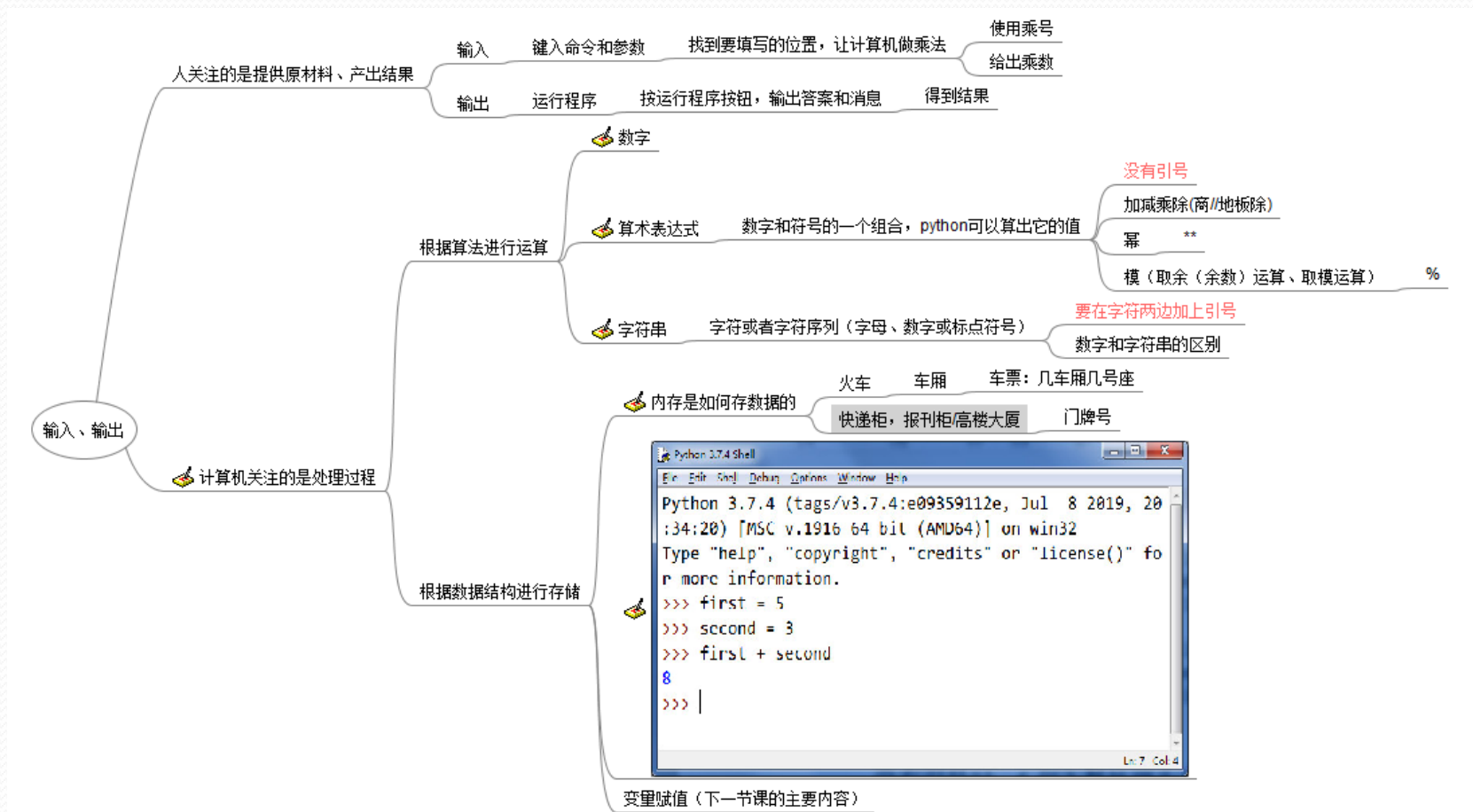


少儿编程

21工作室出品

本节课思维导图



程序的输入、处理、输出

- 人关注输入、输出
 - 输入原材料、输入规则 (点菜)
 - 获得结果、结果不止一个，就需要格式
- 计算机关注处理过程
 - 算法+数据结构 (做菜)

按步骤计算时，中间过程存储到哪里？

- 存储在内存
 - first second third
- 按照什么格式存储
 - 数据结构

语言发展的历史、底层的实现

- 对底层原理懂得越多的人，会犯越少的错误
- 但是底层的实现很复杂，是一代一代人进步改善的结果
- 通往成功的路（实现）有很多，我们应该关注其中的思想

快递柜、高楼大厦门牌号（内部地址）——用C语言理解

一单元

m_102

m_101

二单元

m_202

m_201

三单元

m_302

m_301

存储柜、高楼大厦-标签、名字

一单元

m_102

name

二单元

m_202

student

三单元

m_302

m_301

存储柜、高楼大厦-存放内容（赋值）

一单元



name



a

二单元



student



a

三单元



传值过程

- 先去m_101中找内容，找到之后，重新抄写一份新的放在m_201中。
- 这里采用C语言的值传递来理解赋值思想，不用考虑字面量、重新贴标签等底层实现。用思想去理解结果。
- 将来在设计函数时，对函数传递的参数也是一个副本的思想。

存储柜、高楼大厦-更改内容

一单元



name



二单元



student



三单元



存储柜、高楼大厦-存放内容

一单元

m_102

b

二单元

m_202

a

三单元

m_304

m_303

m_302

m_301

值传递、引用传递、副本——为什么用C语言去理解？

- 底层实现内存存储很复杂、不同语言之间实现的方式也有不同，垃圾回收机制也有区别
- 我们这里首先、最基础、需要关注的是自己设计的部分。然后才需要关注我们手上的工具是如何工作的。让工具满足我们设计的初衷，达到我们劳动的目的。
- 对于变量、函数的参数如何传值，需要具体语言、具体对象、具体分析。

如何把内容（数据）放到内存中去

- `>>> name = "a"`
- `>>> student = name`
- `>>> name = "b"`
- `>>> print(name)`
- `b`
- `>>> print(student)`
- `a`
- `>>>`