变量：

在Python中，变量命名需要遵循以下这些必须遵守硬性规则和强烈建议遵守的非硬性规则。

硬性规则：

变量名由字母（广义的Unicode字符，不包括特殊字符）、数字和下划线构成，数字不能开头；

大小写敏感（大写的 a 和小写的 A 是两个不同的变量）；

不要跟关键字（有特殊含义的单词，后面会讲到）和系统保留字（如函数、模块等的名字）冲突。

PEP 8要求：

用小写字母拼写，多个单词用下划线连接；

受保护的实例属性用单个下划线开头；

私有的实例属性用两个下划线开头。

当然，给变量（事实上应该是所有的标识符）命名做到见名知意也是非常重要的。

数据类型：

计算机中的变量是实际存在的数据或者说是存储器中存储数据的一块内存空间，变量的值可以被读取和修改，这是所有计算和控制的基础。计算机能处理的数据有很多中类型，除了数值之外还可以处理文本、图形、音频、视频等各种各样的数据，那么不同的数据就需要定义不同的存储类型。Python中的数据类型很多，部分如下：

整型：Python中可以处理任意大小的整数（Python 2.x中有int和long两种类型的整数，但这种区分对Python来说意义不大，因此在Python 3.x中整数只有int这一种了），而且支持二进制（如0b100，换算成十进制是4）、八进制（如0o100，换算成十进制是64）、十进制（100）和十六进制（0x100，换算成十进制是256）的表示法。

浮点型：浮点数也就是小数，之所以称为浮点数，是因为按照科学记数法表示时，一个浮点数的小数点位置是可变的，浮点数除了数学写法（如123.456）之外还支持科学计数法（如1.23456e2）。

字符串型：字符串是以单引号或双引号括起来的任意文本，比如'hello'和"hello",字符串还有原始字符串表示法、字节字符串表示法、Unicode字符串表示法，而且可以书写成多行的形式（用三个单引号或三个双引号开头，三个单引号或三个双引号结尾）。

布尔型：布尔值只有True、False两种值，要么是True，要么是False，在Python中，可以直接用True、False表示布尔值（请注意大小写），也可以通过布尔运算计算出来（例如3 < 5会产生布尔值True，而2 == 1会产生布尔值False）。

复数型：形如3+5j，跟数学上的复数表示一样，唯一不同的是虚部的 i 换成了 j 。

数据类型之间的转换：

int()：将一个数值或字符串转换成整数，可以指定进制。

float()：将一个字符串转换成浮点数。

str()：将指定的对象转换成字符串形式，可以指定编码。

chr()：将整数转换成该编码对应的字符串（一个字符）。

ord()：将字符串（一个字符）转换成对应的编码（整数）。

运算符与表达式：



赋值表达式： 变量 = 表达式

表达式为真的情况: **not** 0 -1 等为真
表达式为假的情况: 0 '' **None** **False** [] () {}

中途小练习：

1、将华氏温度转换为摄氏温度 F = 1.8C + 32 ，F表示华氏温度，C表示摄氏温度

f = float(input('请输入华氏温度: '))
c = (f - 32) / 1.8
print('%.1f华氏度 = %.1f摄氏度' % (f, c))

2、输入圆的半径计算计算周长和面积

import math
radius = float(input('请输入圆的半径: '))
perimeter = 2 \* math.pi \* radius
area = math.pi \* radius \* radius
print('周长: %.2f' % perimeter)
print('面积: %.2f' % area)

3、输入年份判断是不是闰年，是则输出 True，否则输出 False

year = int(input('请输入年份: '))
# 如果代码太长写成一行不便于阅读 可以使用\或()折行
is\_leap = (year % 4 == 0 and year % 100 != 0 or
year % 400 == 0)
print(is\_leap)

list（列表）、tuple（元组）、dict（字典）、set（集合）
列表list
格式: 列表名 = [列表值1, 列表值2, 列表值3, ... , 列表值n]
列表是一个有序集合

下面是一些list的操作：
# 创建一个空列表
list1 = []
print(list1)
print(type(list1))
# 创建带有元素的列表
list2 = [1, 2, 3, 4, 5, 6, 7, 8]
print(list2)
# 列表元素的类型可以不同
list3 = [12, 'qwe', 0.87, False, None]
print(list3)
# 访问列表中的元素 格式: 列表名[元素的下标] 注意：下标不能超出列表的范围，最大为len(list)-1
list4 = [1, 2, 3, 4, 5, 6, 7, 8]
print(list4[5])
# 修改元素
list4[3] = 10
print(list4)

# 列表的操作
# 列表拼接
list5 = [1, 2, 3, 4]
list6 = [5, 6, 7]
list7 = list5 + list6
print(list7)
# 列表的重复
list8 = [4, 5, 6]
print(list8 \* 3)
# 判断一个元素是否在列表中
list9 = [1, 2, 3, 4, 5, 6]
# 返回的结果是布尔值 存在为True 不存在为False
print(7 in list9)
print(4 in list9)
# 截取列表 list[start:end:step]
list10 = [1, 2, 3, 4, 5, 6, 7, 8]
print(list10[:6]) # 从左截取，从头开始，到下标为5的元素
print(list10[5:]) # 从左截取，从下标为5的元素，到末尾
print(list10[::-1]) # 将list倒序排列
print(list10[::-2]) # 从右截取，末尾开始，每隔一个元素截取一个
# 二维列表
list11 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
# 获取元素
print(list11[0][0])

# 列表的方法
# append(): 在列表的末尾添加一个元素
list1 = [1, 2, 3, 4, 5, 6]
list1.append(8)
print(list1)
list1.append([7, 9, 10])
print(list1)
# extend(): 在列表的末尾一次性追加另外一个列表中的多个值
list2 = [1, 2, 3, 4, 5, 6]
list2.extend([200, 300, 400])
print(list2)
#insert(index, val):在指定下标处添加一个元素,原来此处的元素往后移动 index：表示列表下标，val：表示下标对应的值
list3 = [1, 2, 3, 4]
list3.insert(1, 250)
print(list3)
# pop([index]):将列表中指定下标的元素删除, 如果不传递参数,则删除的是最后一个元素 index：表示列表下标，返回值：删除的该元素
list4 = [1, 2, 3, 4, 5, 6]
print(list4.pop())
list4.pop()
print(list4)
list4.pop(1)
print(list4)
# remove(val): 移除列表中指定的元素，从左开始，移除第一个 val：表示列表中的值
list5 = [3, 4, 5, 6, 3, 7]
list5.remove(3)
# list5.remove(3)
print(list5)
# clear(): 清除列表中所有的元素
list6 = [1, 2, 3, 4, 5, 6]
list6.clear()
print(list6)
# index(val): 获取列表中该元素所对应的下标 val：列表中的某一个元素 返回值：列表中元素所对应的下标
list7 = [1, 2, 3, 4, 5, 6]
index1 = list7.index(3)
print(index1)
# len(list): 获取列表的长度
print(len([1, 2, 3, 4, 5, 6]))
# max(list): 获取列表中的最大值
print(max([1, 2, 3, 4, 5, 6]))
# min(list): 获取列表中的最小值
print(min([1, 2, 3, 4, 5, 6]))
# count(val): 计算列表中元素出现的次数 val：列表中的某一个元素
list8 = [1, 3, 4, 3, 3, 3, 3, 6, 7]
print(list8.count(3))

# reverse(): 将列表进行倒序排列
list9 = [1, 2, 3, 4, 5, 10, 7, 8, 9]
list9.reverse()
print(list9)
# sort(): 将列表中的元素进行升序排列
list10 = [10, 90, 78, 23, 65, 43]
list10.sort()
print(list10)
enumerate(list[, start = 0]):将列表中元素和其对应的下标一一展示出来
list: 列表
start: [start = 下标值]

list16 = list(enumerate(list15))
list17 = list(enumerate(list15, start = 1))
print(list16)
print(list17)
for i, v in list16:
print(i, v)

元组tuple
元组:也是一种有序集合，一旦定义好之后不能修改
*# 创建一个空元祖*
tuple1 = ()
print(tuple1)
print(type(tuple1))
*# tuple和list一样也可以保存任意数据类型的数据*
tuple2 = (231, 323.14, 'asddf', **None**, **True**)
print(tuple2)
*# 注意：当创建的元组只有一个元素时，需要在元素后面加一个逗号*
tuple3 = (34, ) *# ===> tuple类型 # tuple3 = (34) # ===> int类型*
print(tuple3)
*# 获取元组的元素和list一样，通过下标来获取*
tuple4 = (1, 2, 3, 4, 5, 6)
print(tuple4[-1])
*# 注意：元组定义好之后是不能修改的*
tuple5 = (1, 2, 3, 4, [11, 22, 33])
*# tuple5[0] = 150 # 报错*
print(tuple5[-1][-2]) *# ==> print(tuple5[4][1])*
*# 元组进行操作*
*# 将两个元组合并成一个新的元组*
tuple7 = (1, 2, 3)
tuple8 = (4, 5, 6)
tuple9 = tuple7 + tuple8
print(tuple9)
print(tuple7, tuple8)
*# 将元组重复n次,最后返回一个新的元组*
tuple10 = (1, 2, 3)
print(tuple10 \* 3)
*# in 判断一个元素是否在一个元组中,如果在返回True,否则返回False*
tuple11 = (3, 5, 7)
print(3 **in** tuple11)
print(4 **in** tuple11)
*# 元组的截取*
*# 格式: 元组名[start:end:step]*
tuple12 = (1, 2, 3, 4, 5, 6, 7, 8, 9,10)
print(tuple12[2:5])
print(tuple12[3:])
print(tuple12[:3])
*# 将元组进行逆序排列*
print(tuple12[::-1])
*# 二维元组*
tuple13 = ((2, 3, 4), (5, 6, 7))
print(tuple13)
print(tuple13[1][1])
*# 元组的方法*
*# len(): 求一个元组的长度*
tuple14 = (1, 2, 3, 4, 5)
print(len(tuple14))
*# max(): 求一个元组的元素的最大值*
print(max(tuple14))
*# min(): 求一个元组的元素的最小值*
print(min(tuple14))
*#遍历*
**for** i **in** tuple14:
print(i)

dict字典

在字典里边是以键值对的形式存放. key-value
注意事项:
1.在字典中键(key)的值必须是唯一
2.在字典中可以存放多个键值对
3.在字典中键(key)必须是不可变类型 字符串 、整数都可以作为键(key)
4.list和tuple都是有序集合, 而dict是无序集合
*# 定义字典*
dict1 = {'Alex': 88, 'xiaoming' : 99, 'Ala': 96, 'Tom': 85, 'Jhon': 77}
print(dict1)
*# 元素的访问*
*# 获取的方式: 字典名[key]*
print(dict1['xiaoming'])
print(dict1.get('Alex'))
*# 添加元素*
dict1['HanMeimei'] = 89
print(dict1)
*# 修改*
dict1['HanMeimei'] = '90'
print(dict1)
*# 删除*
dict1.pop('HanMeimei')
print(dict1)
*# 遍历*
**for** k **in** dict1:
print(k, dict1[k])
*# 获取字典所有的value和key*
print(dict1.values())
print(dict1.keys())
*# 遍历value值*
**for** v **in** dict1.values():
print(v)
*#遍历key值*
**for** k **in** dict1.keys():
print(k)
*#遍历key-value值*
**for** k,v **in** dict1.items():
print(k, v)
**for** k,v **in** enumerate(dict1):
print(k, v)

set集合
set(集合):类似于dict, 也是无序的,以key的形式存在,没有value
作用: 是对list,tuple进行去重的, 求交集、并集
1.set是无序集合
2.set集合是不可改变的

# 获取set
set1 = set([1, 2, 2, 3, 3, 3])
print(set1)
print(type(set1))
set2 = set((1, 2, 3, 1, 2, 3, 4, 5))
print(set2)
print(type(set2))
set3 = set({3, 4, 5, 5, 6, 3, 7})
print(set3)
print(type(set3))
*# 添加*
set4 = set([1, 2, 2, 3, 3, 3])
set4.add(55) *# add() #可以添加重复的值,但是没效果*
*# set4.add([10, 9]) #直接报错,不能添加list*
set4.add((10, 9))
*# set4.add({'a':1}) #直接报错,不能添加字典*
*# 总结:list和dict是可改变的, 而tuple是不可改变*
print(set4)
*# 修改*
set5 = set([1, 2, 3, 4, 5])
*# 将list dict tuple 等等整个插入进去*
set5.update([6, 7, 8])
set5.update({9 : 'a', 10 : 'b'}) *#注意：插入字典时，是将key插进去*
set5.update((11, 56))
print(set5)
*# 删除*
set6= set([1, 2, 3, 4, 5])
set6.remove(4)
print(set6)
*# 遍历*
set7 = set((231, 323.14, 'asddf', **None**, **True**))
**for** i **in** set7:
print(i)
*# 交集 & 并集 | 差集 - 对称差 ^*
set8 = set([1, 2, 3, 4])
set9 = set([3, 4, 5, 6])
set10 = set8 & set9
set11 = set8 | set9
print(set10)
print(type(set10))
print(set11)
print(type(set11))

类型转换

*#list-set*
list1 = [1, 2, 3, 4, 4, 5, 5]
set1 = set(list1)
print(set1)
print(type(set1))
*# set->list*
set2 = {1, 2, 3, 4, 5, 6}
list2 = list(set2)
print(list2)
print(type(list2))
*# tuple->set*
tuple1 = (1, 2, 3, 4, 5, 6, 6, 6)
set3 = set(tuple1)
print(set3)
print(type(set3))
*# set->tuple*
set4 = {1, 2, 3, 4, 5, 6}
tuple2 = tuple(set4)
print(tuple2)
print(type(tuple2))
*# dict->set*
dict1 = {'Alex': 88, 'xiaoming' : 99, 'Ala': 96, 'Tom': 85, 'Jhon': 77}
set5 = set(dict1)
print(set5)
print(type(set5))
*# set-dict 注意：set不能转换成dict，因为set只有key，而dict是key-value形式存储的*
*# set6 = {2, 3, 4, 5}*
*# dict2 = dict(set6)*
*# print(dict2)*

流程控制：分支与循环

分支：

在Python中，要构造分支结构可以使用if、elif和else关键字。

结构：

if 表达式1:

语句1

elif 表达式2:

语句2

...

else:

语句n

练习：英制单位与公制单位互换

value = float(input('请输入长度: '))
unit = input('请输入单位: ')
if unit == 'in' or unit == '英寸':
print('%f英寸 = %f厘米' % (value, value \* 2.54))
elif unit == 'cm' or unit == '厘米':
print('%f厘米 = %f英寸' % (value, value / 2.54))
else:
print('请输入有效的单位')

循环：

构造循环语句要使用 for...in... 或者 while 关键字。

1、for...in...

结构：

for 变量 in 可迭代容器:

语句...

eg：求1-100的和

sum = 0
for x in range(101):
sum += x
print(sum)

2、while

结构：

while 表达式：

语句...

eg：猜数字游戏
计算机出一个1~100之间的随机数由人来猜
计算机根据人猜的数字分别给出提示大一点/小一点/猜对了

import random
answer = random.randint(1, 100)
counter = 0
while True:
counter += 1
number = int(input('请输入: '))
if number < answer:
print('大一点')
elif number > answer:
print('小一点')
else:
print('恭喜你猜对了!')
break
print('你总共猜了%d次' % counter)
if counter > 7:
print('你的智商余额明显不足')

练习：打印九九乘法表，要使用双层循环

for i in range(1, 10):
for j in range(1, i + 1):
print('%d\*%d=%d' % (i, j, i \* j), end='\t')
print()