列表推导式

1. 定义：

   使用简易方法，将可迭代对象转换为列表。

2. 语法：

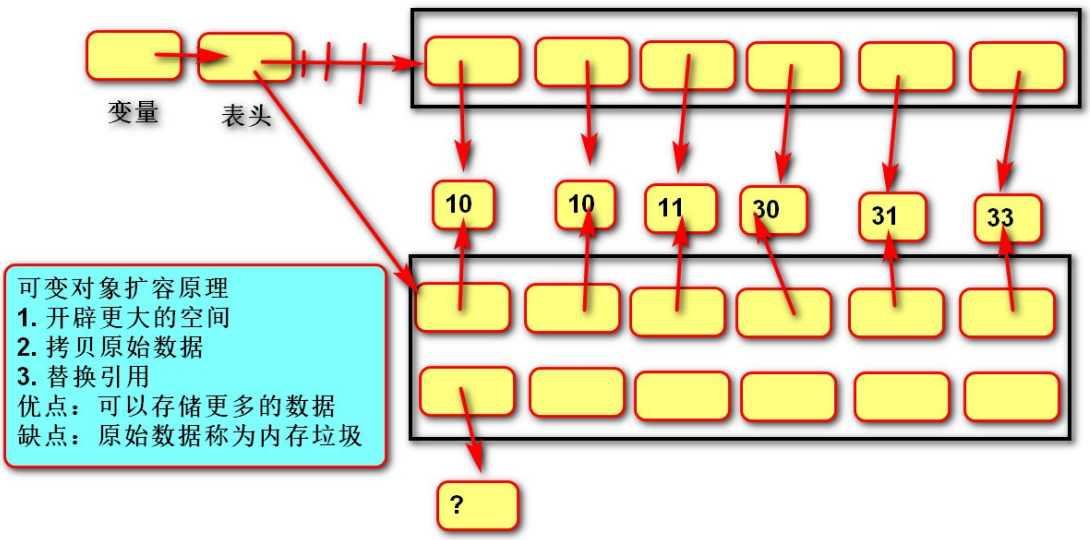
   变量 = [表达式 for 变量 in 可迭代对象]

   变量 = [表达式 for 变量 in 可迭代对象 if 条件]

3. 说明:

如果 if *真值表达式的布尔值为* False,*则可迭代对象生成的数据将被丢弃。*

```
练习：exercise01.py
# 需求1：将lie01中所有数据+1再存储另外一个列表
list01 = [5,15,55,6,6,7]
# list02 = []
# for item in list01:
#     list02.append(item + 1)
#
# list03 = [item + 1 for item in list01]
# print(list02)
# print(list03)
# 需求2：将lie01中大于10的数据，+1再存储另外一个列表
list02 = []
for item in list01:
    if item > 10:
        list02.append(item + 1)
list03 = [item + 1 for item in list01 if item > 10]
print(list02)
print(list03)
```



```python
# 练习1:使用range生成1--10之间的数字,将数字的平方存入list01中

# list01 = []
# for item in range(1,11):
#     list01.append(item ** 2)
list01 = [item ** 2 for item in range(1,11)]
print(list01)
# 练习2:将list01中所有奇数存入list02
# list02 = []
# for item in list01:
#     if item % 2:
#         list02.append(item)
list02 = [item for item in list01 if item % 2]
print(list02)
# 练习3:将list01中所有偶数增加10之后存储list03
list03 = [item + 10 for item in list01 if item % 2 == 0]
print(list03)
```

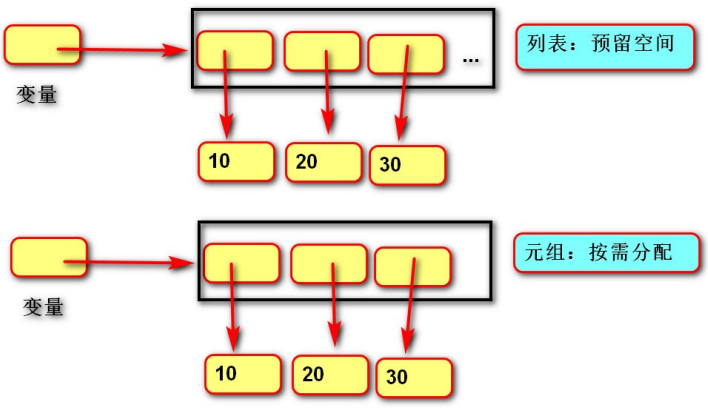
元组 tuple

定义

1. 由一系列变量组成的不可变序列容器。
2. 不可变是指一但创建，不可以再添加/删除/修改元素。

基础操作

1. 创建空元组：
   元组名 = ()
   元组名 = tuple()
2. 创建非空元组：
   元组名 = (20,)
   元组名 = (1, 2, 3)
   元组名 = 100,200,300
   元组名 = tuple(可迭代对象)
3. 获取元素：
   索引、切片
4. 遍历元组：
   正向：
   for 变量名 in 列表名:



变量名就是元素

反向:

for 索引名 in range(len(列表名)-1,-1,-1):

元祖名[索引名]就是元素

作用

1. 元组与列表都可以存储一系列变量，由于列表会预留内存空间，所以可以增加元素。
2. 元组会按需分配内存，所以如果变量数量固定，建议使用元组，因为占用空间更

```
""
    元组
    练习:exercise02.py
    练习:exercise03.py
"""
# 1. 创建元组
# 空元组
tuple01 = ()
tuple01 = tuple()
# 具有默认值元组
tuple02 = (1,2)
# 预留空间的存储机制 --> 按需分配的存储机制
list01 = ["a","b"]
tuple02 = tuple(list01)
print(tuple02)
# 按需分配的存储机制 --> 预留空间的存储机制
list02 = list(tuple02)
print(list02)
# tuple02 = ("a")# 不是元组
tuple02 = ("a",)# 注意:如果元组只有一个元素，必须要在末尾添加逗号
print(tuple02)
# 多个变量 = 序列
name01,name02 = ("无忌","翠山")
print(name01)
print(name02)
tuple03 = ("a","b","c","d")
# 获取元素
#   单个元素
print(tuple03[2])
#   多个元素
```

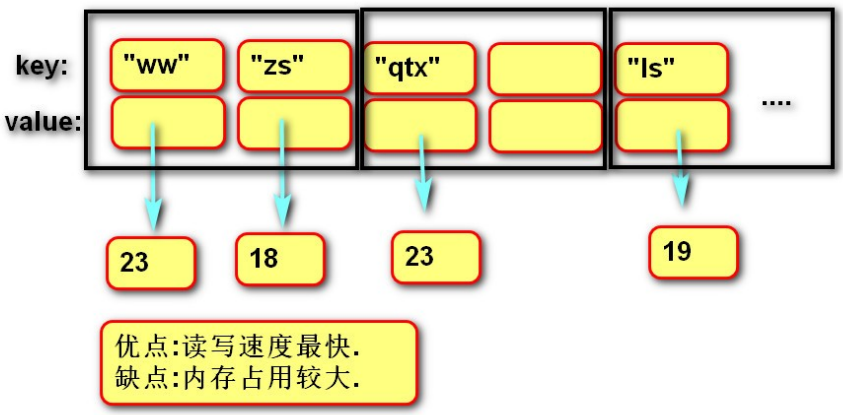
```python
print(tuple03[1:3])# ("b","c")
#   所有元素
#     正向
for item in tuple03:
    print(item)
#     反向
for i in range(len(tuple03)-1,-1,-1):
    print(tuple03[i])


# 使用容器的思想，改造下列代码。
"""
year = int(input("请输入年份："))
month = int(input("请输入月份："))  # 15
if month < 1 or month > 12:
    print("月份输入有误")
elif month == 2:
    if year % 4 == 0 and year % 100 != 0 or year % 400 ==
0:
        print("29天")
    else:
        print("28天")
elif month == 4 or month == 6 or month == 9 or month ==
11:
    print("30天")
else:
    print("31天")
"""
```

```
# 方法1：
year = int(input("请输入年份："))
month = int(input("请输入月份："))  # 15
if month < 1 or month > 12:
    print("月份输入有误")
elif month == 2:
    if year % 4 == 0 and year % 100 != 0 or year % 400 ==
0:
        print("29天")
    else:
        print("28天")
# elif month == 4 or month == 6 or month == 9 or month ==
11:
elif month in (4,6,9,11):
    print("30天")
else:
    print("31天")
"""
year = int(input("请输入年份："))
month = int(input("请输入月份："))  # 15
# if month < 1 or month > 12:
#     print("月份输入有误")
# elif month == 2:
#     if year % 4 == 0 and year % 100 != 0 or year % 400
== 0:
#         print("29天")
#     else:
#         print("28天")
# # elif month == 4 or month == 6 or month == 9 or month
== 11:
# elif month in (4,6,9,11):
#     print("30天")
# else:
#     print("31天")
if month < 1 or month > 12:
    print("月份输入有误")
else:
    day_of_month02 =29 if year % 4 == 0 and year % 100 !=
0 or year % 400 == 0 else 28
    days_of_month =
(31,day_of_month02,31,30,31,30,31,31,30,31,30,31)
    print(days_of_month[month - 1])
```

```
"""
    练习:
    在控制台中录入日期(年，月,日),计算这是这一年的第几天？
    2019,3月10日.--> 1月天数 + 2月天数 +  10
"""
year = int(input("请输入年份："))
month = int(input("请输入月份："))
day = int(input("请输入日："))
if month < 1 or month > 12:
    print("月份输入有误")
else:
    day_of_month02 = 29 if year % 4 == 0 and year % 100 !
= 0 or year % 400 == 0 else 28
    days_of_month =
(31,day_of_month02,31,30,31,30,31,31,30,31,30,31)
    """ 方式一
    # 先累加前几个月的天数
    total_day = 0
    for i in range(month - 1):
        total_day += days_of_month[i]
    """
    total_day = sum(days_of_month[:month - 1])
    # 再累加当月天数
    total_day += day
    print(total_day)
```

字典 dict



定义

1. 由一系列键值对组成的可变散列容器。
2. 散列：对键进行哈希运算，确定在内存中的存储位置，每条数据存储无先后顺序。
3. 键必须惟一且不可变(字符串/数字/元组)，值没有限制。

基础操作

1. 创建字典：

    字典名 = {键 1：值 1，键 2：值 2}

字典名 = dict (可迭代对象)

2. 添加/修改元素：

   语法:

   字典名[键] = 数据

   说明:

   键不存在，创建记录。

   键存在，修改映射关系。

3. 获取元素：

   变量 = 字典名[键] # 没有键则错误

4. 遍历字典：

   for 键名 in 字典名:

   字典名[键名]

   for 键名,值名 in 字典名.items():

   语句

5. 删除元素：

   del 字典名[键]

```
字典：
15:35
  练习:exercise04.py
  练习:exercise05.py
  练习:exercise06.py
"""
# 1. 创建字典
# 空
dict01 = {}
dict01 = dict()
# 具有默认值
dict02 = {"qtx":18,"ls":20,"ww":23}
list01 = [["a","b"],("c","d")]
dict02 = dict(list01)
print(dict02)
# 2. 添加元素
# 第一次(没有该键)添加
dict02["键"] = "值"
# 3. 修改元素
dict02["键"] = "值2"
dict02["a"] = "B"
print(dict02)
# 4. 删除
del dict02["键"]
print(dict02)
# 5.查找单个元素
```

```python
# 在查找元素时,如果字典中不存在该键，则错误.
# 所以查找前，一定通过 in 判断。
if "qtx" in dict02:
    print(dict02["qtx"])
else:
    print("不存在")
# del dict02["qtx"]#  KeyError:
# 获取所有元素
for key in dict02:
    print(key)# 键
    print(dict02[key])# 键 --> 值
# # (键,值)
# for item in dict02.items():
#     print(item[0])
#     print(item[1])
# (键,值)
for key,value in dict02.items():
    print(key)
    print(value)
# 获取字典中所有值
for value in dict02.values():
    print(value)
# 在 python3.6 以后,字典在功能上体现了加入的顺序.
dict02["三丰"] = 128
dict02["无忌"] = 28
dict02["翠山"] = 35
for item in dict02:
    print(item)


"""
    练习1：在控制台中录入商品信息(名称,价格)
        如果名称为空字符串,停止录入.
        -- 将所有商品的名称与价格打印出来(一个商品一行)
        -- 如果录入了"游戏机",则单独打印其价格.
"""
dict_commodity_info = {}
while True:
    name = input("请输入名称:")
    if name =="":
        break
    price = float(input("请输入价格:"))
    dict_commodity_info[name] = price
```

```python
for k,v in dict_commodity_info.items():
    print("%s 的价格是%f"%(k,v))
if "游戏机" in dict_commodity_info:
    value = dict_commodity_info["游戏机"]
    print("游戏机的价格是：%f"%value)



# 练习 2:在控制台中循环录入人的信息(姓名,年龄,性别,体重)

#           如果名称为空字符串,停止录入.
#       -- 将所有人的信息打印出来(一人一行)
# 数据结构：字典内嵌列表
# {
#   “无忌":[28,"男",80],
#   “赵敏":[26,"女",50],
# }
dict_person_info = {}
while True:
    name = input("请输入姓名：")
    if name == "":
        break
    age = int(input("请输入年龄："))
    sex = input("请输入性别：")
    weight = float(input("请输入体重："))
    dict_person_info[name] = [age, sex, weight]
for k_name, v_info in dict_person_info.items():
    # print("%s 的年龄是%d,性别是%s,体重是%f")
    print("%s--%d--%s--%.1f" % (k_name, v_info[0],
v_info[1], v_info[2]))



# 练习:在控制台中循环录入人的信息(姓名,年龄,性别,体重),如果名称为
空字符串,停止录入.
#       -- 将所有人的信息打印出来(一人一行)
#       -- 打印第一个人的信息
#       -- 打印最后一个人的信息
# 数据结构：列表内嵌字典
# [
#   {"name":“无忌","age":28,"sex":"男","weight":80},
#   {"name":“赵敏","age":26,"sex":"女","weight":50},
# ]
```

```
    总结：存储多个数据，使用什么数据结构？
          根据具体需求，结合优缺点，综合考虑(两害相权其轻)
    字典：
        优：根据 key 获取 value，速度最快.
            如果信息较多，通过键(指定的名称)获取,代码可读性偏好。
        缺:不能通过索引/切片获取元素，不灵活.
             内存占用较大
    列表
        优：能通过索引/切片获取元素，灵活.
        缺: 数据过多时，查找元素较慢.
             通过索引(整数)获取，如果信息较多，代码可读性偏差。
"""
list_person_info = []
while True:
    name = input("请输入姓名：")
    if name == "":
        break
    age = int(input("请输入年龄："))
    sex = input("请输入性别：")
    weight = float(input("请输入体重："))
    dict_info =
{"name":name,"age":age,"sex":sex,"weight":weight}
    list_person_info.append(dict_info)
for dict_item in list_person_info:
    print("%s 的年龄是%d,性别是%s,体重是%f"%
(dict_item["name"],dict_item["age"],dict_item["sex"],dict
_item["weight"]))
# 第一个人
dict_item = list_person_info[0]
print("%s 的年龄是%d,性别是%s,体重是%f" % (dict_item["name"],
dict_item["age"], dict_item["sex"], dict_item["weight"]))
# 最后一个人
dict_item = list_person_info[-1]
print("%s 的年龄是%d,性别是%s,体重是%f" % (dict_item["name"],
dict_item["age"], dict_item["sex"], dict_item["weight"]))
```

字典推导式

1. 定义：
   使用简易方法，将可迭代对象转换为字典。
2. 语法:
   {键:值 for 变量 in 可迭代对象}
   {键:值 for 变量 in 可迭代对象 if 条件}

```
字典推导式
    练习：exercise07.py
    练习：exercise08.py
"""
# 需求:range(10)  key 0---9   value 键的平方
dict01 = {}
for item in range(10):
    dict01[item] = item ** 2
dict02 = {item: item ** 2 for item in range(10)}
# 需求:range(10) 大于5的数， key 0---9   value 键的平方
dict01 = {}
for item in range(10):
    if item > 5:
        dict01[item] = item ** 2
dict02 = {item: item ** 2 for item in range(10) if item >
5}
print(dict01)
print(dict02)



# 练习1:list --> 字典,key 是列表元素，值是键的长度.

# ["无忌","翠山","张三丰"] --> {"无忌":2,"翠山":2,"张三丰":3}
list_names = ["无忌", "翠山", "张三丰"]
# dict_names = {}
# for item in list_names:
#     dict_names[item] = len(item)
dict_names = {item: len(item) for item in list_names}
print(dict_names)
```

```python
# 练习2:姓名列表:  ["无忌","周芷若","赵敏"]

#       房间号列表：[101,102,103]
#       将两个列表合并为一个字典,名字作为key,房间号作为value
list_names = ["无忌", "周芷若", "赵敏"]
list_room_number = [101, 102, 103]
# dict_name_room = {}
# for i in range(len(list_names)):
#     dict_name_room[list_names[i]] = list_room_number[i]
dict_name_room = {list_names[i]: list_room_number[i] for
i in range(len(list_names))}
print(dict_name_room)
```