

01 今日内容大纲

1. is == id 用法
2. 代码块
3. 同一代码块下的缓存机制
4. 不同代码块下的缓存机制（小数据池）
5. 总结
6. 集合（了解）
7. 深浅copy

02 昨日回顾以及作业讲解

1. 字典初始：
 - 查询速度快，{'name': '太白'}, 存储大量的关联型数据。
 - 键：必须是不可变的数据类型（int, str, bool, tuple），唯一的。
 - 值：任意数据类型，对象。
 - 字典3.5x 之前无序的，3.6x 按照初始时的顺序排列，3.7之后有序的。
2. 增删改查：
 - 增：setdefault(), dic['age'] = 18
 - 删：pop 键(可以设置返回值)。clear 清空，del dic['name']
 - 改：dic['name'] = 'wusir'
 - dic['name'] dic.get('name') dic.keys() dic.values() dic.items()
3. 字典的嵌套。

03 具体内容

1. id is ==

```
# id 身份证号
# i = 100
# s = 'alex'
# print(id(i))
# print(id(s))
# == 比较的是两边的值是否相等
# l1 = [1, 2, 3]
# l2 = [1, 2, 3]
# print(l1 == l2)
# s1 = 'alex'
# s2 = 'alex '
# print(s1 == s2)

# is 判断的是内存地址是否相同
# l1 = [1, 2, 3]
```

```
# l2 = [1, 2, 3]
# print(id(l1))
# print(id(l2))
# print(l1 is l2)

s1 = 'alex'
s2 = 'alex'
print(id(s1))
print(id(s2))
print(s1 is s2)

# id 相同, 值一定相同
# 值相同, id不一定相同
```

```
# id 身份证号
i = 100
s = 'alex'
```

内存

内存地址: 1232143243

i → 100

内存地址: 7657657567

s → 'alex'

```
17 # print(id(l1))
18 # print(id(l2))
19 # print(l1 is l2)
20
21 s1 = 'alex'
22 s2 = 'alex'
23 print(id(s1))
24 print(id(s2))
25 print(s1 is s2)
```

内存

内存地址: 6576345543

s1 → alex

s2 → alex

2. 代码块

- 代码块: 我们所有的代码都需要依赖代码块执行。
- 一个文件就是一个代码块。
- 交互式命令下一行就是一个代码块。

3. 两个机制: 同一个代码块下, 有一个机制。不同的代码块下, 遵循另一个机制。

4. 同一个代码块下的缓存机制。

- 前提条件: 同一个代码块内。

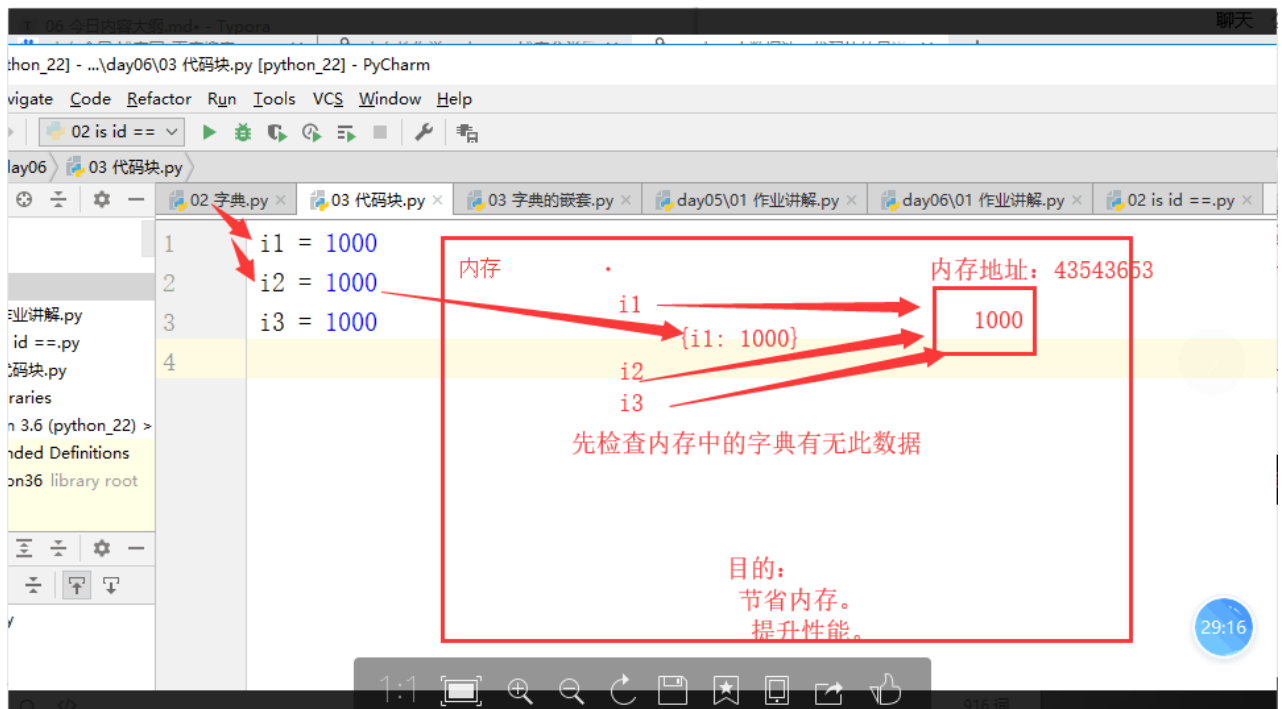
- 机制内容: pass
- 适用的对象: int bool str
- 具体细则: 所有的数字, bool, 几乎所有的字符串。
- 优点: 提升性能, 节省内存。

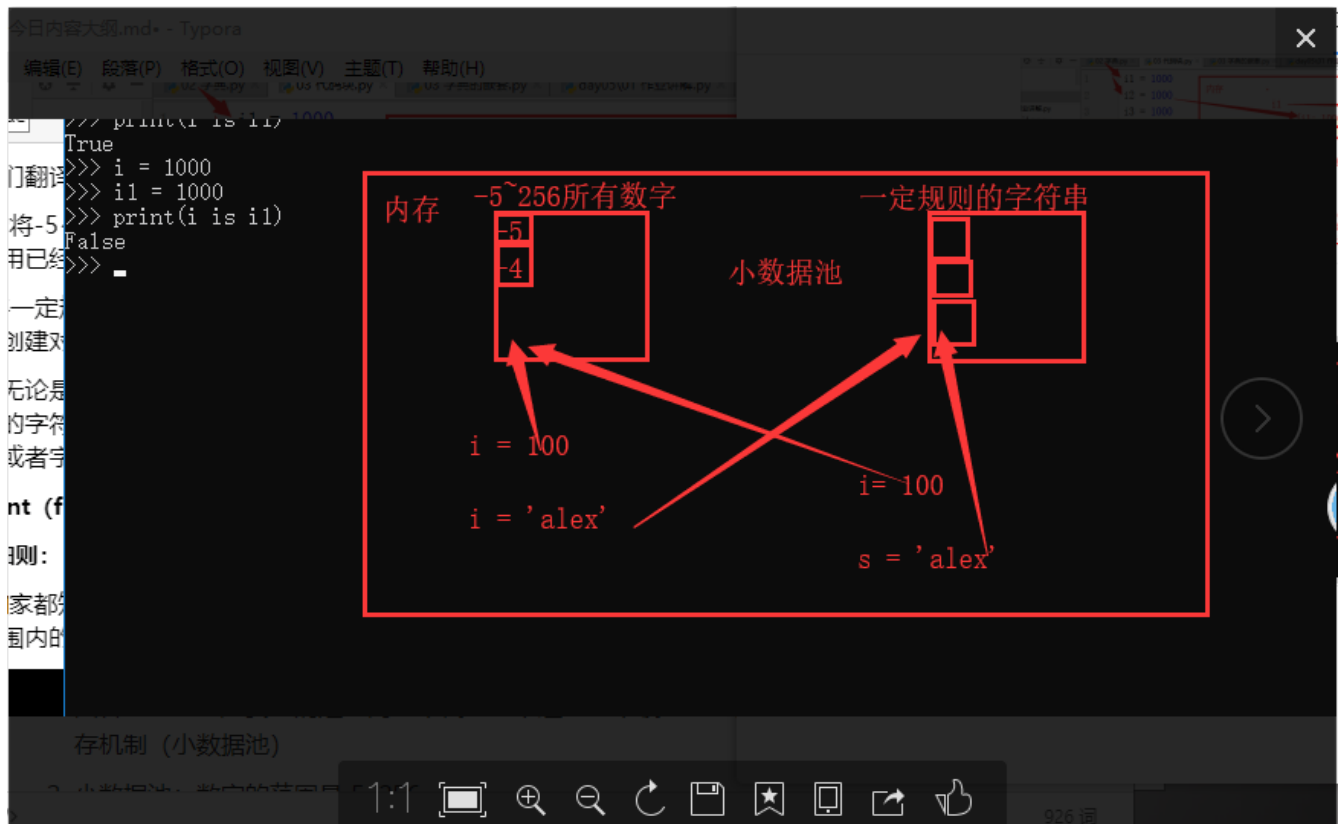
5. 不同代码块下的缓存机制: 小数据池。

- 前提条件: 不同代码块内。
- 机制内容: pass
- 适用的对象: int bool str
- 具体细则: **-5~256数字**, bool, 满足规则的字符串。
- 优点: 提升性能, 节省内存。

```
# i1 = 1000
# i2 = 1000
# i3 = 1000
# l1 = [1,2,3]
# l2 = [1,2,3]
# print(id(l1))
# print(id(l2))
# print(id(i1))
# print(id(i2))
# print(id(i3))

i = 800
i1 = 800
s1 = 'hfdjka6757fds1slgaj@!#fkdljsafjdskl;fjds中国'
s2 = 'hfdjka6757fds1slgaj@!#fkdljsafjdskl;fjds中国'
print(i is i1)
print(s1 is s2)
```





1. 总结: +

1. 面试题考。
 2. 回答的时候一定要分清楚：同一个代码块下适用一个缓存机制。不同的代码块下适用另一个缓存机制（小数据池）
 3. 小数据池：数字的范围是-5~256.
 4. 缓存机制的优点：提升性能，节省内存。
2. python基础数据类型之：集合 set。容器型的数据类型，它要求它里面的元素是不可变的数据，但是它本身是可变的数据类型。集合是无序的。{}。
 - o 集合的作用：
 - 列表的去重。
 - 关系测试：交集，并集，差集，.....
 - pass

```
# 集合的创建：
# set1 = set({1, 3, 'Barry', False})
# set1 = {1, 3, '太白金星', 4, 'alex', False, '武大'}
# print(set1)

# 空集合：
# print({}, type({})) # 空字典
# set1 = set()
# print(set1)

# 集合的有效性测试
# set1 = {[1,2,3], 3, {'name': 'alex'}}
# print(set1)
```

```
# set1 = {'太白金星', '景女神', '武大', '三粗', 'alexb', '吴老师'}
# 增:
# add
# set1.add('xx')
# print(set1)

# update迭代着增加
# set1.update('fdsafgsd')
# print(set1)

# 删
# remove
# remove 按照元素删除
# set1.remove('alexb')
#
# print(set1)
# pop 随即删除
# set1.pop()
# print(set1)

# 变相改值
# set1.remove('太白金星')
# set1.add('男神')
# print(set1)

#关系测试: ***
# 交集
# set1 = {1, 2, 3, 4, 5}
# set2 = {4, 5, 6, 7, 8}
# print(set1 & set2)

# 并集:
# print(set1 | set2)

# 差集 -
# print(set1 - set2)

# 反交集
# print(set1 ^ set2)

# 子集
# set1 = {1,2,3}
# set2 = {1,2,3,4,5,6}
# # print(set1 < set2)
#
# # 超集
# print(set2 > set1)

# 列表的去重 ***
# l1 = [1,'太白', 1, 2, 2, '太白',2, 6, 6, 6, 3, '太白', 4, 5, ]
# set1 = set(l1)
# l1 = list(set1)
```

```
# print(l1)

# 用处：数据之间的关系，列表去重。
```

3. 深浅copy (面试会考)

```
# 赋值运算
# l1 = [1, 2, 3, [22, 33]]
# l2 = l1
# l1.append(666)
# print(l1)
# print(l2)
```

```
# 浅copy
# l1 = [1, 2, 3, [22, 33]]
# l2 = l1.copy()
# l1.append(666)
# print(l1, id(l1))
# print(l2, id(l2))
```

```
# l1 = [1, 2, 3, [22, 33]]
# l2 = l1.copy()
# l1[-1].append(666)
# print(id(l1[-1]))
# print(id(l2[-1]))
# print(id(l1[0]))
# print(id(l2[0]))
# print(l1)
# print(l2)
```

```
# l1 = [1, 2, 3, [22, 33]]
# l2 = l1.copy()
# l1[0] = 90
# print(l1)
# print(l2)
```

```
# 深copy
# import copy
# l1 = [1, 2, 3, [22, 33]]
# l2 = copy.deepcopy(l1)
# # print(id(l1))
# # print(id(l2))
# l1[-1].append(666)
# print(l1)
# print(l2)
```

```
# 相关面试题;
```

```
# l1 = [1, 2, 3, [22, 33]]
# l2 = l1[:]
# l1[-1].append(666)
# print(l1)
# print(l2)
# 浅copy: list dict: 嵌套的可变的数据类型是同一个。
# 深copy: list dict: 嵌套的可变的数据类型不是同一个。
```

06 今日内容大纲.md - Typora

文件(F) 编辑(E) 段落(P) 格式(O) 视图(V) 主题(T) 帮助(H) 12:25:27

05 深浅copy.py

02 字典.py × 03 代码块.py × 04 集合 (了解).py × 05 深浅copy.py × 03 字典的嵌套.py × day05\01 作业讲解.py × day06\01 作业讲解.py ×

```
# 赋值运算
l1 = [1, 2, 3, [22, 33]]
l2 = l1
l1.append(666)
print(l1)
print(l2)
```

机制 (小数据池)

1:1 [放大] [缩小] [刷新] [打印] [分享] 1174 页

固定到“快速访问” 06 今日内容大纲.md - Typora

文件(F) 编辑(E) 段落(P) 格式(O) 视图(V) 主题(T) 帮助(H)

```
# print(l2)
# 浅copy
l1 = [1, 2, 3, [22, 33]]
l2 = l1.copy()
l1.append(666)
print(l1, id(l1))
print(l2, id(l2))
```

聊天 公告 文件

• 集合: 列表去重, 关系测试。
• 深浅copy: 理解浅copy, 深浅copy课上练习题整明白。

2个项目

1:1 [放大] [缩小] [刷新] [打印] [分享] 1184 页

04 今日总结

- id is == 三个方法要会用，知道是做什么的。
- 回答的时候一定要分清楚：同一个代码块下适用一个缓存机制。不同的代码块下适用另一个缓存机制（小数据池）
- 小数据池：数字的范围是-5~256.
- 缓存机制的优点：提升性能，节省内存。
- 集合：列表去重，关系测试。
- 深浅copy：理解浅copy，深浅copy,课上练习题整明白。

05 预习内容

- 数据类型的补充以及编码的进阶：<https://www.cnblogs.com/jin-xin/articles/10577131.html>