

Python程序设计02-计算机基础

北京大学 陈斌

2019.03.04

数据对象及其组织



- 什么是数据
- 多种多样的数据类型
- 数据类型归纳
- 对数据进行组织

什么是数据

- 要用计算机解决问题，首先要把问题表述为计算机能处理的形式。
- 现实世界中的万事万物蕴含着纷繁复杂的内容
- 我们只关注这些事物与所要求解决问题相关的一些性质，表述其中关键的部分。
- 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



什么是数据

- 学生信息表
- 描述了一个学生的各方面属性
 - 70435、张小明
 - 男、19
 - 2016年9月1日
 - 照片图像

学号	70435
姓名	张小明
性别	男
年龄	19
入学日期	2016 年 9 月 1 日
照片	

大数据时代

- 计算机处理的数据越来越多
- 数据获取手段空前增多
- 人类开始广泛收集收据
- 大数据(big data)
 - Volume (大量)
 - Velocity (高速)
 - Variety (多样)
 - Value (低价值密度)
 - Veracity (真实性)

- Python语言是最热门的大数据分析处理语言



多种多样的数据类型

- 描述事物大小、次序的数值类型
- 描述事物各方面特性的文本字符串类型
- 描述事物时间属性的日期时间类型等
- 每种数据类型都有自己的独特的运算

```
>>> 12 * 34.5 + 23.4
437.4
>>> ('abc' + '123') * 3
'abc123abc123abc123'
>>>
>>> import math
>>> math.sqrt(12)
3.4641016151377544
```

多种多样的数据类型

- 复杂数据类型



图形、图像



音频



视频

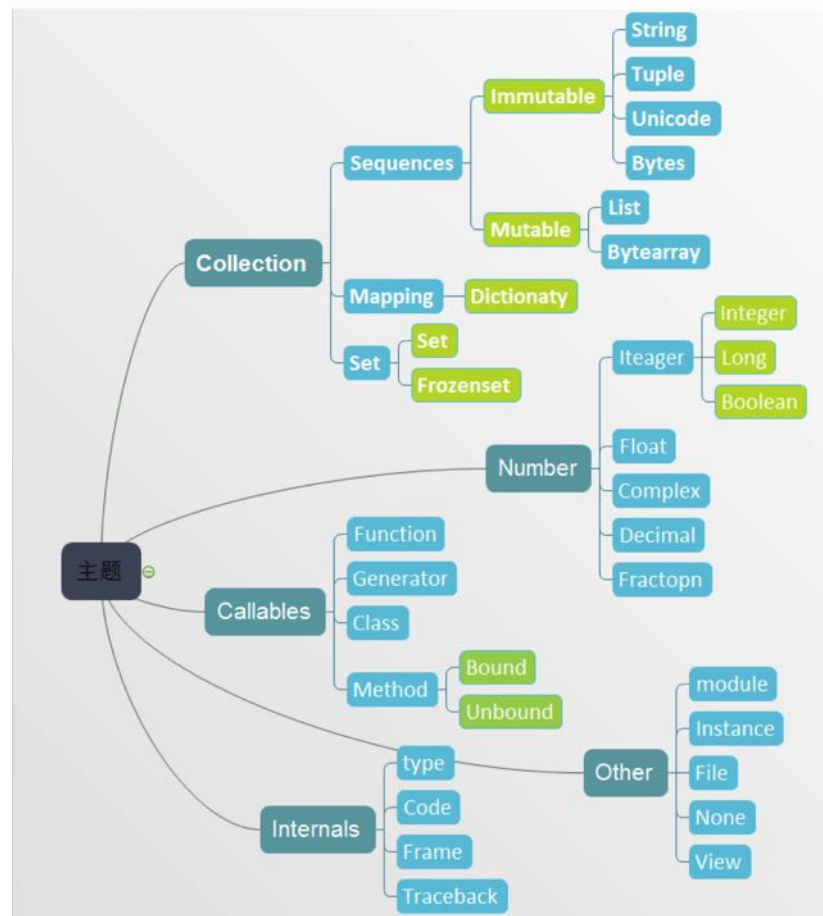
对数据进行组织

- 对大量的数据进行处理的时候，需要建立各种各样的数据组织，以便提高计算效率
- 组织方式：
 - 没有组织
 - 顺序组织数据
 - 标签式组织数据



Python数据类型概览

- 简单类型用来表示值
 - 整数int、浮点数float
 - 复数complex
 - 逻辑值bool、字符串str
- 容器类型用来组织这些值
 - 列表list、元组tuple
 - 集合set、字典dict
- 数据类型之间几乎都可以转换



图书信息的数据分析

- 有哪些数据项?
- 分别是什么数据类型?
- 有容器类型么?



★★★★☆
被引用指数0.0272
被图书引用册数5

计算机科学中的离散结构

作者: [王元元](#), [张桂芸](#)编著
出版发行: 北京: 机械工业出版社, 2004.01
ISBN号: 7-111-12939-3
页数: 302
丛书名: [高等院校计算机专业教育改革推荐教材](#)
原书定价: 28.00
开本: 26cm
主题词: 离散数学(学科: 高等学校) 离散数学
中图法分类号: O158 ([数理科学和化学](#)->[数学](#)->[代数、数论、组合理论](#)->[离散数学](#))
内容提要: 高等院校计算机专业教育改革推荐教材:该教材涵盖了经典的“离散结构”或“离散数学”课程的主要内容, 包括集合论基础、逻辑代数、形式系统与形式推理、组合论基础等内容。
参考文献格式: 王元元, 张桂芸编著. 计算机科学中的离散结构[M]. 北京: 机械工业出版社, 2004.01.

序号	数据项	类型	示例
1	书名	字符串	"计算机科学中的离散结构"
2	作者	字符串的列表	["王元元", "张桂芸"]
3	页数	整数	302

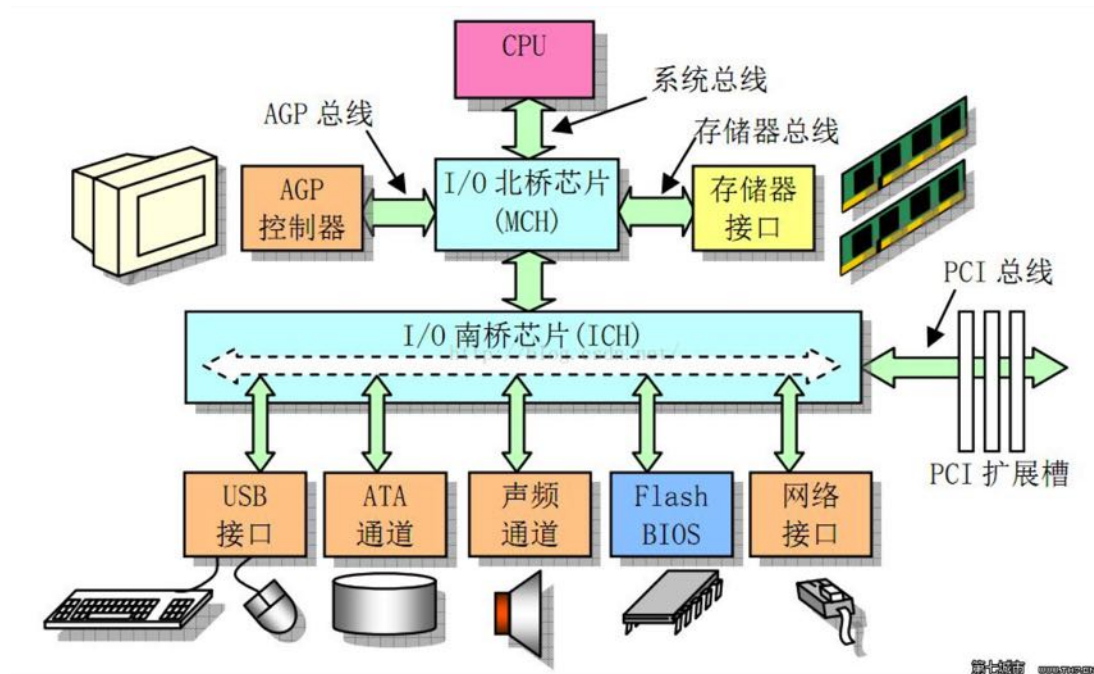
【H2】 生活中的数据分析

- 指出有哪些数据项?
 - 10项以上
- 分别是什么数据类型?
- 指出其中的容器类型?
- 以京东商城为例
 - 对手机进行数据分析
 - 对鞋子进行数据分析
 - 对商品评价进行数据分析

序号	数据项	类型	示例

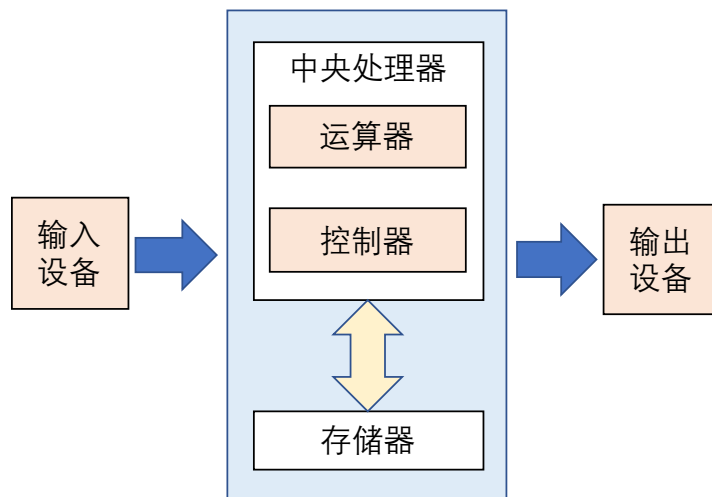
自动计算过程

- “冯·诺依曼结构”计算机
- 计算机内部运行过程
- 基本计算语句



“冯·诺依曼结构”计算机

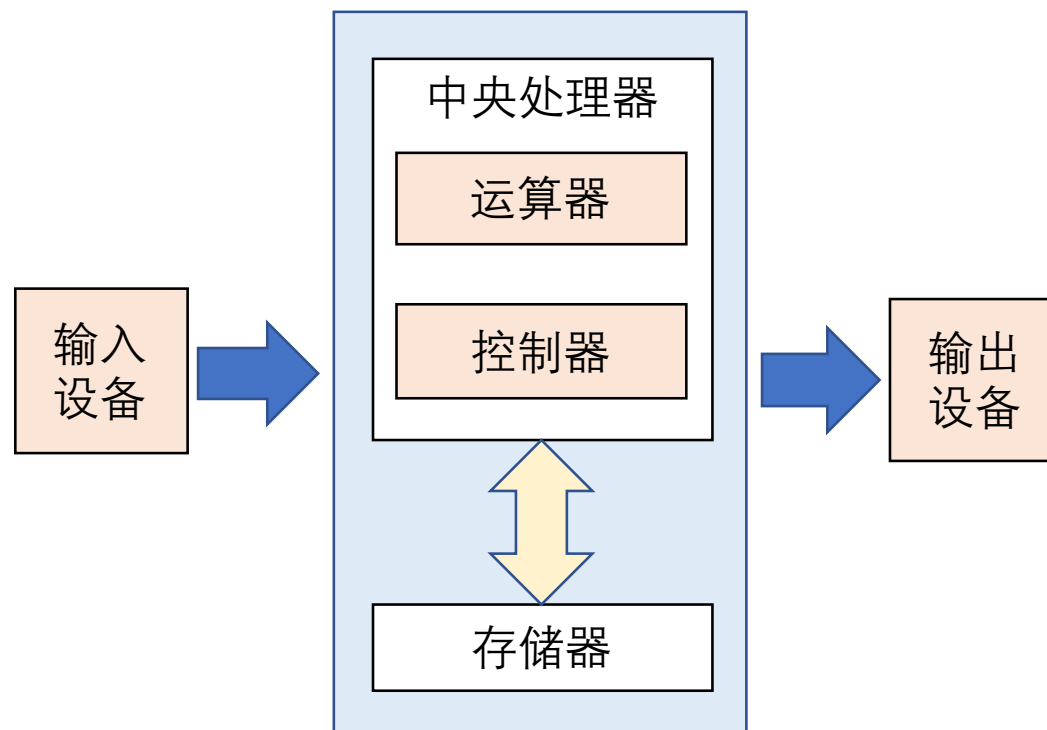
- “计算机之父” 冯·诺依曼
 - 20世纪最重要的数学家之一
 - 现代计算机、博弈论、核武器和生化武器等领域的科学全才
- 设计制造第一台电子计算机ENIAC时提出了“冯·诺依曼结构”



“冯·诺依曼结构” 计算机

- 计算机硬件五大部件

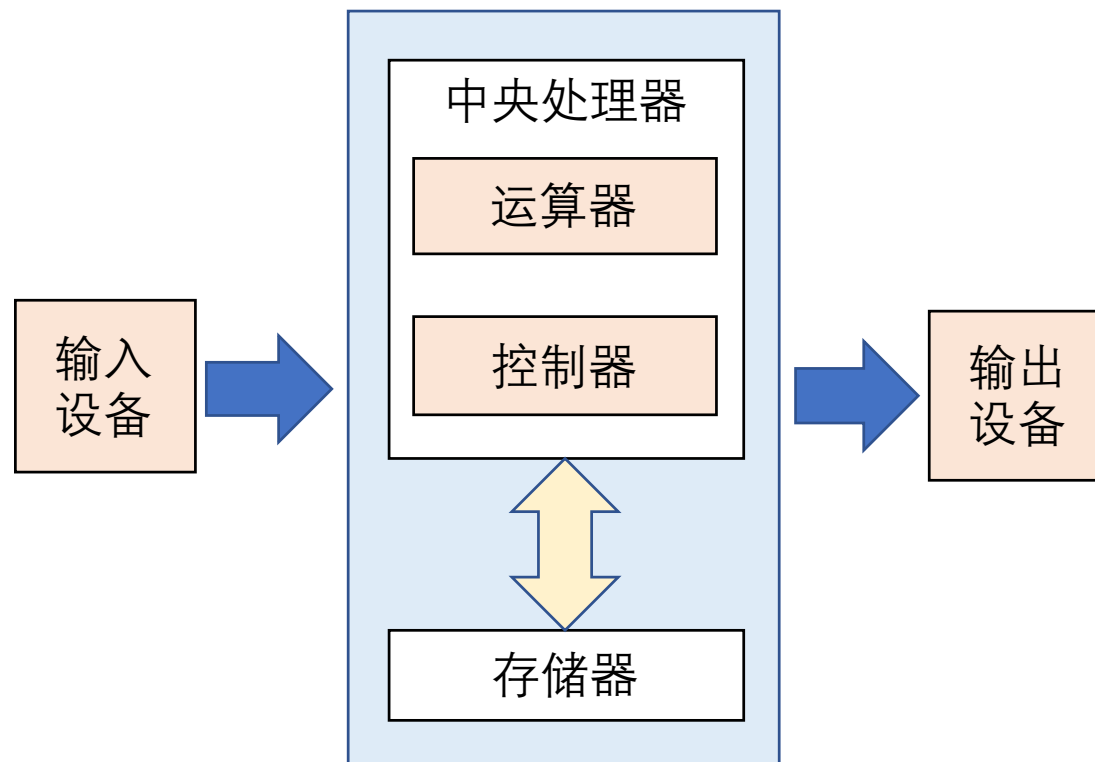
- 运算器：进行算术和逻辑运算
- 控制器：控制计算机持续协调运行
- 存储器：存储数据和程序
- 输入设备：从计算机外部获取数据（如键盘、鼠标）
- 输出设备：将计算结果反馈给外界（如显示器、打印机）



计算机内部运行过程

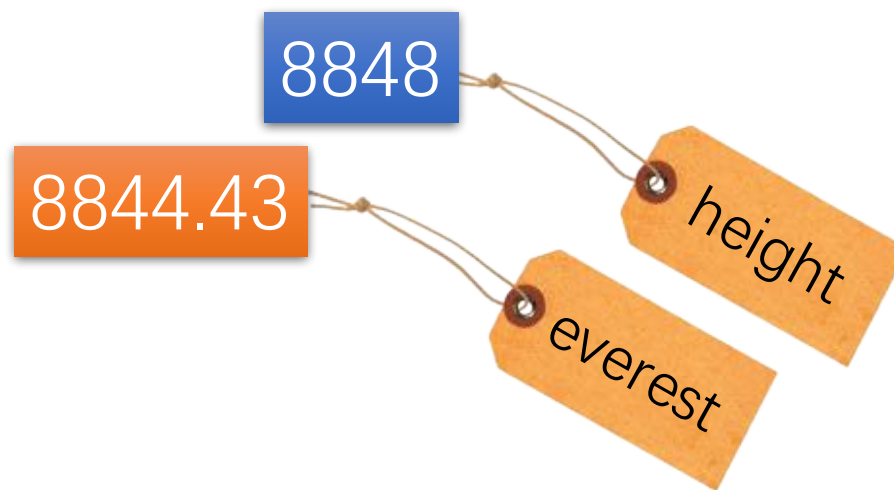
- 基本步骤

- 控制器从存储器中取出程序语句，和所需的额外数据；
- 数据齐全的语句交给运算器进行算术或者逻辑运算；
- 运算结果再存回存储器；
- 控制器确定下一条程序语句，回到步骤1继续。



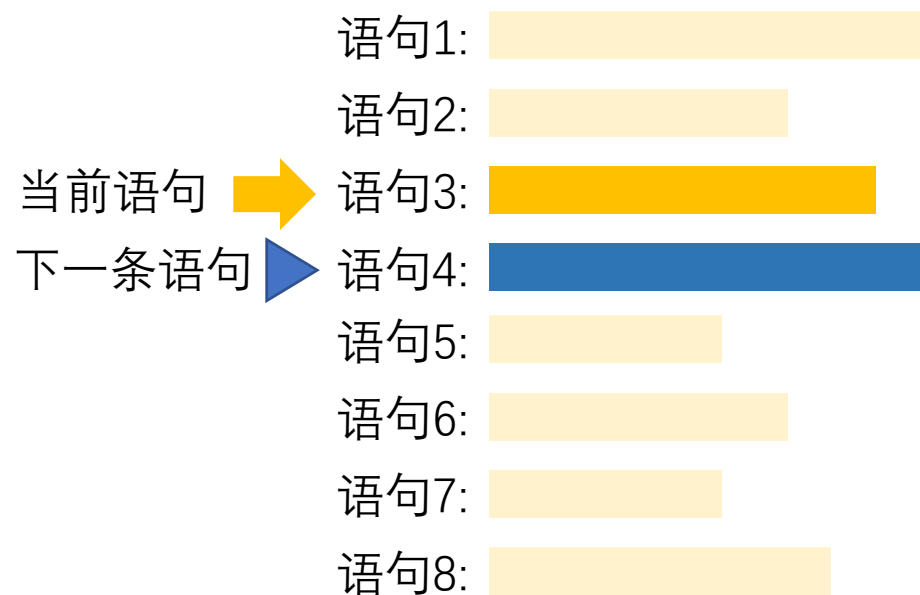
基本计算语句

- 赋值语句
 - $\langle \text{变量} \rangle = \langle \text{表达式} \rangle$
- Python语言的赋值语句很好地对应了“运算”和“存储”
- 赋值语句的执行语义为：
 - 计算表达式的值，存储起来
 - 贴上变量标签以便将来引用
- 与计算机运行过程中的“计算”和“存储”相对应

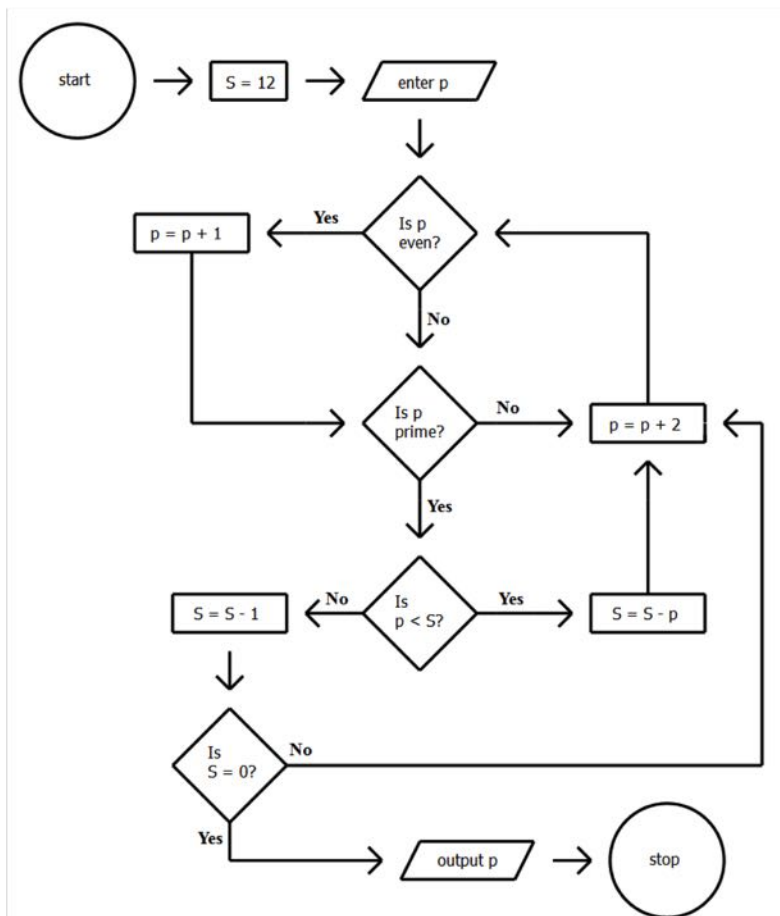


基本计算语句

- “控制器确定下一条程序语句”即对应“控制”
- 一个程序的很多语句，在存储器中的排列，就像在火车站买票一样排成一个队列
- 思考：下一条语句仅仅是“语句队列 中的后一条”一种情况吗？



计算和控制流



- 计算与流程
- 运算语句
- 控制流语句
 - 决定下一条语句

计算与流程

- 数据是对现实世界处理和过程的抽象
- 各种类型的数据对象
- 可以通过各种运算组织成复杂的表达式

```
>>> 12 * 34.5 + 23.4
437.4
>>> ('abc' + '123') * 3
'abc123abc123abc123'
>>>
>>> import math
>>> math.sqrt(12)
3.4641016151377544
```

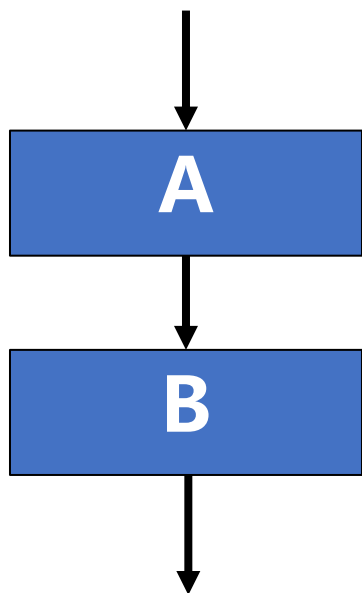
运算语句

- 将表达式赋值给变量进行引用
- 赋值语句用来实现处理与暂存
 - 表达式计算
 - 函数调用
 - 赋值

```
>>> n = 12 * 34
>>> n
408
>>> p2 = math.sqrt(2)
>>> p2
1.4142135623730951
>>> pfg = math.sqrt
>>> pfg
<built-in function sqrt>
>>> pfg(2)
1.4142135623730951
```

控制流语句

- 控制流语句用来组织语句描述过程



顺序结构

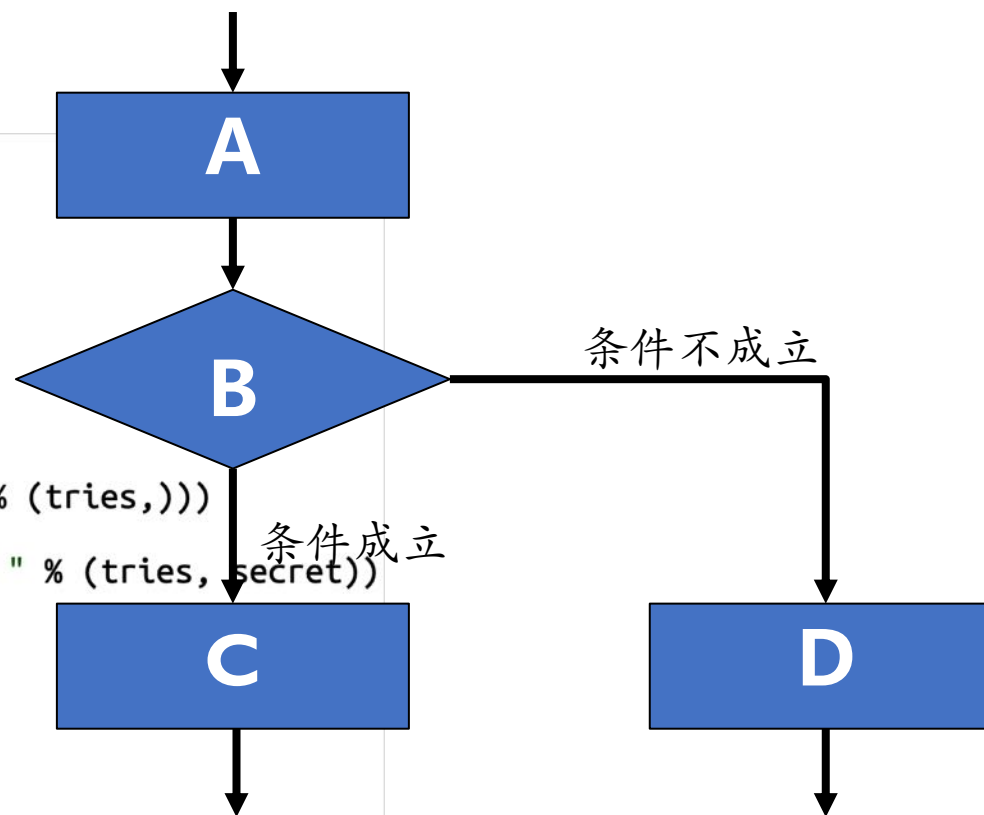
三角形.py ×

```
import turtle
t=turtle.Turtle()
t.color('green')
t.forward(100)
t.right(120)
t.forward(100)
t.right(120)
t.forward(100)
t.right(120)

t.hideturtle()
turtle.done()
```

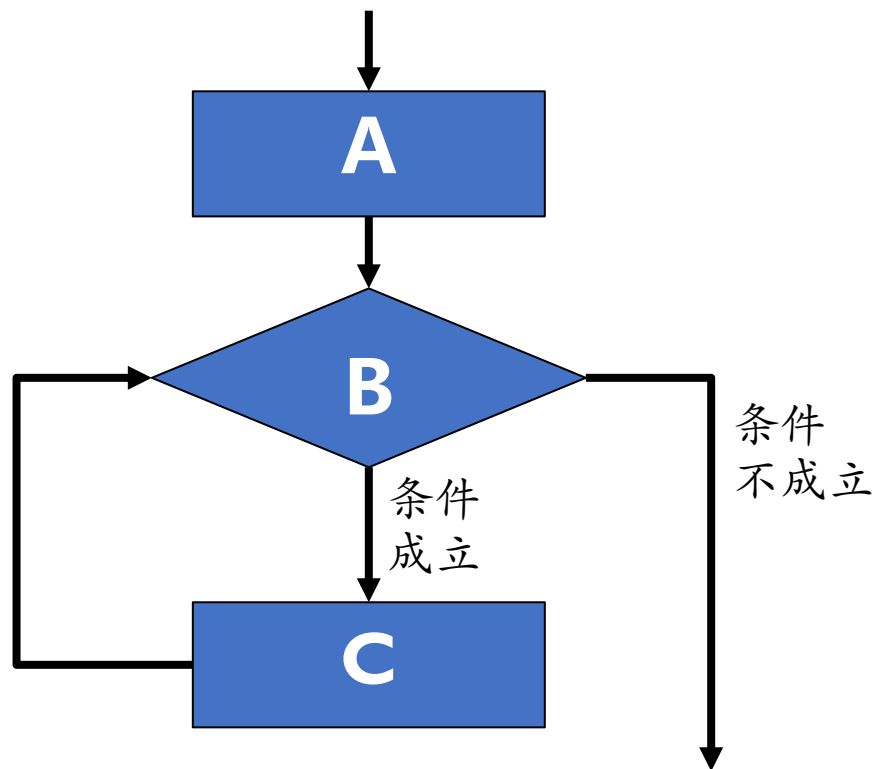
控制流语句

```
1 # 猜数字游戏
2 import random
3
4 secret = random.randint(1, 100)
5 print('''猜数游戏!
6 我想了一个1-100的整数, 你最多可以猜6次,
7 看看能猜出来吗? ''')
8 tries = 1
9 while tries <= 6:
10     guess = int(input("1-100的整数, 第%d次猜, 请输入: " % (tries,)))
11     if guess == secret:
12         print("恭喜答对了! 你只猜了%d次! \n就是这个: %d! " % (tries, secret))
13         break
14     elif guess > secret:
15         print("不好意思, 你的数大了一点儿! ")
16     else:
17         print("不好意思, 你的数小了一点儿! ")
18     tries += 1
19 else:
20     print("哎呀! 怎么也没猜中! 再见! ")
```



条件分支: if

控制流语句



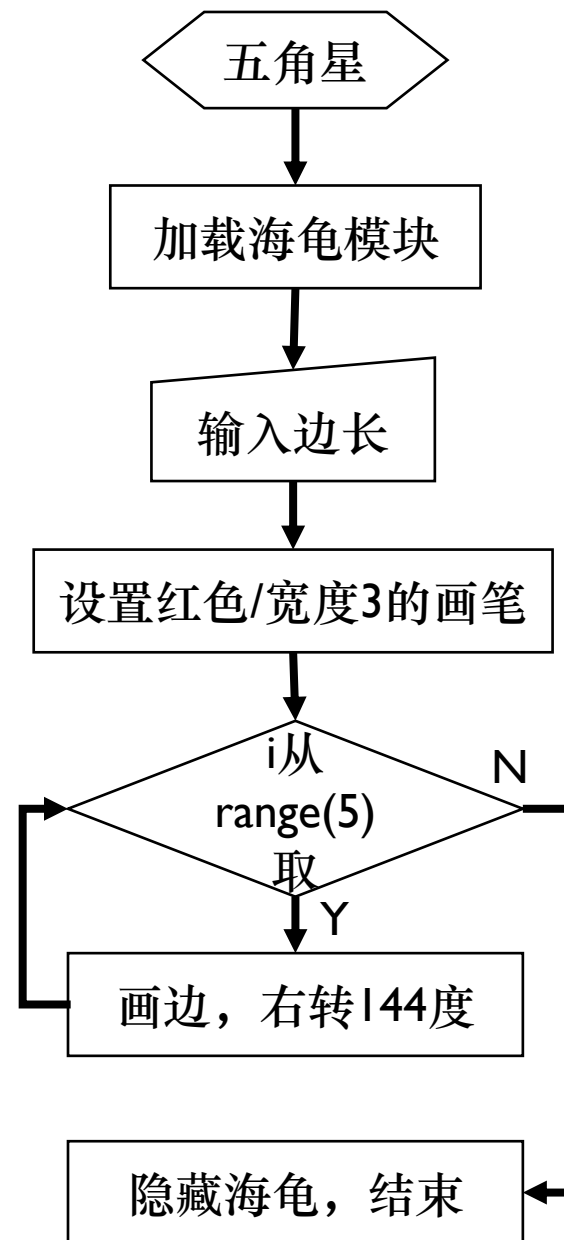
```
1  import turtle
2
3  size = int(input("Please input size:(20~200)"))
4
5  t = turtle.Turtle()
6  t.color('red')
7  t.pensize(3)
8
9  for i in range(5):
10     t.forward(size)
11     t.right(144)
12
13 t.hideturtle()
14 turtle.done()
```

循环结构for、while

分析程序流程

```
1  import turtle
2
3  size = int(input("Please input size:(20~200)"))
4
5  t = turtle.Turtle()
6  t.color('red')
7  t.pensize(3)
8
9  for i in range(5):
10     t.forward(size)
11     t.right(144)
12
13 t.hideturtle()
14 turtle.done()
```

打印显示



【H3】 程序流程图绘制

```
1  import random
2
3  menu = ["coffee", "tea", "cola", "milk", "water"]
4  print("Menu:", menu)
5  name = input("Your name please:")
6  drink = random.choice(menu)
7  print("Hello", name, "! Enjoy your", drink)
```

【H3】 程序流程图绘制

```
1  # 猜数字游戏
2  import random
3
4  secret = random.randint(1, 100)
5  print('''猜数游戏!
6  我想了一个1-100的整数, 你最多可以猜6次,
7  看看能猜出来吗? ''')
8  tries = 1
9  while tries <= 6:
10     guess = int(input("1-100的整数, 第%d次猜, 请输入: " % (tries,)))
11     if guess == secret:
12         print("恭喜答对了! 你只猜了%d次! \n就是这个: %d! " % (tries, secret))
13         break
14     elif guess > secret:
15         print("不好意思, 你的数大了一点儿! ")
16     else:
17         print("不好意思, 你的数小了一点儿! ")
18     tries += 1
19 else:
20     print("哎呀! 怎么也没猜中! 再见! ")
```

作业

- 在希悦提交【H2】和【H3】
- （可选）看视频
 - <http://www.chinesemooc.org/live/685377>
 - 2.1, 2.2, 2.3

