

01 今日内容大纲

- 字典的初识
- 字典的使用（增删改查）
- 字典的嵌套

02 内容回顾以及作业讲解

- 列表：容器型数据类型，可以承载大量的数据，有序的数据。
 - 增：
 - append 追加
 - insert 插入
 - extend 迭代着追加
 - 删：
 - pop 按照索引删除。有返回值，默认删除最后一个
 - remove 按照元素删除
 - clear 清空
 - del 索引，切片（步长）
 - 改：
 - l1[1] = '大壮'
 - l1[1:3] = 'fdsafdsafdhjskf'
 - l1[1:4:2] = '太白'
 - 查：索引，切片，for循环
- 元组：只读列表，(), 拆包
- range：看做：可以自己控制范围的数字列表，但是它不是列表。
-

03 具体内容

- 字典的初识：
 - why：
 - 列表可以存储大量的数据，数据之间的关联性不强
 - ['太白', 18, '男', '大壮', 3, '男']
 - 列表的查询速度比较慢。
 - what：容器型数据类型：dict。
 - how：
 - 数据类型的分类（可变与不可变）：
 - 可变（不可哈希）的数据类型：list dict set
 - 不可变（可哈希）的数据类型：str bool int tuple
 - 字典：{}括起来，以键值对形式存储的容器型数据类型：

```
dic = {'太白':
      {'name': '太白金星', 'age': 18, 'sex': '男'},
      'python22期':
      ['朱光亚', '大壮', '雪飞', '岑哥'],
      }
```

- 键必须是不可变的数据类型：int, str (bool tuple几乎不用) 唯一的。
- 值可以是任意数据类型，对象。
- 字典3.5x版本之前（包括3.5）是无序的。
- 字典3.6x会按照初次建立字典的顺序排列，学术上不认为是有序的。
- 字典3.7x以后都是有序的。
- 字典的优点：查询速度非常快，存储关联性的数据。
- 字典的缺点：以空间换时间。

○ 字典的创建方式：

```
# 面试会考
# 方式一：
# dic = dict (('one', 1), ('two', 2), ('three', 3))
# print(dic) # {'one': 1, 'two': 2, 'three': 3}

# 方式二：
# dic = dict(one=1, two=2, three=3)
# print(dic)

# 方式三：
# dic = dict({'one': 1, 'two': 2, 'three': 3})
# print(dic)
```

○ 验证字典的合法性：

```
dic = {[1,2,3]: 'alex', 1: 666} # 键要不可变的数据类型
print(dic)
dic = {1: 'alex', 1: '太白', 2: 'wusir'} # 键要唯一
print(dic)
```

○ 字典的增删改查

```
dic = {'name': '太白', 'age': 18, 'hobby_list': ['直男', '钢管', '开车']}
# 增：
# 直接增加 有则改之，无则增加
# dic['sex'] = '男'
# dic['age'] = 23 # 改
# print(dic)
#.setdefault 有则不变，无则增加
# dic.setdefault('hobby')
# dic.setdefault('hobby', '球类运动')
# dic.setdefault('age', 45)
# print(dic)
```

```
# 删
# pop 按照键删除键值对，有返回值 ***
# 设置第二个参数则无论字典中有无此键都不会报错
# dic.pop('age')
# ret = dic.pop('age')
# ret = dic.pop('hobby','没有此键')
# print(ret)
# print(dic)

# clear 清空 **
# dic.clear()
# print(dic)

# del **
# # del dic['age']
# del dic['age1']
# print(dic)

# 改
# dic['name'] = 'alex'
# print(dic)

# 查
# print(dic['hobby_list'])
# print(dic['hobby_list1'])

# get ***
# l1 = dic.get('hobby_list')
# l1 = dic.get('hobby_list1','没有此键sb') # 可以设置返回值
# print(l1)

# 三个特殊的
# keys() values() items()
# print(dic.keys(),type(dic.keys()))
# 可以转化成列表
# print(list(dic.keys()))
# for key in dic.keys():
#     print(key)
# for key in dic:
#     print(key)

# values()
# print(dic.values())
# print(list(dic.values()))
# for value in dic.values():
#     print(value)

# items()
# print(dic.items())
```

```
# for key,value in dic.items():
#     print(key,value)
# a,b = ('name', '太白')
# print(a,b)
# 面试题
# a = 18
# b = 12
# a,b = b,a
# # a,b = 12,18
# print(a,b)
```

相关练习题：

```
dic = {'k1': "v1", "k2": "v2", "k3": [11,22,33]}
# 请在字典中添加一个键值对, "k4": "v4", 输出添加后的字典
# 请在修改字典中 "k1" 对应的值为 "alex", 输出修改后的字典
# 请在k3对应的值中追加一个元素 44, 输出修改后的字典
# print(dic['k3'])
# l1 = dic.get('k3')
# print(l1)
# dic.get('k3').append(44)
# print(dic)
# 请在k3对应的值的第 1 个位置插入个元素 18, 输出修改后的字典
```

字典的嵌套

```
dic = {
    'name': '汪峰',
    'age': 48,
    'wife': [{ 'name': '国际章', 'age': 38 },],
    'children': { 'girl_first': '小苹果', 'girl_second': '小怡', 'girl_three': '顶顶' }
}

# 1. 获取汪峰的名字。
#
# 2. 获取这个字典: { 'name': '国际章', 'age': 38 }。
#
# 3. 获取汪峰妻子的名字。
#
# 4. 获取汪峰的第三个孩子名字。
```

04 今日总结

- 字典：查询速度快，数据的关联性强。
 - 键不可变得数据类型，（str int），唯一。
 - 值：任意数据类型，对象。
 - 增删改查（全部都要会默写）***

- 字典的嵌套。***

05 明日预习内容

- 明天讲理论性偏多：id is == 小数据池，集合。