# 第四阶段 Python数据分析

# Python 基础6

## 网校相关视频

Python语言基础

第二章Python初级基础知识

## 明确本次课知识点，明确重点难点

### 【知识点目标】

* Python面向对象
* Numpy入门

### 【重点】

* Python面向对象
* Numpy入门

### 【难点】

* 面向对象：封装和继承
* Numpy ndarray相关操作

## 复习巩固作业讲解

* Python函数

## 本次课程任务讲解

### 【知识点1】Python面向对象

1. 常见的编程方式
   1. 面向过程

根据业务逻辑从上到下书写代码

* 1. 函数式

将相关的业务逻辑代码封装到程序块中，后续只需要调用即可

* 1. 面向对象

对函数进行分类和封装，做为对象的内置属性或方法，在调用时通过对象进行调用

1. 面向对象的基本概念

什么是面向对象

对象

对象有哪些特征、哪些行为、对象和对象之间有什么关系

对象的属性，对象的方法，**封装，继承**

**属性：对象的特征，用来描述对象（笔的材质、长短）**

**对属性可以做的操作：获取属性/设置属性**

**方法：对象的行为，用来处理对象可以实现的功能（笔有写的方法）,方法其实就是封装在对象内部的函数**

**pen.write(“hello”)**

1. 类与对象
   1. 什么是类

类是用来描述具有相同属性和方法的对象的集合

* 1. 什么是对象

对象是对类的实例化

* 1. 类和对象的关系

类是对象的类型，对象是类的实例，类是抽象概念，对象是具体实体

1. 类的声明和实例化
   1. 创建类

class语句

class 类名称(继承类):

类体

* 1. 实例化类

和函数的调用方式类似

asset1 = Asset()

* 1. 如何调用类
     1. 类属性的访问

用点语法进行属性的访问

变量 = 对象.属性 （获取类的属性）

对象.属性 = 表达式 （设置类的属性）

dict.keys()

* + 1. 类方法的调用

使用点语法进行方法的调用

对象.方法(参数)

* 1. 类的构造函数：创建类属性

构造函数是一个属于类的特殊方法，\_\_init\_\_(),前后都有两个下划线，类的构造函数。

每当用类名称实例化创建对象时，都会执行\_\_init\_\_()方法，其作用是初始化已经实例化的对象。

在\_\_init\_\_()方法中，可以设置类的属性，在初始化类的时候，需要为这些属性赋值

* 1. 封装：
     1. 在类中定义方法

通过设定自定义方法名称，在类中创建方法

* + 1. 调用类的方法

通过对象.方法名()来调用

* + 1. 属性私有化

将属性限制在类的内部进行访问，不暴露到类外面

通过在属性前加上\_\_（两个下划线）,将属性转为private属性

私有化属性的调用可以通过类中的方法（函数）来实现

* 1. 继承：子类继承父类的属性和方法

继承的类具有父类的属性和方法，也可以有自己的属性和方法

### 【知识点2】Python科学计算库Numpy简介

1. 导入：针对矩阵/向量的编程

[0,1,4]+[0,1,8]=[0,2,12]

a列表是0-n的平方，b列表是0-n的三次方，结果是将两个列表按位相加

* 1. 传统Python方法
  2. 基于Numpy的方法

1. 为什么要使用Numpy
   1. 精简

Numpy中有直接针对数组和矩阵的操作

* 1. 高效

Numpy对数组和矩阵进行操作的时候可以省略循环

Numpy底层代码使用C写的

1. Numpy是什么
   1. Numeric Python，一个开源的Python科学计算库，包括

* 一个强大的N维数组对象ndarray；
* 比较成熟的函数库；
* 用于整合C/C++和Fortran代码的工具包；
* 实用的线性代数、傅里叶变换和随机数生成函数
  1. NumPy支持**高维度数组与矩阵运算**，此外也针对数组运算提供大量的数学函数库

1. Numpy的安装
   1. 纯Python环境

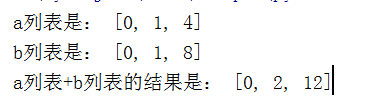
cmd下pip install numpy

* 1. Pycharm

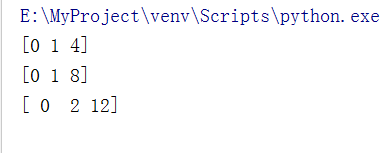
project

1. Numpy初步实践

原生python的执行结果



numpy执行的结构



## 自主学习作业讲解

无

## 课程总结

## 下次自主学习任务布置

见作业文档