**《Python全栈开发》**

**第一单元 Python 基础入门**

1.1 为什么要学习Python

1.1.1 Python的用途

1.1.2 Python的优势

1.2 Python与其他语言的比较

1.2.1 Python与其他语言的区别

1.2.2 Python的优缺点

1.3 Python的简介

1.3.1 Python的起源

1.3.2 解释型语言与编译型语言区别

1.4 Python环境搭建与VIM指令

1.4.1 Linux下Python 的安装

1.4.2 Windows下Python 的安装

1.4.3 VIM常用命令

1.5 数据类型

1.5.1 标准数据类型介绍

1.5.2 数值型

1.5.3 初识字符串

1.5.4 判断数据类型（type）

1.6 第一个Python开发程序

1.6.1“hello world”程序的编写

1.6.2 程序代码讲解

1.7 运算符和表达式

1.7.1 赋值运算符、算术运算符、关系运算符、逻辑运算符

1.7.2 表达式

1.8 扩展模块的安装与导入

1.8.1使用pip安装Python扩展

1.8.2模块导入import、from ...import

**第二单元 Python控制语句**

2.1 if语句

2.1.1 为什么要使用if语句

2.1.2 如何使用if语句

2.1.3 常见的if语句分类

2.2 for语句

2.2.1 为什么要使用for语句

2.2.2 如何使用for语句

2.2.3 常见的for语句分类

2.3 while语句

2.3.1 为什么要使用while语句

2.3.2 如何使用while语句

2.3.3 常见的while语句分类

2.4 案例

2.4.1 九九乘法表

2.4.2 登陆验证

**第三单元 序列之字符串**

3.1 序列

3.1.1为什么要使用序列

3.1.2 什么是序列

3.1.3 序列的常用操作（下标、切片）

3.1.4 序列的遍历

3.2 字符串

3.2.1 为什么使用字符串

3.2.2 什么是字符串

3.2.3 字符串表现形式

3.3 字符编码介绍

3.4 格式化字符串

3.4.1 使用”%”进行格式化输出

3.4.2 使用format方法进行格式化输出

3.5 字符串常用方法（一）

3.5.1 去除空格 strip、rstrip、lstrip、replace

3.5.2 个数统计len、count

3.5.3 大小写转换 lower、upper

**第四单元 字符串常用操作**

4.1 字符串常用方法（二）

4.1.1 编码转换方法 encode、decode

4.1.2 字符串常用判断方法isdigit、isalpha、isupper

4.1.3 其他方法join、find、split

4.2字符串单词次数统计

4.3 字符串替换

4.4 生成验证码

4.5总结与案例：登录注册

**第五单元 序列之列表**

5.1为什么使用列表

5.2 什么是列表

5.3 列表的特征

5.4 创建列表

5.5 引用和修改元素

5.5.1 列表引用

5.5.2 列表修改

5.6 元素添加

5.6.1 追加元素

5.6.2 插入元素

5.6.3 扩展列表

5.7 列表的删除

5.7.1 pop

5.7.2 remove

5.7.3 clear

5.7.4 del

5.8 元素排序与反转

5.8.1 sort、sorted

5.8.2 reverse、reversed

**第六单元 列表嵌套与算法**

6.1 列表嵌套

6.1.1 如何实现嵌套

6.1.2 列表嵌套应用

6.2 双色球之红球生成（一）

6.2.1 思路分析

6.2.2 代码实现

6.3冒泡排序

6.3.1 为什么使用冒泡排序

6.3.2 什么是冒泡排序

6.3.3 冒泡排序的实现

6.4插入排序

6.4.1 为什么使用插入排序

6.4.2 什么是插入排序

6.4.3 插入排序的实现

**第七单元 序列之元组**

7.1 为什么使用元组

7.2 什么是元组

7.3 元组的特征

7.4元组的常用操作

7.4.1元组的创建

7.4.2 元组的删除

7.5生成器推导式

7.6元组的嵌套

7.7综合应用：饭店收银系统

**第八单元 字典**

8.1 为什么使用字典

8.2 什么是字典

8.3 字典特征

8.4 创建字典

8.5 获取字典value

8.5.1 通过key获取

8.5.2 使用get方法获取

8.6 字典的删除

8.6.1 del

8.6.2 pop

8.6.3 clear

8.7 字典的修改与赋值

8.7.1 通过key实现

8.7.2 通过fromkeys实现

8.8 字典的嵌套

8.9 字典的遍历

**第九单元 集合**

9.1 为什么使用集合

9.2 什么是集合

9.3 集合的特征

9.4 创建集合

9.5 集合元素添加

9.6 集合元素删除

9.6.1 remove

9.6.2 discard

9.6.3 pop

9.6.4 clear

9.7 集合查询

9.8 集合间的运算

9.8.1 交集

9.8.2 并集

9.8.3 差集

9.8.4 补集

9.9案例：双色球之红球生成（二）

**第十单元 函数**

10.1 为什么使用函数

10.1.1 函数与过程对比

10.1.2 函数的优势

10.2 函数的定义

10.3 函数的创建

10.3.1 有参函数

10.3.2 无参函数

10.3.3 有返回值函数

10.3.4 无返回值函数

10.4 调用函数

10.5 函数的参数

10.5.1 函数的形参与实参

10.5.2 位置参数

10.5.3 默认参数

10.6 非关键字可变长参数

10.6.1 为什么使用非关键字可变长参数

10.6.2 如何使用非关键字可变长参数

10.7 关键字可变长参数

10.7.1 为什么使用关键字可变长参数

10.7.2 如何使用关键字可变长参数

10.8 案例：用函数改写收银系统

**第十一单元 变量分类与匿名函数、递归函数**

11.1 变量依据作用域分类

11.1.1 局部变量

11.1.2 全局变量

11.2 函数内使用全局变量

11.3 匿名函数

11.3.1 为什么使用匿名函数

11.3.2 什么是匿名函数

11.3.3 如何使用匿名函数

11.4 递归函数

11.4.1 为什么使用递归函数

11.4.2 什么是递归函数

11.4.3 如何使用递归函数

11.5 总结与练习：斐波那契案例

**第十二单元 生成器函数、装饰器函数与闭包函数**

12.1 生成器函数

12.1.1 为什么使用生成器函数

12.1.2 什么是生成器函数

12.1.3 如何使用生成器函数

12.2 函数名作为参数

12.3装饰器函数

12.3.1 为什么使用装饰器函数

12.3.2 什么是装饰器函数

12.3.3 如何使用装饰器函数

12.4 闭包函数

12.4.1 为什么使用闭包函数

12.4.2 什么是闭包函数

12.4.3 如何使用闭包函数

12.5 内置高阶函数使用

12.5.1 filter

12.5.2 map

12.5.3 reduce

12.6 函数综合练习：登录次数记录

**第十三单元 面向对象编程**

13.1 为什么使用面向对象

13.1.1面向过程的取款

13.1.2面向过程的不足

13.1.3面向对象的取款

13.1.4面向对象的优点

13.2 什么是面向对象

13.3 什么是对象

13.3.1 生活中的对象

13.3.2 计算机中的对象

13.4 对象的属性和方法

13.5 什么是类

13.5.1类与对象的关系

13.5.2类的定义

13.6 创建类

13.6.1 创建类属性

13.6.2 使用类属性

13.7 类的实例化

13.8 对象属性和方法

13.8.1对象属性

13.8.2 对象方法

13.8.3 带有位置参数的对象方法

13.8.4 带有默认值参数的对象方法

13.8.5 对象方法中self参数介绍

13.9 总结与案例：双色球类与用户类创建

**第十四单元 私有属性和方法**

14.1 对象的私有属性和私有方法

14.1.1 对象的私有属性

14.1.2 对象的私有方法

14.2 Python的魔法方法

14.2.1 \_\_init\_\_()

14.2.2 \_\_del\_\_()

14.2.3 \_\_str\_\_()

14.2.4 \_\_doc\_\_()

14.3将对象方法包装成属性之@property

14.4 案例：狗的一生

**第十五单元 类方法与静态方法**

15.1类方法的定义和调用

15.1.1 为什么要使用类方法

15.1.2 什么是类方法

15.1.3 如何使用类方法

15.1.4类方法的定义

15.1.5在类内部被调用

15.1.6在类外部被调用

15.2静态方法

15.2.1 为什么要使用静态方法

15.2.2 什么是静态方法

15.2.3 如何使用静态方法

15.2.4静态方法的定义

15.2.5静态方法特性和调用

15.3 总结与练习：出国学生统计

**第十六单元 面向对象的特征--抽象、封装和继承**

16.1面向对象特征

16.2 抽象

16.2.1 什么是抽象

16.2.2 如何实现抽象

16.3封装

16.3.1 为什么要封装

16.3.2 什么是封装

16.3.3 如何实现封装

16.4 继承

16.4.1 为什么要继承

16.4.2 什么是继承

16.4.3 如何实现继承

16.4.4 使用父类方法

16.4.5重写父类方法

16.5 总结与练习：吃鸡游戏

**第十七单元 面向对象的特征--多态**

17.1 多态

17.1.1 为什么使用多态

17.1.2 什么是多态

17.1.3 如何使用多态

17.2 多态性

17.2.1 什么是多态性

17.2.2 如何使用多态性

17.3 多态与多态性区别

17.4 案例：动物园

**第十八单元 鸭子类型与设计模式**

18.1 鸭子类型

18.1.1 什么是鸭子类型

18.1.2 如何使用鸭子类型

18.1.3 鸭子类型与多态区别

18.2 Python的设计模式

18.3 单例模式

18.3.1 为什么使用单例模式

18.3.2 什么是单例模式

18.3.3 如何使用单例模式

18.4 工厂模式

18.4.1 为什么使用工厂模式

18.4.2 什么是工厂模式

18.4.3 如何使用工厂模式

18.5 总结与练习：计算器

**第十九单元 综合案例**

19.1 案例展示

19.2案例分析

19.3 思路图绘制

19.4 系统类的创建

19.5 会员登陆注册登录

19.6 手选彩票

19.7 机选彩票

**第二十单元 综合案例及答辩**

20.1 结果对比

20.2 结果查询

20.3 学生答辩