

01. 클래스

- class 클래스명: 으로 정의한다.
- 클래스는 하나의 틀이다.
- 클래스 안에는 여러가지 함수와 변수가 추가된다.
- 1개의 클래스는 무수히 많은 객체를 만들어 낼 수 있다.
- 클래스안에 구현된 함수를 우리는 다른말로 메서드(Method)라 부른다.

클래스 만들기

```
In [1]: class Cookie:  
        pass
```

객체 만들기

- 결과값을 돌려받은 a, b가 바로 객체이다.
- a는 객체
- a는 Cookie의 인스턴스라고 표현한다.

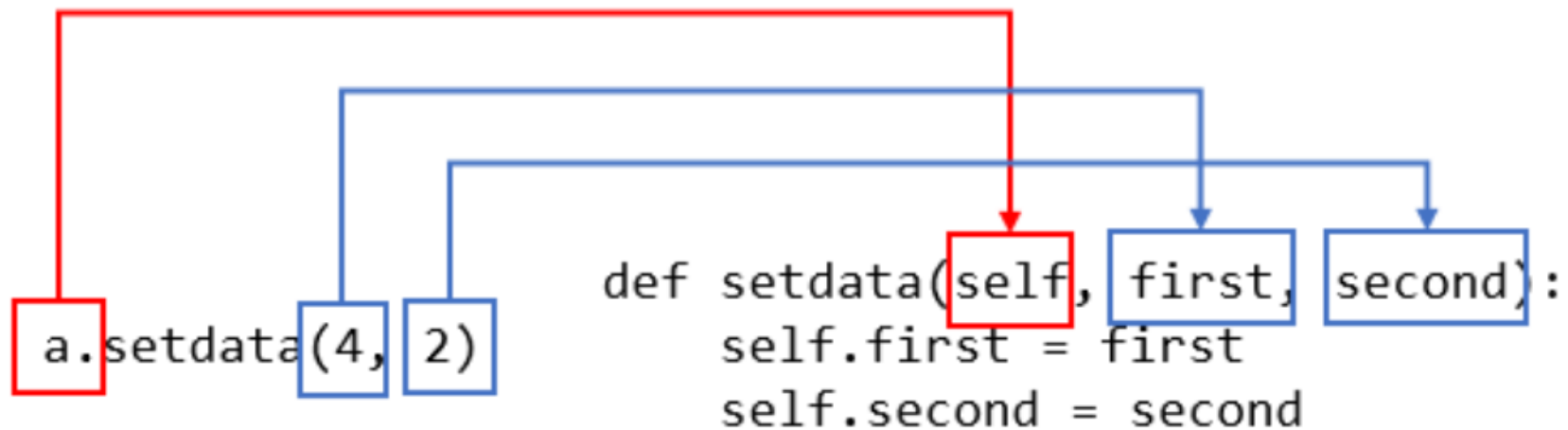
```
In [2]: a = Cookie()  
        b = Cookie()
```

객체들이 메서드에 값이 전달되는 구조

```
In [3]: class FourCal:
        def setdata(self, first, second):
            self.first = first
            self.second = second

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

클래스의 메서드를 호출시에는 `a.setdata(4,2)`와 같이 도트(.)연산자 사용



따라서 다음과 같이 생각해 볼 수 있다.

```

def setdata(self, first, second):
    self.first = first
    self.second = second
-----
--> a.setdata(4,2)
-----
def setdata(self, first, second):
    a.first = 4
    a.second = 2

```

클래스의 생성자

- 생성자(Constructor)란 객체가 생성될 때 자동으로 호출되는 메서드를 의미한다.
- 파이썬 메서드명으로 `__init__`을 사용하면 이 메서드는 생성자가 된다.

```

In [13]: class FourCal:
        def __init__(self, first, second):
            self.first = first
            self.second = second
        def sum(self):
            return self.first + self.second

```

클래스의 상속

- 상속이란 "물려받다"라는 뜻으로 "재산을 상속받다"라고 할때의 상속과 같은 의미이다.
- 클래스의 상속은 다음과 같이 하면 된다.
 - class 클래스명(상속할 클래스명)

```

In [14]: class MoreFourCal(FourCal):
        pass

```

```
In [15]: a = MoreFourCal(4,2)
a.sum()
```

Out[15]: 6

클래스의 변수

- 클래스의 변수는 **클래스명.클래스변수**로 사용가능하다.

```
In [16]: class Family:
        lastname="김"
```

```
In [17]: print(Family.lastname)
```

김

02. 모듈(Module)

- 모듈이란 함수나 변수 또는 클래스 등을 모아 놓은 파일이다.
- 다른 파이썬 프로그램에서 불러와 사용할 수 있게끔 만들어진 파일.
- 우리는 굉장히 많은 파이썬 모듈을 사용한다.

모듈 함수를 불러오기

```
from 모듈이름 import 모듈함수
from mod import sum # 하나의 함수만 사용
from mod import sum, safe_sum # 두개 이상의 함수를 사용
```

name == "main" 을 사용하는 이유

- c> python mod1.py처럼 직접 실행시에 **name == "__main__"** 참이 된다.

- 일반적으로 import 할 때는 거짓이 되어 다른 문장이 나타나지 않음.

```
(base) C:\Users\THJS\Documents\GitHub\PythonBasic\pythonBasic>python
Python 3.6.4 [Anaconda, Inc.] (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import mod
더할 수 없습니다.
None
5
20.4
>>> import modN
>>>
```

내가 만든 module를 다른 것과 동일하게 사용하기.

```
import sys
sys.path
sys.path.append("C:/module")
sys.path
import mod
```

```
(base) C:\>
(base) C:\>
(base) C:\>python
Python 3.6.4 [Anaconda, Inc.] (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import mod
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'mod'
>>>
```

```
In [18]: import sys
sys.path
```

```
Out[18]: ['',
'C:\\Users\\WWW\\THJS\\Anaconda3\\python36.zip',
'C:\\Users\\WWW\\THJS\\Anaconda3\\DLLs',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib',
'C:\\Users\\WWW\\THJS\\Anaconda3',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages\\win32',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages\\win32\\lib',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages\\Pythonwin',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages\\IPython\\extensions',
'C:\\Users\\WWW\\THJS\\.ipython']
```

```
In [19]: sys.path.append("C:/module")
sys.path
```

```
Out[19]: ['',
'C:\\Users\\WWW\\THJS\\Anaconda3\\python36.zip',
'C:\\Users\\WWW\\THJS\\Anaconda3\\DLLs',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib',
'C:\\Users\\WWW\\THJS\\Anaconda3',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages\\win32',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages\\win32\\lib',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages\\Pythonwin',
'C:\\Users\\WWW\\THJS\\Anaconda3\\lib\\site-packages\\IPython\\extensions',
'C:\\Users\\WWW\\THJS\\.ipython',
'C:/module']
```

```
In [20]: import mod
```

```
더할 수 없습니다.
None
5
20.4
```

In []: