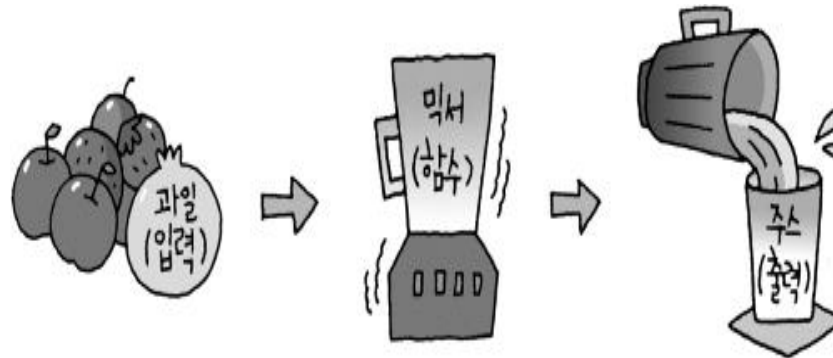


05 함수

■ 함수란 무엇인가?

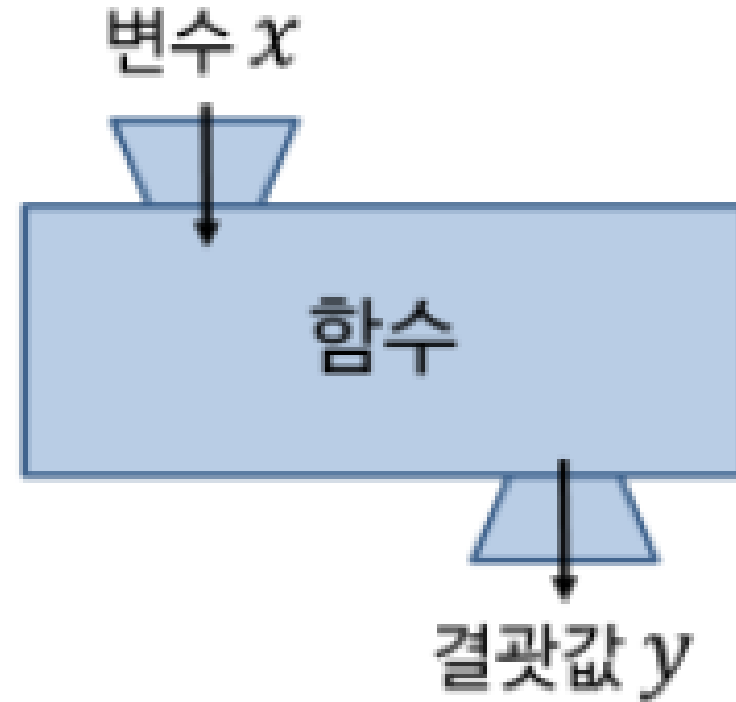
- ✓ 독립적으로 존재하며 일을 수행하는 코드들의 집합
- ✓ 함수의 이름으로 호출하며 재사용이 가능하다.
- ✓ 매개변수를 받을 수도 받지 않을 수도 있다.
- ✓ 결과값(return)이 존재 할 수도 존재하지 않을 수도 있다



05 함수

■ 함수란 무엇인가?

- ✓ $f(x) = 5x + 3$
- ✓ input : $x = 1$
- ✓ output : $f(x)=y=8$



05 함수

■ 함수를 사용하는 이유

- ✓ 반복되는 코드를 줄이기 위해.
- ✓ 프로그램을 함수화 하면 흐름을 일목요연하게 볼 수 있기 때문에.
- ✓ 오류를 찾기에 편하며 유지보수에 용이.

05 함수

■ lista = ['1', '2', '3', '4']

✓ 1. print(lista.sort())

✓ 2. print(lista.count('1'))

✓ 위 둘의 차이는?

05 함수

■ 함수의 기본 구조

✓ def 함수명(매개변수) : << 매개변수 : input값

 실행문1. << 실행문 : f(x)

 실행문2.

 return 결과값 << 리턴값 : output

05 함수

■ 함수의 기본 구조 예시

✓ `def plus(a, b) :`

`result = a + b`

`return result`

<< 매개변수 : input값

<< 실행문 : f(x)

<< 리턴값 : output

05 함수

- 함수 연습1(입출력이 있는 함수)

- ✓ `print(mul(2,3))`을 하면 2와 3을 곱한 값이 출력되는 함수를 구현하라.

05 함수

■ 연습문제 (사용자 정의 함수)

- ✓ 키보드로 반지름의 길이를 입력받고 ,
- ✓ 원의 넓이를 구하는 함수를 만들어 결과를 출력하라.
- ✓ 원의 넓이 = 반지름 * 반지름 * 3.14

[출력화면 예시]

반지름의 길이를 입력하세요. 2

원의 넓이 : 12.56

05 함수

■ 입력값과 결과값에 따른 함수의 다양한 형태

- ✓ 1. 입력값 & 결과값이 있는 함수

결과값을 받을 변수 = 함수명(인수1, 인수2, ...)

- ✓ 2. 입력값이 없는 함수

결과값을 받을 변수 = 함수명()

05 함수

- 입력값과 결과값에 따른 함수의 다양한 형태

- ✓ 3. 결과값이 없는 함수

함수명(인수1, 인수2, ...)

- ✓ 4. 입력값 & 결과값 모두 없는 함수

함수명()

05 함수

■ 함수 연습2(입력값이 없는 함수)

✓ 1) `hello()`라고 함수를 실행하면 "hello python!" 이라고

출력되는 함수를 구현하라.

✓ 2) `print(hello())`라고 실행하면 "hello python!"이라고

출력되는 함수를 구현하라.

05 함수

■ 함수 연습3(출력값이 없는 함수)

✓ 아래 사진은 오름차순 정렬 프로그램 코드이다.

✓ 해당 로직을 함수로 변형하고, lst를 키보드로 입력받아 오름차순으로 정렬하라.

```
# lst = [93, 45, 21, 30, 20, 94, 66, 71, 45]
# #오름차순
# for a in range(0,len(lst)-1):
#     for b in range(a+1,len(lst)):
#         if lst[a] > lst[b]:
#             (lst[a], lst[b]) = (lst[b], lst[a])
#             tmp = lst[a]
#             lst[a] = lst[b]
#             lst[b] = tmp
#
# print(lst)
```

05 함수

- 입력값이 몇 개인지 모를 때의 함수 구조

- ✓ `def 함수명(*args) :` << 여러 개의 값이 튜플 형태로 전달

- 수행할 문장

05 함수

■ 입력값이 여러 개인 함수 예제

✓ `def plus(*args) :` << 일반적으로 args라 쓰나 변경해도 무관

```
    result = 0
    for i in args :
        result += i
    return result
```

`result = plus(1, 2, 3, 4, 5)`
`print(result)`

05 함수

■ 연습문제 (선택적 함수 만들기)

- ✓ 사칙연산을 하는 함수를 만들어보자. 키보드로 +, -, *, /를 입력 받아 어떤 연산을 할 것인지 선택하고 정수 두개를 입력 받아 연산을 처리하고 결과를 돌려주는 함수를 만들어라.

[출력화면 예시]

어떤 연산을 하실건가요? (+,-,*,/) +

첫번째 정수를 입력하세요. 3

두번째 정수를 입력하세요. 7

3과 7의 +연산결과는 10입니다.

05 함수

■ 함수의 return 값이 2개 이상일 경우

- ✓ 튜플 형태의 값으로 리턴이 된다.
- ✓ `def sum(a, b):`
- ✓ `result1 = a + b`
- ✓ `result2 = a * b`
- ✓ `result3 = a // b`
- ✓ `return result1, result2, result3`
- ✓ `result = sum(1,2)`
- ✓ `print(result)`

05 함수

■ 함수에서 return의 용도

- ✓ 특수한 조건일 때 해당 함수를 종료시키고 싶다면 return을 사용해 종료시킬 수 있다.

```
✓ def speak(msg) :  
✓     if msg == "종료" :  
✓         return  
✓         print("{msg}를 입력하셨습니다,")  
✓     else :  
✓         print(msg)  
✓ speak(input("메시지를 입력하세요 :"))
```

05 함수

- 함수의 초기값(default) 미리 세팅하기
 - ✓ Ex) print에서 end의 사례
- 함수 매개변수의 순서를 꼭 다 맞춰야 할까?
 - ✓ Key value 로 설정해 두면 가능

05 함수

■ 예약어 lambda

- ✓ **lambda** 매개변수N개, ... : 매개변수를 이용한 표현식
- ✓ 편의성과 자유도를 높이기 위해 등장
- ✓ lambda로 만든 함수는 return 명령어가 없어도 결과값을 리턴한다.
- ✓

```
def add(a,b):  
    return a+b
```
- ✓

```
add = lambda a, b : a+b
```

05 함수(변수의 효력 범위)

■ 지역변수/전역변수란 무엇인가?

✓ [지역변수]

해당 지역(함수)에서만 활용하는 변수

해당 함수 호출이 끝나면 사라진다.

✓ [전역변수]

프로그램 전체에서 활용하는 변수

05 함수(변수의 효력 범위)

■ 지역변수와 전역변수의 예

- ✓ `a = 2`
- ✓ `def sum(a):`
- ✓ `a += 1`
- ✓ `return a`
- ✓ `result = sum(a)`
- ✓ `print(result)`
- ✓ `print(a)`

05 함수(변수의 효력 범위)

- 메모리 구조를 통해 알아보자.

- ✓ <https://pythontutor.com/>
- ✓ 전역변수는 전체 프로그램이 종료될 때까지 유지
- ✓ 지역변수는 해당 함수의 호출이 종료되면 사라짐

05 함수(변수의 효력 범위)

- 지역변수는 왜 함수 호출이 끝나면 사라지는 변수로 설계했을까?

- ✓ `def add (a, b) :`

`result = a + b`

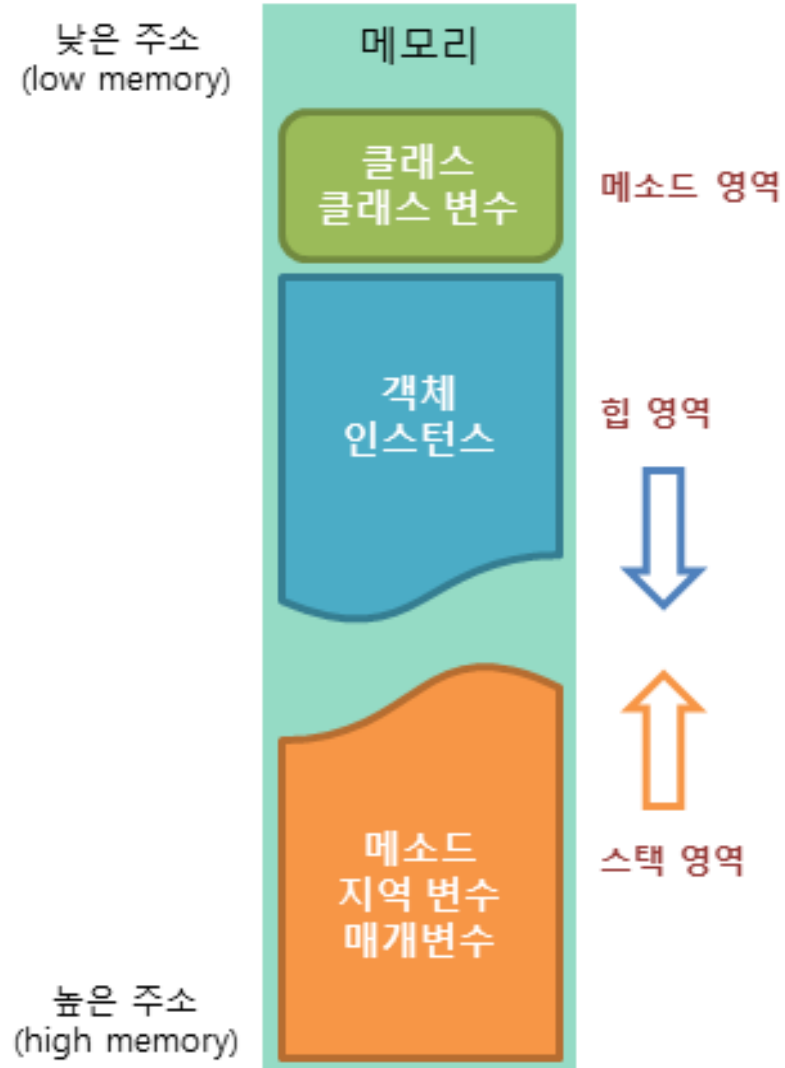
- ✓ `add` 함수를 여러 용도로 여러 번 사용하게 될 때, `result`의 값이 누적된다면 어떻게 될까?

05 함수(변수의 효력 범위)

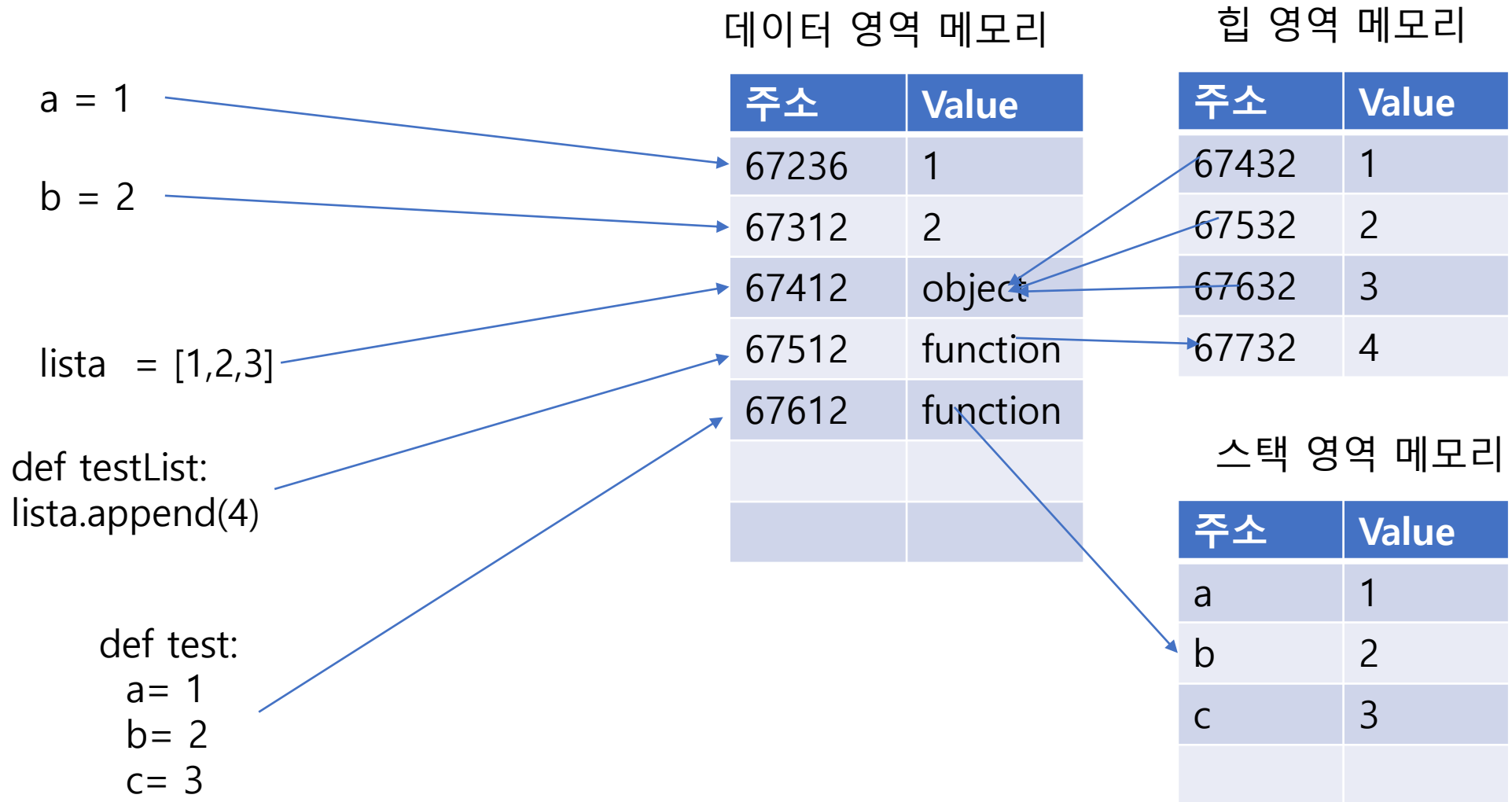
- 지역변수의 값으로 전역변수의 값을 바꾸는 2가지 방법
 - ✓ 해당 변수에 다시 담아두기
 - ✓ global 변수를 선언하기.

05 함수(변수의 효력 범위)

■ 메모리 구조



05 함수(변수의 효력 범위)



05 함수(mutable / immutable)

- immutable : int, string, tuple, bool

- ✓ 객체를 생성 후, 객체의 값을 수정 **불가능**, 변수는 값이 수정된 다른 객체를 가르키게 된다.

- mutable : list, dict, set

- ✓ 객체를 생성 후, 객체의 값을 수정 **가능**, 변수는 값이 수정된 같은 객체를 가르키게 된다.

05 함수(입력과 출력)

- input과 print : 이미 모두 배운 내용
- print 주요 특징 복습
 - 1) `print("python is easy", end = ' ')`
 - 2) `print("you need", "python")`

05 함수

■ 함수 연습4

- ✓ [1, 4, 6]의 값으로 구성된 리스트가 있다.
- ✓ "index(a)" 함수는 a가 리스트의 몇번째에 있는지 찾는 내장함수다.

해당 내장함수를 쓰지 말고, for문을 사용하여 같은 역할을 하는
find라는 함수를 구현하라.

05 함수

■ 함수 연습5

- ✓ layer 변수에 별의 층수를 입력하면 각 층마다 별의 개수가 한 개씩 증가하여 출력되고, 마지막 줄에 별의 개수가 출력되도록 함수를 구현하라.
- ✓ def starCount (layer) : << 함수 선언부

[출력 결과]

layer : 5

*

**

별의 개수 : 15

05 함수

■ 함수 연습6

- ✓ 팩토리얼을 계산하는 재귀함수를 만들어보라
- ✓ `def factorial_func (x) :`

[출력 결과]

1 2 3 4 팩토리얼 결과 : 24