

程序初步 - I

- ❖ **Python 基本元素：数据**
- ❖ **最简单的 Python 程序**
- ❖ **程序和计算**
- ❖ **表达式和求值**
- ❖ **Python 编程环境**
- ❖ **语法、语义和编程**

Python 基本编程元素

■ 数据

□ 数

- 整数、浮点数、复数

□ 字符串

□ 其他

■ 表达式

□ 描述基本计算

■ 语句

□ 描述操作

■ 控制和抽象

本次课只涉及
数和表达式

编程环境 IDLE

- **Python** 系统提供的交互式编程环境
 - 用于交互式计算和编程
 - 很好用，用于编写和执行（运行）**Python** 程序
- 建议始终用它编写 **Python** 程序
 - 实际上可以用任何文本编辑器
 - 用 **IDLE** 更方便
- 启动 **IDLE**，进入了一个 **Python** 计算环境
 - **IDLE** 接受人的输入
 - 注意，**Python** 程序有规定的语法，形式上符合要求的才是合法的程序

基本编程元素：数

■ Python 有三类数

- 整数，模拟数学里的整数
- 浮点数，模拟数学的实数
- 复数，模拟数学的复数

■ 与对应的数学概念不同

- 能表达的数值是有穷的
- 浮点数和复数的精度有限

■ 每一类数有规定的

- 书写形式（语法）
- 表达的意思（语义）

整数

■ 整数

- 语法：数字序列（非 0 不能以 0 开头）
- 语义：相应的整数（简单）

■ 性质

- 可以写任意大的整数
- 计算机的存储器有穷，只能表示有穷大小的整数

■ 最简单的 Python 程序

■ 程序和计算：

- 写出一个程序（描述希望做的计算）
- Python 读入程序，完成计算，输出结果（实现计算）

浮点数

- 语法：数字序列，必须
 或有小数点，
 或有表示指数的 **e**（或**E**）后跟可选正负号和一个整数
- 语义：表示相应的浮点数（指数部分以 **10** 为底）
- 与数学中实数的关系：
 - 浮点数可以表示一定范围里的一些实数
 - 表示范围有限，表示精度有限
- 语法错误，例如：
3a **3e--3** **3+**

类型

- 直接写出的数据称为**字面量**
- **Python** 把各种计算元素统称为“**对象**”（**object**）
对象都有确定的**类型**（**type**）
- 例：整数
 - 整数字面量的计算（求值）得到整数对象
 - 整数对象的类型是 **int**（整数类型的名字）
 - （浮点数类型的名字是 **float**）
- **type(...)** 得到 ... 的类型
- 类型（名）的一个用途是做**类型转换**

int(2.3) 浮点数转换到整数的规则是取整

算术表达式

- 算术表达式用算术运算符构造

- 算术运算符

+ - * / // % **

// 表示整数的整除，** 表示乘幂，% 表示余数

其中 + 和 - 是一元和二元运算符，其他是二元运算符

一元的 + 和 - 看作运算符，不看作是数的符号

- 算术表达式

- 语法：基于运算对象、运算符和括号，符合运算符元数要求（可以严格描述，参看语言手册）

- 语义：表达式描述的计算，算出的值

算术表达式描述的计算

- 合法（满足语法）的表达式有意义（语义）
 - 描述了一个计算过程
 - 计算（求值）表达式，将得到一个值
 - 注意：合法程序在计算中也可能出错。如 **3/0**，**3.0/0**
- 理解求值过程，运算符有优先级和结合方式等
 - 优先级(从低到高): **+** **-**, ***** **/** **//** **%**, 一元运算符, ******
 - 结合方式
 - 一元运算符和 ****** 从右到左，其余算术运算符从左到右
 - 括号用于描述特定的计算顺序
 - 二元运算符，先计算左边的运算对象，再计算右边的

算术表达式

- 一元运算符和 `**` 的特殊规定：
 - `-3**2` 的意义同 `-(3**2)`
 - `3**-2` 同 `3**(-2)`
- 表达式用一维的字符序列表示
 - 所有细节都必须严格定义
 - 计算机容易处理，人可以接受和习惯
- 表达式描述的计算与类型有关
 - 参与计算的对象有类型
 - 计算结果也有类型
 - 类型影响计算的方式和结果

表达式计算和类型

- 整数计算是“精确的”，总得到精确的整数结果（/ 除外）
 - $1/3$ 和 $1//3$ 不同
- 浮点数计算是近似的，给出近似结果
 - 溢出问题（动态错误）
- 混合类型计算和类型转换
 - 不同类型混合计算时，先将两者转换到统一类型
 - 整数和浮点数运算时，先将整数转换到浮点数
 - 转换可能出错：整数可以任意的大，可能超出浮点数类型能表示的范围

课下工作

- 浏览课程网页及其链接
- 下载和安装 **Python** 系统（在自己的计算机上）
 - 从官网下载，或者从课程网页下载
 - 写一些简单算术表达式，试验确认系统可用
 - 设法确定**正**浮点数的最大和最小数值
 - 翻阅 **Python** 系统手册，看看有什么内容
- 登陆教学网
 - 检查你的账户里是否已经有本课程
 - 检查你教学网邮箱是否正常（给自己的其他邮箱发信）