**python运行方式**

交互式解释器

进入交互式解释器

python3

退出交互式解释器

exit()货ctrl+d

文件形式

明确指定解释器

python3 hello.py

赋予python文件可执行权限

chmod +x hello.py

./hello.py

**python语法结构**

语句块缩进

python代码块通过缩进对齐表达式代码缩进而不是用大括号，缩进表达一个语句属于哪个代码块

缩进风格

1或2：可能不够，很难确定代码语句属于哪个代码块

8至10个：可能太多，如果代码内嵌的层次太多，就会使得代码很难读

4哥空格：非常流行。

缩进相同的一组语句构成一个代码块，称为代码组

首行以关键字开始，以冒号：结束，该行至后的一行或多行代码构成代码组，如果代码组只有一行，可以将其直接写在冒号后面，但是这样的可读性比较差，不推荐

注释和续行

注释以#开头，注释科一在一行的任何地方开始，解释器会忽略掉该行#至后的所有内容

一行过长的语句可以用反斜杠\分解成几行

同行多个语句

分号;允许将多个语句写在同一行上，但是这些语句不能在这行开始一个新的代码块，因为可读性会变差，所以不推荐使用

**python语法**

获取帮助

help(print)

输出语句print

print('hello world!') # 字符串必须放在引导中

print('hello', 'world', '!', 123) # 数字不用放在引号中

print('hello', 'world', '!', 123, sep='\*\*\*') # 各项间用\*\*\*分隔

print('hello' + 'world') # 字符串拼接

print('hello', 'world', '!', 123, end='') # 不打印回车

输入语句input

a = input('number:') # input读入的数据是字符类型

# a+5错误，不能把数字和字符相加

print(int(a) + 5) # 将字符转换成数字再加5

print(a + str(5)) # 将数字5转换成字符，再和字符串拼接

**python变量**

变量定义

变量名称约定

第一个字符只能是大小写字母或下划线

后续字符职能是大小写字母货数字或下划线

区分大小写

python是动态类型语言，即不需要预先声明变量的类型

推荐才用全名的办法

变量名全部采用小写字母

简短、有意义

多个单词间用下划线分割

变量名用名词，函数名用谓语（动词+名词）

类名采用驼峰形式

变量赋值

变量的类型喝值在赋值的那一刻被初始化

变量赋值通过等号来执行

python也支持增量赋值

a = 10 + 5 # 变量赋值，自右向左进行

a = a + 1 # 可以简化位以下形式

a += 10

# b+=10错误，因为等价于b=b+10,b没有赋值

运算符

标准算术运算符：+-\*/ // % \*\*

比较运算符：< <= > >= == !=

逻辑运算符：and not or

print(5 / 2) # 2.5

print(5 // 2) # 2

print(5 % 2) # 之遥余数，模运算，1

print(2 \*\* 3) # 2的3次方，8

20 > 10 > 5 # true相当于20>10and10>5

**python数据类型**

数字

基本数字类型

int：有符号整数

bool：布尔值（True：1，False：0）

float：浮点数

数字表示方法

python默认使用十进制数表示；数字以0o或者0O开头表示为8进制数；数字以0x或0X开头表示为16进制数，数字以0b或0B开头表示为2进制数

a=True+2

print(a)#bool类型也是整型

print(0o11)#8进制数11换算成10进制数为9

print(0x11)#16进制数11换算成10进制数是17

print(0b11)#二进制数11换算成10进制数是3

字符串

定义字符串

python种字符串被定义为引号之间的字符集合，可以使用成对的单引号货双引号，两个意义相同，还支持三引号（三个连续的引号或者双引号），可以用来包含特殊字符。不区分字符和字符串

py=’python’

字符串切片（类似与截取）

使用索引运算符[]和切片运算符[:]可得到字符串，第一个字符串的索引是0,最后一个字符的索引是-1，字符串包含切片中的起始下标，但不包含下标，切片没有下标不会报错

py[0]#取出第一个字符

py[-1]#取出最后一个字符

py[2:4]#切片，取出2到4个字符

py[2:]#结尾不写，表示取到最后部分结束

py[:]#取所有

py[::-1]#步长为负，表示从右往左取

字符串连接

使用+号可以将多个字符串拼接在一起，使用\*号可以将一个字符串重复多次

py+’is good’

‘\*’ \* 50#打印\*50次

列表

可以将列表当成普通的“数组”，它能保存任意数量任意类型的python对象，像字符串一样，列表也支持下标和切片操作，列表中的项目可以改变

列表操作

使用in或not in判断成员关系，使用append方法向列表中追加元素

alist=[‘bob’,’alice’,10,[1,2,3]]

[‘tom’]+alist

alist.append(100)#向列表中追加元素

元组

可以认为元组是“静态”的列表，元组一旦定义，不能改变

aaa=(‘bob’,10,28)

aaa[2:]

(100,)+aaa

字典

字典是由键值对构成的映射数据类型，通过键取值，不支持下标操作

adict={‘name’:’zhangsan’,’age’:25}

25 in adict#false

adict[‘name’]#’zhangsan’

数据类型比较

按存储模型分类

标量类型：数值、字符串

容器类型：列表、元组、字典

按更新模型分类

可变类型：列表、字典

不可变类型：数字、字符串、元组

按访问模型分类

直接访问：数字

顺序访问：字符串、列表、元组

映射访问：字典

**python判断语句**

if语句

语法：if expression:

if\_suite

else:

else\_suite

如果表达式的值非0或者为布尔值true，则代码组if suite被执行；否则就去执行ekse suite，只要表达式数字为非零值即为true，空字符串、空列表、空元组、空字典的值均为false

if “”:

print(‘yes’)

else:

print(‘no’)

扩展if语句

语法：if expression1:

if\_suite

elif expression2:

elif\_suite

else:

else\_suite

条件表达式

语法：x if c else y中间是条件，两边是结果

a=10

b=5

s=a if a<b else b

print(s)

**python循环**

python中有while和for循环，循环次数未知的情况下，建议采用while循环，循环次数可以预知的情况下，可以采用for循环

while循环语句结构

while expression:

while\_suite

语句while\_suite会被不断的执行循环，直到表达式的值为-或FLASE

sum=0

counter=1

while counter<=100:

sum+=counter

counter+=1

print(sum)

while进阶

break语句

break跳出循环体，执行下面的语句

continue语句

当遇到continue语句时，程序循环终止当前循环，并忽略剩余的语句，然后回到循环的顶端，如果条件满足，循环体内的语句继续执行，如果不满足则退出

else语句

python中的while语句也支持else子句，else子句只在循环正常完成后执行，break语句会跳过else块

for循环详解

python中的for接受可迭代对象（例如迭代器或序列）作为其参数，每次迭代其中一个参数

for iter\_var in iterable:

suite\_to\_repeat

与while循环一样，支持break、continue、else语句

alist=[‘bob’,’alice’]

for name in alist:

print(name)

range函数

for循环常与range函数一起使用

range函数提供循环语法

range函数的完整语法为：

range(start,end,step=1)

list(range(10))#只写10，表示结束数字10，但是不包含10，开始数字是0

list(range(6,10))#[6,7,8,9,10]

list(range(1,10,2))#[1,3,5,7,9]

list(range(10,0,-1))#[10,9,8,7,6,5,4,3,2,1]

for i in range(1,101):

print(‘hello’)#输出三次hello

列表解析

是一个非常有用、简单、而且灵活的工具，科一动态创建列表

语法：

[expr for i in iterable]

[3+2 for i in range(10)]#循环控制3+2运行多少次,[5,5,5,5,5,5,5,5,5,5]

**文件对象**

文件打开方法

open及file内建函数

内建函数open()以及file()提供了初始化输入/输出（I/O）操作的通用接口

成功打开文件时候会返回一个对象，否则引发一个错误

open()方法和file()方法可以完全相互替换

基本语法：

aaa=open(file\_name,’r’,buffering=-1)

文件对象访问方法

r以读的方式打开（文件不存在则报错）

w以写方式打开（文件存在则清空，不存在则创建）

a以追加模式打开（必要时创建文件）

r+以读写模式打开（参见r）

w+以读写模式打开（参见w）

a+以读写模式打开（参见a）

b以二进制模式打开

文件输入

read方法

read()方法用来直接读取字节到字符串中，最多读取给定数目个字节

如果没有给定size参数（默认值为-1）或者size值为负，文件将读取至末尾

data=fobj.read()

print(data)