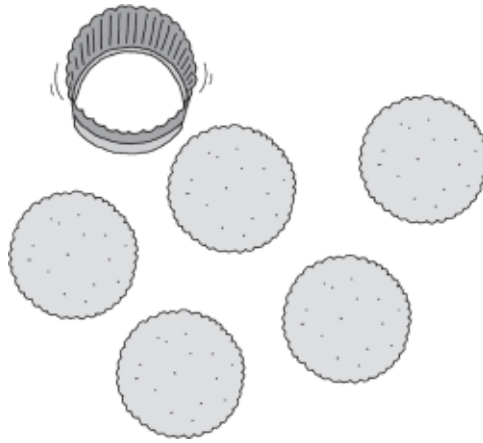


05-1 클래스



- 클래스 = 쿠키틀
- 객체 = 쿠키

→ 동일한 클래스로 만든 객체들은 서로 전혀 영향을 주지 X

사칙연산 클래스 만들기

```
class fourcal:
    def setdata(self, first, second):
        self.first = first
        self.second = second
    def add(self):
        result = self.first + self.second
        return result
    def mul(self):
        result = self.first * self.second
        return result
    def div(self):
        result = self.first / self.second
        return result
    def sub(self):
        result = self.first - self.second
        return result
a = fourcal()
a.setdata(4, 2)
print(a.add()) #6
print(a.mul()) #8
```

생성자(constructor)

객체에 초깃값을 설정해야할 때 지금까지는 a.setdata(x,y) 이런 식으로 setdata메서드를 호출하여 설정함

→ 생성자를 이용하여 초깃값을 설정하는 것이 안전한 방법

```
클래스에 추가로 __init__ 을 사용하면 됨
class fourcal:
    def __init__(self,first,second):
        self.first = first
        self.second = second
    def setdata(self,first,second):
        self.first = first
        self.second = second
    ...
a = fourcal(4,2)
print(a.add())
print(a.mul())

$ 따로 a.setdata(4,2)를 추가할 필요 없어짐
```

클래스 상속

어떤 클래스를 만들 때 다른 클래스의 기능을 물려받을 수 있게 만드는 것

앞에 만들었던 fourcal 클래스의 기능을 이어받아 Morefourcal 클래스를 만들려면

```
class Morefourcal(fourcal):
    pass

-> 일반화: class 클래스이름(상속할 클래스이름):
```

메서드 오버라이딩

부모 클래스(상속한 클래스)에 있는 메서드를 동일한 이름으로 다시 만드는 것

→ 오버라이딩한 메서드가 호출됨

```
class Safefourcal(fourcal):
    def div(self):
        if self.second == 0:
            return 0
        else:
            return self.first / self.second
```

```
a = SafeFourCal(4,0)
print(a.div()) #0
```

클래스 변수

클래스 안에 선언한 변수

```
class family:
    lastname = "강" #이게 클래스 변수
print(family.lastname) #강

또는

a = family()
print(a.lastname) #이렇게 family클래스로 만든 객체를 통해서도 클래스 변수 사용 가능
```

\$ 클래스 변수는 클래스로 만든 모든 객체에 공유됨