

实例 3：生词本

背单词是英语学习中最基础的一环，不少学生在背诵单词的过程中会整理自己的生词本，以不断拓展自己的词汇量。本实例要求编写生词本程序，该程序需具备以下功能。

(1) 查看生词列表功能：输出生词本中全部的单词；若生词本中没有单词，则提示“生词本内容为空”。

(2) 背单词功能：从生词列表中取出一个单词，要求用户输入相应的翻译，输入正确提示“太棒了”，输入错误提示“再想想”。

(3) 添加新单词功能：用户分别输入新单词和翻译，输入完成后展示添加的新单词和翻译，并提示用户“单词添加成功”。若用户输入的单词已经存在于生词本中，提示“此单词已存在”。

(4) 删除单词功能：展示生词列表，用户输入单词以选择要删除的生词，若输入的单词不存在提示“删除的单词不存在”，生词删除后提示“删除成功”。

(5) 清空生词本功能：查询生词列表，若列表为空提示“生词本内容为空”，否则清空生词本中的全部单词，并输出提示信息“生词本已清空”。

(6) 退出生词本功能：退出生词本。

实例目标

- 熟练地创建集合
- 掌握集合的基本操作，能添加、删除、查看、清空集合元素

实例分析

本实例的生词本主要用于保存多个单词及其翻译，其中的单词与翻译是互相对应的，且单词是不能重复的，因此这里可将每组单词与翻译视为一个字典，将生词本视为包含多个单词的集合。但集合中存储的元素必须是可哈希类型的，其中不能包含字典，这时需要将字典转换为字符串之后再添加到集合中。

生词本具有 5 个功能，分别是查看生词本、背单词、添加新单词、删除单词和清空生词本。其中背单词的功能相当于遍历集合元素的操作，添加新单词的功能相当于往集合中添加元素的操作，删除单词和清空生词本的功能相当于删除和清空集合元素的操作。

代码实现

```
words_book = set()
words_only_set = set()
print('=' * 20)
```

```
print('欢迎使用生词本')
print('1.查看生词本')
print('2.背单词')
print('3.添加新单词')
print('4.删除单词')
print('5.清空生词本')
print('6.退出生词本')
print('=' * 20)
while True:
    word_dict = {}
    fun_num = input('请输入功能编号: ')
    if fun_num == '1': # 查看生词本
        if len(words_book) == 0:
            print('生词本内容为空')
        else:
            print(words_book)
    elif fun_num == '2': # 背单词
        if len(words_book) == 0:
            print('生词本内容为空')
        else:
            for random_words in words_book:
                w = random_words.split(':')
                in_words = input("请输入" + w[0] + '翻译'+': \n')
                if in_words == w[1].strip():
                    print('太棒了')
                else:
                    print('再想想')
    elif fun_num == '3': # 添加新单词
        new_words = input('请输入新单词: ')
        # 检测单词是否重复
        if new_words in words_only_set:
            # 添加的单词重复
            print('此单词已存在')
        else:
            # 执行单词添加
            new_chinese = input('请输入单词翻译: ')
            word_dict.update({new_words: new_chinese})
            # 转换成字符串存入 set 集合中
            dict_str = str(word_dict).replace('{', '').replace('}',
```

```
        ' ').replace("'", ' ')
        words_book.add(dict_str)
        print('单词添加成功')
        dict_str = dict_str.replace(',', ' ')
        print(dict_str)
        words_only_set.add(new_words)
elif fun_num == '4': # 删除单词
    if len(words_book) == 0:
        print('生词本为空')
    else:
        temp_list = list(words_book)
        print(temp_list)
        del_wd = input("请输入要删除的单词")
        # 如果要删除的单词不在单词集合中
        if del_wd not in words_only_set:
            print('删除的单词不存在')
        else:
            words_only_set.discard(del_wd)
            for temp in temp_list:
                if del_wd in temp:
                    words_book.remove(temp)
                    print('删除成功')
elif fun_num == '5': # 清空
    if len(words_book) == 0:
        print('生词本为空')
    else:
        words_only_set.clear()
        words_book.clear()
        print('生词本清空成功')
elif fun_num == '6': # 退出
    print('退出成功')
    break
```

以上代码首先创建了表示生词本的集合 `words_book` 和仅保存新单词的集合 `words_only_set`，然后输出了生词本的功能菜单提示用户选择功能序号，最后又创建了一个存储单词与翻译的字典 `word_dict`，接收用户输入的功能序号之后使用 `if-elif` 语句处理了不同序号的情况。

代码测试

运行程序，在控制台输入“3”之后的结果如下所示：

```
=====
欢迎使用生词本
1. 查看生词本
2. 背单词
3. 添加新单词
4. 删除单词
5. 清空生词本
6. 退出生词本
=====
请输入功能编号：3
请输入新单词：new
请输入单词翻译：新的
单词添加成功
new：新的
请输入功能编号：3
请输入新单词：perfect
请输入单词翻译：完美的
单词添加成功
perfect：完美的
```

在控制台输入“1”之后的结果如下所示：

```
请输入功能编号：1
{'perfect：完美的', 'new：新的'}
```

在控制台输入“2”之后的结果如下所示：

```
请输入功能编号：2
请输入 perfect 翻译：
完美的
太棒了
请输入 new 翻译：
陈旧的
再想想
```

在控制台输入“4”之后的结果如下所示：

```
请输入功能编号：4
['perfect：完美的', 'new：新的']
请输入要删除的单词 new
删除成功
```

在控制台输入“5”之后的结果如下所示：

请输入功能编号：5

生词本清空成功