|  |  |
| --- | --- |
| 字典创建申明 | Example\_Dict = {key1:value1,key2:value2} |
| 字典值获取 | 1 example\_dict[key] 可以获取key所对应的值，如果key不存在则报错  2 example\_dict.get(key) 可以获取key所对应的值，如果key不存在则不报错返回None |
| 字典更新 | Dict1.update(Dict2) 使用dict2去更新dict1,如果dict2独有的key则加入dict1,如果有dict1与dict2共有，则dict1的key被覆盖 |
| 字典是否存在key | key in dict1 |
| 字典返回数据 | Dict.keys()返回字典所有key  Dict.values()返回字典所有value  Dict.items()返回字典所有键值对入元组中 |
| 字典删除键值对 | Dict.pop(key) 返回删除key对应的value  Del dict[key] 删除，但不返回对应value |

1. 对list of dict进行排序

例如：

dict1 = {"weight" : 1, "value":3}

dict2 = {"weight" : 2, "value":3}

dict3 = {"weight" : 3, "value":3}

list1 = [dict3, dict2, dict1]

此时如果直接进行sorted(list1)将报错， 因为Python使用list默认的行为进行比较。但是所含元素为字典，无法直接进行比较，所以我们要指定sorted的时候使用具体的什么域作为比较的属性。

Sorted(list1, key= lambda k: k[‘weight’]) # 指定采用元素k字典中的weight作为排序比较的对象