

# 《Python 程序设计基础》

## 一、课程基本信息

课程名称	中文名称		Python 程序设计基础				
	英文名称		The Basic Python language programming				
课程代码				课程性质	选修		
开课学院	软件与物联网工程学院			课程负责人			
授课学期	2022-2023 学年 第一学期			学分	4		
任课教师	张勇						
课程总学时	48	理论学时	32	实验学时 (含上机)	32	课内实践学时	
课程类型	<input type="checkbox"/> 线上课程		<input checked="" type="checkbox"/> 线下课程		<input type="checkbox"/> 混合式课程		线上学时
							线下学时
适用专业及年级	VR 专业一年级			授课语言	中文		
先修课程	无			后续课程	大数据分析、高级程序设计		
课程简介	<p>1. 该课程的目的在于通过学习“Python 程序设计基础”，学生可以熟练掌握 Python 语言的强大又简单的面向对象编程方法，掌握 Python 语言中的基本数据类型：整型、浮点型、复数型、字符串等，和特殊形式的数据类型：列表类型、元组类型、集合类型、字典类型等。灵活运用内置函数、递归函数、装饰器和复合函数，进一步掌握 Python 语言的高级编程技术，例如：类与对象、文件相关操作和图形用户界面设计等，初步具备能够使用 Python 语言实现较大规模软件的能力，为后续进一步学习大数据分析、高级程序设计等技术奠定扎实的基础。 2. 教学目标在微观上体现为：让学生通过一学期的学习与上机实验，掌握 Python 语言基本语法和常用数据类型；熟练应用自定义函数解决简单问题；掌握类与对象的用法；熟练应用 Python 语言的 os 模块实现文件的安全读写操作；掌握 Python 语言的 Tkinter 模块实现简单的图形用户界面设计等。在宏观上体现为：全面掌握和精通 Python 语言的高级编程技术，学会使用计算机语言解决数学问题和实际问题的建模方法，培养学生良好的编程习惯和编程思维。 3. 基本知识教学要求：掌握 Python 语言的基础语法和程序控制结构；灵活运用常用数据类型；掌握 Python 语言的函数与模块；熟练应用类与对象；熟练掌握 os 模块实现文件的安全读写操作，应用 Tkinter 模块实现简单的图形用户界面设计等。能力教育要求：培养学生熟练应用 PyCharm 开发较大规模 Python 应用程序；通过大量的练习和实例，增强学习的阅读、编写和调试程序的能力。</p>						
执笔人				审定人			

讨论参与人	
大纲制定日期	年 月

## 二、课程目标

### 课程目标 1：知识目标

- 1.1 掌握 Python 语言的基础语法、程序控制和数据类型。
- 1.2 掌握 Python 语言的函数定义、类与模块的使用。
- 1.3 掌握 Python 语言的 os 模块和 Tkinter 模块。

### 课程目标 2：能力目标

2.1 学会使用 PyCharm 集成开发环境设计较大规模的 Python 语言程序，掌握 Python 语言面向对象编程技巧和各种模块的功能。

2.2 培养良好的程序编写习惯，培养学生画流程图和注释程序的技巧。

2.3 为后续学习大数据分析、高级程序设计等内容打下扎实的基础。

### 课程目标 3：素质目标

3.1 通过学习和交流讨论，掌握抽象并求解基本计算问题的初步能力。学生能够通过积极参与课堂讨论活动，并能够在课堂讨论活动中，主动思考，提出自己的观点和疑惑，培养敢于猜想、敢于提出问题、敢于挑战权威的科学意识和创造性思维。

3.2 深刻体会计算思维，通过计算思维来解决生活中的问题。

3.3 通过综合性设计性实验及实践环节，让同学们明白细节规划的重要性，培养工匠精神；通过小组讨论，分工合作完成设计性实验项目，培养学生团队合作精神；引导学生参加各类学科竞赛，培养学生公平竞赛、追求卓越的精神。

### 课程目标 4：思政目标

4.1 学生通过学习课程内容，掌握 Python 语言的编程技能，能够使用 Python 语言解决现实生活中遇到的各种问题，树立科技报国志向。

4.2 通过对红色经典文献的数据存储与分析，加深对马克思主义思想及原理的理解，深刻理解社会主义核心价值观的精髓和内涵。

4.3 通过对 Python 语言类与对象的分析，加深对唯物主义辩证法的理解。

### 课程目标与毕业要求指标点对应关系表

序号	毕业要求	支撑毕业要求指标点	课程目标												
			知识目标			能力目标			素质目标			思政目标			
			1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2	3.3	4.1	4.2	4.3	
1	毕业要求1: 具备工程知识: 能够将数学、自然科学、工程基础和专业知识用于解决复杂软件工程问题。	<p><b>指标点 1-2:</b> 掌握程序设计与数据结构等专业基础知识, 培养算法设计与计算思维能力, 对软件复杂工程问题提出解决方案。</p> <p><b>指标点 1-3:</b> 掌握软件理论专业知识, 用于软件系统及复杂工程问题的需求分析、设计、开发与实现。</p>	√	√	√	√	√	√							
2	毕业要求3: 设计/开发解决方案: 能够设计针对复杂工程问题的解决方案, 设计满足特定需求的系统、部件或流程, 并能够在设计环节中体现创新意识, 考虑社会、健康、安全、法律、文化以及环境等因素。	<p><b>指标点 3-1:</b> 能够基于软件复杂工程问题的需求分析, 确定其设计目标及具体方案。</p> <p><b>指标点 3-2:</b> 在设计目标与具体方案确定过程中, 考虑社会、健康、安全、法律、文化以及环境等因素的影响。</p> <p><b>指标点 3-3:</b> 在设计目标与具体方案确定过程中, 具有创新的理念、意识和方法, 并融合软件工程领域相关前沿知识与最新发展趋势。</p>				√	√	√							

3	<p><b>毕业要求4:</b> 研究:能够基于科学原理并采用科学方法对复杂工程问题进行研究,包括设计实验、分析与解释数据,并通过信息综合得到合理有效的结论。</p>	<p><b>指标点4-1:</b> 能够基于科学原理并采用科学方法针对计算机软硬件系统进行实验设计并确定实验方案与流程。</p> <p><b>指标点4-2:</b> 能够采集数据并运用相关模型与工具对数据进行整理分析,并做出解释说明。</p> <p><b>指标点4-3:</b> 通过数据挖掘、关联分析、数据可视化、信息综合等方法与手段,呈现实验数据及结果,得到合理有效的结论。</p>									√	√	√	√
4	<p><b>毕业要求9:</b> 个人和团队:能够在多学科背景下的团队中承担个体、团队成员以及负责人的角色。</p>	<p><b>指标点9-1:</b> 能够在多学科背景下的团队中开展工作,承担个体和团队成员的角色,并具备良好的沟通能力。</p> <p><b>指标点9-2:</b> 能够承担团队负责人的角色,并具备良好的组织能力,协调团队成员积极有效地开展工作。</p>			√	√		√	√	√				√

### 三、教学内容及进度安排

#### 教学内容及进度安排表

序号	章节内容	学生学习 预期成效	学时	授课 形式	教学方式	支撑 课程 目标
1	<p>第一章 Python 编程基础</p> <p>1. Python 语言输入与输出</p> <p>2. Python 基本数据类型</p> <p>3. Python 运算符</p> <p>4. 列表</p> <p>5. 字符串</p> <p><b>【重点】</b> 熟练掌握 Python 语言的基本数据类型和运算符，灵活运用列表类型。</p> <p><b>【难点】</b> 列表的元素访问以及处理方法。</p>	<p>1. 熟练掌握 Python 语言的基本数据类型</p> <p>2. 熟悉 Python 语言中的运算符</p> <p>3. 熟练掌握 Python 语言的列表类型的使用</p> <p>4. 掌握 Python 语言中字符串的相关处理方法</p>	6	线下多媒体	<p>1. 使用课件课堂讲授 Python 编程基础知识点</p> <p>2. 通地实例介绍 Python 语言中的基本数据类型和运算符</p> <p>3. 通过实例 Python 语言中的列表类型和字符串类型</p> <p>4. 布置课后练习，强化学生对 Python 编程基本的知识</p> <p>5. 通地实例开展课上讨论</p>	1.1 2.1 2.2 3.1 3.3
2	<p>课内实验： 实验一、输入与输出实验</p>	<p>1. 掌握 Python 语言的输入和输出语句</p> <p>2. 了解程序调试过程</p> <p>3. 灵活运用列表和字符串的使用</p>	6	线下	<p>学生上机做实验 教师指导</p>	1.1 2.1 2.2 3.1 3.3
3	<p>第二章 程序控制</p> <p>1. 顺序结构</p> <p>2. 分支结构</p> <p>3. 循环结构</p> <p>4. 排序实例</p> <p><b>【重点】</b> 顺序结构、分支结构、循环结构</p> <p><b>【难点】</b> 熟练使用各种程序控制结构解决问题。</p>	<p>1. 了解 Python 语言中基本的程序控制结构</p> <p>2. 掌握顺序结构、分支结构、循环结构定义</p> <p>3. 学会使用 Python 语言的程序控制结构处理简单问题</p> <p>4. 学会使用 Python 语言进行排序的基本思想</p>	6	线下多媒体	<p>1. 借助于 PPT 讲授 Python 语言程序控制的知识点</p> <p>2. 通过程序实例介绍顺序结构</p> <p>3. 通过程序实例介绍分支结构</p> <p>4. 通过程序实例介绍循环结构</p> <p>5. 布置课后练习，强化学生对 Python 语言程序控制的认识</p> <p>6. 通地实例开展课上讨论</p>	1.2 1.3 2.1 2.2 3.2 3.1 3.2 4.1

4	课内实验 实验二、冒泡排序实验	1. 掌握 random 模块的使用 2. 熟练掌握各种程序控制结构的使用	6	线下	学生上机做实验 教师指导	1.2 1.3 2.1 2.2 3.2 3.1 3.2 4.1
5	第三章 数据表示 1. 元组 2. 集合 3. 字典 【重点】元组、集合和字典类型的定义和使用 【难点】熟练使用元组、集合和字典类型解决一些实际问题	1. 了解Python语言中基本数据类型与特殊数据类型的区别 2. 理解元组、集合和字典的相关概念 3. 掌握元组、集合和字典的定义 4. 熟练使用元组、集合和字典的元素访问和内置函数 5. 理解书本给出的实例的基本思想	6	线下多媒体	1. 通过多媒体讲解 Python 语言的数据表示 2. 借助多个实例介绍元组、集合和字典的定义、元素访问以及内置函数 3. 通过对课本实例分析与讨论，加强学生对 Python 语言的数据表示的理解 4. 布置课后练习，强化学生对数据表示的认识 5. 通地实例开展课上讨论	1.2 2.2 2.3 3.2 3.3 4.1 4.3
6	课内实验 实验三、奶茶店点单实验	1. 掌握字典的使用 2. 掌握使用 lambda 函数	6	线下	学生上机做实验 教师指导	1.2 2.2 2.3 3.2 3.3 4.1 4.3
7	第四章 函数与模块 1. 常用内置函数 2. 自定义函数 3. 递归函数 4. 复合函数 5. 包与模块 【重点】常用内置函数、递归函数以及包与模块 【难点】可变参数函数、函数闭包以及装饰器	1. 了解常用的内置函数 2. 学会自定义函数并能实现简单的功能 3. 理解递归函数的几个实例原理，进一步掌握递归函数的使用 4. 初步理解关于包与模块的概念	6	线下多媒体	1. 通过多媒体讲解函数与模块的知识点 2. 借助实例介绍常用内置函数的使用 3. 借助实例介绍自定义函数的格式 4. 借助实例说明递归函数 5. 布置课后练习，强化学生对函数与模块的认识 6. 通地实例开展课上讨论	1.3 2.1 2.3 3.2 3.3 4.1 4.3
8	课内实验		6	线下	学生上机做实验	1.3

	实验四、快速排序实验	1.掌握自定义函数的格式 2.掌握递归函数 3.掌握快速排序算法的原理			教师指导	2.1 2.3 3.2 3.3 4.1 4.3
9	第五章 类与对象 1. 类与对象的定义 2. 类中的属性与方法 3. 特殊方法 4. 继承 【重点】类中的属性与方法、继承 【难点】类中的数据成员与方法成员、类之间的继承	1. 理解类与对象的定义 2. 理解并掌握类中的属性与方法的定义 3. 掌握类中特殊方法的定义和使用 4. 学习类之间继承	6	线下多媒体	1. 通过多媒体讲解类与对象的知识点 2. 通过多个实例介绍类与对象中的属性与方法定义 3. 通过程序实例介绍类中特殊方法的定义和使用 4. 通过实例介绍类之间的继承 5. 布置课后练习，强化学生对类与对象 6. 通地实例开展课上讨论	1.3 2.1 2.3 3.1 3.3 4.2 4.3
10	课内实验 实验五、类的继承实验	1. 掌握类与对象的定义 2. 掌握类中的属性与方法的使用 3. 掌握类之间的继承关系	6	线下	学生上机做实验 教师指导	1.3 2.1 2.3 3.1 3.3 4.2 4.3
11	第六章 文件操作与异常 1. 磁盘文件读写操作 2. os 模块 3. Excel 文件读写操作 4. 异常 【重点】os 模块的使用、异常处理结构 【难点】异常处理结构的使用	1. 掌握磁盘文件的读取操作 2. 熟悉使用 os 模块 3. 掌握 Excel 文件读写操作 4. 学会灵活使用异常处理结构	6	线下多媒体	1. 通过多媒体讲解文件操作与异常的知识点 2. 通过实例介绍文件读取操作和 os 模块 3. 通过程序实例介绍异常处理结构的使用 4. 布置课后练习，强化学生对文件操作与异常 5. 通地实例开展课上讨论	1.3 2.1 2.3 3.1 3.3 4.1 4.2
12	课内实验 实验六、文件安全读写操作实验	1. 掌握文件的读写操作 2. 掌握常用的异常处理结构 3. 灵活地将异常处理结构运用到文件读写操作	6	线下	学生上机做实验 教师指导	1.3 2.1 2.3 3.1 3.3 4.1 4.2
13	第七章	1.	6	线下	1.	1.1

	<p>图形用户界面设计</p> <ol style="list-style-type: none"> <li>1. 视窗设计</li> <li>2. 界面布局设计</li> <li>3. “复数计算器”程序算法设计</li> <li>4. 常用控件</li> <li>5. 画布与绘图技术</li> <li>6. 事件绑定与自定义事件</li> </ol> <p><b>【重点】</b>界面布局设计、常用控件、事件绑定与自定义事件</p> <p><b>【难点】</b>使用控件实现图形用户界面设计</p>	<p>了解视窗设计的基本流程</p> <ol style="list-style-type: none"> <li>2. 掌握三种界面布局方法</li> <li>3. 理解“复数计算器”程序算法</li> <li>4. 理解各种控件的使用</li> </ol>		多媒体	<p>通过多媒体讲解图形用户界面设计的知识点</p> <ol style="list-style-type: none"> <li>2. 通过实例介绍视窗设计的流程和界面布局设计</li> <li>3. 通过多个实例介绍常用控件的使用</li> <li>4. 通过程序实例介绍画布与绘图技术</li> <li>5. 通过实例介绍事件绑定与自定义事件</li> <li>6. 布置课后练习，强化学生对图形用户界面设计的认识</li> <li>7. 通地实例开展课上讨论</li> </ol>	<p>1.3</p> <p>2.2</p> <p>2.3</p> <p>3.1</p> <p>3.3</p> <p>4.1</p> <p>4.3</p>
14	<p>课内实验</p> <p>实验七、“复数计算器”实验</p>	<ol style="list-style-type: none"> <li>1. 理解视窗设计流程</li> <li>2. 会使用各种界面布局方法</li> <li>3. 掌握各种控件的使用</li> </ol>	6	线下	<p>学生上机做实验</p> <p>教师指导</p>	<p>1.1</p> <p>1.3</p> <p>2.2</p> <p>2.3</p> <p>3.1</p> <p>3.3</p> <p>4.1</p> <p>4.3</p>
15	<p>大作业</p> <p>设计一个学生信息管理系统，包括以下两项功能：</p> <p><b>(1) 管理人员登录</b></p> <p>以信息库管理员登录本系统，具有权限：1) 查看全部学生的基本信息：学号、姓名、性别、科目成绩、学生账号、密码等；2) 能够对学生信息进行录入和修改，例如用户名和密码（初始密码为 123456）。3) 查询单个学生的相关信息。</p> <p><b>(2) 学生登录</b></p>	<p>通过本次大作业，使学生综合运用Python语言设计一个大型应用软件——学生信息管理系统，实现学生信息的入库、修改和日常维护等功能，使用类和对象管理学生信息。</p>	4	课外	<p>课外完成</p> <p>教师指导</p>	<p>2.2</p> <p>2.3</p> <p>3.1</p> <p>3.2</p> <p>4.1</p> <p>4.3</p>

以学生身份登录本系统，具有权限：1) 显示本人基本信息；2) 选修课程；3) 退选课程；4) 查看选修课程状态（是否满员）；5) 查询成绩单；6) 更新自己的登录密码。						

## 具体章节内容

### 第一章 Python 编程基础

#### 第一节 Python 语言输入与输出

#### 第二节 Python 基本数据类型

#### 第三节 Python 运算符

#### 第四节 列表

#### 第五节 字符串

#### 【重点难点】

【重点】熟练掌握 Python 语言的基本数据类型和运算符，灵活运用列表类型。

【难点】列表的元素访问以及处理方法。

### 第二章 程序控制

#### 第一节 顺序结构

#### 第二节 分支结构

#### 第三节 循环结构

#### 第四节 排序实例

#### 【重点难点】

【重点】顺序结构、分支结构、循环结构

【难点】熟练使用各种程序控制结构解

### 第三章 数据表示

#### 第一节 元组

#### 第二节 集合

#### 第三节 字典

#### 【重点难点】

【重点】元组、集合和字典类型的定义和使用

【难点】熟练使用元组、集合和字典类型解决一些实际问题

### 第四章 函数与模块

#### 第一节 常用内置函数

#### 第二节 自定义函数

#### 第三节 递归函数

第四节 复合函数

第五节 包与模块

**【重点难点】**

**【重点】** 常用内置函数、递归函数以及包与模块

**【难点】** 可变参数函数、函数闭包以及装饰器

第五章 类与对象

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/586210140010010145>