# 引用

## 课程目标

* 能够说出什么是引用
* 能够说出Python中哪些数据类型是可变类型
* 能够说出Python中哪些数据类型是不可变类型

## 课程内容

* 引用

### 引用

#### 1.1引用概念

引用是一种变量指向数据存储空间的现象

内存地址是数据在物理内存中的存储位置

引用地址是对象在内存中的描述性地址，该地址与内存地址有区别

#### 1.2引用特征

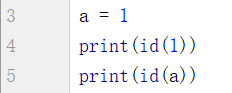
相同的数据在内存空间中仅占用一个存储空间，不同的变量使用相同的数据则指向相同的存储空间。

#### 1.3获取内存存储地址（编号）

语法格式:

id（数据）

id（变量名）



#### 1.4地址存储特殊性

使用固定内存地址存储数据如下：

* -5到256的整数
* True和False
* 由字母、数字、下滑线组成的字符串

使用临时内存地址存储数据如下：

* + - * 小于-5后大于256的整数
      * 所有小数
      * 包含字母、数字、下滑线之外的字符组成的字符串

### 列表引用

#### 2.1列表存储空间

* 无变量引用

列表保存在临时引用地址中，该引用地址可以反复使用

* 有变量引用

列表保存在独立引用地址中，该引用地址专用，每个列表对象具有独立的引用地址

#### 2.2列表存储空间变更

1.列表变量指向引用地址，该引用地址中保存有存储列表信息的内存地址。

2.当列表中存储数据的空间不足时，会申请新的内存地址，并更新引用地址中对应的内存地址。

3.重新申请的内存地址将复制原始内存地址中的数据。

#### 2.3列表中的数据存储

1.列表中存储的数据都是数据对应的内存地址，没有真实数据

2.列表中的数据是通过引用地址的形式查找真实数据

### 元组引用

#### 3.1元组存储空间

* 空元组（无数据）

空元组指向统一的独立引用地址，所有空元组共用同一个引用地址

* 非空元组（有数据）

非空元组保存在独立引用地址中，每个元组对象具有独立引用地址

注：由于元组不支持修改，不存在重新申请空间的问题，元组创建的同时存储的数据已经固定，内存结构也固定。

#### 3.2元组中的数据存储（同列表）

1.元组中存储的数据都是数据对应的内存地址，没有真实数据

2.元组中的数据是通过引用地址的形式查找真实数据

注：元组不支持修改，但是元组中的数据如果是引用类型的，对应的数据可以在自身的内存地址中进行数据变更，对元组不影响。

### 集合引用（同列表）

#### 4.1集合存储空间

* 无变量引用

集合保存在临时引用地址中，该引用地址可以反复使用

* 有变量引用

集合保存在独立引用地址中，该引用地址专用，每个集合对象具有独立的引用地址

#### 4.2集合存储空间变更

1.集合变量指向引用地址，该引用地址中保存有存储集合信息的内存地址。

2.当集合中存储数据的空间不足时，会申请新的内存地址，并更新引用地址中对应的内存地址。

3.重新申请的内存地址将复制原始内存地址中的数据。

#### 4.3集合中的数据存储

1.集合中存储的数据都是数据对应的内存地址，没有真实数据

2.集合中的数据是通过引用地址的形式查找真实数据

### 字典引用（同列表）

#### 5.1字典存储空间

* 无变量引用

字典保存在临时引用地址中，该引用地址可以反复使用

* 有变量引用

字典保存在独立引用地址中，该引用地址专用，每个字典对象具有独立的引用地址

#### 5.2字典存储空间变更

1.字典变量指向引用地址，该引用地址中保存有存储字典信息的内存地址。

2.当字典中存储数据的空间不足时，会申请新的内存地址，并更新引用地址中对应的内存地址。

3.重新申请的内存地址将复制原始内存地址中的数据。

#### 5.3字典中的数据存储

1.字典中存储的数据都是数据对应的内存地址，没有真实数据

2.字典中的数据是通过引用地址的形式查找真实数据

### 对象引用

#### 6.1对象存储空间

* 无变量引用

对象保存在临时引用地址中，该引用地址可以反复使用

* 有变量引用

对象保存在独立引用地址中，该引用地址专用，每个列表对象具有独立的引用地址

#### 6.2对象中的变量数据

1.对象中的变量存储的都是对应数据的引用地址，没有真实数据

2.对象中的变量是通过引用地址的形式查找真实数据

3.对象中的变量存储形式是一个字典的形式，对变量的操作参照字典的操作方式，变量名为字典的key，变量值为字典的value

### 可变类型与不可变类型

#### 7.1基本概念

数据存储空间中的数据可以发生变化的数据称为可变类型数据

数据存储空间中的数据不能发生变化的数据称为不可变类型数据

#### 7.2分类

#### 7.3函数调用时的实参

* 函数的实参使用可变类型数据时，在函数内部对形参进行数据操作，实参将发生改变
* 函数的实参使用不可变类型数据时，在函数内部对形参进行数据操作，实参将不会发生改变

#### 7.4函数定义时的形参

条件1：函数形参使用默认参数

条件2：函数调用时未对默认参数传值

条件3：函数的形参使用可变类型数据

上述3个条件同时满足时，多次调用对应函数时，操作的形参是同一个对象