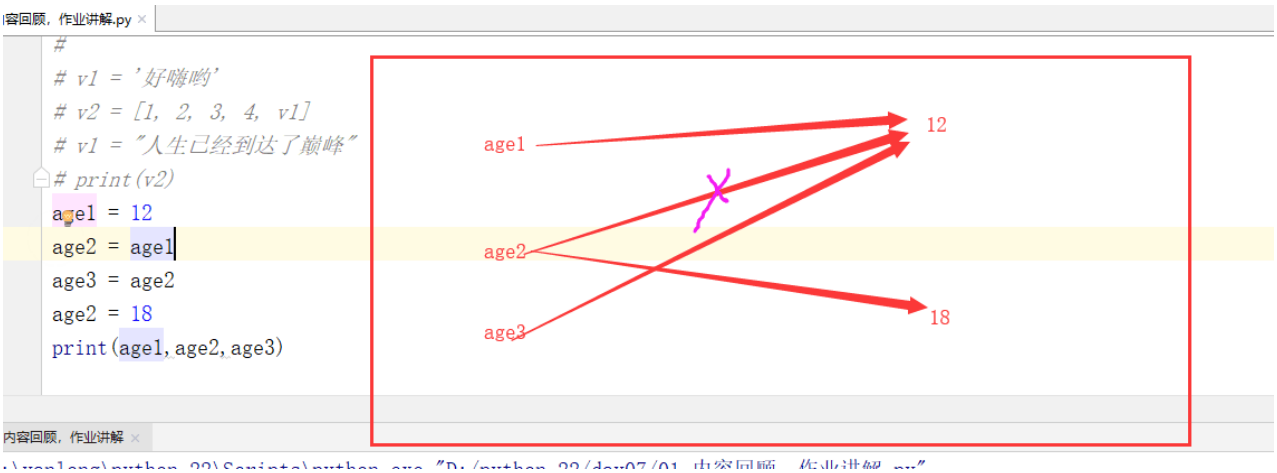
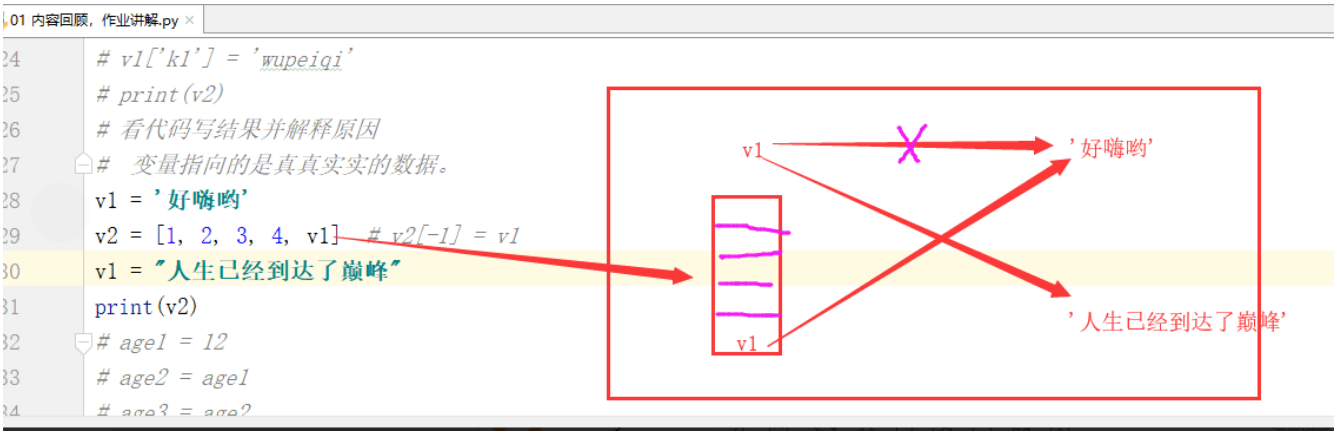
01 今日内容大纲

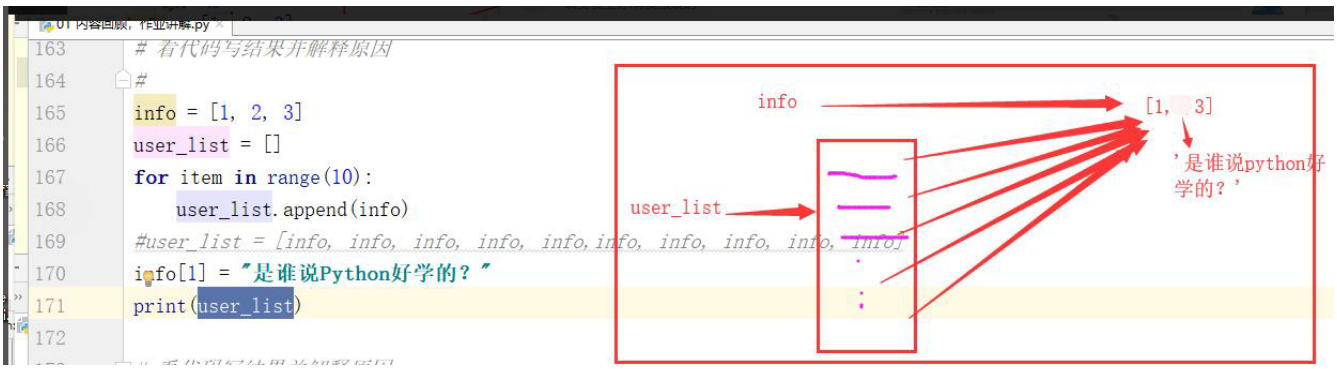
1. 基础数据类型的补充
2. 数据类型之间的转换
3. 编码的进阶

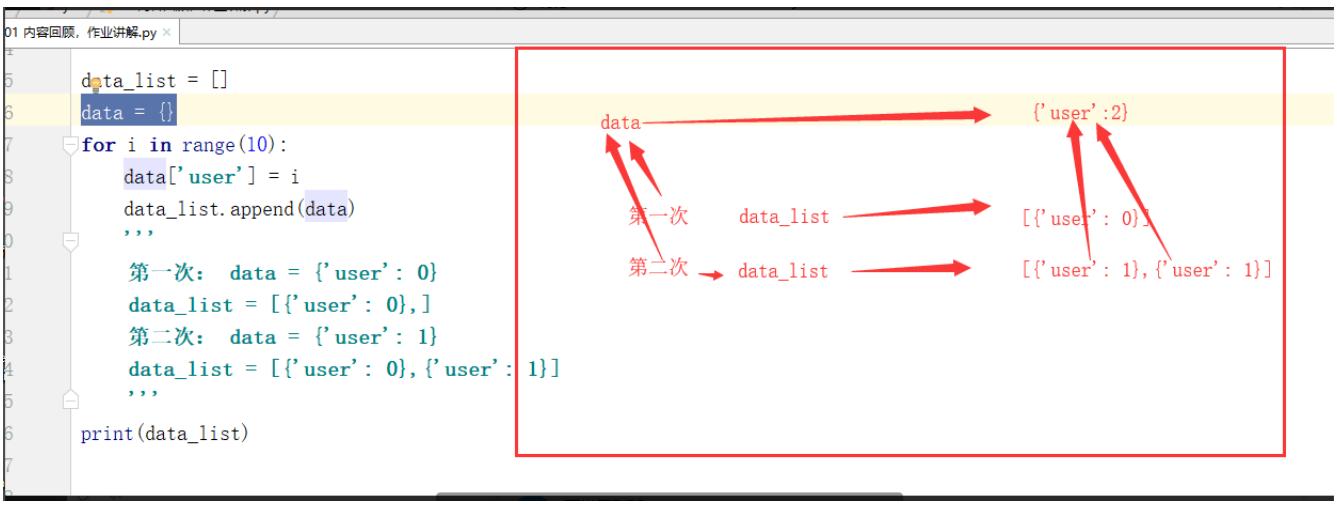
02 昨日内容回顾以及作业讲解

1. id == is：
    - == ： 数值是否相同 is：内存地址，id 获取对象的内存地址
2. 代码块：一个文件，交互式命令一行就是一个代码块。
3. 同一代码块下缓存机制（字符串驻留机制）：
    - 所有数字，bool 几乎所有的字符串
    - 优点：提升性能，节省内存空间。
4. 不同代码块的缓存机制（小数据池）：在内存中开辟两个空间，一个空间存储-5~256的int，一个空间存储一定规则的字符串，如果你的代码中遇到了满足条件的数据，直接引用提前创建的。
    - -5~256 int，bool，满足一定规则的字符串。
    - 优点：提升性能，节省内存空间。
5. 集合：列表去重，关系测试 交并差。
6. 深浅copy：
    - 浅copy：在内存中开辟一个新的空间存放copy的对象（列表，字典），但是里面的所有元素与被copy对象的里面的元素共用一个。









03 具体内容

- 数据类型的补充
  - str

```
# str : 补充的方法练习一遍就行。
# s1 = 'taiBAi'
# capitalize 首字母大写，其余变小写
# print(s1.capitalize())
# swapcase  大小写翻转
# print(s1.swapcase())
# title
# msg= 'taibai say3hi'
# print(msg.title()) #每个单词的首字母大写
```

```
s1 = 'barry'
# 居中
# print(s1.center(20))
# print(s1.center(20,'*'))

# find :通过元素找索引，找到第一个就返回，找不到 返回-1
# index:通过元素找索引，找到第一个就返回，找不到 报错
# print(s1.find('a'))
# print(s1.find('r'))
# print(s1.find('o'))
# print(s1.index('o'))
```

- 元组

```
# tuple
# 元组中如果只有一个元素，并且没有逗号，那么它不是元组，它与改元素的数据类型一致。 ***
# tu1 = (2,3,4)
# tu1 = (2)
# tu1 = ('太白')
# tu1 = ([1,2,3])
# tu1 = (1,)
# print(tu1,type(tu1))
# tu = (1,2,3,3,3,2,2,3,)
# # count 计数
# print(tu.count(3))
# tu = ('太白', '日天', '太白')
# # index
# print(tu.index('太白'))
```

- 列表

```
# l1 = ['太白', '123', '女神', '大壮']
# count pass
# index
# print(l1.index('大壮'))
# sort  **
# l1 = [5, 4, 3, 7, 8, 6, 1, 9]
# # l1.sort()  # 默认从小到大排序
# # l1.sort(reverse=True)  # 从大到小排序  **
# l1.reverse()  # 反转  **
# print(l1)
# 列表可以相加
# l1 = [1, 2, 3]
# l2 = [1, 2, 3, '太白', '123', '女神']
# print(l1 + l2)

# 列表与数字相乘
# l1 = [1, 'daf', 3]
# l2 = l1*3
# print(l2)
```

```
l1 = [11, 22, 33, 44, 55]
# 索引为奇数对应的元素删除（不能一个一个删除，此l1只是举个例子，里面的元素不定）。
#  *** 重要
# 正常思路：
# 先将所有的索引整出来。
# # 加以判断，index % 2 == 1:  pop (index)
# for index in range(len(l1)):
#     if index % 2 == 1:
#         l1.pop(index)
# print(l1)
# 列表的特性：
# l1 = [11, 22, 33, 44, 55]
# 最简单的：
# del l1[1::2]
# print(l1)
# l1 = [11, 22, 33, 44, 55]
# # 倒序法删除元素
# for index in range(len(l1)-1,-1,-1):
#     if index % 2 == 1:
#         l1.pop(index)
# print(l1)

# 思维置换
# l1 = [11, 22, 33, 44, 55]
# new_l1 = []
# for index in range(len(l1)):
#     if index % 2 ==0:
#         new_l1.append(l1[index])
# # print(new_l1)
# l1 = new_l1
# print(l1)

# 循环一个列表的时，最好不要改变列表的大小，这样会影响你的最终的结果。
```

- 字典

```
# 字典的补充
# update ***
# dic = {'name': '太白', 'age': 18}
# # dic.update(hobby='运动', hight='175')
# # dic.update(name='太白金星')
# dic.update([(1, 'a'),(2, 'b'),(3, 'c'),(4, 'd')])  # 面试会考
# print(dic)
# dic1 = {"name":"jin","age":18,"sex":"male"}
# dic2 = {"name":"alex","weight":75}
# dic1.update(dic2)  # 更新，有则覆盖，无责添加
# print(dic1)  # {'name': 'alex', 'age': 18, 'sex': 'male', 'weight': 75}
# print(dic2)
```

```
# fromkeys
# dic = dict.fromkeys('abc', 100)
# dic = dict.fromkeys([1, 2, 3], 'alex')
# 坑：值共有一个,面试题
# dic = dict.fromkeys([1,2,3],[])
# dic[1].append(666)
# print(dic)
dic = {'k1': '太白', 'k2': 'barry', 'k3': '白白', 'age': 18}
# 将字典中键含有'k'元素的键值对删除。
# for key in dic:
#     if 'k' in key:
#         dic.pop(key)
# print(dic)

# 循环一个字典时，如果改变这个字典的大小，就会报错。
# l1 = []
# for key in dic:
#     if 'k' in key:
#         l1.append(key)
# print(l1)
# for i in l1:
#     dic.pop(i)
# print(dic)

# for key in list(dic.keys()):  # ['k1', 'k2', 'k3','age']
#     if 'k' in key:
#         dic.pop(key)
# print(dic)
```

- 数据类型的转换

```
# 0,''(),[],{},set(),None  转换成bool值为False
```

- 数据类型的分类（了解）

- 编码的进阶
  - **ASCII码：包含英文字母，数字，特殊字符与01010101对应关系。**

    a 01000001 一个字符一个字节表示。
  - **GBK：只包含本国文字（以及英文字母，数字，特殊字符）与0101010对应关系。**

    a 01000001 ascii码中的字符：一个字符一个字节表示。

    中 01001001 01000010 中文：一个字符两个字节表示。
  - **Unicode：包含全世界所有的文字与二进制0101001的对应关系。**

    a 01000001 01000010 01000011 00000001

    b 01000001 01000010 01100011 00000001

中 01001001 01000010 01100011 00000001

- **UTF-8:包含全世界所有的文字与二进制0101001的对应关系（最少用8位一个字节表示一个字符）。**

a 01000001 ascii码中的字符：一个字符一个字节表示。

To 01000001 01000010 (欧洲文字：葡萄牙，西班牙等)一个字符两个字节表示。

中 01001001 01000010 01100011 亚洲文字；一个字符三个字节表示。

1. 不同的密码本之间能否互相识别？不能。
2. 数据在内存中全部是以Unicode编码的，但是当你的数据用于网络传输或者存储到硬盘中，必须是以非Unicode编码（utf-8,gbk等等）。

英文：

str： 'hello '

内存中的编码方式： Unicode

表现形式： 'hello'

bytes ：

内存中的编码方式： 非Unicode

表现形式：b'hello'

中文：

str：

内存中的编码方式： Unicode

表现形式：'中国'

bytes ：

内存中的编码方式： 非Unicode # Utf-8

表现形式：b'\xe4\xb8\xad\xe5\x9b\xbd'

```
# str ---> bytes\
# s1 = '中国'
# b1 = s1.encode('utf-8')  # 编码
# print(b1,type(b1))  # b'\xe4\xb8\xad\xe5\x9b\xbd'
# # b1 = s1.encode('gbk')  # 编码  # b'\xd6\xd0\xb9\xfa' <class 'bytes'>
# # bytes---->str
# b1 = b'\xe4\xb8\xad\xe5\x9b\xbd'
# s2 = b1.decode('utf-8')  # 解码
# print(s2)
```

```
# gbk ---> utf-8
b1 = b'\xd6\xd0\xb9\xfa'
s = b1.decode('gbk')
# print(s)
b2 = s.encode('utf-8')
print(b2)  # b'\xe4\xb8\xad\xe5\x9b\xbd'
```

04 今日总结

- 数据类型的补充： list（sort,revrse,列表的相加，乘，循环问题），dict （update 循环问题） ***
- 编码的进阶：
    - bytes为什么存在?
    - str --->bytes(Unicode ---> 非Unicode)
    - gbk <-----> utf-8

05 预习内容

预习内容：

文件操作，博客地址：http://www.cnblogs.com/jin-xin/articles/8183203.html