

字典 (dict)

新华字典----厚

1. 字典是Python中唯一——一个映射类型
2. dict构成：
 - 键:key
 - 值:value
 - 键值对---项
3. 字典是无序的
4. 不是序列类型，是映射类型
5. 是可迭代对象

- 字典的创建及访问

1. 直接创建
 - `d={'one':1,'jiang':'江'}`
2. dict()
 - 1. dict() 创建空字典
 - 2. dict(mapping)
 - 利用mapping (字典) 创建对象
 - `d=dict({'one':1,'two':2})`
 - 3. dict(iterable)
 - `d2=dict([(1,'one'),(2,'two')])`
 - `d2=dict(['ab','cd'])`
 - 4. dict(**kwargs)
 - `d3=dict(a=10,b=20)`

字典的访问

1. 基本形式
 - 变量名[键名]----值
 - 如果键不存在，则报错
2. 添加一个键值对
 - 当键不存在的时候
 - 变量名[键名]=值
3. 修改一个键所对应的的值
 - 当键存在时
 - 变量名[键名]=新值

- 字典的方法

1. clear()
 - 清空字典
2. copy()
 - 浅拷贝
3. fromkeys (可迭代对象, value=None)
 - 根据可迭代对象的元素创建字典
 - key: 从可迭代对象中的元素

v : 从value获取
d.fromkeys([1,2,3])
d.fromkeys([1,2,3],('one','two','three'))

4. get(k,[d])

k:键
d:提示信息
{'one': 1, 'jiang': '江'}
>>> d.get('one')
1
>>> d.get('two')
>>> d.get('two','没有查出')
'没有查出'

5. items()

将字典中键值对封装成元组，并把它放到类集合中
dict_items([('one', 1), ('jiang', '江')])
>>> list(d.items())
[('one', 1), ('jiang', '江')]

6. keys()

同上取键

7. pop(k,[d])

k:键
d:提示信息
根据k删除一个键值对，并把键值对进行返回
如果键不存在，则返回提示信息，如果d值没有设置，则报错

8. popitem()

删除某个键值对，并返回
如果字典为空则报错

9. setdefault(k,[v=None])

k:键
v:值
如果键不在字典中就会创建一个键值对
>>> d
{'one': 1, 'jiang': '江', 'hui': None}
>>> d.setdefault('hu','辉')
'辉'

如果键存在，不会创建键值对，也不会修改
{'one': 1, 'jiang': '江', 'hui': None, 'hu': '辉'}
>>> d.setdefault('hu','kk')

10. update([E],**F)

E:是一个字典或者是一个可迭代对象（元素必须是两个元素的序列）
F:可变成参数
如果键存在，则修改值，如果不存在则创建键值对
>>> d.update({'a':10,'b':20})
>>> d
{'one': 1, 'jiang': '江', 'a': 10, 'b': 20}
>>> d.update(a=10,b=20)
>>> d
{'one': 1, 'jiang': '江', 'a': 10, 'b': 20}
>>> d.update({'a':10,'b':20},c=30)
>>> d
{'one': 1, 'jiang': '江', 'a': 10, 'b': 20, 'c': 30}

11. values()

```
返回一个类集合对象，包含字典的值
>>> d.values()
dict_values([1, '江', 10, 20, 30])
>>> list(d.values())
[1, '江', 10, 20, 30]
>>>
```

- 字典是可迭代对象

1. 如果用for循环返回键

```
>>> for i in d:
    print(i)
```

2. 如何输出键值对

```
>>> for i in d.items():
    print(i)
```

3. in not in

只支持键的查询，不支持值的查询