1. 把今天讲的列表，元组，字典，字符串的所有接口基本功能练习一遍，并用思维导出总结好（不清楚哪些是基础功能，就是照着上课代码练习一遍）
2. 编写代码理解 局部变量与全局变量
3. 求两个有序数字列表的公共元素
4. 给定一个n个整型元素的列表a，其中有一个元素出现次数超过n / 2，求这个元素
5. 列表、元组，字典的相同点，不同点有哪些，请罗列
6. 将元组 (1,2,3) 和集合 {4,5,6} 合并成一个列表。
7. 在列表 [1,2,3,4,5,6] 首尾分别添加整型元素 7 和 0。
8. 反转列表 [0,1,2,3,4,5,6,7] 。
9. 反转列表 [0,1,2,3,4,5,6,7] 后给出中元素 5 的索引号。
10. 分别统计列表 [True,False,0,1,2] 中 True,False,0,1,2的元素个数，发现了什么？
11. 从列表 [True,1,0,‘x’,None,‘x’,False,2,True] 中删除元素‘x’。
12. 从列表 [True,1,0,‘x’,None,‘x’,False,2,True] 中删除索引号为4的元素。
13. 删除列表中索引号为奇数（或偶数）的元素。
14. 清空列表中的所有元素。
15. 对列表 [3,0,8,5,7] 分别做升序和降序排列。
16. 将列表 [3,0,8,5,7] 中大于 5 元素置为1，其余元素置为0。
17. 遍历列表 [‘x’,‘y’,‘z’]，打印每一个元素及其对应的索引号。
18. 将列表 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] 拆分为奇数组和偶数组两个列表。
19. 分别根据每一行的首元素和尾元素大小对二维列表 [[6, 5], [3, 7], [2, 8]] 排序。
20. 从列表 [1,4,7,2,5,8] 索引为3的位置开始，依次插入列表 [‘x’,‘y’,‘z’] 的所有元素。
21. 快速生成由 [5,50) 区间内的整数组成的列表。
22. 若 a = [1,2,3]，令 b = a，执行 b[0] = 9， a[0]亦被改变。为何？如何避免？----讲了深COPY和浅COPY再做
23. 将列表 [‘x’,‘y’,‘z’] 和 [1,2,3] 转成 [(‘x’,1),(‘y’,2),(‘z’,3)] 的形式。
24. 以列表形式返回字典 {‘Alice’: 20, ‘Beth’: 18, ‘Cecil’: 21} 中所有的键。
25. 以列表形式返回字典 {‘Alice’: 20, ‘Beth’: 18, ‘Cecil’: 21} 中所有的值。
26. 以列表形式返回字典 {‘Alice’: 20, ‘Beth’: 18, ‘Cecil’: 21} 中所有键值对组成的元组。
27. 向字典 {‘Alice’: 20, ‘Beth’: 18, ‘Cecil’: 21} 中追加 ‘David’:19 键值对，更新Cecil的值为17。
28. 删除字典 {‘Alice’: 20, ‘Beth’: 18, ‘Cecil’: 21} 中的Beth键后，清空该字典。
29. 判断 David 和 Alice 是否在字典 {‘Alice’: 20, ‘Beth’: 18, ‘Cecil’: 21} 中。
30. 遍历字典 {‘Alice’: 20, ‘Beth’: 18, ‘Cecil’: 21}，打印键值对。
31. 若 a = dict()，令 b = a，执行 b.update({‘x’:1})， a亦被改变。为何？如何避免？----讲了深COPY和浅COPY再做
32. 以列表 [‘A’,‘B’,‘C’,‘D’,‘E’,‘F’,‘G’,‘H’] 中的每一个元素为键，默认值都是0，创建一个字典。
33. 将二维结构 [[‘a’,1],[‘b’,2]] 和 ((‘x’,3),(‘y’,4)) 转成字典。
34. 将元组 (1,2) 和 (3,4) 合并成一个元组。
35. 将空间坐标元组 (1,2,3) 的三个元素解包对应到变量 x,y,z。
36. 返回元组 (‘Alice’,‘Beth’,‘Cecil’) 中 ‘Cecil’ 元素的索引号。
37. 返回元组 (2,5,3,2,4) 中元素 2 的个数。
38. 判断 ‘Cecil’ 是否在元组 (‘Alice’,‘Beth’,‘Cecil’) 中。
39. 返回在元组 (2,5,3,7) 索引号为2的位置插入元素 9 之后的新元组。
40. 创建一个空集合，增加 {‘x’,‘y’,‘z’} 三个元素。
41. 删除集合 {‘x’,‘y’,‘z’} 中的 ‘z’ 元素，增j加元素 ‘w’，然后清空整个集合。
42. 返回集合 {‘A’,‘D’,‘B’} 中未出现在集合 {‘D’,‘E’,‘C’} 中的元素（差集）。
43. 返回两个集合 {‘A’,‘D’,‘B’} 和 {‘D’,‘E’,‘C’} 的并集。
44. 返回两个集合 {‘A’,‘D’,‘B’} 和 {‘D’,‘E’,‘C’} 的交集。
45. 返回两个集合 {‘A’,‘D’,‘B’} 和 {‘D’,‘E’,‘C’} 未重复的元素的集合。
46. 判断两个集合 {‘A’,‘D’,‘B’} 和 {‘D’,‘E’,‘C’} 是否有重复元素。
47. 判断集合 {‘A’,‘C’} 是否是集合 {‘D’,‘C’,‘E’,‘A’} 的子集。
48. 去除数组 [1,2,5,2,3,4,5,‘x’,4,‘x’] 中的重复元素。
49. 返回字符串 ‘abCdEfg’ 的全部大写、全部小写和大下写互换形式。
50. 判断字符串 ‘abCdEfg’ 是否首字母大写，字母是否全部小写，字母是否全部大写。
51. 返回字符串 ‘this is python’ 首字母大写以及字符串内每个单词首字母大写形式。
52. 判断字符串 ‘this is python’ 是否以 ‘this’ 开头，又是否以 ‘python’ 结尾。
53. 返回字符串 ‘this is python’ 中 ‘is’ 的出现次数。
54. 返回字符串 ‘this is python’ 中 ‘is’ 首次出现和最后一次出现的位置。
55. 将字符串 ‘this is python’ 切片成3个单词。
56. 返回字符串 ‘blog.csdn.net/xufive/article/details/102946961’ 按路径分隔符切片的结果。
57. 将字符串 ‘2.72, 5, 7, 3.14’ 以半角逗号切片后，再将各个元素转成浮点型或整形。
58. 判断字符串 ‘adS12K56’ 是否完全为字母数字，是否全为数字，是否全为字母？
59. 将字符串 ‘there is python’ 中的 ‘is’ 替换为 ‘are’。
60. 清除字符串 ‘\t python \n’ 左侧、右侧，以及左右两侧的空白字符。
61. 将三个全英文字符串（比如，‘ok’, ‘hello’, ‘thank you’）分行打印，实现左对齐、右对齐和居中对齐效果。
62. 将三个字符串（比如，‘Hello, 我是David’, ‘OK, 好’, ‘很高兴认识你’）分行打印，实现左对齐、右对齐和居中效果。
63. 将三个字符串 ‘15’, ‘127’, ‘65535’ 左侧补0成同样长度。
64. 将列表 [‘a’,‘b’,‘c’] 中各个元素用’|'连接成一个字符串。
65. 将字符串 ‘abc’ 相邻的两个字母之间加上半角逗号，生成新的字符串。
66. 从键盘输入手机号码，输出形如 ‘Mobile: 186 6677 7788’ 的字符串。
67. 从键盘输入年月日时分秒，输出形如 ‘2019-05-01 12:00:00’ 的字符串。
68. 给定两个浮点数 3.1415926 和 2.7182818，格式化输出字符串 ‘pi = 3.1416, e = 2.7183’。
69. 将 0.00774592 和 356800000 格式化输出为科学计数法字符串。
70. 将列表 [0,1,2,3.14,‘x’,None,’’,list(),{5}] 中各个元素转为布尔型。
71. 返回字符 ‘a’ 和 ‘A’ 的ASCII编码值。
72. 返回ASCII编码值为 57 和 122 的字符。
73. 将列表 [3,‘a’,5.2,4,{},9,[]] 中 大于3的整数或浮点数置为1，其余置为0。
74. 将二维列表 [[1], [‘a’,‘b’], [2.3, 4.5, 6.7]] 转为 一维列表。
75. 将等长的**键列表和值列表**转为字典。
76. 数字列表求和。