构造方法

1. __init__(self):
    初始化方法
2. 自动调用
    在创建实例的时候已经调用
3. 如果一个类中没有构造方法会自动父类的构造方法
4. 类中可以书写多个构造方法，但是只会执行最后一个
    方法名相同，覆盖掉前面的方法
5. 构造方法有返回值，但是只能返回None
6. 构造方法可以传递买多个参数
    在创建实例的时候传递参数

创建对象

1. 变量名=类名（构造参数）

2. 类名（构造参数）---也在创建对象----没有引用
    使用一次的时候创建
3. 创建出来的对象---实例----实例对象
    和实例相关的属性：实例属性
    和实例相关的方法：实例方法

组合

大对象是由小对象组成
大对象的属性也是一个对象
1. 如果一个类中的属性是一个对象，则称为组合

```
class Student:
    age=6

class Teacher:
    salary = 99999

class School:
    s=Student()
    t=Teacher()
    def getClass(self):
        print('上课！！！')


school=School()
print(school.s.age)
print(school.t.salary)
```

- 补充

```
1. 类属性和方法名同名，后执行的会把前面的覆盖掉
        class Dog:

            def run(self):
                print('狗狗会run')
            run='10km/h'

        d=Dog()
        print(d.run)
2. 如果实例属性和方法名发生命名冲突，实例属性创建后会覆盖掉同名的方法对象

        class Dog:

            def run(self):
                self.run='10km'
                print(self.run)

        d=Dog()
        d.run()

        d.run()
```

共有私有

```
1. 属性：共有属性
2. _属性：在属性名前加一个下划线
        私有化属性或方法，禁止通过from modules import *
3. __属性：在属性名前加两个下划线
        私有化属性或方法
        在外部无法访问
        class Dog:
            __name='旺财'
            def run(self):

                print(self.__name)
        d=Dog()
        d.run()
4. Python为了实现私有化，利用名字重造的 方式把名字进行了更改
        调用形式：
                实例对象._类名属性名/方法名

        class Dog:
            __name='旺财'
            def __run(self):

                print(self.__name)
        d=Dog()
        print(d._Dog__name)
        print(d._Dog__run())
```

5. 属性私有化+提供get set方法

```
class Card:
    __pwd=123

    def getPwd(self):
        return self.__pwd

    def setPwd(self,newPwd):
        self.__pwd=newPwd

c=Card()
c.setPwd(110)
print(c.getPwd())
```