클래스

형식

```
class 클래스이름:
    def 메서드(self):
        코드
```

- 인스턴스 변수 : 메소드 안에 정의되는 변수
    - 클래스 내부에서는 self.변수명 과 같이 접근
    - 클래스 외부에서는 객체변수.인스턴스변수 와 같이 접근
- 인스턴스 메소드 : 클래스에 정의되는 메소드
    - 첫번째 파라미터에 반드시 self라는 파라미터를 지정해야 한다.
    - 이는 인스턴스 변수에 항상 액세스할 수 있도록 하기 위함이다.

생성자(initializer)

- 클래스로부터 객체를 만들때 인스턴스 변수를 초기화한다. init양쪽에 언더바를 2개씩 붙여 사용한다.

```
class 클래스이름:
    def __init__(self, 인자1, 인자2):
        인스턴스 변수 초기화 코드
```

정적메소드

- 클래스명으로 바로 접근할 수 있는 메소드를 의미한다.
- @staticmethod 데코레이터를 사용한다.

클래스메소드

- 정적메소드와 비슷하다.
- 객체 인스턴스를 의미하는 self대신 cls라는 클래스를 의미하는 파라미터를 전달받는다.
- cls를 통해 클래스변수를 엑세스할 수 있다.
- @classmethod 데코레이터를 사용한다.

정보은닉

멤버변수 혹은 멤버메소드 이름이 __(언더바 2개)로 시작하면 클래스 내부에서만 접근할 수 있는 private멤버가 된다.

예제] 14class.py

```
class FourCalculator:
    def setdata(self, first, second):
        self.first = first
        self.second = second
    def addition(self):
        result = self.first + self.second
        return result
    def subtraction(self):
        result = self.first - self.second
        return result
    def multiplication(self):
        result = self.first * self.second
        return result
    def division(self):
        result = self.first / self.second
        return result

a = FourCalculator()
b = FourCalculator()

a.setdata(4, 2)
b.setdata(3, 8)

print("객체a 덧셈", a.addition())
print("객체a 곱셈", a.multiplication())
print("객체b 뺄셈", b.subtraction())
print("객체b 나눗셈", b.division())

class CalculatorInit:
    count = 0
    '''
    def __init__(self):
        self.first = 1
        self.second = 2'''
    def __init__(self, first, second):
        self.first = first
        self.second = second
        CalculatorInit.count += 1
    def addition(self):
        result = self.first + self.second
        return result
```

```
    @staticmethod
    def staticArea(pFirst, pSecond):
        result = pFirst * pSecond
        print("static메소드", result)
    @classmethod
    def showInfo(cls):
        print('class메소드', cls.count)

```

```
#fCal = FourCalculatorInit() -> 에러발생
fCal = CalculatorInit(2010, 43)
print(fCal.addition())
fCal.showInfo()
CalculatorInit.staticArea(5, 8)

```

여기까지 작성하세요.

```
class moreCalulator(CalculatorInit):
    def pow(self):
        result = self.first ** self.second
        return result
    def addition(self):
        return (self.first + self.second) * 2

moreCal = moreCalulator(4, 3)
print("상속후", moreCal.pow())


p1 = CalculatorInit(100, 200)
p2 = moreCalulator(100, 200)
print("부모객체로호출", p1.addition())
print("자식객체로호출", p2.addition())
```

```
class Person:
    def __init__(self, n, a, pw):
        self.name = n
        self.age = a
        self.__passwd = pw
    def secret_info(self):
        return self.__passwd
```

```
p1 = Person('my', 22, 'qwer1234')
print("이름", p1.name)
print("나이", p1.age)
print("비밀번호1", p1.__passwd)
print("비밀번호2", p1.passwd)
print("비밀번호", p1.secret_info())
```