# 第四阶段 Python数据分析

# Python 基础7

## 网校相关视频

Python数据分析

第二章Numpy

## 明确本次课知识点，明确重点难点

### 【知识点目标】

* Numpy基础之Ndarray

### 【重点】

* Numpy基础之Ndarray

### 【难点】

* Numpy基础之Ndarray

## 复习巩固作业讲解

* Python面向对象
* Numpy介绍

## 本次课程任务讲解

### 【知识点1】Numpy中的多维数组ndarray

1. Python在数组和矩阵运算上的缺陷和Numpy的改进
   1. 标准安装的Python中用列表(list)可以用来当作数组使用，但是列表中所保存的是对象(任意对象)的指针。对于数值运算来说这种结构显然比较浪费内存和CPU计算时间。
   2. Python还提供了一个array模块，它直接保存数值。但是由于它只支持一维数组，也没有各种运算函数，因此也不适合做数值运算。
   3. NumPy提供了以下对象，解决标准Python的不足：
      1. ndarray（N-dimensional array object）N维数组(简称数组)对象，存储**单一数据类型**的N维数组
      2. ufunc（universal function object）通用函数对象，对数组进行处理的函数。
2. 什么是ndarray

NumPy中的ndarray是一个多维数组对象，它是一个快速而灵活的大**数据集容器**，它由两部分组成：

（1）实际的数据；

（2）描述这些数据的元数据。

大部分的数组操作仅仅修改元数据部分，而不改变底层的实际数据。另ndarray的**所有元素的类型必须是一致的**。

1. 创建ndarray对象
   1. 创建一维数组

通过np.array方法创建

* 1. 获取数组的维度和类型

ndarray.shape属性：维度

ndarray.dtype属性：数据类型

注意ndarray中的数据类型由兼容性最强的值数据类型决定

* 1. 创建多维数组
     1. 直接创建

通过np.array方法创建

* + 1. 用其他方式创建

zeros(全0数组)

ones(全1数组)

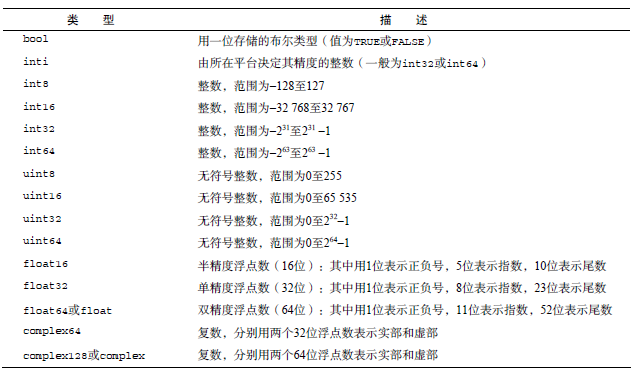
full(全填充数组，可以自行设定填充的数字)

eye(对角数组)

random(随机数组)

1. Numpy数据类型

Python支持的数据类型不足以满足科学计算的需求，因此NumPy添加了很多其他的数据类型，它们占用的内存空间也是不同的。在NumPy中，大部分数据类型名是以数字结尾的，这个数字表示其在内存中占用的位数。下面的表格列出了NumPy中支持的数据类型。



1. 数组的索引和切片
   1. 基本概念
      1. 索引即通过一个无符号整数值获取数组里的值。Python索引是从0的位置开始。
      2. 切片即对数组里某个片段的描述。
   2. 数组的索引
      1. 一维数组的索引：类似于Python的列表
      2. 二维数组的索引
         1. 以一维数组的索引方式访问时，获取的是一个一维数组
         2. 以二维数组的索引方式访问时，获取的是一个值
      3. 多维（三维）数组的索引

参考二维数组

利用多维数组的多维索引，访问效率较高

array[v1,v2,v3...]，通过,分隔维度

* 1. 数组的切片
     1. 一维数组的切片：与Python列表类似
        1. 下标形式：a[begin:end] 取begin到end-1位置上的元素（步长1）
        2. 下标+步长形式：a[begin:end:step] 取begin到end-1位置上的元素（步长step）
        3. 负数下标翻转数组：a[::负数]
     2. 多维数组的切片：在一维基础上增加维度的处理
        1. 下标形式 b[m,n,o]
        2. 下标+步长: b[b1:e1:s1,b2:e2:s2,b3:e3:s3...]

在对多维数组进行切片的时候，[m,n,o],前后的索引都可以接受切片形式的操作，逗号分隔维度，冒号分隔切片的需求

1. 改变数组的维度
   1. ndarray.reshape方法：通过一个元组，设定数组的新维度，注意改变时数组的**元素数量必须匹配**
   2. ndarray.shape属性：通过一个元组设定数组的新维度
   3. ndarray.resize方法：和reshape效果相同，但resize是改变原来的数组，不返回值
   4. 以上三个扩展维度的方法，resize和shape是直接作用在数组自身，reshape是作用在数组映射出的视图上（即获取的新数组和原数组共享内存空间），因此原数组发生变化，新数组也会受到影响，反之也一样（可以用copy获得一份新数组，不受该影响）

## 自主学习作业讲解

无

## 课程总结

#### ndarray的创建和基本操作

## 下次自主学习任务布置

见作业文档