交互式解释器

提示符：>>>

出错：

不以提示符开头的行是解释器的输出

(从属)次提示符：…

有以一个(从属)次提示符独占一行时，加入一个空行;用来表示多行命令的结束

python注释以#号开头

数字和表达式

操作符+，-，\*，/

一个整数被另一个整数除，计算结果的小数部分被截除，只留下整数部分；若希望python只执行普通的除法，

1. 用实数，浮点数；

2. 让python改变除法的执行方式，在程序前加上以下语句，或直接在解释器里执行它：from \_\_future\_\_ import division；

3. 若通过命令行运行python，使用命令开关Qnew?

整除操作：双斜线 //

取余：%

幂、乘方：\*\*

幂运算符比取反(-)的优先级高，-3\*\*2 = -(3)\*\*2

长整型，结尾L

长整型和普通整型可以混用

变量

变量名包括字母、数字和下划线(\_)，变量不能以数字开头，

赋值：=

语句

赋值语句

输入

x = input(“x: ”)

if语句：if 条件，用冒号:将其和后面的语句隔开，当条件为真时，执行后面的语句；在交互式解释器中使用if语句，需要按两次回车，if语句才能被执行

函数

标准函数(内建函数)

乘方：幂运算符\*\*，pow( , )函数

绝对值：abs()函数

把浮点数四舍五入为最接近的整数值：round()函数

调用函数，提供参数，返回值

模块

import命令导入模块，按照“模块.函数”格式使用这个模块的函数；

将给定的值向下取整：floor()函数

将给定的数转换成大于或等于它的最小整数：ceil()函数

在确定不会导入多个同名函数(从不同模块导入)的情况下，不希望每次调用时，都写上模块的名字，使用“from模块import函数”后可以直接使用函数，无须模块名作为前缀

可以使用变量来引用函数，

cmath和复数

nan：“not a number”

import cmath

虚数均以j(J)结尾

交互式解释器里输入的一切都会在退出时丢失

文件名以.py结尾

通过命令提示符运行python脚本

python \*.py

让脚本像普通程序一样运行，不显式使用python解释器

在脚本首行前面加上#!(pound bang或shebang)，在其后加上用于解释脚本的程序的绝对路径

#!/usr/bin/env python

在实际运行前，让脚本具有可执行属性

chmod a+x \*.py

./\*.py

也可以去掉.py