

对比多层字典是否相同函数

现有2个字典且是多层级的，找出2个字典的不同且输出，实现如下：

```
ok_req={

    "version": "9.0.0",

    "is_test": True,

    "store": "",

    "urs": "",

    "device": {

        "os": "android",

        "imei": "99001062198893",

        "device_id": "CQliMWEyYTEzNTYyYzk5MzJmCTJlNmY3Zjlkx",

        "mac": "02:00:00:00:00:00",

        "galaxy_tag": "CQliMWEyYTEzNTYyYzk5MzJmCTJlNmY3Zjlkx",

        "udid": "a34b1f67dd5797df93fdd8b072f1fb8110fd0db6",

        "network_status": "wifi"

    },

    "adunit": {

        "category": "VIDEO",

        "location": "1",

        "app": "7A16FBB6",

        "blacklist": ""

    },

    "ext_param":{

        "is_start" : 0,

        "vId":"W0BRMTEV"

    }

}

not_ok={

    "version": "9.0.0",

    "is_test": True,

    "urs": "",

    "store": "",

    "device": {

        "os": "android",

        "imei": "99001062298893",
```

```

    "device_id": "CQlIMWEyYTEzNTYyYzk5MzJmCTJlNmY3Zjlkx",
    "mac": "02:00:00:00:00:00",
    "galaxy_tag": "CQlIMWEyYTEzNTYyYzk5MzJmCTJlNmY3Zjlkx",
    "udid": "a34b1f67dd5797da93fdd8b072f1fb8110fd0db6",
    "network_status": "wifi"
},
"adunit": {
    "category": "VIDEO",
    "location": "1",
    "app": "7A16FBB6",
    "blacklist": ""
}, "ext_param": {
    "is_start": 0,
    "vid": "VWOBRMTEV"
}
}

```

#对比多层字典函数

```

def dic_compare(dic_name, compar_dic):

    for key, value in dic_name.items(): #遍历字典

        if key not in compar_dic.keys(): #如果KEY不在对比的字典中，输出字典

            print(key, value)

        if type(value) != dict: #如果value不是多层

            if value not in compar_dic.values(): #不是多层的，如果value和对比字典不一致输出

                print(key, value)

            if type(value) == dict: #如果value是多层的，那么通过递归再次走上面代码进行对比

                dic_compare(value, compar_dic[key])

dic_compare(ok_req, not_ok) #调用函数

```