Python程序设计

第五讲 批量数据类型与操作概述

张 华
WHU

概述

- 批量数据的概念
- Python的批量数据类型

引例

- 问题：输入一组数，计算它们的标准差。
  - 对n个数X={x_1,x_2,…,x_n}，标准差的计算公式如下

$$s=\sqrt{\frac{\sum(\bar{x}-x_i)^2}{n-1}}$$

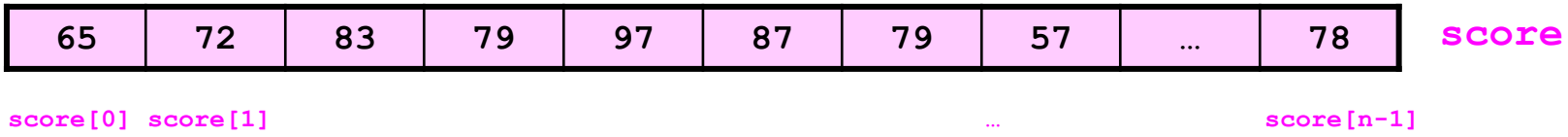
  - 算法
    - 输入n个数
    - 计算平均值
    - 计算标准差
    - 输出标准差

需要再次引用n个数，怎么保存批量数据？

批量数据的概念

- 计算机语言支持数据类型相似的批量数据的存储。
- 用统一的名称管理一批数据。
- 在内存的存储上是连续的。
- 可通过下标访问批量数据中的元素。

例如，用score保存n个学生的考试成绩



score[i]　i=0,1,…,n-1

用一个名字和下标去引用这些变量。

批量数据的概念

批量数据的优势

- 一批批量数据只需定义一个名称，程序的通用性更强。
  - 一个单变量只可以控制一个数据，使用单变量程序可控制的数据的个数是固定的。
- 使用方便，可以构造循环结构，通过控制下标的值访问一批数据（的每一项数据）。

大多数程序设计语言提供了数组来组织批量数据的存储与操作。

面向对象的程序语言还提供了功能更为强大的列表类、向量类等，在定义批量对象的存储之外，同时提供对批量数据的常规操作。

Python的批量数据类型

- Python支持批量数据存储和操作的常用组合数据类型：
  - 列表（list）
  - 元组（tuple）
  - 字符串（str）
  - 字典（dict）
  - 集合（set）
- 这些数据类型按组成元素在内存中是否连续排列可分为两类：
  - 有序的数据集合体
  - 无序的数据集合体。

序列

- 有序的数据集合体，也称为序列
  - 包括字符串(str)、元组(tuple)和列表(list)等。
- 序列的数据成员之间存在排列次序
  - 因此可以通过各数据成员在序列中所处的位置，即索引或下标来访问数据成员。
- Python提供序列的一些通用操作
  - 实现对序列的索引、连接、复制、检测成员等。

序列的基本操作

操作	描述
x1+x2	连接序列x1和x2，生成新序列
x*n	将序列x复制n次，生成新序列
x[i]	引用序列x中下标为i的序列成员，i从0开始计数
x[i:j]	引用序列x中下标从i到j-1的子序列
x[i:j:k]	引用序列x中下标从i到j-1，间隔k的子序列
len(x)	计算序列x中成员的个数
max(x)	序列x中最大数据项
min(x)	序列x中最小数据项
v in x	检测v是否在序列x中，返回布尔值
v not in x	检测v是否不在序列x中，返回布尔值

无序的数据集合体

- 无序的数据集合体包括集合、字典等。
- 无序的数据集合体中的数据成员之间不存在先后关系。