运算相关的魔法方法

思考:

```
class Person:
  def __init__(self,name,age):
    self.name = name
    self.age = age

p1 = Person('zhangsan',18)
p2 = Person('zhangsan',18)
print(p1 == p2)
```

上述代码中，使用 `==` 运算符比较两个对象，结果是True还是False? `==` 到底比较的是什么?

比较运算符相关魔法方法

```
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __eq__(self, other):
        return self.name == other.name and self.age == other.age

    # def __ne__(self, other):

    def __lt__(self, other):
        return self.age < other.age

    # def __gt__(self, other):

    def __le__(self, other):
        return self.age <= other.age
    # def __ge__(self, other):


s1 = Student('zhangsan', 18)
s2 = Student('zhangsan', 18)
s3 = Student('lisi', 20)
print(s1 == s2)
print(s1 != s2)
print(s1 > s2)
print(s1 >= s2)
print(s1 <= s2)
print(s1 <= s2)
```

算数运算符相关魔法方法

```
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __add__(self, other):
        return self.age + other

    def __sub__(self, other):
        return self.age - other

    def __mul__(self, other):
        return self.age * other

    def __truediv__(self, other):
        return self.age / other

    def __mod__(self, other):
        return self.age % other

    def __pow__(self, power, modulo=None):
        return self.age ** power


s = Student('zhangsan', 18)
print(s + 1)  # 19
print(s - 2)  # 16
print(s * 2)  # 36
print(s / 5)  # 3.6
print(s % 5)  # 3
print(s ** 2)  # 324
```

类型转换相关魔法方法

```
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __int__(self):
        return self.age

    def __float__(self):
        return self.age * 1.0

    def __str__(self):
        return self.name

    def __bool__(self):
        return self.age > 18


s = Student('zhangsan', 18)
print(int(s))
print(float(s))
print(str(s))
print(bool(s))

if s:
    print('hello')
```