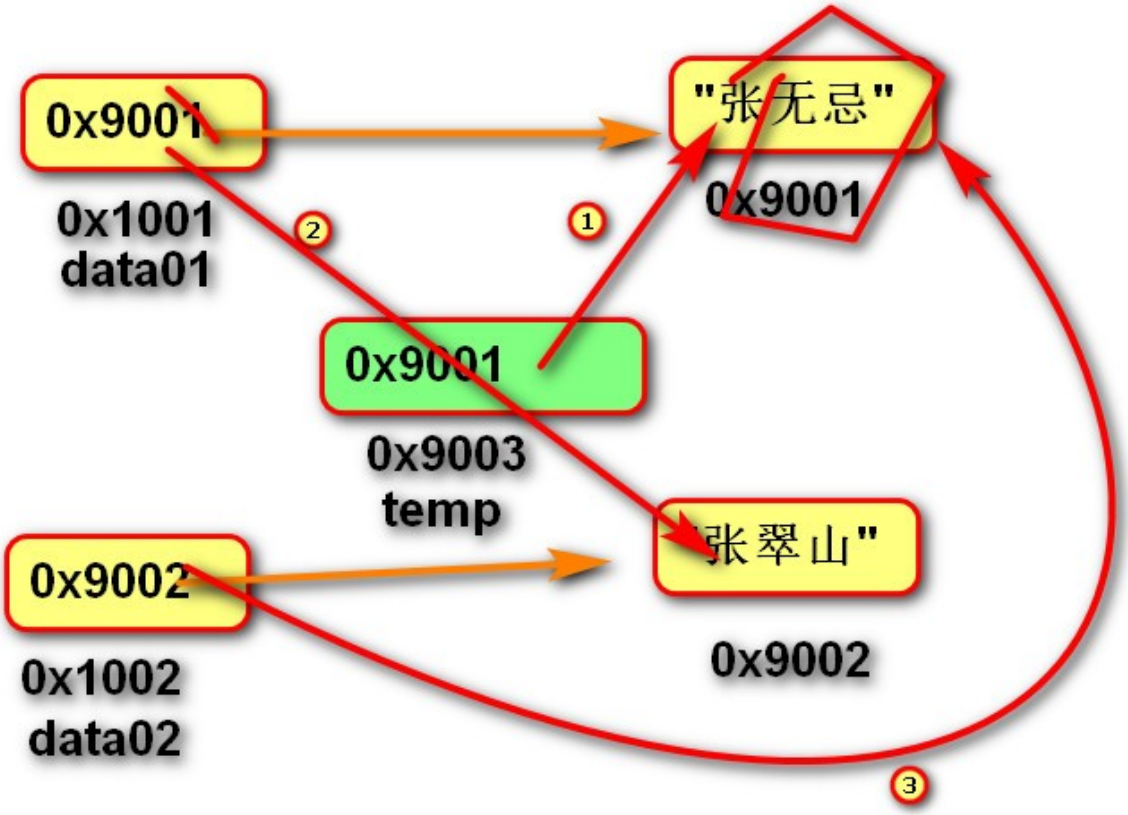
变量

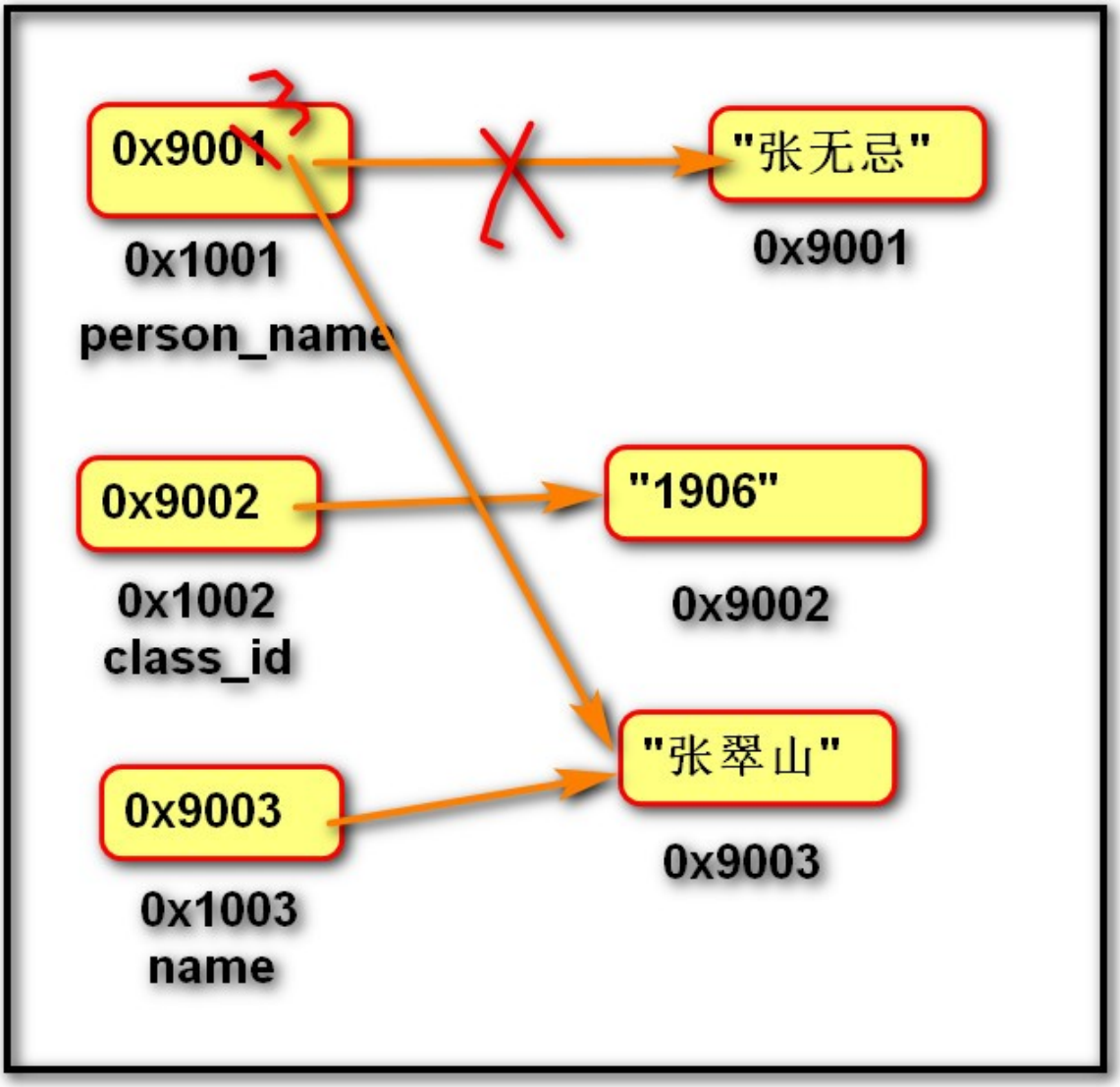
1. 定义：关联一个对象的标识符。
2. 命名：必须是字母或下划线开头，后跟字母、数字、下划线。
   不能使用关键字(蓝色)，否则发生语法错误：SyntaxError: invalid syntax。
3. 建议命名：字母小写，多个单词以下划线隔开。
   class_name
4. 赋值：创建一个变量或改变一个变量关联的数据。
5. 语法：变量名 = 数据

变量名 1 = 变量名 2 = 数据

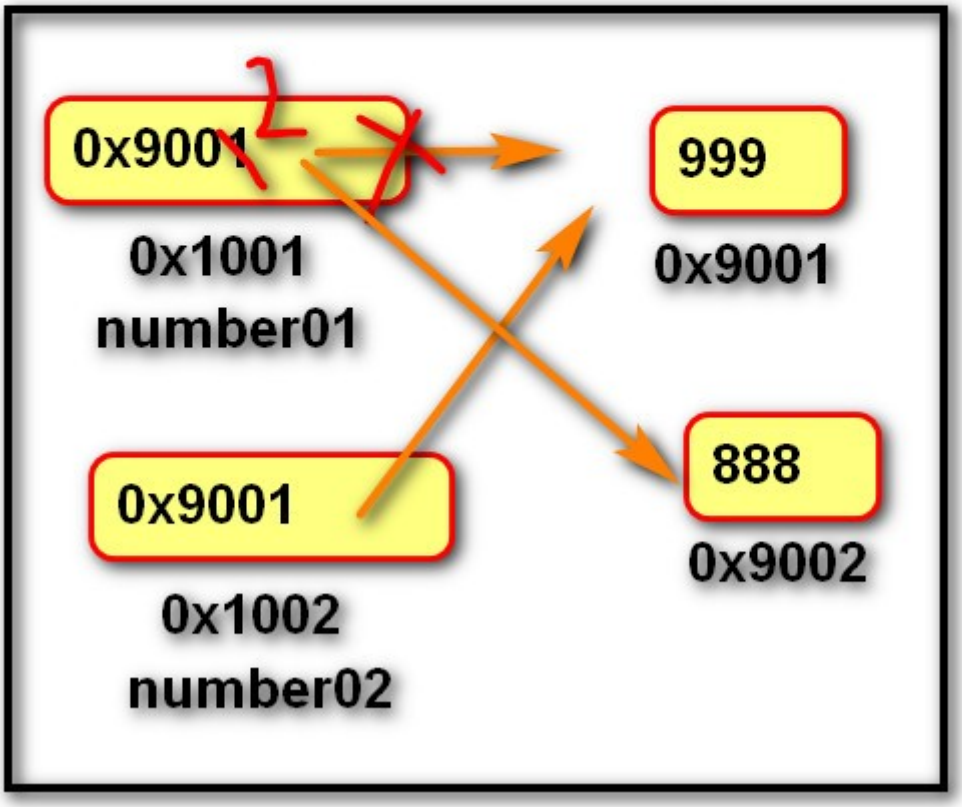
变量名 1, 变量名 2, = 数据 1, 数据 2



person_name = "张无忌"
class_id = "1906"
person_name = "张翠山"
name = person_name



```
number01 = 999
number02 = number01
number01 = 888
print(number02)# ?
```

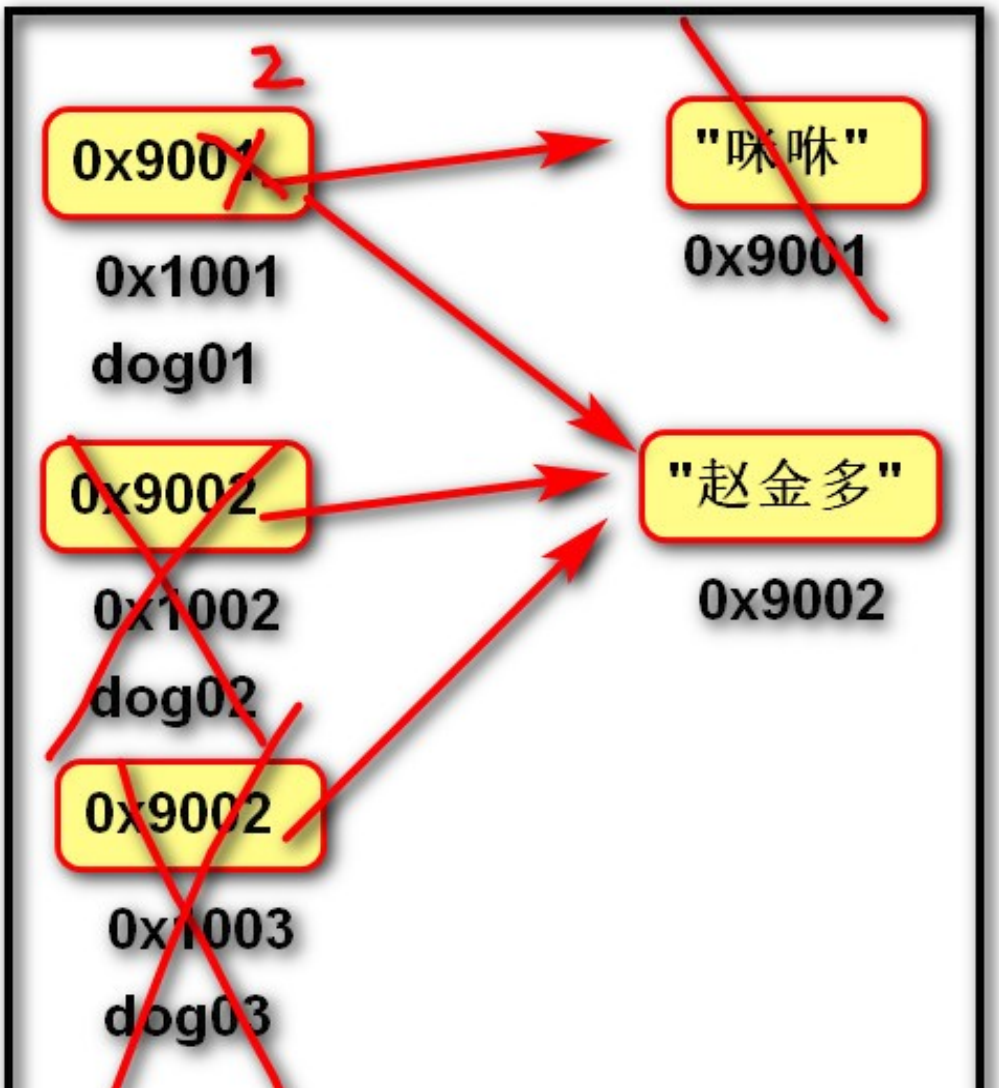


```
"""
    变量
        字面意思：可以变化  的  数据
        赋值号=：将右边的对象复制一份给左边
        语法:
            名称  = 对象
        练习:exercise01.py
             exercise02.py
    10:55
"""
person_name = "张无忌"
class_id = "1906"
person_name = "张翠山"
name = person_name
# 将括号中的内容输出到终端中
print(person_name)
# 格式：xx班级的xx.
print(class_id + "班级的" + person_name)
```

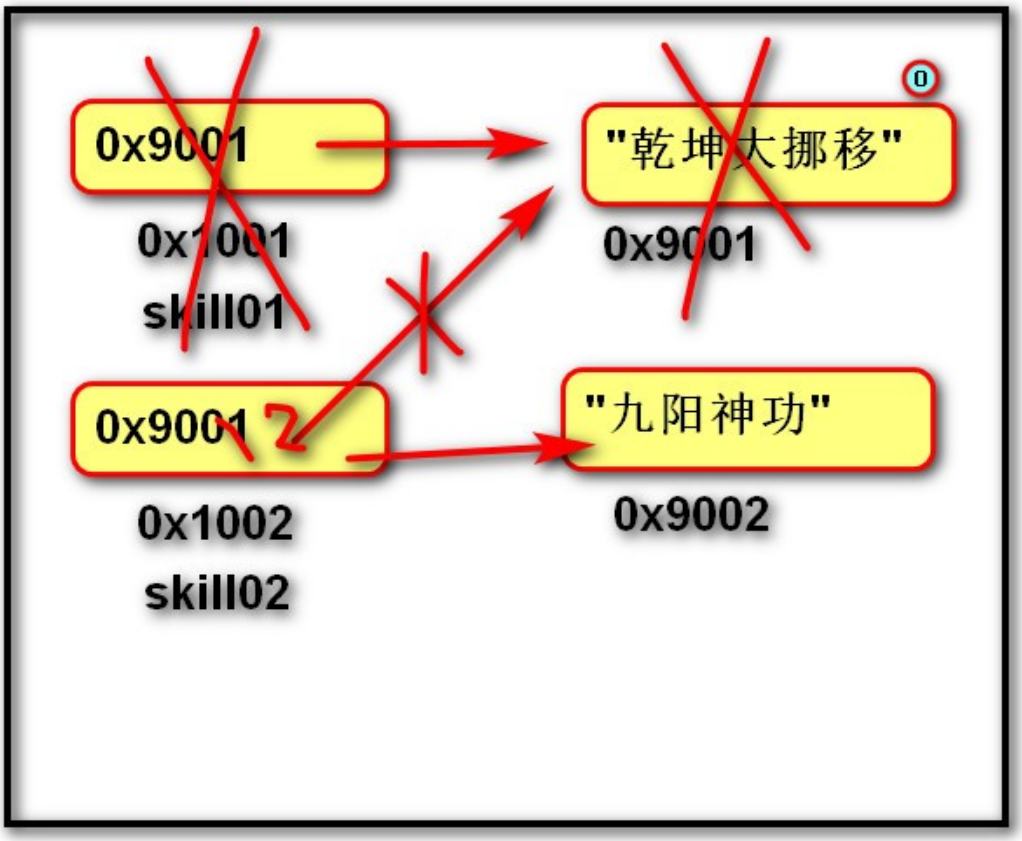
del 语句

1. 语法:
   del 变量名 1, 变量名 2
2. 作用:
   用于删除变量,同时解除与对象的关联.如果可能则释放对象。
3. 自动化内存管理的引用计数:
   每个对象记录被变量绑定(引用)的数量,当为 0 时被销毁。

dog01 = "咪咻"
dog02 = "赵金多"
dog01 = dog02
dog03 = dog02
del dog02,dog03



```
skill01 = "乾坤大挪移"
skill02 = skill01
# 删除变量
del skill01
skill02 = "九阳神功"
```




```
"""
    变量语法与del
    练习:exercise03.py
        exercise04.py
"""
# 语法1:  名称  =  对象
name = "无忌"
# 语法2:  名称1,名称2 = 对象1,对象2
person_name, class_id = "无忌", "1906"
# 语法3:名称1 = 名称2 = 对象
number01 = number02 = 999
# 启发:两个变量交换
# 变量1,变量2 = 变量2,变量1
#--------------------
```

```
skill01 = "乾坤大挪移"
skill02 = skill01
# 删除变量
del skill01
skill02 = "九阳神功"
```

核心数据类型

1. 在 python 中变量没有类型，但关联的对象有类型。
2. 通过 type 函数可查看。

空值对象 None

1. 表示不存在的特殊对象。
2. 作用：占位和解除与对象的关联。

整形 int

1. 表示整数，包含正数、负数、0。
   如： -5, 100, 0
2. 字面值：
   十进制：5
   二进制：0b 开头，后跟 1 或者 1
   八进制：0o 开头，后跟 0~7
   十六进制：0x 开头，后跟 0~9,A~F,a~f
3. 小整数对象池：CPython 中整数 -5 至 256,永远存在小整数对象池中,不会被释放并可重复使用。

浮点型 float

1. 表示小数，包含正数、负数，0.0)。
2. 字面值：
   小数：1.0　2.5
   科学计数法：e/E (正负号) 指数
   1.23e-2 (等同于 0.0123)
   1.23456e5(等同于 123456.0)

字符串 str

是用来记录文本信息(文字信息)。
字面值：双引号

复数 complex

由实部和虚部组成的数字。
虚部是以 j 或 J 结尾。
字面值： 1j 1+1j 1-1j

布尔 bool

用来表示真和假的类型
True 表示真(条件满足或成立)，本质是 1
False 表示假(条件不满足或不成立)，本质是 0

数据类型转换

1. 转换为整形: int(数据)
2. 转换为浮点型:float(数据)
3. 转换为字符串:str(数据)
4. 转换为布尔:bool(数据)
   结果为 False：bool(0) bool(0.0) bool(None)
5. 混合类型自动升级：
   1 + 2.14 返回的结果是 3.14
   1 + 3.0 返回结果是：4.0
   """

```
    核心数据类型
        变量没有类型，关联的对象才有类型.
"""
# 1. None 空 NoneType
# 占位:只希望有个变量，指向的对象还不确定。
name = None
skill01 = "乾坤大挪移"
# 解除"乾坤大挪移" 与  变量 skill01 的关系
skill01 = None
# 2. 整形(整数)int
# 十进制：每逢十进一位  0 1 2 3 ...10
number01 = 250
```

```python
# 二进制：每逢二进一位  0 1  10  11  100  101
number02 = 0b100
# 八进制：每逢八进一位  0 1  2 ..  7 10  11
number03 = 0o10
# 十六进制：每逢十六进一位0 1 3 ...9   a(10) f(15)
number04 = 0xf
print(number04)
# 3. 浮点型(小数)float
number05 = 10.5
# 科学计数法
number06 = 1.23456e5
print(number06)  # 123456.0
number07 = 0.00000000000000000000000005
number08 = 5e-26
print(0.00001)
# 4. 字符串str
message01 = "我爱编程"
number09 = "100.5"
print(type(name))
# 5. *复数 complex
num01 = 10 + 1.5j
print(type(num01))
# 6. 类型转换
# input 函数的结果是字符串类型
str_age = input("请输入年龄：")  # "18"  + 1
# str --> int
int_age = int(str_age)  # 19
# int --> str
print("明年是：" + str(int_age))  # "明年是："  19
```

运算符

算术运算符

+	加法
-	减法
*	乘法
/	除法：结果为浮点数
//	地板除：除的结果去掉小数部分
%	求余

** 幂运算

优先级从高到低：()

**

* / % //

+ -

增强运算符

y += x	等同于 y = y + x
y -= x	等同于 y = y - x
y *= x	等同于 y = y * x
y /= x	等同于 y = y / x
y //= x	等同于 y = y // x
y %= x	等同于 y = y % x
y **= x	等同于 y = y ** x

```
"""
    运算符
        算数运算符
            + - * /   // %  **
        增强运算符
            += -=  *=  /=   //=  %=  **=
    练习:exercise05.py
        exercise06.py
        exercise07.py
        exercise08.py
        exercise09.py
"""
number01 = 5
number02 = 2
# result = number01 / number02 # 2.5
# result = number01 // number02 # 2
result = number01 % number02 # 2
print(result)  # ?
number03 = 4 ** 2
print(number03)# 4 * 4
#  增强运算符
number04 = 100
# print(number04 + 10)# 110
# print(number04)# ?100
temp = number04 + 10 # 110
print(number04)# 100
# # 变量 = 变量  + 数据
number04 = number04 + 10
print(number04)# 110
# 变量 += 数据(改变自身变量)
```

```
number04 += 10
print(number04)# 110
```

比较运算符

< 小于
<= 小于等于
> 大于
>= 大于等于
== 等于
!= 不等于
返回布尔类型的值
比较运算的数学表示方式:0 <= x <= 100

逻辑运算符

与 and

表示并且的关系，一假俱假。
示例:

```
True and True  # True
True and False # False
False and True # False
    False and False    # False
```

或 or

表示或者的关系，一真俱真
示例:

```
    True or True       # True
    True or False          # True
    False or True          # True
    False or False         # False
```

非 not

表示取反
例如:

```
not True   # 返回 False
```

not False # 返回 True

短路运算

一但结果确定，后面的语句将不再执行。

身份运算符

语法:

x is y

x is not y

作用：

is 用于判断两个对象是否是同一个对象,是时返回 True,否则返回 False。

is not 的作用与 is 相反

优先级

高到低：

算数运算符

比较运算符

快捷运算符

身份运算符

逻辑运算符