**特许全球金融科技师（CGFT）一级**

**《Python语言基础》考试大纲**

（修订日期： 2022年09月18日）

[一、 考试性质 2](#_Toc114676037)

[二、 考查目标 2](#_Toc114676038)

[三、 考试形式和试卷结构 3](#_Toc114676039)

[1. 试卷满分 3](#_Toc114676040)

[2. 答题方式 3](#_Toc114676041)

[3. 试卷内容结构 3](#_Toc114676042)

[4. 试卷题型结构 3](#_Toc114676043)

[四、 考察内容与考点解析 4](#_Toc114676044)

[1. 章节内容考点清单 4](#_Toc114676045)

[2. 重要高频考点提示 12](#_Toc114676046)

[五、 历年真题与考试样题 13](#_Toc114676047)

# 考试性质

特许全球金融科技师（CGFT）一级包含《Python语言基础》科目，Python编程是金融科技从业基础工具和技能，特许全球金融科技师（CGFT）一级考试是技能认证性考试，按照金融科技人才知识技能能力标准，以标准化考试的方式对考生对于Python工具技能的掌握程度进行评价。

# 考查目标

1. **知识技能要求**

CGFT一级对考生Python相关知识技能要求如下：

* 掌握Python编程环境的安装搭建与维护管理，熟悉常用Python编程环境IDE工具使用。
* 熟练掌握Python编程语言基础概念，理解Python语言基本原理，理解Python基础数据类型，理解函数式编程思想，掌握面向对象的程序设计方法，包括：语法基础、数据类型与对象、运算符与表达式、条件结构、循环结构、函数与函数式编程、面向对象与类、输入输出、异常机制、模块与包、正则表达式、数据库访问、内置函数、内置模块与标准库等。
* 熟练掌握Python常用外部第三方包的使用，包括：numpy包、pandas包、matplotlib包、statsmodels包、scipy包的使用。

1. **能力素质要求**

CGFT一级对考生Python相关能力素质要求如下：

* 可以灵活使用相关工具包完成数据处理、数据可视化、数据分析建模、应用开发等工作编程实现。

# 考试形式和试卷结构

## 试卷满分

CGFT一级考试包含七门课程，其中《Python编程基础》包含40题，该部分满分为100分。

## 答题方式

机考，闭卷，可以跨区答题

## 试卷内容结构

不对内容结构分布做明确限定。

## 试卷题型结构

以客观题方式进行考察，题型包括：判断题、单选题、多选题。

不对题型结构分布做明确限定。

# 考察内容与考点解析

## 章节内容考点清单

本表中考察要求的定义如下

* 基础识记：主要是事实知识，考察对事实性知识的记忆，“知道”即可。
* 一般理解：主要是概念定义，仅需一般理解。
* 深入理解：主要是核心原理，需要较深入理解并能灵活应用。
* 熟练操作：主要是操作技能，需要熟悉基本操作流程。
* 不做要求：不作为考试内容，考生可以自主选择阅读。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 章 | 节 | 考点 | 考查要求 | 重要性 | 难度 |
| 1. Python简介 | 1.1 Python基础 | * Python优点包括：可移植性、可解释性、面向对象 * Python可以做什么 * Python是解释性语言（脚本型语言） * Python是动态语言（变量无需提前定义且随时可赋值） | 一般理解 | 低 | 低 |
| 1.2 开发环境安装 | * 熟悉在Windows操作系统下Python安装流程 * 熟悉使用命令行、IDE(spyder)、JupyterNotebook三种方式运行Python代码 | 熟练操作 | 低 | 低 |
| 1. Python语法基础 | 2.1 行结构与缩进 | * 代码行续行符 * 代码缩进规则 | 识记 | 低 | 低 |
| 2.2 代码注释 | * 单行注释使用# * 多行注释使用’’’ | 识记 | 低 | 低 |
| 2.3 标识符命名规则 | * 标识符的定义 * 标识符命名规则 * 标识符命名规范（建议性） | 识记 | 低 | 低 |
| 2.4 系统保留字 | * 掌握系统保留字 | 识记 | 低 | 低 |
| 1. 数据类型和对象 | 3.1 变量与常量 | * 掌握编程中的变量与常量的基本概念 | 识记 | 低 | 低 |
| * 1. 基础类型——数值类型 | * 理解基础数值类型基本概念，包括：空值型、布尔型、整型、浮点型、复数型，熟悉数值类型变量定义方式，熟悉数值类型对象方法 | 一般理解 | 高 | 低 |
| 3.3 序列类型——字符串(str) | * 理解字符串类型基本概念、定义方式 * 掌握字符串类型操作方法，包括：索引与切片、查找子串find、定位index、计数count、替换replace、切分split、转换大写upper、转换小写lower、检查是否以指定字符串开始startwith、检查是否以指定字符串结束endwith | 一般理解 | 高 | 中 |
| 3.4序列类型——列表(list) | * 理解列表类型基本概念、定义方式 * 掌握列表类型操作方法，包括：求长度、索引与切片、添加元素(append、extend、insert)、删除、修改、查找、排序、遍历 | 一般理解 | 高 | 中 |
| 3.5 序列类型——元组(tuple) | * 理解元组类型基本概念、定义方式 * 掌握元组类型操作方法，包括：定义、索引切片 * 理解单元素元组的定义方式 | 一般理解 | 高 | 中 |
| 3.6映射类型——字典(dict) | * 理解字典类型基本概念、定义方式 * 掌握字典类型操作方法，包括：添加元素、修改元素、删除元素、元素个数、所有键keys、所有值values、所有元素items | 一般理解 | 高 | 中 |
| 3.7 集合类型——集合(set) | * 理解集合类型基本概念、定义方式 * 掌握集合类型操作方法，包括：不支持索引、用于list元素去重 | 一般理解 | 高 | 中 |
| 3.8 集合类型——冻结集合(frozenset) | * 理解冻结集合类型基本概念、定义方式 * 掌握冻结集合类型操作方法，复制、集合交、集合并、集合差 | 一般理解 | 低 | 中 |
| 1. 运算符和表达式 | 4.1运算符概述 | * 理解运算符基本概念定义 * 理解运算符分类：按操作数据个数分类、按运算类型分类 * 理解运算符的内部实现机制 | 识记 | 高 | 低 |
| 4.2算术运算符 | * 算术运算符：+、-、\*、/、//、%、\*\*，掌握概念定义和基本操作，特别是：字符串+、字符串\*、列表+、列表\*的理解。 | 识记 | 高 | 低 |
| 4.3赋值运算符 | * 赋值运算符：=，掌握概念定义和基本操作 | 识记 | 高 | 低 |
| 4.4比较（关系）运算符 | * 比较（关系）运算符：==、!=、>、<、>=、<=，掌握概念定义和基本操作 | 识记 | 高 | 低 |
| 4.5逻辑运算符 | * 逻辑运算符：and、or、not，掌握概念定义和基本操作 | 识记 | 高 | 低 |
| 4.6位运算符 | * 位运算符：&、|、^、<<、>>，掌握概念定义和基本操作 | 识记 | 高 | 低 |
| 4.7成员运算符 | * 成员运算符：in、not in，掌握概念定义和基本操作 | 识记 | 高 | 低 |
| 4.8身份运算符 | * 身份运算符：is、is not，掌握概念定义和基本操作 | 识记 | 高 | 低 |
| 4.9运算优先级 | * 理解运算优先级 | 识记 | 高 | 低 |
| 4.10字符串的格式化输出 | * 字符串的格式化输出：%占位符、.format()方法 | 识记 | 低 | 低 |
| 1. 程序流：条件与循环 | 5.1 入门的简单例子Tutorial |  | 不做要求 |  |  |
| 5.2条件分支 | * 条件分支包含三种形式：If结构、If-else结构、If-elif-else结构 | 一般理解 | 高 | 低 |
| 5.3循环语句 | * 循环的两种形式：for、while * break、continue循环控制，理解两者区别 | 一般理解 | 高 | 低 |
| 5.4 with与上下文管理器 | * with语句 * contextmanager装饰器的使用 | 一般理解 | 中 | 低 |
| 1. 生成器与迭代器 | 6.1 迭代与可迭代对象 | * For循环可以使用任何可迭代对象 * Enumerate内置函数 | 深入理解 | 高 | 低 |
| 6.2 列表生成式 | * 理解列表生成式的基本概念 * 列表生成式的定义方式：[ exp for iter\_var in iterable ] | 一般理解 | 高 | 低 |
| 6.3 生成器 | * 理解生成器基本概念 * 生成器定义方式1：生成器表达式：( exp for iter\_var in iterable ) * 生成器定义方式2：生成器函数：yield语句 | 深入理解 | 高 | 高 |
| 6.4 迭代器 | * 理解迭代器概念 * 理解生成器、迭代器、可迭代对象关系 | 深入理解 | 高 | 高 |
| 1. 函数与函数式编程 | 7.1 入门的简单例子Tutorials |  | 不做要求 | 低 | 低 |
| 7.2 定义函数 | * 理解函数概念 * 函数定义方式 * Pass语句占位符的概念 * 函数的参数检查 * 多返回值 | 一般理解 | 高 | 低 |
| 7.3 调用函数 | * 函数调用方式 * 为函数提供说明文档，用在help()中 | 一般理解 | 高 | 低 |
| 7.4 函数的参数 | * 位置参数 * 默认参数 * 可变参数 * 关键字参数 * 命名关键字参数 * 参数顺序 | 深入理解，需要重点掌握 | 高 | 高 |
| 7,5 递归函数 | * 理解递归函数基本概念 * 递归函数的定义方式 | 一般理解 | 中 | 高 |
| 7.6 高阶函数 | * 理解高阶函数的概念 * Sorted()函数是高阶函数 | 一般理解 | 中 | 中 |
| 7.7 返回函数 | * 将函数名作为函数的返回值 | 一般理解 | 低 | 低 |
| 7.8 匿名函数 | * 理解匿名函数的定义方式：lambda | 一般理解 | 高 | 低 |
| 7.9 函数装饰器 | * 理解函数装饰器概念定义 * 理解函数装饰器的作用 | 一般理解 | 高 | 高 |
| 7.10 偏函数 | * 理解偏函数的定义 | 一般理解 | 低 | 低 |
| 1. 面向对象与类 | 8.1 入门的简单例子Tutorials |  | 不做要求 | 低 | 低 |
| 8.2 类和实例 | * 类和对象的基本概念 * 类的定义方式 * 类的继承 * 数据的封装 | 深入理解，需要重点掌握 | 高 | 高 |
| 8.3 访问限制 | * 理解私有变量概念，为什么需要私有变量 * 访问私有变量使用get()和set()方法 | 深入理解 | 高 | 高 |
| 8.4 继承和多态 | * 类的继承概念 * 类的多态概念：子类同名方法会覆盖父类的方法 | 深入理解 | 高 | 高 |
| 8.5 获取对象信息 | * type()获得对象类型 * isinstance()获得对象是否类的实例化对象 * dir()获得对象的属性和方法 * 获取属性的函数 | 识记 | 中 | 低 |
| 8.6实例属性和类属性 | * 理解实例属性概念 * 理解类属性概念 | 深入理解 | 高 | 高 |
| 8.7 使用\_\_slots\_\_ | * 给对象动态添加属性和方法 * 给类动态添加属性和方法 * 使用\_\_slots\_\_限定添加属性和方法 | 深入理解 | 高 | 高 |
| 8.8使用@property | * 使用@property将对象方法变成只读属性 * 使用setter将对象方法变成writable属性 | 深入理解 | 中 | 低 |
| 8.9 多重继承 | * 理解多重继承的概念 | 深入理解 | 中 | 低 |
| 8.10定制类（类的特殊方法） | * \_\_str\_\_() * \_\_iter\_\_() * \_\_getitem\_\_() * \_\_call\_\_()，理解对象callable概念 | 一般理解 | 低 | 低 |
| 8.11枚举类 | * 理解枚举类的特点 * 掌握枚举类的定义方式 | 一般理解 | 中 | 中 |
| 1. 输入输出I/O与文件 | 9.1 文件读写 | * 打开文件：open函数 * 读取文件：read方法 * 写入文件：write方法 | 一般理解 | 高 | 中 |
| 9.2操作文件和目录 | * 熟悉使用os模块进行文件和目录操作 | 一般理解 | 高 | 中 |
| 9.3序列化 | * 理解序列化和反序列化概念 * 使用Pickle模块进行序列化和反序列化 * 使用Json模块进行序列化和反序列化 | 一般理解 | 高 | 中 |
| 1. 错误与异常处理机制 | 10.1什么是错误和错误处理机制 | * 理解异常概念 * 理解异常处理机制 | 深入理解 | 高 | 中 |
| 10.2异常处理机制 | * 使用try…except…finally | 深入理解 | 高 | 高 |
| 10.3 调用栈 | * 了解异常处理中的栈机制 |  |  |  |
| 10.4 记录错误 | * 使用logging模块记录 | 一般理解 | 中 | 低 |
| 10.5主动raise error | * 理解主动raise error的概念和方法 | 一般理解 | 高 | 低 |
| 1. 模块与包 | 11.1 模块与包简介 | * 理解包和模块的概念定义，理解包和模块的区别 | 一般理解 | 高 | 低 |
| 11.2 使用标准库模块 | * 内部标准化模块的定义和调用方法 | 一般理解 | 高 | 低 |
| 11.3 安装第三方模块 | * 第三方模块的定义和调用方法 | 一般理解 | 高 | 低 |
| 11.4 自定义模块和使用 | * 自定义模块的定义和调用方法 | 一般理解 | 低 | 低 |
| 12. 正则表达式 | 12.1 正则表达式规则 | * 理解正则表达式基本概念 * 掌握正则表达式语法规则 | 深入理解 | 高 | 高 |
| 12.2 re模块 | * 熟练使用re模块完成正则表达式，重点掌握.math，.search方法 |  |  |  |
| 13. 半结构化数据处理 | 13.1半结构化数据简介 | * 半结构化数据简介：XML、JSON、HTML | 一般理解 | 高 | 高 |
| 13.2处理XML半结构化数据 |  | 不做要求 |  |  |
| 13.3处理JSON半结构化数据 |  | 不做要求 | 低 | 低 |
| 14. 数据库连接与访问 | 14.1 数据库概述 |  | 一般理解 | 高 | 低 |
| 14.2数据库连接包概述 |  | 不做要求 | 低 | 低 |
| 14.3数据库安装 |  | 不做要求 | 低 | 低 |
| 14.4数据库连接包安装 |  | 不做要求 | 低 | 低 |
| 14.5通过数据库连接包访问数据库 |  | 不做要求 | 低 | 低 |
| 15. 内置函数、类型与异常 | 15.1 内置函数 | * 理解内置函数的概念，并了解常用内置函数 | 识记 | 低 | 低 |
| 15.2 内置类型 | * 理解内置类型的概念，并了解常用内置类型 | 识记 | 低 | 低 |
| 15.3 内置异常类 | * 理解内置异常的概念，并了解常用内置异常 | 识记 | 低 | 低 |
| 16. 内置模块与标准库 | 16.1内置模块与标准库简介 | * 理解内置模块与标准库的概念 * 熟悉常见内置模块与标准库 | 一般理解 | 低 | 低 |
| 16.2 文本处理 | * re | 一般理解 | 低 | 低 |
| 16.3 数据类型 | * datatime * collections * copy | 一般理解 | 低 | 低 |
| 16.4 数值计算与数学模块 | * math * random * statistics | 一般理解 | 低 | 低 |
| 16.5 文件和目录 | * glob * shutil | 一般理解 | 低 | 低 |
| 16.6 数据持久化 | * pickle | 一般理解 | 低 | 低 |
| 16.7 操作系统服务 | * os | 一般理解 | 低 | 低 |
| 16.8 运行时服务 | * sys * warnings | 一般理解 | 低 | 低 |
| 17. 分析建模常用外部第三方包 | 17.1外部第三方包简介 | * 了解常用外部第三方包的分类以及常用第三方包 * 掌握外部第三方包的安装方式 | 识记 | 低 | 低 |
| 17.2数据处理和数据分析包 | * 掌握常用数据处理和数据分析包 | 识记 | 低 | 低 |
| 17.3机器学习与深度学习包 | * 掌握常用机器学习与深度学习包 | 识记 | 低 | 低 |
| 17.4量化金融与金融计算包 | * 掌握常用量化金融与金融计算包 | 识记 | 低 | 低 |
| 18. numpy基础 | * 1. numpy包简介 | * 了解numpy的功能 * 掌握numpy的安装 | 一般理解 | 低 | 低 |
| 18.2数组对象 | * 数组对象的结构 * 数组对象的创建和生成方式 * 生成简单的数组：全0、全1、单位矩阵、 空矩阵 * 使用arrange()和linspace()函数生成等差序列 * 使用genfromtxt函数读取外部数据文件 | 一般理解 | 低 | 低 |
| 18.3数组对象属性 | * 掌握常用数组对象属性 | 一般理解 | 低 | 低 |
| 18.4数组对象方法 | * 掌握常用数组对象方法 | 一般理解 | 低 | 低 |
| 18.5数组索引切片 | * 脚标索引 * 布尔索引 | 一般理解 | 低 | 低 |
| 18.6数组形状控制 | * 变形 * 拉直 * 转置 * 维度交换 * 去掉维度 * 拼接 | 一般理解 | 低 | 低 |
| 18.7数组基本运算 | * 数组的算术运算，运算符方式、模块函数方式、对象方法方式 * 数组的比较运算，运算符方式、模块函数方式、对象方法方式 | 一般理解 | 低 | 低 |
| 18.8数组广播机制 | * 理解数组广播机制 | 一般理解 | 低 | 低 |
| 18.9数组函数 | * 了解数组函数类别 * 了解各类别包含的函数 | 一般理解 | 低 | 低 |
| 19. pandas基础 | 19.1 pandas模块简介 | * 了解pandas的功能 * 掌握pandas的安装 | 一般理解 | 低 | 低 |
| 19.2 一维数据结构：Series对象 | * 掌握Series对象的结构 * 掌握Series对象的创建 | 一般理解 | 低 | 低 |
| 19.3 Series对象基础操作 | * 掌握数据文件的读取和写入 * 查看基本属性 * 数据描述统计 * 索引和切片 * 序列运算符 * 序列的串钩函数 | 一般理解 | 低 | 低 |
| 19.4 Series对象属性 | * 了解Series对象常用属性 | 一般理解 | 低 | 低 |
| 19.5 Series对象方法 | * 了解Series对象常用方法 | 一般理解 | 低 | 低 |
| 19.6 二维数据结构：DataFrame对象 | * 掌握DataFrame对象的结构 * 掌握DataFrame对象的创建 | 一般理解 | 低 | 低 |
| 19.7 DataFrame对象基础操作 | * 掌握数据文件的读取和写入 * 查看基本属性 * 数据描述统计 * 索引和切片 * 表间的合并和连接 * 聚合汇总操作 * 排序 * 数据透视表 | 一般理解 | 低 | 低 |
| 19.8 DataFrame对象属性 | * 了解DataFrame对象常用属性 | 一般理解 | 低 | 低 |
| 19.9 DataFrame对象方法 | * 了解DataFrame对象常用属性 | 一般理解 | 低 | 低 |
| 19.10 Pandas主要数据类型和对象 | * 了解Pandas主要数据类型和对象 * 了解index对象 * 了解window对象 * 了解groupby对象 | 一般理解 | 低 | 低 |
| 19.11 Pandas常用函数 | * 掌握Pandas常用函数 | 一般理解 | 低 | 低 |
| 19.12 Pandas数据可视化 | * 掌握Series对象绘图方法 * 掌握DataFrame对象绘图方法 | 一般理解 | 低 | 低 |
| 20. matplotlib基础 | 20.1 matplotlib模块简介 | * 了解matplotlib的功能 * 掌握matplotlib的安装 | 一般理解 | 低 | 低 |
| 20.2 绘图基本概念 | * 理解绘图基本概念和要素 | 一般理解 | 低 | 低 |
| 20.3 两种绘图方式 | * 绘图方式1：基于pyplot模块函数 * 绘图方式2：基于Axes对象方法 | 一般理解 | 低 | 低 |
| 20.4 常见图表绘制 | * 了解常用图表 * 饼状图pie * 柱状图bar和barh * 折线图plot * 散点图scatter * 气泡图scatter * 箱线图boxplot * 面积图stackplot * 统计直方图hist * 小提琴图violinplot | 一般理解 | 低 | 低 |
| 20.5 常见三维图表绘制 | * 了解常用三维图表 * 三维曲线图plot * 三维曲面图plot\_surface | 一般理解 | 低 | 低 |
| 20.6 绘图的修饰选项 | * 了解绘图的修饰选项和定义方法 | 一般理解 | 低 | 低 |
| 20.7 绘图的colarmap | * 了解绘图的colormap和定义方法 | 一般理解 | 低 | 低 |
| 21. statsmodels基础 | 21.1 statsmodels模块简介 | * 了解statsmodels的功能 * 掌握statsmodels的安装 | 一般理解 | 低 | 低 |
| 21.2 statsmodels模块的类和函数 | * 了解statsmodels模块的类和函数 | 一般理解 | 低 | 低 |
| 21.3 描述性统计与统计绘图 | * 掌握使用statsmodels进行描述性统计绘图 | 一般理解 | 低 | 低 |
| 21.4 线性回归模型与广义线性模型 | * 掌握使用statsmodels进行线性回归模型 | 一般理解 | 低 | 低 |
| 21.5 时间序列分析与状态空间模型 | * 掌握使用statsmodels进行时间序列分析 | 一般理解 | 低 | 低 |
| 22. scipy基础 | 22.1 scipy简介 | * 了解scipy的功能 * 掌握scipy的安装 | 一般理解 | 低 | 低 |
| 22.2 线性代数：scipy.linalg | * 掌握scipy.linalg进行常见矩阵操作的函数与类 * 矩阵基本运算 * 矩阵分解 | 一般理解 | 低 | 低 |
| 22.3 概率统计：scipy.stats | * 掌握scipy.stats进行统计的函数与类 * 常见统计分布 * 计算统计量 * 假设检验 | 一般理解 | 低 | 低 |
| 22.4 插值：scipy.interpolate | * 掌握scipy.interpolate进行插值操作的函数与类 * 一维插值 * 高维插值 | 一般理解 | 低 | 低 |
| 22.5 最优化：scipy.optimize | * 掌握scipy.optimize进行优化计算的函数与类 * 数据拟合 * 数值优化 * 方程求根 * 线性规划 | 一般理解 | 低 | 低 |
| 22.6 积分：scipy.integrate | * 掌握scipy.integrate进行数值积分的函数与类 * 函数积分 * 采样点积分 | 一般理解 | 低 | 低 |
| 22.7 稀疏矩阵：scipy.sparse | * 掌握scipy.sparse进行稀疏矩阵计算的函数与类，理解稀疏矩阵 * 掌握常用稀疏矩阵存储方式 | 一般理解 | 低 | 低 |

## 重要高频考点题目

* Python是解释性语言（脚本型语言）
* Python是动态语言（变量无需提前定义且随时可赋值）
* 单个元素元组的定义方式
* 通过集合实现列表去重
* 字符串+、字符串\*、列表+、列表\*的理解
* 运算符优先级
* break、continue循环控制，理解两者区别
* 理解生成器概念
* 函数的参数顺序
* 理解类属性和实例属性概念及区别
* 理解异常的概念
* 理解内置函数概念

# 历年真题与考试样题

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 题面 | 选项列 | | | | 正确答案 |
| A | B | C | D |
| Python 是一种跨平台、开源、免费的高级动态编程语言。 | 正确 | 错误 |  |  | A |
| Python 变量使用前必须先声明，并且一旦声明就不能再当前作用域内改变其类型。 | 正确 | 错误 |  |  | B |
| 已知 x = 3，那么赋值语句 x = 'abcedfg' 是无法正常执行的。 | 正确 | 错误 |  |  | B |
| 加法运算符可以用来连接字符串并生成新字符串。 | 正确 | 错误 |  |  | A |
| 如果仅仅是用于控制循环次数，那么使用 for i in range(20)和 for i in range(20, 40)的作用是等价的。 | 正确 | 错误 |  |  | A |
| python源程序执行的方式是（）。 | 编译执行 | 解析执行 | 直接执行 | 边编译边执行 | B |
| 使用pip安装pandas的命令是 | pip install pandas | pip upgrade pandas | pip put pandas | pip use pandas | A |
| 下列哪个语句在Python中是非法的？ | x = y = z = 1 | x = (y = z + 1) | x, y = y, x | x += y  x=x+y | B |
| Python语言语句块的标记是（）。 | 分号 | 逗号 | 缩进 | / | C |
| Python语句print('\x48\x41!')的运行结果是（）。 | x48\x41!' | 4841! | 4841 | HA! | D |
| 以下是字符转换成字节的方法是（）。 | decode() | encode() | upper() | rstrip() | B |
| print(100 - 25 \* 3 % 4)应该输出什么？ | 1 | 97 | 25 | 0 | B |
| Python中用来告知解释器跳过当前循环中的剩余语句，然后继续进行下一轮循环，此关键词是（）。 | go | break | continue | next | C |
| 函数如下： def showNnumber(numbers): for n in numbers: print(n) 下面哪些在调用函数时会报错？ | showNnumber([2,4,5]) | showNnumber('abcesf') | showNnumber((12,4,5)) | showNnumber(3.4) | D |
| 调用以下函数返回的值是（）。 def myfun():  pass | 0 | 出错不能运行 | 空字符串 | None | D |
| 关于python类，说法错误的是（）。 | 类的实例方法必须创建对象后才可以调用 | 类的实例方法必须创建对象前才可以调用 | 类的类方法可以用对象和类名来调用 | 类的静态属性可以用类名和对象来调用 | B |
| Python的输出函数是（）。 | input() | print() | math() | turtle() | B |
| 有关异常说法正确的是（）。 | 程序中抛出异常终止程序 | 程序中抛出异常不一定终止程序 | 拼写错误会导致程序终止 | 缩进错误会导致程序终止 | B |
| 导入模块的方式错误的是（）。 | import mo | from mo import \* | import mo as m | import m from mo | D |
| Python语句d={1:'a',2:'b',3:'c'}; print(len(d))的运行结果是（）。 | 0 | 1 | 3 | 6 | C |
| 下面关于time库引用不正确的是（）。 | from time import strftime | from time import \* | import time | from \* import time | D |
| Python机器学习方向的第三方库是 | PIL | PyQt5 | TensorFlow | random | C |
| 计算numpy中元素个数的方法（）。 | np.sqrt() | np.size() | np.identity() | np.itemsize() | B |
| DataFrame对象的.ix[idx] 与 .ix[[idx]] 有什么区别？ | 没有区别 | .ix[idx] 的结果是Series，.ix[[idx]]的结果是DataFrame。 | .ix[idx] 的结果是DataFrame，.ix[[idx]]的结果是Series。 | 以上都不对 | B |
| matplotlib绘制饼图的方法或函数是 | pie | bar | scatter | boxplot | A |
| statsmodels包中，建立线性一般回归模型的函数是 | OLS | GLS | WLS | RollingOLS | A |
| 进行线性代数计算，可以使用的包包括scipy.linalg 与 numpy.linalg，更推荐使用哪一个？ | numpy.linalg | scipy.linalg |  |  | B |
| “ab”+”c”\*2 结果是（）。 | abc2 | abcabc | abcc | ababcc | C |
| 迭代输出序列时（如：列表）使用 for 比 while 更好？ | 错误，while 比 for 更好。 | 正确 | 错误，while 不能用于迭代系列。 | 错误，for 和 while 都不能用于迭代系列。 | B |
| 以下选项不是函数作用的是（）。 | 提高代码执行速度 | 复用代码 | 增强代码可读性 | 降低编程复杂度 | A |
| 对于生成器对象 x = (3 for i in range(5))，连续两次执行 list(x)的结果是一样的。 | 正确 | 错误 |  |  | B |
| 定义函数时，即使该函数不需要接收任何参数，也必须保留一对空的圆括号来表示这是一个函数。 | 正确 | 错误 |  |  | A |
| 在Python中定义类时，如果某个成员名称前有 2 个下划线则表示是私有成员。 | 正确 | 错误 |  |  | A |
| 对文件进行读写操作之后必须显式关闭文件以确保所有内容都得到保存。 | 正确 | 错误 |  |  | A |
| 在try...except...else 结构中，如果 try 块的语句引发了异常则会执行else 块中的代码。 | 正确 | 错误 |  |  | B |
| Python程序的\_\_name\_\_的作用是什么？ | 标识模块的名字的变量 | 查看变量名字 | 普通自定义变量 |  | A |
| 以下选项中说法不正确的是 | C语言是静态语言，Python语言是脚本语言 | 编译是将源代码转换成目标代码的过程 | 解释是将源代码逐条转换成目标代码同时逐条运行目标代码的过程 | 静态语言采用解释方式执行，脚本语言采用编译方式执行 | D |
| 关于Python语言的特点，以下选项描述正确的是 | Python语言不支持面向对象 | Python语言是解释型语言 | Python语言是编译型语言 | Python语言是非跨平台语言 | B |
| 以下不是python中的关键字（）。 | raise | with | import | final | D |
| Python语句 print(0xA + 0xB)的输出结果是（）。 | 0xA + 0xB | A + B | 0xA0xB | 21 | D |
| 以下关于Python的描述错误的是（）。 | Python 的语法类似 PHP | Python 可用于 Web 开发 | Python 是跨平台的 | Python 可用于数据抓取（爬虫） | A |
| 关于Python中的复数，下列说法错误的是（）。 | 表示复数的语法是real + image j | 实部和虚部都是浮点数 | 虚部必须后缀j，且必须是小写。 | 方法conjugate返回复数的共轭复数 | C |
| 关于字符串下列说法错误的是哪个？ | 字符应该视为长度为1的字符串 | 字符串以\0标志字符串的结束 | 既可以用单引号，也可以用双引号创建字符串 | 在三引号字符串中可以包含换行回车等特殊字符 | B |
| 执行mean([1,2,3])的结果是（）。 | 2 | NameError: name 'mean' is not defined | 2 | 以上都不对 | B |
| Python中==运算符比较两个对象的值，下列选项中哪一个是is比较对象的因素？ | id() | sum() | max() | min() | A |
| 执行 not 1 and 1的结果为（）。 | A True | B False | C 0 | D 1 | B |
| 在python中，字符串s = 'abc',那么执行表达式s+'d'之后，s的打印结果是什么？ | 'abc' | 'abcd' | 'abc+d' | 报错 | A |
| 以下哪个描述是正确的？ | break 语句用于终止当前循环。 | continue 语句用于跳过当前剩余要执行的代码，执行下一次循环。 | break 和 continue 语句通常与 if, if...else 和 if...elif...else 语句一起使用。 | 以上说法都是正确的。 | D |
| [i\*\*i for i in range(3)]执行的结果是？ | [0, 1, 4] | [1, 1, 4] | [1, 2, 3] | (1, 1, 4) | B |
| 下列哪种函式参数定义不合法？ | def myfunc(\*args): | def myfunc(arg1=1): | def myfunc(\*args, a=1): | def myfunc(a=1, \*\*args): | C |
| 阅读下面代码： lst=[1,2,3] lst.append(4)  代码中append是\_\_\_\_\_? | 属性 | 方法 | 函数 | 模块 | B |
| Python 语言中，以下表达式输出结果为11的选项是 | print(“1+1”) | print(1+1) | print(eval(“1+1”)) | print(eval(“1” + “1”)) | D |
| 以下哪个关键字是与 try 语句一起使用来处理异常的？ | catch | exception | catch(a) | except | D |
| 如果我们需要从 math 模块中输出 pi 常量，以下代码正确的是（）。 | print(math.pi) | print(pi) | from math import pi print(pi) | from math import pi print(math.pi) | C |
| 以下哪个代码是将字符串转换为浮点数？ | int(x [,base]) | long(x [,base] ) | float(x) | str(x) | C |
| 关于random.uniform(a,b)的作用描述，以下选项中正确的是 | 生成一个[a, b]之间的随机小数 | 生成一个均值为a，方差为b的正态分布 | 生成一个(a, b)之间的随机数 | 生成一个[a, b]之间的随机整数 | A |
| 以下选项中是 Python 中文分词的第三方库的是 | jieba | itchat | time | turtle | A |
| Numpy中创建全为0的矩阵使用（）。 | zeros | ones | empty | arange | A |
| 如何删除DataFrame对象df中的空行？ | df.dropna(axis=0,how='all') | df.dropna(axis=0,how='any') | df.dropna(axis=0) | 以上都不对 | A |
| matplotlib绘制柱状图的方法或函数是 | pie | bar | scatter | boxplot | B |
| 尽管可以使用 import 语句一次导入任意多个标准库或扩展库，但是仍建议每次只导入一个标准库或扩展库。 | 正确 | 错误 |  |  | A |
| 正则表达式模块 re 的 match()方法是从字符串的开始匹配特定模式，而 search()方法是在整个字符串中寻找模式，这两个方法如果匹配成功则返回 match 对象，匹配失败则返回空值 None。 | 正确 | 错误 |  |  | A |
| Python 只能使用内置数据库 SQLite，无法访问 MS SQLServer、ACCESS 或 Oracle、MySQL 等数据库。 | 正确 | 错误 |  |  | B |
| 使用内置函数open()且以”w”模式打开的文件，文件指针默认指向文件尾。 | 正确 | 错误 |  |  | B |
| 只有Python扩展库才需要导入以后才能使用其中的对象，Python 标准库不需要导入即可使用其中的所有对象和方法。 | 正确 | 错误 |  |  | B |
| 以下哪项python能正常运行？ | 拼写错误 | 错误表达式 | 缩进错误 | 手动抛出异常 | D |
| 关于Python内存管理，下列说法错误的是（）。 | 变量不必事先声明 | 变量无须先创建和赋值而直接使用 | 变量无须指定类型 | 可以使用del释放资源 | B |
| 下列字符中对应ASCII码数值最小的是哪个选项？ | b | B | a | M | B |
| 在python中的占位符中，请你选出不属于占位符的选项（）。 | %d | %e | %E | %z | D |
| 在 if...elif...else 的多个语句块中只会执行一个语句块？ | 正确 | 错误 | 根据条件决定。 | Pyhton 中没有 elif 语句。 | A |
| 递归函数有什么特点？ | 函数名称作为返回值 | 包含一个循环结构 | 函数内部包含对本函数的再次调用 | 函数比较复杂 | C |
| 给定对象，使用哪个函数查看该对象的方法和属性？ | type() | id() | dir() | isinstance() | C |
| 以下选项中不是Python对文件的写操作方法的是 | writelines | write | writetext | write 和 seek | C |
| 关于try-except，哪个选项的描述是错误的？ | 表达了一种分支结构的特点 | 用于对程序的异常进行捕捉和处理 | NameError是一种异常类型 | 使用了异常处理，程序将不会再出错。 | D |
| 以下关于模块说法错误的是（）。 | 一个xx.py就是一个模块 | 任何一个普通的xx.py文件可以作为模块导入 | 模块文件的扩展名不一定是 .py | 运行时会从指定的目录搜索导入的模块，如果没有，会报错异常。 | C |
| Python语句nums=set([1,2,2,3,3,3,4]);print(len(nums))的输出结果是（）。 | 1 | 2 | 4 | 7 | C |
| 函数表达式 all([1,True,True]) 的结果是： | 无输出 | FALSE | 出错 | TRUE | D |
| 表达式 divmod(20,3) 的结果是 | 6, 2 | 6 | 2 | (6, 2) | D |
| 执行后可以查看Python的版本的是 | import sys print(sys.version) | import system print(system.version) | import system print(system.Version) | import sys print(sys.Version) | A |
| random库的seed(a)函数的作用是 | 生成一个[0.0, 1.0)之间的随机小数 | 生成一个k比特长度的随机整数 | 设置初始化随机数种子a | 生成一个随机整数 | C |
| 关于time库的描述，以下选项中错误的是 | time库提供获取系统时间并格式化输出功能 | time.sleep(s)的作用是休眠s秒 | time.perf\_counter()返回一个固定的时间计数值 | time库是Python中处理时间的标准库 | C |
| 以下选项中使 Python 脚本程序转变为可执行程序的第三方库的是 | pygame | PyQt5 | PyInstaller | random | C |
| Python数据分析方向的第三方库是 | pdfminer | beautifulsoup4 | time | numpy | D |
| 以下选项中，不是Python中用于进行数据分析及可视化处理的第三方库是 | Pandas | mayavi2 | Mxnet | numpy | C |
| arr1=np.array([1,2,3]) arr1[[0]] 输出结果为？ | 1 | array([1]) | [1] | 代码有错误 | B |
| 对DataFrame对象df，执行df.sort\_values('A', ascending=False)后，df有什么变化？ | 按A这一列升序排列 | 按A这一列降序排列 | 没有变化 | 以上都不对 | C |
| df.tail()这个函数是用来（）。 | 用来创建数据 | 用来展现数据 | 用来分析数据 | 以上都不对 | B |
| 以下哪个选项可用于在Pandas中创建DataFrame？ | scalar | ndarray | dict | 以上都正确 | D |
| matplotlib绘制散点图的方法或函数是 | pie | bar | scatter | boxplot | C |
| statsmodels包中，使用OLS建立线性回归模型，查看模型统计结果使用（包含统计检验结果）哪个方法 | fit() | summary() | params属性 | rsquared属性 | B |