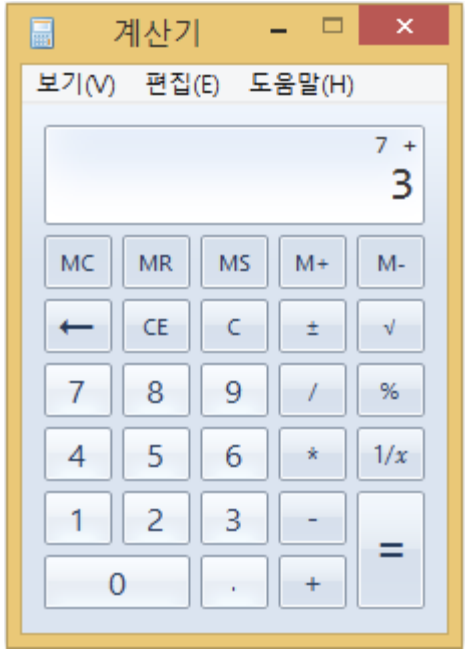
Object-Oriented Programming in Python

Created by 이 윤 준 (yoonjoon.lee@gmail.com)

May, 2019

Why Class?



계산기의 "더하기" 기능을 구현

```
result = 0

def add(num):
  global result
  result += num
  return result

print(add(3))
print(add(4))
```

한 프로그램에서 2개의 계산기가 필요한 상황

```
result1 = 0
result2 = 0

def add1(num):
  global result1
  result1 += num
  return result1

def add2(num):
  global result2
  result2 += num
  return result2

print(add1(3))
print(add1(4))
print(add2(3))
print(add2(7))
```

```
class Calculator:
  def __init__(self):
    self.result = 0

  def add(self, num):
    self.result += num
    return self.result

cal1 = Calculator()
cal2 = Calculator()

print(cal1.add(3))
print(cal1.add(4))
print(cal2.add(3))
print(cal2.add(7))
```

빼기 기능을 추가

```
class Calculator:
  def __init__(self):
    self.result = 0

  def add(self, num):
    self.result += num
    return self.result

  def sub(self, num):
    self.result -= num
    return self.result
```

Class & Object

```
>>> class Cookie:
>>>    pass
>>>
>>> a = Cookie()
>>> b = Cookie()
>>> type(a)
?
>>> type(Cookie)
?
```

객체와 인스턴스의 차이

- 클래스에 의해서 만들어진 객체는 인스턴스.
- 객체와 인스턴스의 차이는?
- `a = Cookie()`,
- a는 객체, 그리고 a라는 객체는 Cookie의 인스턴스.
- 인스턴스라는 특정 객체(a)가 어떤 클래스(Cookie)의 객체인지를 설명하는 관계.
- 즉, "a는 인스턴스" 보다는 "a는 객체"라는 표현이, "a는 Cookie의 객체" 보다는 "a는 Cookie의 인스턴스"라는 표현이 정확.

Class Design: Calculation

```
a = FourCal()
```

```
a = set_data(4, 2)
```

```
a.add()
```

```
a.sub()
```

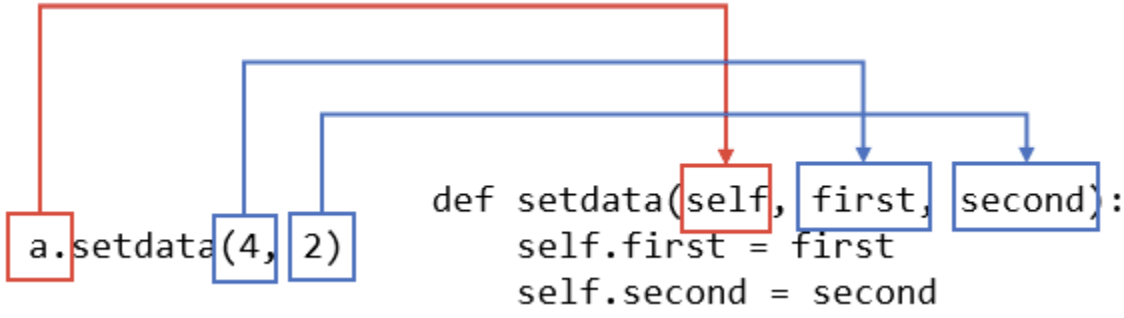
```
a.mul()
```

```
a.div()
```

```
class FourCal:

  def setdata(self, first, second):
    self.first = first
    self.second = second
```

```
a = FourCal()
a.setdata(4, 2)
```



```
a = FourCal()
FourCal.setdata(a, 4, 2)
```

```
a = FourCal()
a.setdata(4, 2)
```

Add an Operation

```
class FourCal:

  def setdata(self, first, second):
    self.first = first
    self.second = second

  def add(self):
    result = self.first + self.second
    return result
```

Constructor

```
class FourCal:

  def __init__(self, first, second):
    self.first = first
    self.second = second

  def setdata(self, first, second):
    self.first = first
    self.second = second
```

Inheritance

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

```
class MoreFourCal(FourCal):
  pass
```

FourCal 클래스에 ab (a의 b승)을 구할 수 있는 기능을 추가

```
class MoreFourCal(FourCal):

  def pow(self):
    result = self.first ** self.second
    return result
```

Override

```
class MoreFourCal(FourCal):

  def div(self):
    if self.second == 0:  # 나누는 값이 0인 경우 0을 리턴하도록 수
      return 0
    else:
      return self.first / self.second
```

Class Variable

```
class Family:

  lastname = "김"
```

```
a = Family()
b = Family()
print(a.lastname)
print(b.lastname)
```

```
Family.lastname = "박"
print(a.lastname)
print(b.lastname)
```