

## Day 06 作业

### 简答题

#### 1. 描述类的组成部分?

1, 类名 2, 属性 3, 方法

#### 2. 书写定义类和创建对象的语法?

```
class 类名:  
    def 方法名(self):  
        pass
```

变量 = 类名()

#### 3. 简述对 self 的理解?

`self` 是一个形参,不需要手动传递实参值,python 解释器会自动将调用该方法的对象作为实参值进行传递  
`self` 就是对象自己

#### 4. 简述对 `__init__` 方法的理解

调用时机: 创建对象之后会自动调用

应用: 给对象添加属性的

注意点: 如果存在 `self` 之外的形参, 在创建对象的时候, 必须传递实参值

#### 5. 简述对 `__str__` 方法的理解

`print(对象)` 的时候会自动调用, 必须返回一个字符串

定义(使用): 我们自己想要使用 `print` 来打印对象, 显示对象的属性信息

### 代码题

#### 题目1

定义一个学生类(Student):

1. 包含属性 姓名 `name`, 年龄 `age`.

2. 包含方法:

1. 吃饭的方法 `eat`, 在方法中输出 `xx 要吃饭`, `xx` 为学生具体的名字

2. 睡觉的方法 `sleep`，在方法输出 `xx 要睡觉`，xx 为学生具体的名字
  3. 过年的方法 `year`，要求，年龄增加一岁
3. 打印对象的时候，输出 学生的 姓名和年龄信息格式如下
- `姓名：xxx，年龄：xx 岁`，xx 为具体的名字和年龄
4. 创建两个对象，并分别调用 吃饭和睡觉和过年的方法
- 小明 18 岁
  - 小红 17 岁

```
"""
类名：学生类 Student
属性：姓名 name，年龄 age
方法：吃饭 eat 睡觉 sleep 过年 year 打印对象信息 __str__ 添加属性 __init__
"""
class Student:
    def __init__(self, name, age):
        self.name = name # 姓名
        self.age = age # 年龄

    def __str__(self):
        return f'姓名：{self.name}，年龄：{self.age} 岁'

    def eat(self):
        print(f'{self.name} 要吃饭')

    def sleep(self):
        print(f'{self.name} 要睡觉')

    def year(self):
        self.age += 1 # 修改属性值

xm = Student('小明', 18)
print(xm)
xm.eat()
xm.sleep()
xm.year()
print(xm)
xh = Student('小红', 17)
xh.eat()
xh.sleep()
xh.year()
print(xh)
```

## 题目 2

定义一个水果类，包含 名称、颜色和价格属性，定义展示水果信息的方法 `show`，打印信息的格式：`水果名称：苹果，颜色：红色，价格：3.5`。然后通过水果类创建苹果对象、西瓜对象，并调用展示水果信息的方法

```

class Fruit:
    def __init__(self, name, color, price):
        self.name = name # 名称
        self.color = color # 颜色
        self.price = price # 价格

    def show(self):
        print(f"水果名称: {self.name}, 颜色: {self.color}, 价格: {self.price}")

if __name__ == '__main__':
    apple = Fruit('苹果', '红色', 3.5)
    apple.show()

```

### 题目 3

定义一个电脑类(computer),

电脑有品牌(brand),有价格(price),能播放电影(play\_movie)。

分别创建2个对象"小米电脑" `mi` 和 "苹果电脑" `mac`。分别调用放电影的动作, 输出内容格式如下: `xx 播放电影 oo`, xx 为 电脑品牌, oo 为电影的名字, 电影名字作为参数传递即可

- 小米电脑播放 `葫芦娃`
- 苹果电脑 播放 `变形金刚`

```

"""
类名: 电脑类 Computer
属性: 品牌 brand  价格 price
方法: 放电影 play_movie
"""

class Computer:
    def __init__(self, brand, price):
        """初始化方法"""
        self.brand = brand # 品牌
        self.price = price # 价格

    def play_movie(self, movie_name):
        """播放电影的方法"""
        print(f'{self.brand} 电脑在播放 {movie_name}')

# 使用类模板创建对象
mi = Computer("小米", 5000)
mi.play_movie("葫芦娃")

apple = Computer("苹果", 8000)

```

```
apple.play_movie("变形金刚")
```

## 题目 4 课上代码 - 摆放家具

```
class HouseItem:
    """家具类"""
    def __init__(self, name, area):
        self.name = name # 家具名字
        self.area = area # 家具的占地面积

    def __str__(self):
        return f'{self.name} 占地面积为{self.area} 平米'

class House:
    """房子类"""
    def __init__(self, h_type, area):
        self.h_type = h_type # 户型
        self.total_area = area # 总面积
        self.free_area = area # 剩余面积和总面积相等
        self.item_list = [] # 新房子没有任何家具

    def __str__(self):
        return f"户型: {self.h_type},总面积:{self.total_area}平米, 剩余面积: {self.free_area} 平米, 家具名称列表: {self.item_list}"

    # 添加家具, 房子对象(self)添加家具对象(item)
    def add_item(self, item): # 将 item 作为家具对象使用
        """添加家具"""
        if self.free_area > item.area:
            # 添加家具对象
            self.item_list.append(item.name) # 只要有对象,就可以获取属性
            # 修改剩余面积 当前的剩余面积 - 家具的面积
            self.free_area -= item.area
            print(f'添加家具: {item.name} 成功')
        else:
            print('剩余面积不足,换个大房子吧....')

if __name__ == '__main__':
    # 创建 家具对象, 席梦思
    bed = HouseItem('席梦思', 4)
    # 创建 家具对象, 衣柜
    chest = HouseItem('衣柜', 2)
    # 创建 家具对象, 餐桌
    table = HouseItem('餐桌', 1.5)
    print(bed)
    print(chest)
    print(table)
    # 创建房子对象
    house = House('别墅', 1000)
    print(house)
```

```
# 添加席梦思
house.add_item('bed')
print(house)
# 添加餐桌
house.add_item('table')
print(house)
```