

第二章 程序基础知识

a = 1

一、程序的构成 (p9)

1.1 python程序的构成

表达式 -> 语句 -> 模块 -> 程序

表达式

语句，按照语句顺序依次执行。语句是python中的构造单元，用于创建对象、变量赋值、调用函数、控制语句等

模块，python源文件，后缀.py

1.2 使用行连接符\

\可以在字符串（' '）内出现

二、标识符 (p12)

2.1 标识符的基本用法

标识符：用于变量、函数、类、模块等的名称

标识符的**命名规则**：

- ① 区分大小写
- ② 第一个字母必须是数字或下划线，后面的可以有数字
- ③ 不能使用关键字（keyword）
- ④ 避免用双下划线开头写标识符

标识符的**约定命名规则**：

类型	规则
模块、包	全小写字母，尽量简单，多个单词之间用下划线
函数名	全小写字母，单词间用下划线
类名	首字母大写，驼峰原则
常量名	全大写字母，多个单词使用下划线隔开

2.2 查看关键字

```
import keyword # 添加keyword模块
print(keyword.kwlist)
# 查看keyword模块中kwlist (keyword list) 内容
```

输出结果：

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for',
```

```
'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or',  
'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

【操作3】使用 Python 帮助系统查看关键字

```
>>> help()  
help> keywords  
False          def          if          raise  
None           del          import       return  
True           elif         in          try  
and            else         is          while  
as             except        lambda       with
```

三、对象的基本组成 (p10)

3.1 对象

在python中，一切都是对象，每个对象由标识（地址）（identity）、类型（type）、值（value）组成（在哪里，什么样子，谁）

1. 标识：标识用于唯一标识对象，通常对应于对象在计算机内存中的地址。使用内置函数`id(obj)`可返回对象在内存中的地址

`id(obj)`

2. 类型：使用内置函数`type()`返回对象的类型

`type(obj)`

四、引用 (p11)

4.1 引用

变量：对象的引用 —— 变量存储对象的地址，变量通过地址引用对象

下面的代码显示了一个对象地址和变量地址的关系：

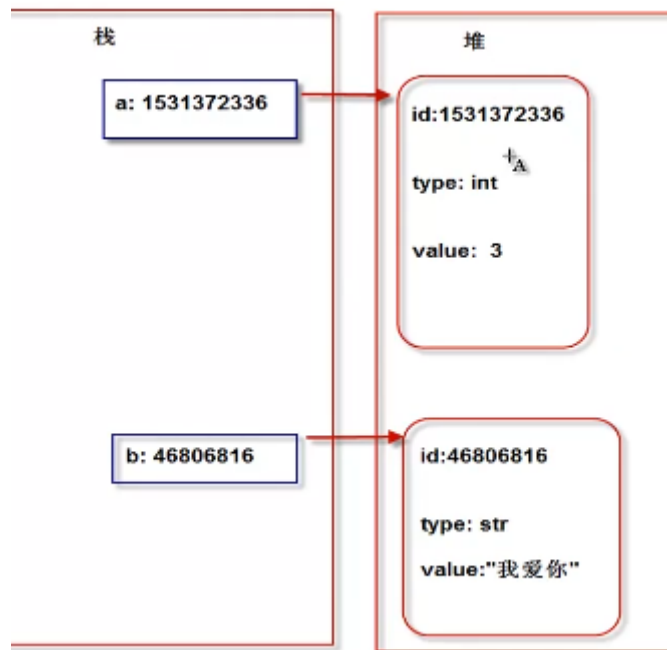
```
a = 3 # 3称为一个对象，a获得了3这个对象的地址-> a == 3  
print(id(3)) # 3  
print(type(3))  
print(3)  
print(id(a)) # a  
print(type(a))  
print(a)
```

```
b = 'python'  
print(id('python')) # 'python'  
print(type('python'))  
print('python')  
print(id(b)) # b  
print(type(b))  
print(b)
```

4.2 堆栈内存

变量位于：栈内存（可变化的）

对象位于：堆内存（成堆存储的）



4.3 python数据类型的特点

动态类型语言：变量不需要显式声明类型，根据变量引用的对象，python解释器自动确定数据类型

强类型语言：每个对象都有数据类型，只支持该类型的操作

五、特殊赋值方式与不存在常量

5.1 链式赋值

链式赋值用于同一个对象赋值给多个变量

`a = b = c = 1`

5.2 系列解包赋值

系列数据赋值给对应相同的变量（个数必须相同）

`a, b, c = 1, 2, 3`

实例：python中互换两个变量的数据（真的方便很多）

`a, b = b, a`

5.3 常量

python不支持常变量，只能通过逻辑上控制不对常量做出修改（全大写标记）