**Python程序设计**

**前导课程： 计算机基础，若学过其他高级语言更好**

**课程地位：五星**

**教学环节：上课+上机 （32+32）**

**教学及进度计划：见课程计划+课时安排**

**学习方法：**

**1、学习基础知识**

**2、多读+多写**

**3、上机实践**

**参考书：**

**3、Python3 基础语法 | 菜鸟教程 https://www.runoob.com/python3/python3-basic-syntax.html?tdsourcetag=s\_pcqq\_aiomsg**

1. **Python入门**

**一、简介：（前世今生）**

**1、谁开发？啥时间？**

**2、本语言在发展过程中地位如何演变？（应用需求+时代特征）**

**3、应用范围？**

**4、本语言特点？ 优势 + 缺陷**

**5、语言本身的演变？ 2.x ---- 3.0up**

**二、平台搭建：3.0up （自主选择）**

**1、常用平台：6个**

**2、下载与安装：**

**3、第三方库安装： 基础： pip install （numpy）**

**三、程序开发：**

**1、文件类型： .py .pyc**

**2、编码规范：**

**A、标识符（大小写敏感）与保留字**

**B、代码缩进（严格缩进）与语句块**

**C、一句多行、多句一行和空行（功能分隔）**

**D、注释添加（单行和多行）**

**E、其他：模块（模块对象）导入、字符编码及转换**

**四、基础对话语句：输入、输出**

**第二章python基础**

**一、基础数据类型**

**A、数字类型**

**1、整型 （1）正整数、负整数和0**

**（2）二进制、八进制、十进制、十六进制**

**Bin（）：其他置换成二进制**

**Oct（）：其他转换成8进制**

**Int（）：其他转换成10进制**

**Hex（）：其他转换成16进制**

**2、浮点型 （1）普通小数型或科学计数法**

**（2）运算可能和预期产生差异**

**3、布尔型 （1）true或false**

**（2）为真：非0数字、非空字符串、非空列表、**

**非空元祖，非空字典、非空集合。**

**为假：数字0或0.0、空字符串、空列表、**

**空元祖，空字典、空集合**

**4、复数型 z.real：获得实部；z.imag：获得虚部**

**5、数字类型转换 （1）int（x）: x转换为整数**

**（2）float（x）： x转为浮点数**

**（3）complex（x） 实部为x 虚部为0**

**complex（x,y） 实部 x 虚部 y**

**（4）bool（）：转为布尔型**

**6、分数和高精度型：**

**标准库模块Fraction对象和decimal对象分别支持**

**B、字符串型：**

**（1）定义：是有序的字符集合，存储基于文本的信息**

**\*python不支持字符类型，故单个字符也是字符串**

**（2）索引：**

**正 0 1 2 3 4 5 6 7**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** |

**逆 -8 -7 -6 -5 -4 -3 -2 -1**

**（3）转换： encode（）；decode（）；str（）；bytes（）**

**（4）创建： （a）赋值创建**

**（b）str（），repr（）**

**（5）访问： （a）字符串名**

**（b）字符串名【index】**

**（c）切片str【【start】：【end】：【step】】**

**（6）运算： +、\*、关系、包含、不包含**

**（7）字符串函数：**

**（a）查找**

**（b）替换**

**（c）拆分**

**（d）转换**

**（e）格式化**

**（f）其他函数**

**（8）转义字符**

**二、常量与变量**

**常量：通常采用大写变量名**

**变量：无需声明，直接赋值**

**（1）\*交换2变量的值 a，b=b，a**

**（2）内存管理和变量管理**

**三、运算符与表达式**

**（1）基础运算**

**1、算术 2、关系 3、赋值 4、逻辑 5、位运算**

**6、成员运算 7、身份运算**

**（2）运算符优先级**

**四、特殊内置函数**

**1、range（） p43 2、type（）和isinstance（） p44**

**3、eval（） 4、map（） 5、filter（）**

**6、zip（） 7、enumerate（）**

**五、程序调试**

**1、错误类型：（1）句法 （2）运行 （3）逻辑**

**2、调试**

**第三章 程序设计**

**一、概述**

**二、顺序程序设计**

**三、分支程序设计**

**1、单分支 if 表达式：**

**语句块**

**2、二分支 if 表达式：**

**语句块**

**else：**

**语句块**

**3、多分支及嵌套**

**四、循环**

**1、while循环： 通常搭配input（）使用**

**while 表达式：**

**语句块**

**【else：**

**Else的子块】**

**2、for循环： 通常搭配 range（）使用**

**for 变量 in 序列或迭代对象：**

**语句块**

**【else：**

**Else的子块】**

**3、break、continue、和 pass**

**4、循环的嵌套**

**第四章 组合数据**

**一、概述：类似于其他语言的数组，但类型多、功能强大**

**二、列表：最频繁、最灵活**

**1、list为有序集合，元素0个（空列表）或多个，可相同，其类型亦可不同。**

**2、索引：支持双向索引**

**正 0 1 2 3 4 5 6**

**【 A， b， c， d， e， f， g 】**

**逆 -7 -6 -5 -4 -3 -2 -1**

**3、创建**

**4、访问**

**5、复制和删除**

**6、运算**

**7、统计**

**8、查找与排序**

**三、元组**

**1、定义： 小括号（）逗号隔离元素集合，建立后无变动、**

**无排序、可索引**

**2、创建**

**3、访问**

**4、复制和删除**

**5、运算**

**6、统计**

**四、字典**

**1、定义：映射类型，用{ }标识，无序的“键：值”对集合；**

**键为不可变类型，值有多类型；**

**键是唯一，而值可以有多个；**

**字典的元素通过键访问和操作**

**2、创建**

**3、访问**

**4、复制和删除**

**五、集合**

**1、定义：小括号、逗号分隔、无序、不重复的元素集合**

**无索引、无切片操作，无键可获取集合元素的值**

**2、创建**

**3、访问**

**4、复制和删除**

**5、集合运算**

**6、集合统计**

**六、嵌套组合数据**

**七、案例：**

**1、查找： （1）顺序查找**

**（2）二分查找**

**2、排序： （1）冒泡排序**

**（2）选择排序**

**3、推导式：**

**4、列表作为堆栈和队列： （1）作为堆栈**

**（2）作为队列**

**第五章 函数。**

1. **定义与调用。**
2. **定义函数： def 函数名（参数表）：**

**“““函数说明”””**

**函数体**

**\*参数表中各参用逗号分隔**

**2、调用： 函数名（参数表）**

1. **参数传递。**
2. **特殊函数： （1）匿名函数**

**（2）递归函数**

**（3）嵌套函数**

1. **函数装饰器： （1）定义和调用（用来包装函数的函数）**

**def decorator（func）**

**pass**

**@decorator**

**def function（）：**

**pass**

**（2）带参数的装饰器**

1. **变量作用域： 全局变量和局部变量**

**第六章 面向对象程序设计**

1. **[概述](#概述)：**

**二、****[类与对象](#类定义)： 1、类定义 2、对象的创建与使用**

**三、[类的成员](#类的成员)： 1、成员类型 2、内置成员**

**四、****[类的方法](#类的方法：)： 1、类的方法类型 2、属性 3、特殊方法**

**五、****类的继承与多态： 1、类的继承 2、类的多态**

**概述：**

**A、面向对象和面向过程：**

**面向过程： （1）分析解决问题所需步骤 （2）通过实现对应函数完成上述步骤**

**（3）使用时调用函数完成**

**优点： 性能优于面向对象 缺陷： 难以维护、复用和扩展**

**面向对象： （1）把问题先抽象 （2）以对象为中心封装成不同的类**

**（3）实例化具体对象 以此来描述整个解决问题的行为**

**优点：易于维护、复用和扩展 缺陷：性能劣于面向过程**

**B、基本概念：**

**对象、类、消息、封装、继承和多态**

**C、python在支持面向过程的同时也支持面向对象**

[**返回**](#概述：)

1. **类定义： class 类名：**

**“““类说明”””**

**类体**

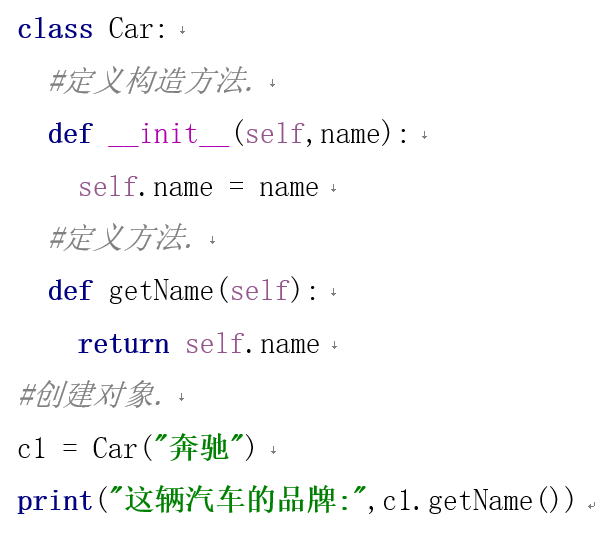
**（1）类头+类体 组成 类头以class为关键字，后接类名，类名遵标识符，首字母大写**

**（2）类体：数据成员（对象特征）+成员方法（对成员进行的操作）组成**

**2、对象的创建与使用**

**使用赋值方式创建类的对象，格式为： 对象名 = 类名（[参数列表]）**

**Ex6.1 创建一个类的对象，调用类中的方法**

**** [**返回**](#概述：)

**类的成员：**

**1、成员类型： （1）公有成员：类的内外均可访问**

**（2）私有成员：只能在类的内部访问，以单下划线或双下划开头，单下划线类和派生类可**

**以访问，而2个以上下划线开头的却不以2个或以上下划线结束的，只有自己访问。**

**Ex6.2 创建及使用类的公有和私有成员：**

*#定义类.***class** Woman:  
 **def** \_\_init\_\_(self,name,sex,age):  
 self.name = name *#定义公有成员.* self.\_sex = sex *#定义单下划线私有成员.* self.\_\_age = age *#定义双下划线私有成员.* **def** getAge(self):  
 **return** self.\_\_age  
*#创建对象.*w = Woman(**"小芳"**,**"Female"**,18)  
**print**(**"姓名: %s, 性别: %s, 年龄: %d."**%(w.name,w.\_sex,w.getAge()))

**2、内置成员：前后均有2个下划线的成员**

**类的常用内置成员： （1）\_\_name\_\_： 类的名字，字符串表示**

**（2）\_\_doc\_\_： 类的文档字符串**

**（3）\_\_bases\_\_： 所有父类组成的元组**

**（4）\_\_dict\_\_： 类的成员组成的字典**

**（5）\_\_module\_\_： 类所属模块** [**返回**](#概述：)

**类的方法：**

**1、类的方法类型： （1）公有方法：名字不用下划线开头，用类名或对象名调用**

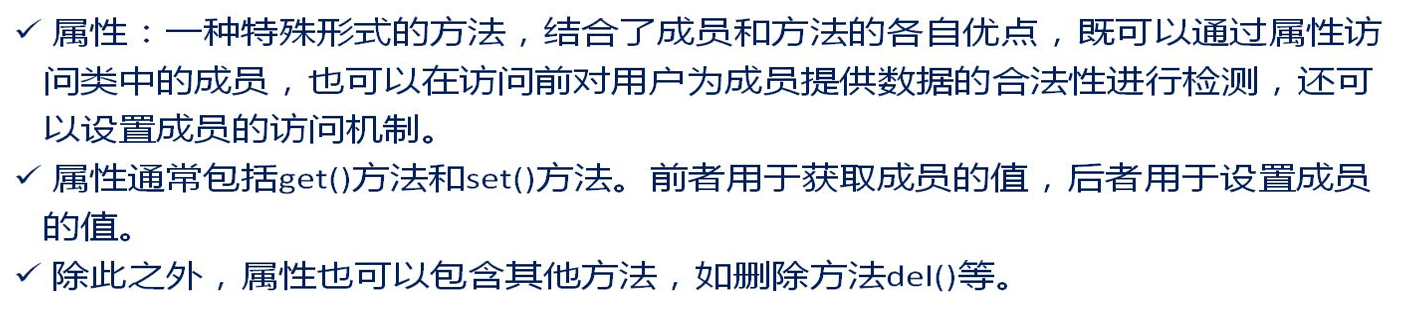
**（2）私有方法：2个或更多下划线开头，用self调用，不能在类外调用**

**（3）静态方法和类方法：不能直接访问属于对象的成员，类成员可以访问**

**Ex6.6 使用类的方法**

*#定义类.***class** A(object):  
 **def** function\_p(self): *#定义公有方法.* **print**(**"在公有方法中调用:"**,self.\_\_function()) *#调用私有方法.* **return "公有方法 'function\_p'"  
 def** \_\_function(self): *#定义私有方法.* **return "私有方法 '\_\_function'"** @classmethod  
 **def** function\_c(cls): *#定义类方法.* **return "类方法 'function\_c'"** @staticmethod  
 **def** function\_s(): *#定义静态方法.* **return "静态方法 'function\_s'"***#创建对象.*a1 = A()  
**print**(**"对象调用: "** + a1.function\_p())  
**print**(**"对象调用: "** + a1.function\_c())  
**print**(**"对象调用: "** + a1.function\_s())  
**print**(**"类名调用: "** + A.function\_p(a1)) *#传递对象a1作为参数.***print**(**"类名调用: "** + A.function\_c())  
**print**(**"类名调用: "** + A.function\_s())

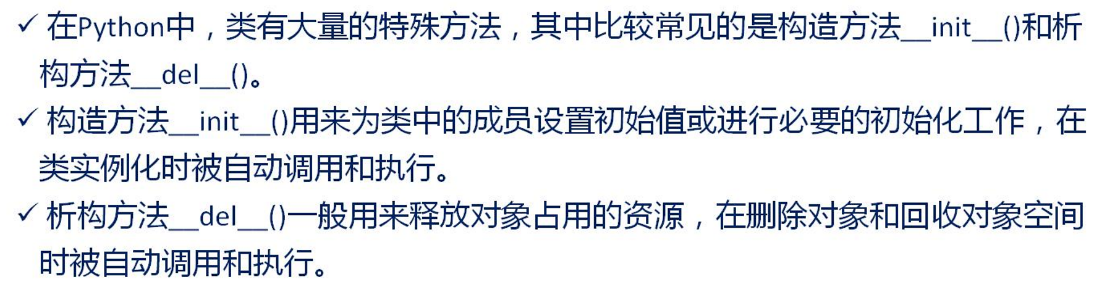
1. **属性**



**Ex6.7使用属性访问并检查私有成员值的合法性**

*#定义类.***class** Circle:  
 **def** set(self,radius):  
 **if** radius > 0:  
 self.\_\_radius = radius  
 **print**(**"圆的面积为: {0}."**.format(3.14 \* self.\_\_radius \*\* 2))  
 **else**:  
 **print**(**"半径 %f 不在规定范围内(>=0)，请重新设置!"**% radius)  
 **def** get(self):  
 **return** self.\_\_radius *#私有实例成员.  
#创建对象.*c = Circle()  
c.set(2.5)  
c.set(-2.5)

**3、特殊方法**

****

**Ex6.10构造方法和析构方法的使用**

**class** Rectangle(object):  
 *#定义构造方法.* **def** \_\_init\_\_(self,w,h):  
 self.w = w  
 self.h = h  
 **print**(**'执行构造方法...'**)  
 *#定义求面积方法.* **def** getArea(self):  
 **return** self.w \* self.h  
*#定义析构方法.* **def** \_\_del\_\_(self):  
 **print**(**'执行析构方法...'**)  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 rect = Rectangle(3,4) *#创建对象，调用构造方法\_\_init\_\_().* **print**(**"面积为:"**,rect.getArea())  
 **del** rect *#删除对象，调用析构方法\_\_del\_\_().*

[**返回**](#概述：)

**第七章 模块、包与库。**

**一、概述。**

**二、标准库模块。**

**1、turtle模块：绘图模块，创建画笔、绘制图形**

**2、random模块：生成随机数**

**（1）random. Random（）：生成【0,1）间的随机浮点数。**

**（2）random. uniform（）：生成指定范围的随机浮点数。**

**random. uniform（a，b）**

**（3）random. randint（）：生成指定范围的整数。**

**random. randint（a，b）**

**（4）random. Randrange（）：指定范围、指定步长整数。**

**（5）random. choice（）：从序列队象中获随机元素。**

**（6）random. shuffle（）：打乱序列对象中的元素。**

**（7）random. sample（）：从指定序列对象中随机获取指定长度片段。**

**3、time和 datetime 模块**

**4、os模块：访问操作系统功能模块，可实现目录、文件操作；调用系统命令。**

**5、sys模块**

**6、timeit模块**

**7、zlib 模块**

**三、第三方库模块**

**1、numpy 库：支持高维数组运算、大型矩阵处理、矢量运算等**

**2、pandas库：用以解决数据分析任务，提供搞笑操作大型数据所需工具。**

**3、scipy库：转为科学和工程设计的工具库，包含统计、优化、整合、线性代数、傅里叶变换、信号和图像处理、常微分方程求解等。**

**4、matplotlib库：基于python、跨平台、交互式2D绘图库。**

**5、jieba 库：**

**6、pyinstaller 库：打包python应用程序。**

**四、自定义模块。**

**第八章 正则表达式**

1. **概述**
2. **正则表达式语法**
3. **处理字符串**
4. **re模块中的常用函数**
5. **常用正则表达式**

**第九章 文件访问**

1. **概述，**
2. **文本文件访问**
3. **打开文件： open（path\filename，mode）**

**File.close（）**

**With open（path\filename）**

1. **文件操作：**

**（1）写文件：file.write（）或file.writelines（）**

1. **二进制文件访问**
2. **用pickle模块读写**
3. **用struct模块读写**
4. **用marchal模块读写**
5. **marchal.dump（）：数据对象序列化后写入二进制文件。**
6. **marchal.load（）：二进制文件内容反序列化为数据对象。**
7. **用shelve模块读写：以字典的方式操作二进制文件**

**Shelve.open（filename，mode，protocol=none，writeback=false）**

**\*四、目录操作。**

**第十章 异常处理和单元测试**

**一、异常类和异常处理**

**二、单元测试**

**第十一章 数据库访问**

1. **概述**
2. **关系数据库访问**
3. **非关系数据库访问**

**第十二章 图形用户界面编程**

1. **wxPython库**
2. **简介：wxpython：开源、跨平台、支持GUI的第三方库**
3. **安装：**
4. **建立GUI程序步骤：**

**（1）导入wx： import wx**

**（2）创建wx.app： app=wx.app（）**

**（3）创建顶层窗体： frame=wx.frame（none）**

**（4）显示窗体： frame.show（true）**

**（5）进入循环，开始处理： app.mainloop（）**

**二、事件处理**

**1、事件处理是其工作的基本机制，相関概念：**

**2、事件绑定：**

**Self.bind（eventbinder，handler，sourcectrl）**

**或**

**Self.sourcectrl.bind（eventbinder.handler）**

**三、常用控件**

**1、窗体：wx.frame.\_init\_(parent,id,title,pos,size,style)**

**2、按钮、标签和文本框**

**按钮：wx.button(parent,id,lable,pos,size,style)**

**标签：wx.statictext(parent,id,lable,pos,size,style)**

**文本框：：wx.textctrl(parent,id,lable,pos,size,style)**

1. **单选按钮、复选框**
2. **列表、组合框**
3. **菜单**
4. **菜单栏**
5. **菜单**
6. **菜单项**
7. **工具栏、状态栏**
8. **对话框**

**四、布局**

**第十三章 多进程与多线程**

1. **多进程**
2. **多线程**

**第十四章 网络程序设计**

1. **网络协议**
2. **套接字编程**
3. **Web编程**