

함수

1. 함수 정의 및 사용

- 입력 값을 받아 다른 값을 출력하도록 미리 만들어져 있는 것
- 반복해서 사용한 코드들을 묶어 놓고 그것에 이름을 붙인 것

▶ 함수 정의

```
def function_name(x):  
  
    expression
```

def : 함수를 정의하기 위한 키워드

function_name : 함수 이름. 숫자로 시작할 수 없다.

x : 매개변수.

▶ 매개변수와 인수

- **매개변수** : 함수 정의 시 지정하는 함수가 실행을 위해 필요하는 값을 받을 변수들

함수 안에서만 참조할 수 있다.

- **인수** : 함수 호출 시 함수 실행을 위해 전달하는 값

▶ 매개변수가 없는 함수 정의

함수 안에서 실행하는 값을 사용하지 않을 때 매개변수를 선언하지 않는 함수를 정의한다.

▶ 매개변수가 있는 함수 정의

- 함수를 실행하기 위해 어떤 값을 받아야 한다면 매개변수를 선언한다.
- 함수 안에서 사용할 값을 받는다.
- 매개변수는 여러 개 사용할 수 있으며, 함수 호출 시 인수의 개수는 함수 정의 시 매개변수의 개수와 일치해야 한다.

▶ docstring

- 함수 본문의 첫번째 문장에 문자열을 포함한다.
- 곱따옴표 3개(""") 또는 홑따옴표 3개('')를 이용한다.

2. 함수의 실행결과를 반환하는 return

- 함수가 실행한 결과를 함수를 호출한 곳에 전달하기 위해 사용
- 여러 개의 값을 반환하면 그 값들은 튜플에 저장되어 반환

3. 함수 매개변수

▶ 기본 값을 갖는 매개변수

- 하나 이상의 인수에 대해 기본값을 지정한다.
- 기본값은 함수가 정의되는 지점에 평가됨.

▶ 변경 가능한 객체를 기본변수로 갖는 매개변수

기본 변수가 리스트, 딕셔너리 등의 변경 가능한 객체일 때, 호출시마다 전달된 인수를 사용한다.

```
def func(a, L=[ ]):  
    L.append(a)  
    return L  
  
print(func(1))  
-> [1]  
  
print(func(2))  
-> [1, 2]
```

▶ 튜플 매개변수를 이용한 가변인수 설정

- 매개변수 앞에 *을 붙여 정의
- 인수들이 튜플에 저장되어 전달된다

▶ 딕셔너리 인수

- 매개변수 앞에 **을 붙여 정의

▶ 매개변수의 순서

순서 인자 > 튜플 인자 > 키워드 인자 > 딕셔너리 인자

4. 람다식

- 람다식은 작은 익명함수를 의미한다.

lambda variable_define : statements

variable_define : 함수의 인수를 정의

statements : 함수가 실행할 문장을 작성

- 한 문장만 작성할 수 있음
- return 구문이 없어도 statement의 결과를 반환
- 람다식은 함수 객체가 필요한 곳이면 어디서든지 사용할 수 있다.

5. 파이썬 내장함수

- import하지 않고 즉시 사용 가능한 함수
- 내장 함수명은 일종의 키워드로 간주되므로 식별자로 사용하는 것은 피한다.

▶ 입출력 관련 함수

print(x)	객체를 문자열로 표시한다.
input([prompt])	사용자 입력을 문자열로 반환한다.
help([x])	x에 대한 도움말을 출력한다
globals()	전역 변수의 리스트를 반환한다.
locals() / vars()	지역 변수의 리스트를 반환한다.
del(x) / del x	객체를 변수 공간에서 삭제한다
eval(expr)	값을 구한다.
exec(obj)	파이썬 명령을 실행시킨다.
open(filename,[,mode]))	파일을 연다.

▶ 기본 자료형의 생성과 변환 함수

object()	새로운 object를 생성한다.
bool(obj)	객체의 진리값을 반환한다.
int(obj)	문자열 형태의 숫자나 실수를 정수로 변환한다.
float(obj)	문자열 형태의 숫자나 정수를 실수로 변환한다.
complex(re,[,img])	문자열이나 주어진 숫자로 복소수를 생성한다.

▶ 기본 자료형의 정보를 얻는 함수

type(obj)	객체의 형을 반환한다.
dir(obj)	객체가 가진 함수와 변수들을 리스트 형태로 반환한다.
repr(obj)	eval()함수로 다시 객체를 복원할 수 있는 문자열 생성
id(obj)	객체의 고유번호(int형)을 반환한다.
hash(obj)	객체의 해시값을 반환 (같은 값이면 해시도 같다)
isinstance(obj, className)	객체가 클래스의 인스턴스인지를 판단한다.

▶ 열거형 정보를 얻는 함수

len(seq)	시퀀스형을 받아서 그 길이를 반환한다.
enumerate(iterable, start=0)	이터러블에서 enumerate형을 반환한다. 입력값으로 시퀀스자료형 (리스트,튜플,문자열)을 반환한다.
sorted(iterable[,key][,reverse])	정렬된 리스트를 반환한다.
reversed(seq)	역순으로 된 iterator를 반환한다.
filter(func, iterable)	iterable의 각 요소 중 func()의 반환 값이 참인 것만을 묶어서 이터레이터로 반환
map(func, iterable)	iterable의 각 요소를 func()의 반환값으로 매핑해서 이터레이터로 반환

▶ 산술/논리 연산과 관련된 함수

hex(n)	정수 n의 16진수 값을 구해서 문자열로 반환한다.
oct(n)	정수 n의 8진수 값을 구해서 문자열로 반환한다.
bin(n)	정수 n의 2진수 값을 구해서 문자열로 반환한다.
abs(n)	절대값을 구한다. 복소수의 경우 크기를 구한다.
pow(x,y[,z])	거듭제곱을 구한다. pow(x,y)는 x**y와 같다.
divmod(a,b)	a를 b로 나눈 (몫, 나머지)를 구한다. 튜플 반환
all(iterable)	iterable의 모든 요소가 True일 경우 True를 반환
any(iterable)) iterable의 하나 이상의 요소가 True일 경우 True를 반환
round()	반올림을 한다.