# file path: data\1\_dict\_if\_kv.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、检查字典中是否有特定内容的 键  
2、检查字典中是否有特定内容的 值  
"""  
  
achiev = {'小张':80, '小王':92, '小李':75,  
 '小赵':100, '小刘':60}  
  
print(achiev.keys())  
print(achiev.values())  
  
print('-'\*20)  
  
name = '小王'  
if name in achiev.keys(): # 检查字典中是否有 键 等于字符串 name  
 print('成绩单上有' + name + '的成绩')  
  
score = 100  
if score in achiev.values(): # 检查字典中是否有 值 等于数字100  
 print('这次有人得了' + str(score) + '分')

# file path: data\2\_dict\_if\_items.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、检查字典中是否有特定内容的 键值对  
2、字典自带的items()函数返回的是元组的列表  
"""  
  
achiev = {'小张':80, '小王':92, '小李':75, '小赵':100, '小刘':60}  
  
print(achiev.items()) # 注意items()返回的是元组的列表  
  
print('-'\*20)  
  
name = '小王'  
score = 100 # 尝试92等不同数值  
  
## 检查字典中是否有 键值对  
# 因为items()是元组的列表，这里必须要用圆括号(name,score)  
if (name,score) in achiev.items():   
 print('这次' + name + '得了' + str(score) + '分。')  
else:  
 print('这次' + name + '没有得到' + str(score) + '分。')

# file path: data\website\1\_dict\_lookup.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、字典由“键-值对”（key-value）组成  
2、形式和日常的“英汉字典”形式类似  
3、掌握根据“键”查找字典中对应的“值”  
"""  
  
# 声明一个字典  
UN\_dict = { '中国':'China',   
 '美国':'United States',   
 '俄罗斯':'Russian Federation',   
 '法国':'France',   
 '英国':'United Kingdom'}  
# UNSC: United Nations Security Council  
  
print(UN\_dict)  
print('------------------')  
print(UN\_dict['美国']) # “查字典”的方式为：字典名[键]  
print(UN\_dict['俄罗斯'].upper())  
  
print('------------------')  
#print(UN\_dict['德国']) # 字典名[不存在的键] 会导致程序出错  
  
# 不确定的情况下，应该用if语句先检查键是否在字典中（后续讲解）  
# 或者使用更安全的“查字典”方式：字典名.get(键, 键不存在时返回的信息)  
print(UN\_dict.get('德国', '未查到'))   
print(UN\_dict.get('法国', '未查到'))

# file path: data\website\大报\2\_dict\_change.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、更新字典内容：修改、添加、删除、清空等  
2、修改和添加用同样的方式，这也保证了列表的“键”不会重复  
"""  
  
UN\_dict = { '中国':'China',   
 '美国':'United States',  
 '俄罗斯':'Russian Federation',   
 '法国':'France',   
 '英国':'United Kingdom'}  
  
print('字典长度：' + str(len(UN\_dict)))  
print('-'\*20)  
  
# 更新字典中的内容  
UN\_dict['美国'] = 'USA'  
print(UN\_dict['美国'])  
  
print('-'\*20)  
# 向字典添加内容，方式和更新字典内容一样，区别在于使用了新的“键”  
UN\_dict['巴西'] = 'Brazil'  
print(UN\_dict['巴西'])  
  
print(UN\_dict)  
print('字典长度：' + str(len(UN\_dict)))  
  
print('-'\*20)  
# 删除字典中的“键值对”。注意，删除不存在的“键”会导致程序报错  
del UN\_dict['英国']  
print(UN\_dict)  
print('字典长度：' + str(len(UN\_dict)))  
  
print('-'\*20)  
# 清空字典  
UN\_dict.clear()  
print(UN\_dict)  
print('字典长度：' + str(len(UN\_dict)))

# file path: data\website\大报\3\_dict\_update.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
合并两个字典  
"""  
  
UN\_dict = { '中国':'China',   
 '美国':'United States',   
 '俄罗斯':'Russian Federation',   
 '法国':'France',   
 '英国':'United Kingdom'}  
brics = {'巴西':'Brazil',   
 '俄罗斯':'RUSSIAN FEDERATION',  
 '印度':'India',   
 '中国':'CHINA',   
 '南非':'South Africa'}  
  
UN\_dict.update(brics) # 合并字典  
print(UN\_dict) # 注意重复键'俄罗斯'和'中国'的变化  
print('-'\*30)  
print(brics) # brics字典仍然存在

# file path: data\website\省报\4\_dict\_alias.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、复制字典不能直接用等号赋值，这样只是建立一个“别名”（alias）  
2、“别名”可以理解成对同一个字典起了个新名字，新老名字都在用  
3、要达到复制的效果，可以先建立一个空字典，再用update()函数合并字典  
"""  
  
bric = {'巴西':'Brazil',   
 '俄罗斯':'Russian Federation',  
 '印度':'India',   
 '中国':'China'}  
  
new\_bric = bric # new\_bric只是一个别名  
new\_bric['南非'] = 'South Africa' # 改变new\_bric，其实就是改变bric  
print(new\_bric)  
print(bric)  
  
print('-'\*20)  
  
# 真正的复制字典  
bric = {'巴西':'Brazil',   
 '俄罗斯':'Russian Federation',  
 '印度':'India',  
 '中国':'China'}  
new\_bric = {} # 建立一个空字典new\_bric  
new\_bric.update(bric) # 将bric的内容复制到new\_bric，现在new\_bric和bric内容相同  
new\_bric['南非'] = 'South Africa' # 改变new\_bric，不会影响bric  
print(new\_bric)  
print(bric)

# file path: data\实习生\4\_dict\_for\_sorted.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、用for循环遍历字典，顺序是不确定的  
2、可以用sorted()函数对键或值进行排序  
3、注意.keys()和.values()返回的都是列表  
"""  
  
# 用字典保存成绩表  
achiev = {'Zhang':80,  
 'Wang':92,  
 'Li':75,  
 'Zhao':100,  
 'Liu':60  
 }  
  
print('---按升序输出“键”---')  
for key in sorted(achiev.keys()):  
 print(key)  
  
print('---按降序输出“键”---')  
for key in sorted(achiev.keys(), reverse=True):  
 print(key)  
   
print('---按升序输出“值”---')  
for value in sorted(achiev.values()):  
 print(value)  
  
print('---按降序输出“值”---')  
for value in sorted(achiev.values(), reverse=True):  
 print(value)  
  
print('---“键值对”是按什么排序呢？---')  
for key,value in sorted(achiev.items()):  
 print(key, value)

# file path: data\实习生\rrr\5\_dict\_for\_set.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、用for循环遍历字典  
2、values()函数返回字典里所有的值，可能会有重复  
3、set()函数生成集合，类似列表，但没有重复值  
"""  
  
# 用字典保存各人喜欢的水果  
favor = { '小张':'苹果',  
 '小王':'葡萄',  
 '小李':'桔子',  
 '小赵':'苹果',  
 '小刘':'桔子'  
 }  
  
print('---打印“值”---')  
print(favor.values())  
print()  
print('---打印“值”的集合---')  
print(set(favor.values())) # set()函数返回的是“集合”  
  
print('='\*20)  
print('大家爱吃的水果包括：')  
for value in set(favor.values()): # 去除了重复值  
 print(value)

# file path: data\实习生\xxx\2\_dict\_in\_list.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
字典可以作为列表的元素  
"""  
  
stu\_0 = {'身高':175, '体重':78, '年龄':22}  
stu\_1 = {'身高':188, '体重':94, '年龄':19}  
stu\_2 = {'身高':163, '体重':82, '年龄':25}  
  
stu\_list = [stu\_0, stu\_1, stu\_2]  
  
for stu in stu\_list:  
 print(stu)  
   
print('-'\*20)  
print('1号学生的体重是：' +  
 str(stu\_list[1]['体重']) +  
 '公斤。')

# file path: data\实习生\xxx\3\_dict\_for.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、用for循环遍历字典  
2、keys()函数返回字典里所有的键  
3、values()函数返回字典里所有的值  
4、items()函数返回字典里所有的“键值对”  
"""  
  
# 用字典保存成绩表  
achiev = {'小张':80, '小王':92, '小李':75,  
 '小赵':100, '小刘':60}  
  
print('---打印“键”---')  
print(achiev.keys())  
for key in achiev.keys():  
 print(key)  
  
print('---打印“值”---')  
print(achiev.values())  
for value in achiev.values():  
 print(value)  
   
print('---打印“键值对”---')  
print(achiev.items()) # items()函数返回的是元组的列表  
#for (key,value) in achiev.items(): 这里可以加圆括号，也可以不加  
for key,value in achiev.items():  
 print(key + '的成绩是' + str(value) + '分。')

# file path: data\实习生\yyy\1\_dict\_type.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、字符串、数字、列表、字典都可以作为字典的“值”  
2、字符串、数字都可以作为字典的“键”  
3、列表、字典不能作为字典的“键”  
4、还有一些特殊的类型可以作为字典的“键”，并不常用，在此不做介绍  
"""  
  
# 可以用来记录个人信息  
info1 = {'身高':175, '体重':80, '电话':'1234'}  
info2 = {'身高':190, '体重':75, '电话':'4567'}  
  
# 字典也可以作为另一个字典的值  
record = {'小张':info1, '小王':info2}  
print('小张的全部信息：', record['小张'])  
print('小张的身高：', record['小张']['身高'])  
print('小王的电话：', record['小王']['电话'])  
  
print('-'\*20)  
# 键：数字；值：列表  
# 记录每天的考试科目  
exam = {1:['语文','物理'], 2:['数学'], 3:['英语','历史','地理']}  
print('第1天所有考试：', exam[1])  
print('第1天第1门考试：' + exam[1][0])  
print('第3天第3门考试：' + exam[3][2])

# file path: data\移动硬盘\1\3\_dict\_in\_list\_range.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、生成若干位学生的信息  
2、随机生成身高、体重、年龄  
3、假想场景为学校篮球队选择球员，根据身高安排位置  
"""  
import random as r  
  
stu\_list = [] # 空列表，用于添加学生信息  
  
# 逐个生成学生信息，每个学生的信息是一个字典  
for i in range(15):  
 h = r.randrange(155,199) # 随机产生身高  
 w = r.randrange(45,95) # 随机产生体重  
 age = r.randrange(18,25) # 随机产生年龄  
 new\_stu = {'身高':h, '体重':w, '年龄':age, '位置':'未分配'}  
 stu\_list.append(new\_stu) # 添加到列表中  
  
## 打印列表中的信息  
#for stu in stu\_list:  
# print(stu)  
  
# 根据身高粗略地分配位置  
for stu in stu\_list:  
 if stu['身高'] >= 190:  
 stu['位置'] = '中锋'  
 elif stu['身高'] >= 175:  
 stu['位置'] = '前锋'  
 else:  
 stu['位置'] = '后卫'  
   
# 打印列表中的信息  
for stu in stu\_list:  
 print(stu)

# file path: data\移动硬盘\3\1\_dict\_\_homework.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
练习题：做一个成绩查询和录入系统  
1、建立一个字典，包含若干“姓名:成绩”  
2、用户输入学生姓名（拼音或英文），可以查询成绩  
3、如果系统中没有该学生姓名，则可以为该学生录入成绩，按回车跳过录入成绩  
4、可以反复查询，直到按q退出  
5、退出时，打印如下信息  
（1）按姓名排序，打印成绩系统中的所有条目  
（2）打印学生总人数和平均成绩  
"""  
  
achiev = {'Zhang':80, 'Wang':92, 'Li':75,  
 'Zhao':100, 'Liu':60}  
  
print('---欢迎登录成绩查询和录入系统---')  
  
while True:  
 name = input('请输入姓名（按q退出）：')  
 if name.lower() == 'q':  
 break  
 if name == '':  
 continue  
   
 if name in achiev.keys():  
 print(name + '的成绩是：' + str(achiev[name]))  
 else:  
 print('没有查到' + name + '的成绩')  
 score = input('请录入' + name + '的成绩（按回车跳过）:')  
 if score == '':  
 continue  
 elif score.isdigit():  
 achiev[name] = int(score)  
 print('已录入：' + name + '，' + str(achiev[name]) + '分')  
 else:  
 print('输入不合规范，忽略……')  
  
print('---打印全部成绩---')  
print('姓名\t成绩') # '\t'为制表符，相当于按一次TAB键  
sum = 0  
for name,score in sorted(achiev.items()):  
 print(name + '\t' + str(score))  
 sum = sum + score  
  
num = len(achiev)  
print('---总人数：' + str(num))  
print('---平均分：' + str(sum//num))

# file path: data\移动硬盘\4\_dict\_in\_list\_choice.py

"""  
Date: 2020.11.10  
Author: Justin  
  
要点说明：  
1、生成若干位学生的爱好信息，如最喜欢的颜色、数字、符号等  
2、掌握random.choice()函数的运用  
"""  
  
import random as r  
# 空列表，用于添加学生信息  
stu\_list = []  
  
# “最喜欢的XX”候选信息  
color\_list = ['Red','Yellow','Green','Blue','Purple']  
num\_list = [0,1,1,2,3,5,8,13]  
char\_list = 'helloworld!@#$%^&' # 字符串可以看作特殊的列表  
  
# 逐个生成学生信息，每个学生的信息是一个字典  
for i in range(15):  
 color = r.choice(color\_list) # 随机选取颜色  
 num = r.choice(num\_list) # 随机选取数字  
 char = r.choice(char\_list) # 随机选取字符  
 new\_stu = {'颜色':color, '数字':num, '字符':char}  
 stu\_list.append(new\_stu) # 添加到列表中  
  
# 打印列表中的信息  
for stu in stu\_list:  
 print(stu)