基本要素

函数和递归

- 为什么需要定义函数?
 - 代码复用提高可读性、可扩展性
- 函数定义def语句
- 函数调用
- 函数的参数和返回值
- 什么是递归?
- 利用递归计算函数
- 绘制自相似图形

```
1 def my_abs(n):
2     if n >= 0:
3         return n
4     else:
5         return -n
6
7
8 # 函数对象
9 s = my_abs
10 print("function object:", s)
11
12 # 加括号和参数, 调用函数
13 s = my_abs(-10)
14 print("call function:", s)
```

```
1 def fibonacci(n):
2     if n == 1:
3         return 0
4     elif n == 2:
5         return 1
6     else:
7         return fibonacci(n - 1) + fibonacci(n - 2)
8
9
10 for i in range(1, 12):
11     print(i, fibonacci(i))
```

硬件编程基础：microbit

- microbit单片机的基本知识
 - 完整独立的微型计算机
 - 性能较低
 - 具备各种传感器和信号输出
- 按钮的编程
- LED显示屏的编程
- 单片机编程基本程序结构
 - 无限循环
 - 按钮事件检测

