

面向过程和面向对象(重点)

- 1 面向过程:代码顺序执行、模块之前尽可能的独立
- 2 面向对象:根据不同的需求执行代码
- 3 注意:面向对象只是一种思想,并不是一门编程语言

类的定义(重点)

```
1 1.类是什么?
2 类是一种自定义的数据类型,这个数据类型定义的变量就叫对象
3
4 2.类有什么用?
5 可以将世间万物抽象成代码,然后通过代码来解决问题或需求
6 3.怎么抽象?
7 任务事务或物品都有属性和方法,那么类就是把属性和方法包含起来
8 属性(用变量表示),方法(用户函数表示)
9
10 4.类的定义
11 格式:
12 class Maker():
13     类体
14
15 说明:
16     1.python中定义 使用class关键字
17     2.类名只要是合法的标识符就可以,一般会使用大驼峰命名法(Maker),小驼峰(maker)
18     3.类体是属性(变量)和方法(函数)
19
20 注意:
21     1.类中的成员方法区别于普通方法:参数部分一定包含self,而且最好self出现在参数列表的第一个
22     2.调用函数的时候,self不需要被传参
23
24 #!/usr/bin/env python
25 # -*- coding:utf-8 -*-
26 #====#====#====#====
27 #Author:
28 #CreatDate:
29 #Version:
30 #====#====#====#====
31
32 #定义一个类,抽象人
33 class Maker():
34     #属性
35     name="张三",
36     age=18,
37     sex="男"
38     #方法
39     def myfunc(self):
40         print("是个海王")
```

```

41
42
43 #定义一个类,抽象汽车
44 class Car():
45     name="BMW",
46     color="白色",
47     type="7系",
48
49     def myfunc(self):
50         print("可以装")
51
52     def myfunc2(self):
53         print("可以代步")
54
55
56
57 #案例:抽象一个汽车,写成类
58
59 #案例:把电脑抽象出来,写成一个类
60
61

```

对象的创建(重点)

```

1  1. 对象的创建
2      格式:对象名=类名()
3      创建对象和创建类可以在同一个.py文件,也可以不在同一个.py文件
4
5      案例:自己写个类,在本文件中创建对象
6
7  2. 调用属性
8      对象名.属性名
9
10     #案例:定义一个学生类,类属性有name,age,然后通过对象修改类属性的值,然后给对象添加属性score,并打印
11
12  3. 调用方法
13     对象名.方法名()
14
15     #!/usr/bin/env python
16     # -*- coding:utf-8 -*-
17     #====#====#====#====
18     #Author:
19     #CreatDate:
20     #Version:
21     #====#====#====#====
22     import MyMaker
23
24     class Maker2():
25         name="李四",
26

```

```

27     def myfunc(self):
28         print("我是Maker2的函数")
29
30 #创建对象
31 m=Maker2()
32 #通过对象获取类里的成员变量,成员函数
33 print(m.name)
34 m.myfunc()
35
36 c=MyMaker.Car()
37 c.myfunc()
38
39
40 案例:定义一个学生类。有下面的
41 属性:
42 1 姓名
43 2 年龄
44 3 成绩(语文,数学,英语)[每课成绩的类型为整数]
45 类方法:
46 1 获取学生的姓名:get_name() 返回类型:str
47 2 获取学生的年龄:get_age() 返回类型:int
48 3 返回3门科目中最高分数的课程。get_course()
49 4 返回该学生的平均成绩get_avg()
50 #!/usr/bin/env python
51 # -*- coding:utf-8 -*-
52 #====#====#====#====
53 #Author:
54 #CreatDate:
55 #Version:
56 #====#====#====#====
57 import MyMaker
58
59 class Maker2():
60     name="李四",
61
62     def myfunc(self):
63         print("我是Maker2的函数")
64
65 #创建对象
66 m=Maker2()
67 #通过对象获取类里的成员变量,成员函数
68 print(m.name)
69 m.myfunc()
70
71 c=MyMaker.Car()
72 c.myfunc()
73
74 print("-----")
75
76 class Student():
77     '''
78     属性:
79     1 姓名

```

```

80 2 年龄
81 3 成绩（语文，数学，英语）[每课成绩的类型为整数]
82 类方法：
83 1 获取学生的姓名：get_name() 返回类型:str
84 2 获取学生的年龄：get_age() 返回类型:int
85 3 返回3门科目中最高分数的课程。get_course()
86 4 返回该学生的平均成绩get_avg()
87     '''
88     name="海王",
89     age=18,
90     score=[88,72,90]
91
92     def get_name(self):
93         return self.name
94
95     def get_age(self):
96         return self.age
97
98     def get_course(self):
99         return max(self.score)
100
101     def get_avg(self):
102         return sum(self.score)/3
103
104 #创建对象
105 s=Student()
106 name=s.get_name()
107 print(name[0])
108 print(s.get_age())
109 print(s.get_course())
110 print(s.get_avg())
111
112
113 名称说明：
114 类中定义的变量：类属性,类变量
115 对象自身定义的变量：对象属性,实例属性,实例变量
116 成员变量包含类变量和实例变量
117 全局变量和局部变量是相对于作用域来说的。
118
119

```

self介绍(重点,难点)

```

1 1.self,是对象本身,哪个对象去调用成员函数,那么成员函数中的self就是这个对象
2 2.传递是python去做的,不是程序员去做的
3 3.作用:
4 #!/usr/bin/env python
5 # -*- coding:utf-8 -*-
6 #====#====#====#====
7 #Author:
8 #CreatDate:
9 #Version:

```

```

10 #====#====#====#====
11
12 class Maker():
13     name="张三",
14
15     def mytest(self):
16         self.age=20#1.在成员函数内定义实例对象
17         print(self.age)
18
19     def mytest02(self,sex):
20         self.sex=sex#2.当局部变量(形参也是局部变量)和对象变量同名时,用self区分
21
22         print(self.sex)
23
24     def mytest03(self,name):
25         self.name=name#3.当局部变量和类变量同名时,用self区分
26         print(self.name)#当self点出的变量名和类变量同名时,就是类变量,不是变量
27
28     def mytest04(self):
29         print(self.age)#4.用self可以在各个成员函数中传递数据
30
31
32 m=Maker()
33 m.mytest()
34 m.mytest02("男")
35 m.mytest03("李四")
36 m.mytest04()
37 #案例:有对象属性(姓名,年龄,分数),设计一个类,类中有修改对象属性值的方法,打印对象属性的方法
38 #其中分数属性在修改函数内定义,修改年龄时要判断是否合法,如果不合法打印不合法

```

构造函数(重点,难点)

```

1  1.构造函数是给对象的实例变量赋初始值
2  2.当一个对象被创建的时候,第一个被自动调用的函数
3  3.语法:
4  __init__(参数列表):
5      函数体
6  4.构造函数的参数在定义对象时传递,如:m=Maker(实参) ->实参传递到__init__(形参)
7  5.如果没有显示定义构造函数,那么系统默认提供一个无参的构造函数,这个无参的构造函数是空函数体
8  手动添加了有参的构造函数之后,系统将不再提供无参的构造函数
9  #!/usr/bin/env python
10 # -*- coding:utf-8 -*-
11 #====#====#====#====
12 #Author:
13 #CreatDate:
14 #Version:
15 #====#====#====#====
16
17 #以下代码,如果不name和age正确的值,那么就会出现问
18 # class Maker():

```

```

19 #     name="",
20 #     age=-1,
21 #
22 #     def getAge(self):
23 #         return self.age
24 #
25 #     def getName(self):
26 #         return self.name
27 #
28 # m=Maker()
29 # print(m.getAge())
30 # print(m.getName())
31
32 class Maker():
33     name="",
34     age=-1,
35
36     #构造函数,自动调用
37     def __init__(self,name,age):
38         print("创建对象的时候调用我")
39         if len(name)>=1:
40             self.name=name
41         else:
42             print("姓名未赋合法值")
43
44         if age>=0 and age<=150:
45             self.age=age
46         else:
47             print("年龄不合法")
48
49     def getAge(self):
50         return self.age
51
52     def getName(self):
53         return self.name
54
55 m=Maker("小甜甜",2)
56 print(m.getAge())
57 print(m.getName())
58
59
60 案例:
61 声明一个Dog类,
62 1)拥有属性:名字、年龄、性别和品种;
63
64 要求创建对象的时候名字、品种必须赋值,
65
66 年龄和性别可以赋值也可以不赋值
67
68 2)拥有两个对象方法,一个用来打印狗的信息(显示信息的方法)
69
70 一个叫唤的方法:以'xxx:汪汪汪汪'的格式打印叫唤信息
71

```

析构函数

```
1  1.当删除一个对象时,python解释器也会调用一个函数,做清理工作,这个函数就叫析构函数
2  2.格式:
3  __del__(self):
4  3.time模块中的sleep(时间)可以暂停代码运行
5  4.删除对象用del
6  5.当1个变量保存了对象的引用,此时对象引用计数器就加1,
7  当使用del删除变量指向的对象时,引用计数器就减1,当减到0时,就调用析构函数
8  6.作用:通常是销毁/删除临时的变量,主要对那些长期占用内存的临时变量进行销毁.有些程序是不结束的或运行
   很久
9  #!/usr/bin/env python
10 # -*- coding:utf-8 -*-
11 #====#====#====#====
12 #Author:
13 #CreateDate:
14 #Version:
15 #====#====#====#====
16 import time
17 class Maker():
18     def __del__(self):
19         print("对象完蛋,我完蛋")
20
21 m=Maker()
22 m2=m
23 del m#删除对象
24 del m2
25 time.sleep(10)#代码运行到这,会暂停10秒
26 #1.程序结束,那么这个程序中unix都会被删除
27
28
29
30 #案例:写一个运行后10秒结束的程序,程序中定义两个对象,当其中一个对象删除时,另一个对象也要被一起删除
```