

# Python实验报告3--参考答案

1 班级:

姓名:

学号:

要求:

1. 在文件开头部分填写自己的信息;
2. 在每个题下方的代码块中书写该题的代码，并运行出结果;
3. 在2节课的时间内完成前6个题；打印为pdf文件并提交，文件名改为“Python实验3+班级姓名.pdf”。

一、实验目的:

- 掌握元组、字典、集合的创建及操作方法；
- 掌握可迭代对象的一些常用内置函数；
- 能合理使用Python组合数据类型解决一些简单的实际问题。

二、实验内容（在各题后书写程序和运行结果）

- 1 1. 填充题：目前，我们接触到的Python数据类型有：  
2 A. 数值、 B. 字符串、 C. range对象、 D. 列表、 E. 元组、 F. 字典、 G. 集合。  
3 (1) 上述的数据类型中，可变数据类型有 ( D、F、G ) ；  
4 (2) 上述的数据类型中，哪些可以使用双向索引访问 ( B、C、D、E ) ；  
5 (3) 上述的数据类型中，哪些可以作为字典的键 ( A、B、C、E ) ；  
6 (4) 上述的数据类型中，哪些是可迭代对象 ( B、C、D、E、F、G ) 。

2. 已知下列的两个元组，试产生一个字典，键来自元组1，值来自元组2的对应位置。

```
In [1]:  
1 day = ('Sun', 'Mon', 'Tues', 'Wed', 'Thur', 'Fri', 'Sat')  
2 index = ('日', '一', '二', '三', '四', '五', '六')  
3 date = {key:val for key, val in zip(day, index)}  
4 date
```

```
Out[1]: {'Sun': '日',  
         'Mon': '一',  
         'Tues': '二',  
         'Wed': '三',  
         'Thur': '四',  
         'Fri': '五',  
         'Sat': '六'}
```

```
In [2]:  
1 day = ('Sun', 'Mon', 'Tues', 'Wed', 'Thur', 'Fri', 'Sat')  
2 index = ('日', '一', '二', '三', '四', '五', '六')  
3 date = dict(zip(day, index))  
4 date
```

```
Out[2]: {'Sun': '日',  
         'Mon': '一',  
         'Tues': '二',  
         'Wed': '三',  
         'Thur': '四',  
         'Fri': '五',  
         'Sat': '六'}
```

3. 对下列的一个16位随机正整数，选出是奇数的各位数字，放到一个列表中。要求使用`map()`。

In [3]:

```
1 from random import randint
2
3 num = randint(1e15, 1e16)
4 print('16位随机整数为:', num)
5 nums = list(map(int, str(num)))
6 alist = [n for n in nums if n%2]
7 print('其中的奇数为:', alist)
```

16位随机整数为: 6980252924838012  
其中的奇数为: [9, 5, 9, 3, 1]

4. 已知某六人宿舍目前有四个同学，他们是小陈，小何，小梁，小谢，他们分别选了4号床，3号床，5号床，2号床。

(1) 以姓名为键，床位为值建立一个字典stu1。

(2) 后来又来了小王和小李，他们分别选了1号床和6号床，且小何和小梁换了床位，对字典stu1更新信息。

In [4]:

```
1 # (1)
2 stu1 = {"小陈": "4号床", "小何": "3号床", "小梁": "5号床", "小谢": "2号床"}
3 # (2)
4 stu1.update({"小王": "1号床", "小李": "6号床", "小何": "5号床", "小梁": "3号床"})
5 stu1
```

Out[4]: {'小陈': '4号床', '小何': '5号床', '小梁': '3号床', '小谢': '2号床', '小王': '1号床', '小李': '6号床'}

In [5]:

```
1 # (1)
2 stu1 = {"小陈": "4号床", "小何": "3号床", "小梁": "5号床", "小谢": "2号床"}
3 # (2)
4 stu1["小王"] = "1号床"
5 stu1["小李"] = "6号床"
6 stu1["小何"], stu1["小梁"] = stu1["小梁"], stu1["小何"]
7 stu1
```

Out[5]: {'小陈': '4号床', '小何': '5号床', '小梁': '3号床', '小谢': '2号床', '小王': '1号床', '小李': '6号床'}

5. 已知某宿舍同学小陈、小何、小梁、小谢、小王、小李的学号为202311921123～202311921128。以姓名为键，学号为值，创建一个字典stu2。尝试多种方法创建。

In [6]:

```
1 # 循环中对空字典添加元素
2 names = ["小陈", "小何", "小梁", "小谢", "小王", "小李"]
3 stu2 = {}
4 for i in range(len(names)):
5     stu2[names[i]] = 202311921123+i
6 stu2
```

Out[6]: {’小陈’: 202311921123,
’小何’: 202311921124,
’小梁’: 202311921125,
’小谢’: 202311921126,
’小王’: 202311921127,
’小李’: 202311921128}

In [7]:

```
1 # 字典推导式
2 names = ["小陈", "小何", "小梁", "小谢", "小王", "小李"]
3 stu2 = {names[i]:202311921123+i for i in range(6)}
4 stu2
```

Out[7]: {’小陈’: 202311921123,
’小何’: 202311921124,
’小梁’: 202311921125,
’小谢’: 202311921126,
’小王’: 202311921127,
’小李’: 202311921128}

In [8]:

```
1 # 循环中利用enumerate
2 names = ["小陈", "小何", "小梁", "小谢", "小王", "小李"]
3 stu2 = {}
4 for num, name in enumerate(names, 202311921123):
5     stu2[name] = num
6 stu2
```

Out[8]: {’小陈’: 202311921123,
’小何’: 202311921124,
’小梁’: 202311921125,
’小谢’: 202311921126,
’小王’: 202311921127,
’小李’: 202311921128}

In [9]:

```
1 # 字典推导式中利用 enumerate
2 names = ["小陈", "小何", "小梁", "小谢", "小王", "小李"]
3 stu2 = {name:num for num, name in enumerate(names, 202311921123)}
4 stu2
```

Out[9]: {’小陈’: 202311921123,
’小何’: 202311921124,
’小梁’: 202311921125,
’小谢’: 202311921126,
’小王’: 202311921127,
’小李’: 202311921128}

6. 已知问题5中创建的字典stu2，将其各项的键值对颠倒，以键为值、以值为键，产生另一个字典stu3。

In [10]:

```

1 # 用循环将键值对颠倒
2 stu3 = {}
3 for i, v in stu2.items():
4     stu3[v] = i
5 stu3

```

Out[10]: {202311921123: '小陈',  
202311921124: '小何',  
202311921125: '小梁',  
202311921126: '小谢',  
202311921127: '小王',  
202311921128: '小李'}

In [11]:

```

1 # 用字典推导式将原来的键值对颠倒
2 stu3 = {v:i for i,v in stu2.items()}
3 stu3

```

Out[11]: {202311921123: '小陈',  
202311921124: '小何',  
202311921125: '小梁',  
202311921126: '小谢',  
202311921127: '小王',  
202311921128: '小李'}

In [12]:

```

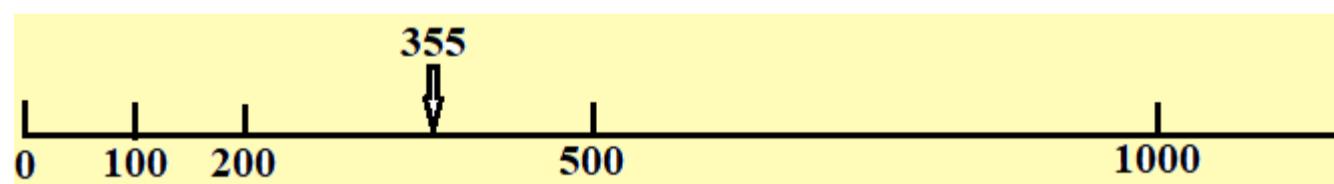
1 # 用zip将颠倒的键值对组合成字典
2 stu3 = dict(zip(stu2.values(), stu2.keys()))
3 stu3

```

Out[12]: {202311921123: '小陈',  
202311921124: '小何',  
202311921125: '小梁',  
202311921126: '小谢',  
202311921127: '小王',  
202311921128: '小李'}

7. 某商场促销活动，总价低于100元内不打折， $[100, 200)$ 打9折， $[200, 500)$ 打8.5折， $[500, 1000)$ 打8折，1000以上打7.5折。现有顾客数人，他们购买商品总价分别为 $[355, 704, 256, 883, 1521, 197, 556]$ ，请以字典形式保存商品总价和实际收款数，并按格式输出：“总价为s的商品，打折后为t”，精确到小数点后2位。

显然，可以用 `if-elif-else` 结构确定每个总价对应的折扣。但下面不准备使用这个结构，而用与阶梯阈值的大小关系来确定折扣。



x	$\geq 0$	$\geq 100$	$\geq 200$	$\geq 500$	$\geq 1000$
折扣	1.0	0.9	0.85	0.8	0.75
355	1	1	1	0	0

如：总价355满足上述不等式中的前3个，因此，对应的折扣是第3个：0.85.

In [13]:

```
1 thr = [0, 100, 200, 500, 1000] # 阶梯阈值
2 dis = [1, 0.9, 0.85, 0.8, 0.75] # 对应的折扣
3 total = [355, 704, 256, 883, 1521, 197, 556]
4 real = [] # 实际付款
5 for t in total:
6     geq = [t>=n for n in thr]
7     r = dis[sum(geq)-1]
8     real.append(round(r*t, 2))
9 {t:r for t,r in zip(total, real)}
```

Out[13]: {355: 301.75,  
704: 563.2,  
256: 217.6,  
883: 706.4,  
1521: 1140.75,  
197: 177.3,  
556: 444.8}

8. 使用random模块的randint函数产生[1,10]上的100个随机整数，然后用一个字典记录各不同的数字及其出现次数。

In [14]:

```
1 from random import randint
2
3 nums = [randint(1, 10) for i in range(100)]
4 print('100个随机整数为: \n', nums)
5 n_set = set(nums)
6 ns = []
7 # 在循环中记录各数的次数
8 for n in n_set:
9     ns.append(nums.count(n))
10 adict = dict(zip(n_set, ns))
11 print('各数字出现的次数为: \n', adict)
```

100个随机整数为:

```
[9, 3, 5, 2, 9, 8, 6, 3, 6, 6, 6, 4, 4, 6, 3, 3, 10, 6, 3, 8, 9, 3, 6, 10, 6, 6, 10, 4, 7, 3, 2, 5, 10, 9, 6, 10, 3, 6, 6, 3, 5, 1, 3, 5, 7, 8, 2, 6, 1, 4, 8, 3, 2, 9, 5, 3, 1, 1, 10, 5, 8, 7, 1, 1, 0, 2, 1, 7, 10, 10, 4, 7, 7, 9, 7, 7, 4, 8, 6, 7, 2, 4, 10, 3, 9, 1, 10, 10, 4, 3, 10, 2, 10, 7, 2, 9, 8, 8, 4, 4]
```

各数字出现的次数为:

```
{1: 7, 2: 8, 3: 14, 4: 10, 5: 6, 6: 15, 7: 10, 8: 8, 9: 8, 10: 14}
```

In [15]:

```
1 from random import randint
2
3 nums = [randint(1, 10) for i in range(100)]
4 print('100个随机整数为: \n', nums)
5 n_set = set(nums) # 去掉相同的元素
6 adict = {n:nums.count(n) for n in n_set} # 字典推导式
7 print('各数字出现的次数为: \n', adict)
```

100个随机整数为:

```
[10, 10, 7, 5, 7, 8, 10, 4, 1, 5, 6, 3, 5, 10, 2, 6, 9, 8, 1, 6, 9, 9, 2, 10, 5, 7, 10, 3, 9, 4, 7, 7, 9, 2, 10, 10, 8, 7, 7, 9, 1, 5, 4, 2, 4, 8, 8, 10, 4, 1, 2, 9, 9, 7, 5, 5, 8, 7, 3, 3, 3, 6, 3, 9, 8, 10, 8, 10, 8, 3, 5, 4, 9, 3, 6, 4, 3, 8, 1, 3, 4, 6, 5, 8, 6, 1, 1, 9, 7, 1, 8, 2, 8, 1, 9, 7, 5, 1, 3]
```

各数字出现的次数为:

```
{1: 10, 2: 6, 3: 12, 4: 8, 5: 10, 6: 7, 7: 11, 8: 13, 9: 12, 10: 11}
```

In [16]:

```
1 from random import randint
2
3 dict1 = {i:0 for i in range(1,51)} # 初始化字典
4 # 以下在循环中产生随机整数，并直接累计各数出现的次数
5 for i in range(100):
6     x = randint(1,10)
7     dict1[x] = dict1[x]+1
8 dict2 = {i:v for i,v in dict1.items() if v} # 去掉值为0的键值对
9 print('各数字出现的次数为: \n', adict)
```

各数字出现的次数为:

```
{1: 10, 2: 6, 3: 12, 4: 8, 5: 10, 6: 7, 7: 11, 8: 13, 9: 12, 10: 11}
```

9. (选做题) 对问题4中的字典stu1中的元素逐个输出，要求：

(1) 每行输出一个人名及其床位，中间用":"分隔。

(2) 按床位从小到大的顺序输出。

In [17]:

```
1 # 按床位从小到大的顺序输出，假定已知床位的序号是1-6
2 for v in range(1, 7):
3     for i in stu1:
4         if stu1[i][0]==str(v):
5             print(i, ":", v, "号床")
6             break
```

```
小王 : 1 号床
小谢 : 2 号床
小梁 : 3 号床
小陈 : 4 号床
小何 : 5 号床
小李 : 6 号床
```

In [18]:

```
1 # 按床位从小到大的顺序输出，假定未知床位的序号
2 # 利用zip的排序特性产生一个新的字典
3 tmp = zip(stu1.values(), stu1.keys())
4 val, ind = zip(*sorted(tmp))
5 stu0 = dict(zip(ind, val))
6 for i, v in stu0.items():
7     print(i, ':', v)
```

```
小王 : 1号床
小谢 : 2号床
小梁 : 3号床
小陈 : 4号床
小何 : 5号床
小李 : 6号床
```

In [19]:

```
1 # 可用sorted()函数直接对字典按值排序
2 stu0 = sorted(stu1.items(), key=lambda x: x[1]) # 将产生一个列表
3 stu0 = dict(stu0)
4 for i, v in stu0.items():
5     print(i, ':', v)
```

```
小王 : 1号床
小谢 : 2号床
小梁 : 3号床
小陈 : 4号床
小何 : 5号床
小李 : 6号床
```

10. (选做题) 问题4和问题5中的同学是同一批人, 试将stu1和stu2合并成1个字典stu4, 以姓名为键, 值为元组, 包含该生的学号和床位。

In [20]:

```
1 stu4 = {}
2 for i, v in stu1.items():
3     stu4[i] = (stu2[i], v)
4 stu4
```

Out[20]: {’小陈’: (202311921123, ’4号床’),
’小何’: (202311921124, ’5号床’),
’小梁’: (202311921125, ’3号床’),
’小谢’: (202311921126, ’2号床’),
’小王’: (202311921127, ’1号床’),
’小李’: (202311921128, ’6号床’)}

In [21]:

```
1 stu4 = {i: (stu2[i], stu1[i]) for i in stu1}
2 stu4
```

Out[21]: {’小陈’: (202311921123, ’4号床’),
’小何’: (202311921124, ’5号床’),
’小梁’: (202311921125, ’3号床’),
’小谢’: (202311921126, ’2号床’),
’小王’: (202311921127, ’1号床’),
’小李’: (202311921128, ’6号床’)}

1 以上也可以将按床号顺序排列, 请自行尝试。

11. (选做题) 从问题9中的字典stu4出发, 将其键和值重新组合, 以床号为键, 姓名和学号组成的元组为值, 创建新字典stu5。

In [22]:

```
1 # 先分离姓名、学号、床号, 再组合
2 names = list(stu4.keys())
3 info = dict(stu4.values())
4 nums = list(info.keys())
5 bed = list(info.values())
6 stu5 = {bed[i]: (names[i], nums[i]) for i in range(len(bed))}
7 stu5
```

Out[22]: {’4号床’: (’小陈’, 202311921123),
’5号床’: (’小何’, 202311921124),
’3号床’: (’小梁’, 202311921125),
’2号床’: (’小谢’, 202311921126),
’1号床’: (’小王’, 202311921127),
’6号床’: (’小李’, 202311921128)}

In [23]:

```
1 # 使用字典推导式
2 stu5 = {v[1]: (i, v[0]) for i, v in stu4.items()}
3 stu5
```

Out[23]: {’4号床’: (’小陈’, 202311921123),
’5号床’: (’小何’, 202311921124),
’3号床’: (’小梁’, 202311921125),
’2号床’: (’小谢’, 202311921126),
’1号床’: (’小王’, 202311921127),
’6号床’: (’小李’, 202311921128)}