第 **1** 天的课后作业

基础知识

## [1] 分别解释"=","==","+="的含义

"="（赋值操作符）：在Python中，"="被用于将一个值赋给一个变量。它表示将右侧的值赋给左侧的变量。

"=="（相等操作符）：在Python中，"=="被用于比较两个值是否相等。它返回一个布尔值（True或False），表示两个值是否相等。

"+="（增量赋值操作符）：在Python中，"+="用于将右侧的值与左侧的变量相加，并将结果赋给左侧的变量。

1. 执行如下代码，n1 和 n2 两个变量之间有何关系？

n1 = 123456

n2 = n1

在给变量n2赋值时，使用了变量n1的值。这意味着n2和n1具有相同的值。

## 现有如下两个变量,请根据执行结果解释原因

n1 = 123456

n2 = n1 n1 = 333

print(n1,n2)

輸出結果為333,123456,这是因为在第一行代码中，n1被赋值为123456，然后将n1的值赋给n2，使得n2也等于123456。接着，在第三行代码中，n1的值被修改为333。由于n2最初是根据n1的值赋值的，所以它并不受n1的后续修改的影响。

## 简述 Python 中的常用的几种数据类型，重点叙述其区别

## 整数（int）是用来表示整数值的数据类型，例如计算年龄或计数。

## 浮点数（float）是用来表示带有小数部分的数值，例如计算圆的面积或处理浮点数运算。

## 字符串（str）用于处理文本数据，例如存储姓名或表示句子或段落。

## 列表（list）是用来存储一组有序的元素，方便进行添加、删除和修改操作。

## 元组（tuple）类似于列表，但它们是不可变的，适用于存储不可更改的数据。

## 字典（dict）用于存储键-值对的数据，方便通过键进行快速查找和访问。

## 集合（set）是用于存储唯一元素的无序集合，方便进行集合操作（如并集、交集、差集）。

## 这些数据类型之间的区别主要在于它们的特性和用途。其中，整数和浮点数用于处理数字，字符串用于处理文本，列表和元组用于存储有序的数据集，字典用于存储键-值对的数据，集合用于存储唯一的元素集合。

1. 什么是可变与不可变的数据类型？可变与不可变分别有哪些？

可变数据类型是指在创建后可以更改其内容的数据类型。这意味着可以添加、删除或修改可变对象的元素，而不会改变其身份（identity）。在对可变对象进行更改时，不需要重新创建一个新的对象。常见的可变数据类型包括：列表（list）字典（dict）集合（set）

不可变数据类型是指一旦创建就无法更改其内容的数据类型。这意味着不可变对象的值是固定的，不能直接修改。如果需要对不可变对象进行更改，必须创建一个新的对象。常见的不可变数据类型包括：整数（int）浮点数（float）字符串（str）元组（tuple）

## 哪些变量或数据类型的布尔值为 False？（如：0）

（1）定义为False的对象：

None和False

（2）值为0的数字类型：

0，0.0 ，oj, Decimal(0), Fraction(0,1)

（3）空的序列和合集：

' ' ，（），[]，{}，set(),range(0)

1. python 中，变量命名规范有哪些？

◼字符串中：“”与’’不区分。

◼ 变量名只能是 字母、数字或下划线的任意组合

◼ 变量名的第一个字符不能是数字

◼ 以下关键字不能声明为变量名['and', 'as', 'assert', 'break', 'class',

'continue', 'def', 'del', 'elif', 'else', 'except', 'exec', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'not', 'or', 'pass', 'print', 'raise', 'return', 'try', 'while', 'with', 'yield']

## 请写出 and 、or、not 的作用，并用代码来演示

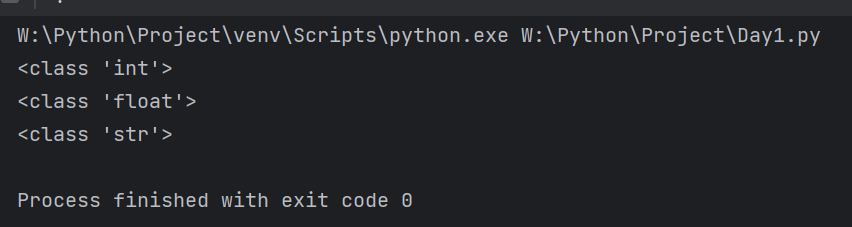
## and：逻辑与操作，返回 True 当且仅当两个表达式都为真。

## or：逻辑或操作，返回 True 如果至少有一个表达式为真。

## not：逻辑非操作，对表达式的结果进行取反。

# 使用 and 运算符  
result = (x < y) and (y < z)  
print(result) # 输出 True  
  
result = (x > y) and (y < z)  
print(result) # 输出 False  
  
# 使用 or 运算符  
result = (x > y) or (y < z)  
print(result) # 输出 True  
  
result = (x > y) or (y > z)  
print(result) # 输出 False  
  
# 使用 not 运算符  
value = True  
result = not value  
print(result) # 输出 False  
  
value = False  
result = not value  
print(result) # 输出 True

1. 编程查看 2、2.22、“Tsinghua University”分别是什么数据类型？
2. a = 2  
   b = 2.22  
   c = "Tsinghua University"  
     
   print(type(a))  
   print(type(b))  
   print(type(c))



# 实践应用（必做）

## 有如下变量，请实现要求的功能

tu = ("amao", [11, 22, {"k1": 'v1', "k2": ["age", "name"], "k3": (11,22,33)}, 44])

* + 讲述元组的特性

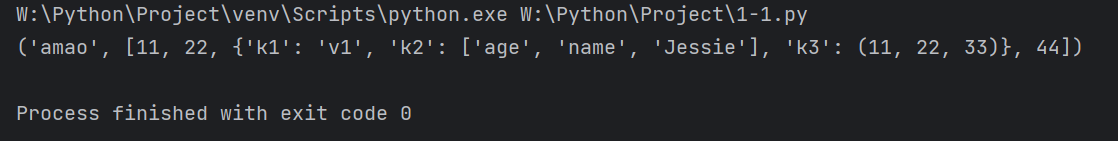
元组是不可变的、有序的数据结构，用于存储和组合多个元素。

* + 请问 tu 变量中的第一个元素“amao”是否可被修改？

不可以, 元组的元素不可被直接修改

* + 请问 tu 变量中的"k2"对应的值是什么类型？是否可以被修改？如果可以，请在其中添加一个元素“Jessie”

"k2" 对应的值是一个列表 ["age", "name"]。列表中的值可以被修改

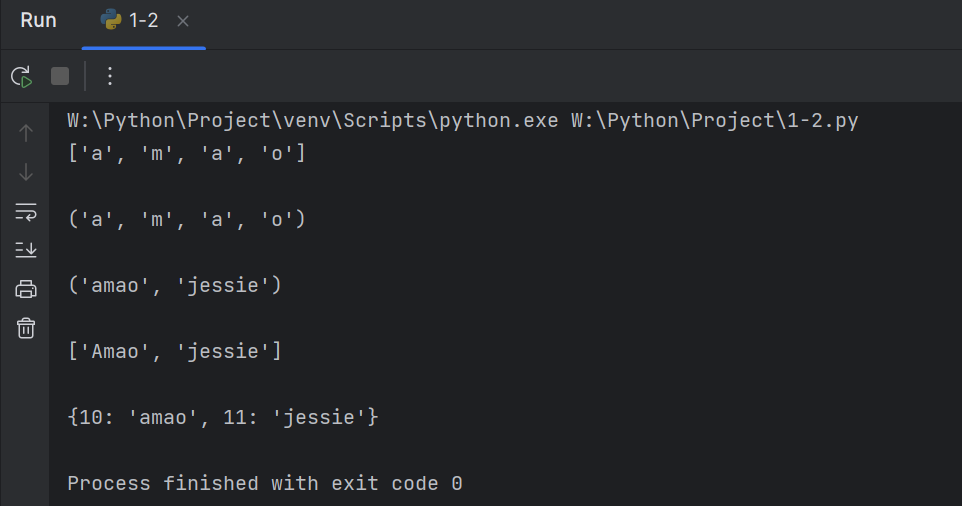


* + 请问 tu 变量中的"k3"对应的值是什么类型？是否可以被修改？如果可以，请在其中添加一个元素“Jessie”

"k3" 对应的值是一个元组 (11, 22, 33)。元組中的值不能被修改

## 变量间转换

* + 将字符串 s = "amao"转换成列表
  + 将字符串 s = "amao"转换成元祖
  + 将列表 li = ["amao", "jessie"]转换成元组
  + 将元祖 tu = ('Amao', "jessie")转换成列表
  + 将列表 li = ["amao", "jessie"]转换成字典且字典的 key 按照 10 开始向后递增



## 编写一个接受句子的程序，并计算大写字母和小写字母的数量。

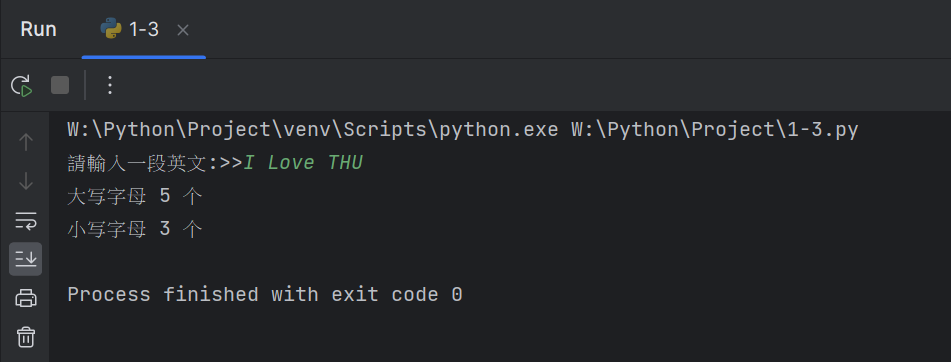
假设为程序提供了以下输入：

Hello world!

输出：

大写字母 1 个

小写字母 9 个



# 实践应用（选做）

## 网站要求用户输入用户名和密码进行注册。编写程序以检查用户输入的密码的有效性， 输出密码是否符合要求，若不符合要求，还应同时输出为何不符合要求。以下是检查密码的标准：

* + 至少有一个小写字母
  + 至少有一个数字
  + 至少有一个大写字母
  + 至少有一个“@、#、$、%、&”字符
  + 密码的最小长度：6
  + 密码的最大长度：12

## 双色球选购：

需求描述：

* + 双色球（假设一共八个球，6 个红球，球号 1-32、2 个蓝球，球号 1-16）
  + 确保用户不能重复选择，不能超出范围
  + 用户输入有误时有相应的错误提示
  + 最后展示用户选择的双色球的号码