DAY 2

목차

- 함수
- Class
- 모듈과 패키지
- 파이썬 내장함수

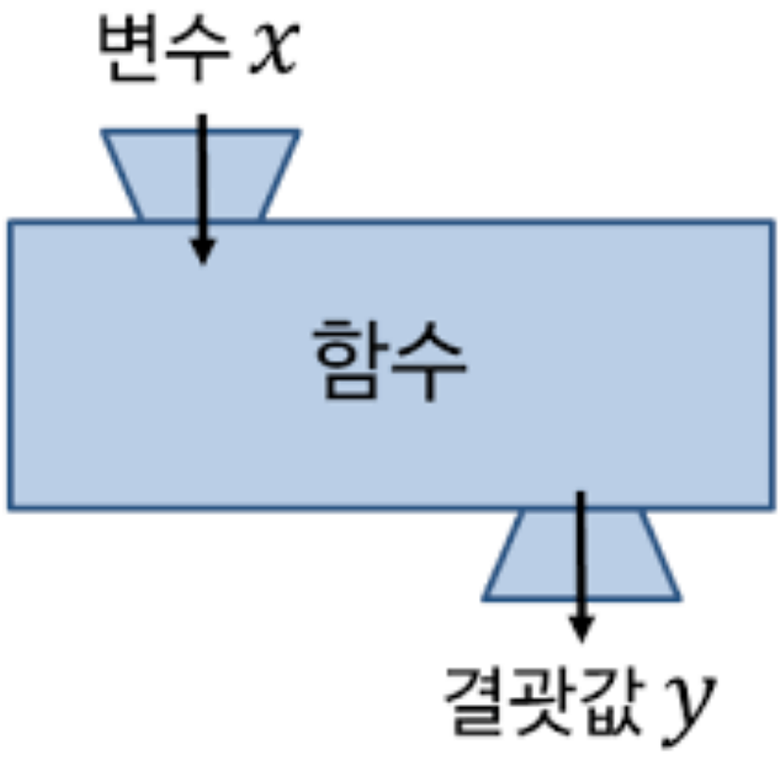
함수

- 함수란?
- 함수 정의하기
- 함수와 변수

함수

- function(함수)

함수란?




```
In [ ]: def 함수이름(함수인자) :
            수행할 문장1
            수행할 문장2
            return 반환값
```

일정한 프로세스를 할 수 있도록 하는 것

함수

- function(함수) 정의하기

def 함수이름(매개변수) :
함수의 내용
return 반환값

- **함수이름** : 사용자가 정의하는 함수 이름, 기존에 사용되는 함수나 예약어들을 제외하고 사용
- **매개변수** : 함수 안에서 사용 할 변수들 (생략 가능)
- **return** : 함수 안에서 모든 연산을 마친 후 반환할 값(생략 가능)

함수

- function(함수) 정의하기

1. 매개변수와 return값이 모두 있는 함수

```
In [1]: def add(a, b) :
            result = a + b
            return result
```

```
In [2]: c = add(3, 5)
        c
```

```
Out[2]: 8
```

2. return값이 비어있는 함수

```
In [3]: def sub(a, b) :
            print('뺄셈의 결과는 %d입니다.' %(a-b))
            return
```

```
In [4]: d = sub(1, 2)
```

```
뺄셈의 결과는 -1입니다.
```

```
In [5]: print(d)
```

```
None
```

함수

- function(함수) 정의하기

3. return이 없는 함수

```
In [6]: def mul(a, b) :
            print('%d 와 %d의 곱은 %d입니다.' %(a, b, a*b))

In [7]: mul(3, 2)
3 와 2의 곱은 6입니다.
```

4. 매개변수와 return값이 모두 없는 함수

```
In [8]: def start():
            print("Hello World!")

In [9]: start()
Hello World!
```

함수

- 더 알아보기

1. 매개변수의 초기값 미리 설정하기

```
In [14]: def function(a, n=2) :
             print("%d의 제곱은 %d 입니다." %(a, a**n))

In [15]: function(4)
         4의 제곱은 16 입니다.
```

2. 매개변수가 몇 개가 필요할지 모를 때

```
In [26]: def add_all(*args) :
             result = 0
             for i in args :
                 result += i
             return result

In [27]: add_all(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
Out[27]: 55
```

→ 매개변수에 *를 붙이면 함수가 받은 input 모두를 tuple로 묶어줌

함수

- 함수와 변수

Quiz! 함수 안에 선언한 변수를 함수 밖에서도 사용할 수 있을까? **NO!**

함수 안의 블록은 함수 고유의 영역임.
따라서, **함수 안에서 선언한 변수는 함수 밖에서 사용할 수 없음.**

함수

- 함수와 변수

예시 1)

```
In [28]: def myfunction() :
             in_val = '함수 안의 변수'

In [29]: myfunction()

In [30]: in_val
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-30-a23adeea90d3> in <module>
----> 1 in_val

NameError: name 'in_val' is not defined
```

예시 2)

```
In [38]: a = 1

In [39]: def myfunction2(a) :
             a += 1
             print(a)

In [40]: myfunction2(a)

In [41]: a
```

함수 실행 시 출력값은 무엇일까요?

a는 어떤 값일까요?

함수

- 함수와 변수

- local variable : 함수 안에 있는 변수
- global variable : 함수 밖에 있는 변수

함수 안에 있는 변수를 밖에서도 쓰고 싶다면?

1. return 사용하기
2. global 사용하기

함수

- 함수와 변수

함수 안에 있는 변수를 밖에서도 쓰고 싶다면?
1. return 사용하기

In [45]:
```
def myfunction3(a) :
    a += 1
    print(a)
    return a
```

In [46]:
```
b = myfunction3(a)
```

```
2
```

In [47]:
```
b
```

Out[47]:
```
2
```

함수

- 함수와 변수

함수 안에 있는 변수를 밖에서도 쓰고 싶다면?

2. global 사용하기

In [51]:
```
a = 1
```

In [52]:
```
def myfunction4():
    global a
    a += 1
    print(a)
```

In [54]:
```
myfunction4()
```

```
2
```

In [55]:
```
a
```

Out[55]:
```
2
```

정리 문제 2-1

1. ‘students.txt’ 파일을 읽어서 딕셔너리로 학생들의 정보를 저장하라
   해당 파일은 이름, 국어점수, 영어점수, 수학점수 순으로 되어있으며, 각 컬럼의 구분자는 tab(\t)으로 이루어져있다.
   이름을 key로 하고, 국어점수, 영어점수, 수학점수를 순서대로 담은 리스트를 value로 하는 딕셔너리로 저장하라.

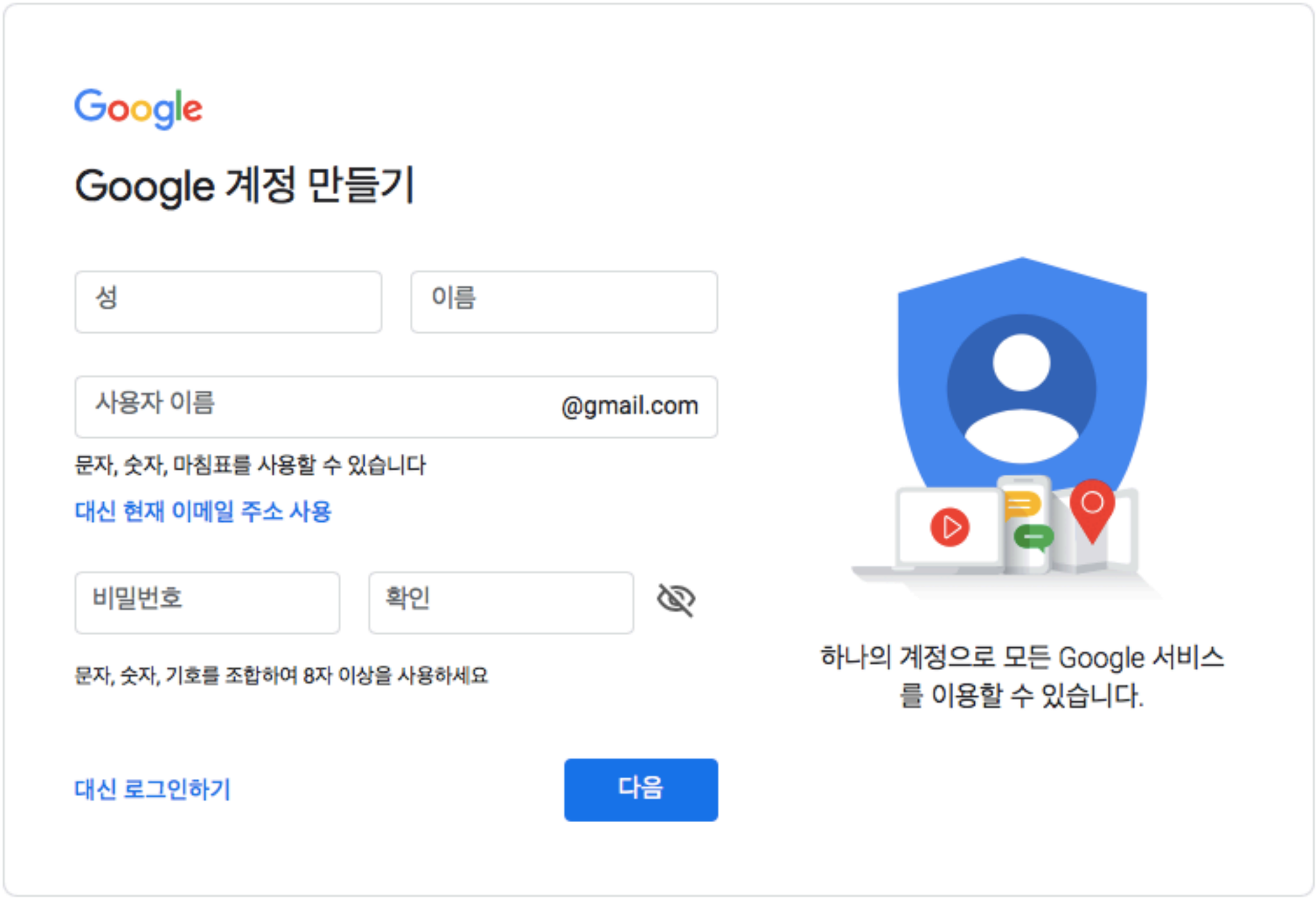
{이름 : [국어, 영어, 수학], }

정리 문제 2-1

2. 저장한 딕셔너리를 토대로 학생들의 평균을 구하라
   전체 학생의 국어, 영어, 수학 점수의 평균을 구해서 각각 kor, eng, math라는 변수에 저장하고, 소수점 첫째자리까지 출력하라.

3. 학생들의 평균을 구하는 함수를 작성하라.
   이름을 넣으면 국어, 영어, 수학 3과목의 평균을 계산하여 반환하는 함수를 작성하라.

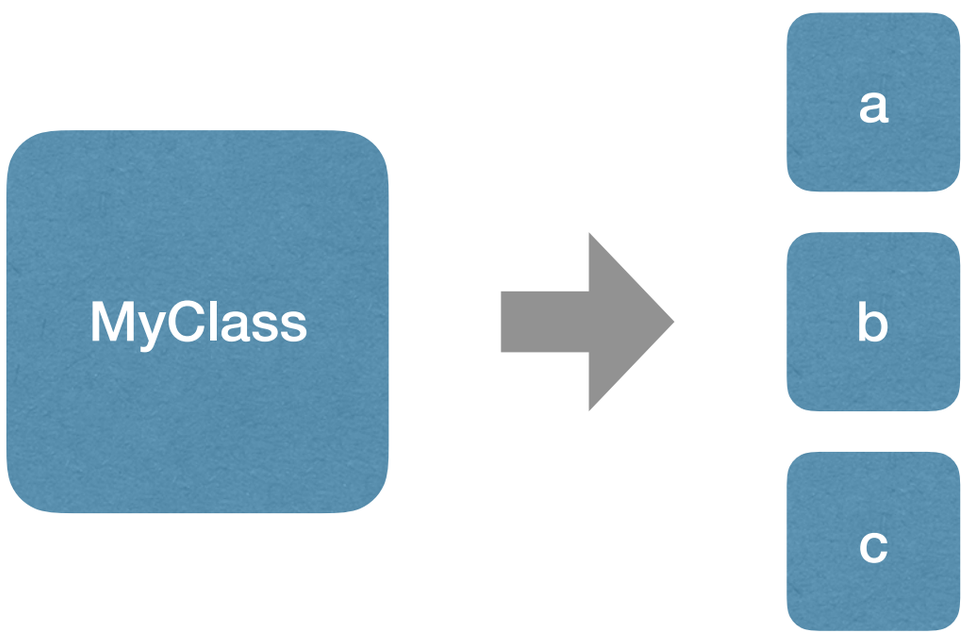
클래스


Google
Google 계정 만들기
성
이름
사용자 이름
@gmail.com
문자, 숫자, 마침표를 사용할 수 있습니다
대신 현재 이메일 주소 사용
비밀번호
확인
문자, 숫자, 기호를 조합하여 8자 이상을 사용하세요
대신 로그인하기
다음
하나의 계정으로 모든 Google 서비스를 이용할 수 있습니다.
한국어
도움말
개인정보 보호
약관

클래스

- Class

- 클래스(Class) : 객체를 만드는 구조 / 틀
- 객체(Object) : 클래스로 만들어진 것
- 인스턴스(Instance) : 객체와 클래스사이의 관계를 이야기할 때 사용하는 용어



a, b, c는 각각 객체

a, b, c는 MyClass의 인스턴스

클래스

- Class

In [25]:
```
class MyClass() :
    class_var = '클래스 변수'

    @classmethod
    def class_method(cls):
        print('클래스의 메소드')
```

In [26]:
```
MyClass.class_method()
```

```
클래스의 메소드
```

In [27]:
```
MyClass.instance_method()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-27-65c83cf9e6e8> in <module>
----> 1 MyClass.instance_method()

TypeError: instance_method() missing 1 required positional argument: 'self'
```

In [28]:
```
cls = MyClass()
```

In [29]:
```
cls.instance_method()
```

```
인스턴스의 메소드
```

클래스

- Variable(변수)
  1. class variable(클래스 변수)
     클래스를 통해서 호출 가능
  2. instance variable(인스턴스 변수)
     인스턴스가 있어야 사용이 가능

```
In [17]: class Room() :

             room = []

         def add_person(self, name):
             self.room.append(name)
```

클래스 변수

```
In [19]: r1 = Room()
         r2 = Room()

         r1.add_person('뽀로로')
         r2.add_person('펭수')
```

```
In [20]: Room.room
Out[20]: ['뽀로로', '펭수']
```

```
In [21]: class Room2() :

             def __init__(self) :
                 self.room = []

             def add_person(self, name):
                 self.room.append(name)
```

인스턴스 변수

```
In [22]: r1 = Room2()
         r2 = Room2()

         r1.add_person('뽀로로')
         r2.add_person('펭수')
```

```
In [23]: r1.room
Out[23]: ['뽀로로']
```

```
In [24]: r2.room
Out[24]: ['펭수']
```

클래스

- Method(메소드)
  1. class method(클래스 메소드)
     클래스 자체만 있어도 사용 가능
  2. instance method(인스턴스 메소드)
     인스턴스가 있어야 사용이 가능

```
In [26]: MyClass.class_method()
```

인스턴스가 필요 없음

```
클래스의 메소드
```

```
In [27]: MyClass.instance_method()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-27-65c83cf9e6e8> in <module>
----> 1 MyClass.instance_method()

TypeError: instance_method() missing 1 required positional argument: 'self'
```

```
In [28]: cls = MyClass()
```

인스턴스가 필요

```
In [29]: cls.instance_method()
```

```
인스턴스의 메소드
```

클래스

- Method(메소드)

2. instance method(인스턴스 메소드)

1) __init__ (초기화자) : 객체 생성시 객체의 초기값을 설정하는 메소드로, 자동으로 호출됨

2) __del__ (소멸자) : 객체 소멸시 자동으로 호출되는 메소드

```
In [14]: class TestClass1():
             def __init__(self, v1, v2):
                 self.v1 = v1
                 self.v2 = v2

In [15]: t_cls1 = TestClass1(10, 20)

In [16]: t_cls1.v1
Out[16]: 10
```

→ 초기화자를 통해서 객체의 초기값을 자동으로 설정

클래스

- Method(메소드)

  2. instance method(인스턴스 메소드)

     주의 인스턴스 메소드의 매개변수 중 가장 첫번째는 **인스턴스**의 자리임

```
In [14]: class TestClass1():
             def __init__(self, v1, v2):
                 self.v1 = v1
                 self.v2 = v2

In [15]: t_cls1 = TestClass1(10, 20)

In [16]: t_cls1.v1

Out[16]: 10
```

인스턴스

인스턴스 메소드의 매개변수

→ 파이썬은 인스턴스를 자동으로 넘겨줌

클래스

- Method(메소드)

2. instance method(인스턴스 메소드)

주의 인스턴스 메소드의 매개변수 중 가장 첫번째는 **인스턴스**의 자리임

```
In [6]: class TestClass2():
            def __init__(v1, v2):
                self.v1 = v1
                self.v2 = v2
```

```
In [7]: t_cls2 = TestClass2(10, 20)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-7-32e19ab7347c> in <module>
----> 1 t_cls2 = TestClass2(10, 20)

TypeError: __init__() takes 2 positional arguments but 3 were given
```

정리 문제 2-2

1. MyPhone이라는 이름으로 본인의 핸드폰에 관한 클래스를 만들어보자.
   1) MyPhone의 속성으로는 어떠한 것이든 좋다.
   2) MyPhone에 set_name과, set_number라는 인스턴스 메소드를 포함하여야 한다.

- set_name은 사용자의 이름을 받아서 user라는 변수에 저장한 후 이를 출력하는 함수이다.
- set_number는 번호를 입력으로 받아서 number라는 변수에 저장하는 함수이다.

클래스

- 클래스 상속(inheritance)

- 상속 : 부모의 클래스를 물려받음
  → 부모 클래스가 가지고 있는 함수/변수를 그대로 사용할 수 있음
- 상속의 장점 : 기존 클래스를 변형하지 않고 추가/변경이 가능함

상속받고자 하는 클래스 (부모 클래스)

```
In [4]: class MyPhone2(MyPhone):
            def has_case(self, val=False) :
                self.case = val
```

자식 클래스의 새로운 메소드

클래스

- 클래스 상속(inheritance)

```
In [8]: class MyPhone():
            def __init__(self, model, color) :
                self.model = model
                self.color = color

            def set_name(self, name):
                self.user = name
                print("사용자의 이름은 : %s" %self.user)

            def set_number(self, number):
                self.number = number
```

```
In [9]: class MyPhone2(MyPhone):
            def has_case(self, val=False) :
                self.case = val
```

```
In [10]: p2 = MyPhone2('갤럭시S8', 'black')
```

```
In [11]: p2.set_name("MJ")
```

```
사용자의 이름은 : MJ
```

부모 클래스의 메소드 사용 가능

```
In [54]: p2.has_case(True)
```

```
In [55]: p2.case
```

```
Out[55]: True
```

자식 클래스의 새로운 메소드도 사용 가능

클래스

- 메소드 수정

- 상속 받은 클래스의 메소드는 덮어쓰기처럼 수정하면 됨

```
class MyPhone():
    def __init__(self, model, color) :
        self.model = model
        self.color = color

    def set_name(self, name):
        self.user = name
        print("사용자의 이름은 : %s" %self.user)

    def set_number(self, number):
        self.number = number
```

```
p1 = MyPhone('갤럭시S8', 'black')
p1.set_number("010-xxxx-xxxx")
```

```
class MyPhone3(MyPhone) :
    def set_number(self, number) :
        self.number = number
        print("이 핸드폰의 번호는 : %s" %self.number)
```

```
p3 = MyPhone3('갤럭시S8', 'black')
p3.set_number("010-xxxx-xxxx")
```

```
이 핸드폰의 번호는 : 010-xxxx-xxxx
```

모듈과 패키지

- 모듈(module)

1. 모듈
   - 함수나 변수, 클래스 등을 가진 파일 (.py)
   - 모듈 안에는 함수, 클래스 또는 변수들이 정의되어 있음
   - 파이썬은 많은 표준 라이브러리 모듈을 제공함

```
In [1]: import math

In [2]: math.factorial(4)
Out[2]: 24
```

import 모듈이름

```
In [3]: from math import factorial

In [4]: factorial(4)
Out[4]: 24
```

from 모듈이름
import 변수/함수/클래스 이름

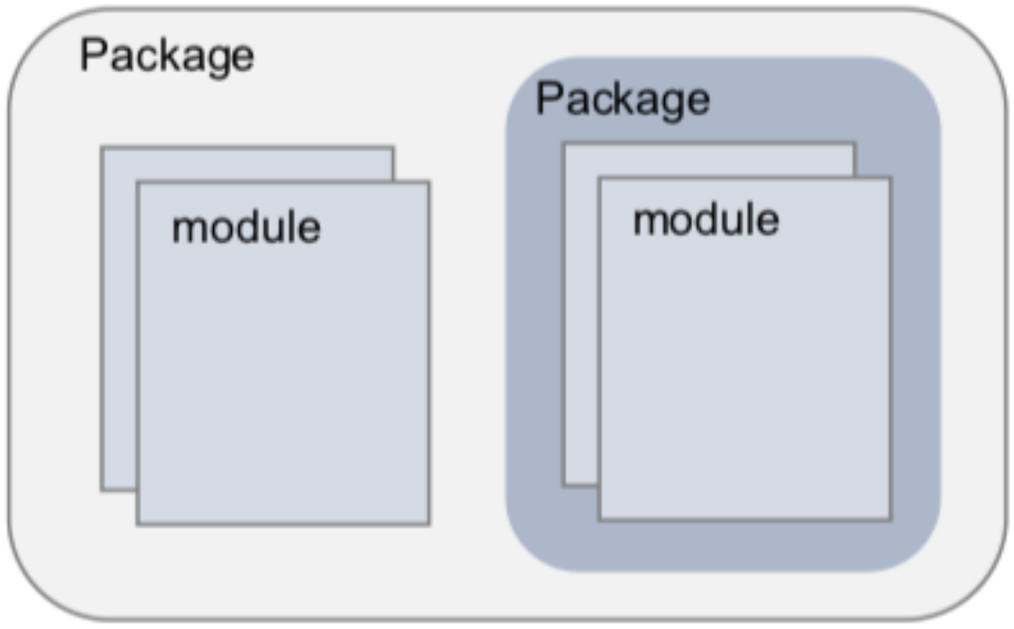
모듈과 패키지

- 모듈(module)

2. 패키지(pacakage)
   - 모듈을 효율적으로 관리하기 위한 모듈의 상위 개념
   - 공동 작업이나 코드의 유지 보수 등에 유리

```
In [ ]: import 패키지.모듈
        import 패키지.모듈.변수
        import 패키지.모듈.함수
        import 패키지.모듈.클래스
```

```
In [ ]: from 패키지.모듈 import 변수/함수/클래스
```



- 나

기본

수들

abs()	dict ()	help ()	min ()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmetho
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	**import**()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

있는 함

파이썬 내장함수

- 내장함수

1) input( ) : 외부로부터 입력 받아오기

```
In [9]: a = input()
        내장함수 알아보기
```

2) len( ) : 자료형의 길이 재기

```
In [10]: len(a)
Out[10]: 9
```

3) abs( ) : 절대값

```
In [11]: abs(-1)
Out[11]: 1
```

파이썬 내장함수

- 내장함수

4) 범위 지정하기 : range( )

```
In [16]: for i in range(5) :
             print(i + 1)

1
2
3
4
5
```

5) 최대값 / 최소값 : max( ) / min( )

```
In [12]: b = [1,2,3,4,5,6,100]

In [13]: max(b)
Out[13]: 100

In [15]: min(b)
Out[15]: 1
```

정리 문제 2-3

235

아래의 에러가 발생하는 이유에 대해 설명하라.

```
hello()
def hello():
    print("Hi")
```

실행 예

```
NameError: name 'hello' is not defined
```

정리 문제 2-3

1. Human 이라는 클래스를 만들어보자. Human 클래스는 아래와 같은 특징이 있다.
   - Human 클래스는 객체 선언 시 birth_date, sex, nation을 인스턴스 변수로 가지고 있다.
   - birth_data는 생년월일을 뜻하며 yymmdd 6자리로 되어있는 숫자이다.
   - sex는 성별을 의미하며 남자는 M, 여자는 F로 표시한다.
   - nation은 태어난 국가를 의미하며 한국어로 되어있다.

정리 문제 2-3

1. Human 이라는 클래스를 만들어보자. Human 클래스는 아래와 같은 특징이 있다.
   - give_name은 인스턴스 메소드로 이름을 매개변수로 받아서 name이라는 인스턴스 변수로 저장하고 화면에 이름을 출력하는 역할을 하는 함수이다.
   - can_sing이라는 함수는 True/False값을 input으로 받으며, 참이면 “Sing a song”을 화면에 출력하는 함수이다.

클래스를 다 만들었으면 인스턴스를 만들어보자.

정리 문제 2-3

2. Human 이라는 클래스를 상속하는 Child라는 클래스를 만들어보자. Child의 클래스에는 아래와 같은 변수와 함수들이 추가된다.
   - 눈동자 색깔을 나타내는 변수 eye를 인스턴스 선언시 사용할 수 있게 추가해보자.
   - Child의 클래스는 노래하는 능력이 없다. 따라서 can_sing이라는 메소드가 호출되면 “Can’t Sing”이고 출력하도록 바꿔보자.
   - Child는 노래 대신 춤을 출 수 있다. can_dance라는 메소드가 호출되면 “Dance Time!”을 출력하도록 메소드를 작성해보자.

정리 문제 2-3

3. 입력으로 들어오는 모든 수의 평균 값을 계산해 주는 함수를 작성해 보자. (단 입력으로 들어오는 수의 개수는 정해져 있지 않으나, 최소한 1개 이상이다.)

4. 다음 중 출력 결과가 다른 것 한 개를 골라 보자.
   - print(“you" "need" “python")
   - print(“you"+"need"+"python")
   - print("you", "need", “python")
   - print("".join(["you", "need", "python"]))