请完成下面的练习，在做每个练习时，都编写一个独立的程序。保存每个程序时，使用符合标准 Python 约定的文件名：使用小写字母和下划线，如 simple\_message.py 和

simple\_messages.py 。

2-1 简单消息： 将一条消息存储到变量中，再将其打印出来。

2-2 多条简单消息： 将一条消息存储到变量中，将其打印出来；再将变量的值修改为一条新消息，并将其打印出来。

2-3 个性化消息： 将用户的姓名存到一个变量中，并向该用户显示一条消息。显示的消息应非常简单，如 “Hello Eric, would you like to learn some Python today?” 。

2-4 调整名字的大小写： 将一个人名存储到一个变量中，再以小写、大写和首字母大写的方式显示这个人名。

2-5 名言： 找一句你钦佩的名人说的名言，将这个名人的姓名和他的名言打印出来。输出应类似于下面这样（包括引号）：

Albert Einstein once said, “A person who never made a mistake never tried anything new.”

2-6 名言 2 ： 重复练习 2-5 ，但将名人的姓名存储在变量 famous\_person 中，再创建要显示的消息，并将其存储在变量 message 中，然后打印这条消息。

2-7 剔除人名中的空白： 存储一个人名，并在其开头和末尾都包含一些空白字符。务必至少使用字符组合 "\t" 和 "\n" 各一次。

打印这个人名，以显示其开头和末尾的空白。然后，分别使用剔除函数 lstrip() 、 rstrip() 和 strip() 对人名进行处理，并将结果打印出来。

2-9 最喜欢的数字： 将你最喜欢的数字存储在一个变量中，再使用这个变量创建一条消息，指出你最喜欢的数字，然后将这条消息打印出来。

2-10 添加注释： 选择你编写的两个程序，在每个程序中都至少添加一条注释。如果程序太简单，实在没有什么需要说明的，就在程序文件开头加上你的姓名和当前日

期，再用一句话阐述程序的功能。

3-1 姓名： 将一些朋友的姓名存储在一个列表中，并将其命名为 names 。依次访问该列表中的每个元素，从而将每个朋友的姓名都打印出来。

3-2 问候语： 继续使用练习 3-1 中的列表，但不打印每个朋友的姓名，而为每人打印一条消息。每条消息都包含相同的问候语，但抬头为相应朋友的姓名。

3-3 自己的列表： 想想你喜欢的通勤方式，如骑摩托车或开汽车，并创建一个包含多种通勤方式的列表。根据该列表打印一系列有关这些通勤方式的宣言，如 “I would

like to own a Honda motorcycle” 。

3-4 嘉宾名单 ：如果你可以邀请任何人一起共进晚餐（无论是在世的还是故去的），你会邀请哪些人？请创建一个列表，其中包含至少 3 个你想邀请的人；然后，使用

这个列表打印消息，邀请这些人来与你共进晚餐。

3-5 修改嘉宾名单 ：你刚得知有位嘉宾无法赴约，因此需要另外邀请一位嘉宾。

以完成练习 3-4 时编写的程序为基础，在程序末尾添加一条 print 语句，指出哪位嘉宾无法赴约。

修改嘉宾名单，将无法赴约的嘉宾的姓名替换为新邀请的嘉宾的姓名。

再次打印一系列消息，向名单中的每位嘉宾发出邀请。

3-6 添加嘉宾 ：你刚找到了一个更大的餐桌，可容纳更多的嘉宾。请想想你还想邀请哪三位嘉宾。

以完成练习 3-4 或练习 3-5 时编写的程序为基础，在程序末尾添加一条 print 语句，指出你找到了一个更大的餐桌。

使用 insert() 将一位新嘉宾添加到名单开头。

使用 insert() 将另一位新嘉宾添加到名单中间。

使用 append() 将最后一位新嘉宾添加到名单末尾。

打印一系列消息，向名单中的每位嘉宾发出邀请。

3-7 缩减名单 ：你刚得知新购买的餐桌无法及时送达，因此只能邀请两位嘉宾。

以完成练习 3-6 时编写的程序为基础，在程序末尾添加一行代码，打印一条你只能邀请两位嘉宾共进晚餐的消息。

使用 pop() 不断地删除名单中的嘉宾，直到只有两位嘉宾为止。每次从名单中弹出一位嘉宾时，都打印一条消息，让该嘉宾知悉你很抱歉，无法邀请他来共进

晚餐。

对于余下的两位嘉宾中的每一位，都打印一条消息，指出他依然在受邀人之列。

使用 del 将最后两位嘉宾从名单中删除，让名单变成空的。打印该名单，核实程序结束时名单确实是空的。

3-8 放眼世界 ：想出至少 5 个你渴望去旅游的地方。

将这些地方存储在一个列表中，并确保其中的元素不是按字母顺序排列的。

按原始排列顺序打印该列表。不要考虑输出是否整洁的问题，只管打印原始 Python 列表。

使用 sorted() 按字母顺序打印这个列表，同时不要修改它。

再次打印该列表，核实排列顺序未变。

使用 sorted() 按与字母顺序相反的顺序打印这个列表，同时不要修改它。

再次打印该列表，核实排列顺序未变。

使用 reverse() 修改列表元素的排列顺序。打印该列表，核实排列顺序确实变了。

使用 reverse() 再次修改列表元素的排列顺序。打印该列表，核实已恢复到原来的排列顺序。

使用 sort() 修改该列表，使其元素按字母顺序排列。打印该列表，核实排列顺序确实变了。

使用 sort() 修改该列表，使其元素按与字母顺序相反的顺序排列。打印该列表，核实排列顺序确实变了。

3-9 晚餐嘉宾 ：在完成练习 3-4~ 练习 3-7 时编写的程序之一中，使用 len() 打印一条消息，指出你邀请了多少位嘉宾来与你共进晚餐。

3-10 尝试使用各个函数 ：想想可存储到列表中的东西，如山岳、河流、国家、城市、语言或你喜欢的任何东西。编写一个程序，在其中创建一个包含这些元素的列

表，然后，对于本章介绍的每个函数，都至少使用一次来处理这个列表。