

Lab06

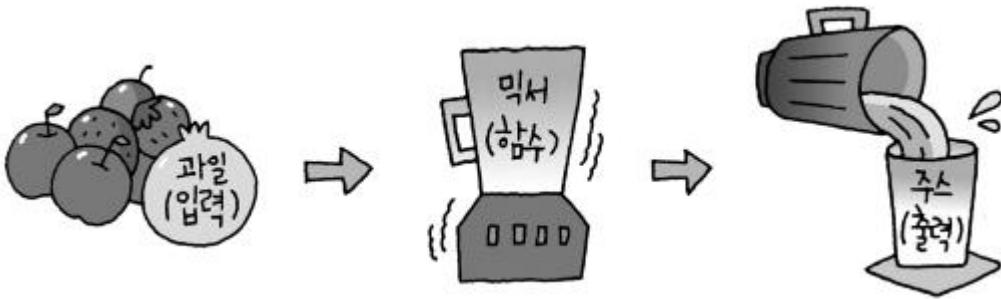
소프트웨어의 이해



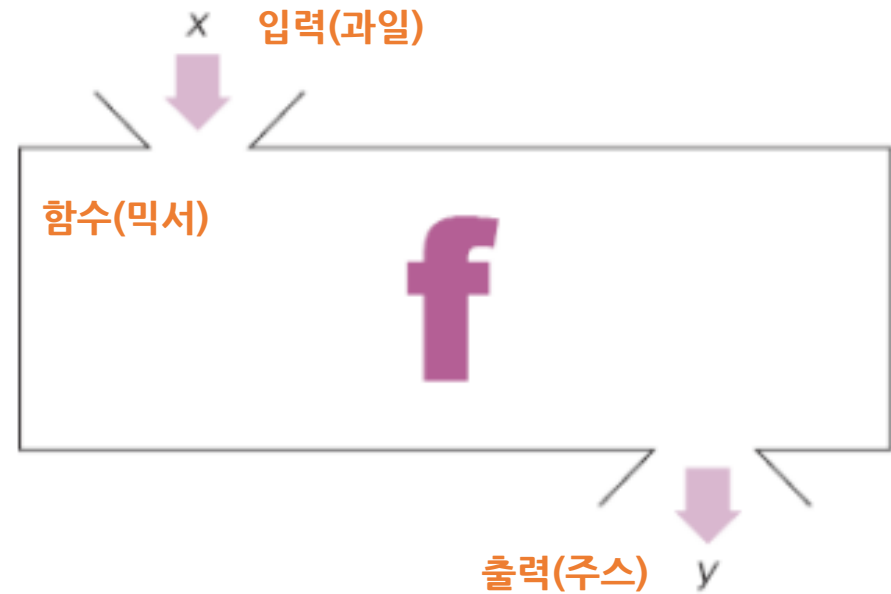
1.1 함수

- 함수 f

값 x 를 받아서 어떤 계산을 수행하여 결과값 $y = f(x)$ 를 돌려준다.



출처 : <https://wikidocs.net/24>



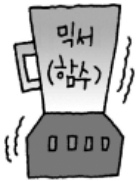
1.2 함수의 구조



함수 선언문

함수 이름 매개변수
`def salePrice(price):`

함수 내부



`result = price - price * 0.1`

`return result` 외부로 출력할 변수

외부로 출력 선언 / 함수 종료



매개변수 : 함수에 필요한 외부 변수를 받는 변수

Return : 외부로 출력 선언 / 함수 종료 선언

result(변수) : 외부로 출력할 변수. 언제든지 변경될 수 있음.



1.3 함수 정의 및 호출

가격과 할인율을 매개변수로 받는 할인 가격 계산 함수

```
def salePrice(price, rate):  
    result = price * (1 - rate/100)  
    return result
```

실행 결과

```
>>> salePrice(48000, 30)  
33600.0  
>>> salePrice(120000, 20)  
96000.0
```

salePrice(48000, 30)

매개변수 전달

```
def salePrice(price, rate):  
    result = price * (1 - rate/100)  
    return result
```



1.4 결과값을 반환하지 않는 함수

가격과 할인율을 매개변수로 받아 할인 가격을 출력하는 함수

```
def salePrice(price, rate):  
    result = price * (1 - rate/100)  
    print("할인 가격:", result)
```

← return 이 없는 것을 확인 가능

실행 결과

```
>>> salePrice(48000, 30)
```

```
할인 가격: 33600.0
```



1.5 매개변수 전달 예시

```
# 나를 소개하는 함수 매개변수
def myintro(name, univ, grade):
    grade = grade + 1
    print("나의 이름은 %s입니다." %name)
    print("%s대학교 %d학년 학생입니다." %(univ, grade))
```

실행 결과

```
>>> mygrade = 1
>>> myintro("홍길동", "한국", mygrade)
나의 이름은 홍길동입니다.
한국대학교 2학년 학생입니다.
>>> print(mygrade)
1
```



1.6 디폴트 매개변수 전달 예시

가격과 할인율을 매개변수로 받아 할인 가격을 출력하는 함수

```
def salePrice(price, rate=10):  
    result = price * (1 - rate/100)  
    print("할인 가격:", result)
```

디폴트 매개변수

← 따로 선언하지 않을 경우,
디폴트 값으로 설정.

실행 결과

```
>>> salePrice(48000)
```

```
할인 가격: 43200.0
```



1.7 가변 인수 예시

인수들의 합을 계산하는 함수

```
def sum_all(*args) ←  
    result = 0  
    for i in args  
        result += i  
    return result
```

정해지지 않은 여러 개의 변수 입력 가능
* 를 달아서 표기함

실행 결과

가변적인 개수

```
>>> sum_all(1, 2, 3, 4, 5)
```

15

```
>>> sum_all(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

55



2.1 지역 변수와 전역 변수

프로그램

전역변수 inum

함수 1

지역변수 inum_v1

함수 2

지역변수 inum_v2

사용 범위

지역변수 inum_v1 → 함수 1

지역변수 inum_v2 → 함수 2

전역변수 inum → 함수1, 함수2

지역변수는 선언된 함수 내에서,
전역변수는 프로그램 내에서 사용 가능



2.2 전역 변수 활용 예시

```
# 할인 가격 계산 함수를 활용하는 프로그램
rate = 20
# 할인율을 전역 변수로 사용한 할인 가격 계산
def salePrice(price):
    result = price * (1 - rate/100)
    return result

original = int(input("가격을 입력하세요:"))
print("원래 가격:", original)
print(rate, "% 할인 가격:", salePrice(original))
rate = 30          ←      Global에서 전역 변수 수정
print(rate, "% 할인 가격:", salePrice(original))
```

주의사항

함수 내에서 전역 변수를 사용할 때
수정은 불가하므로 주의 요망.

실행 결과

가격을 입력하세요:48000

원래 가격: 48000

20 % 할인 가격: 38400.0

30 % 할인 가격: 33600.0



2.3 함수 내 전역 변수 수정

전역 변수 변경을 시도하는 프로그램 2

```
rate = 20
```

```
def salePrice(price) :
```

```
    global rate    ←
```

Global의 전역 변수를 가져와서 수정하기 위한 선언

```
    rate = 30
```

혹은 rate = rate + 10

```
    result = price * (1 - rate/100)
```

```
    return result
```

```
print(salePrice(50000))
```

```
print(rate)
```

실행 결과

35000.0

30



3.1 내장 함수

자주 사용하는 연산자들을 제공

세 수 중 큰 수를 결정하는 프로그램

```
print("세 개의 수를 입력하세요")
val1 = float(input())
val2 = float(input())
val3 = float(input())
max_value = max(val1, val2, val3)
print("가장 큰 수: ", max_value)
```

실행 결과

세 개의 수를 입력하세요

5890.2

5980.4

3567.0

가장 큰 수: 5980.4



3.2 평균과 중앙값 계산 함수

평균 계산 함수

```
def mean(x):  
    return sum(x) / len(x)
```

실행 결과

```
>>> incomes = [8800, 3500, 5600,  
7500, 3900, 6000, 5200, 4100,  
9000, 6500]
```

```
>>> mean(incomes)  함수를 선언한 뒤,  
6010.0             이처럼 함수 실행까지 해야 함
```

```
>>> median(incomes)  
5800.0
```

중앙값 계산 함수

```
def median(x):  
    n = len(x)  
    x = sorted(x)  
    mid = n // 2  
    if n % 2 == 1:  
        return x[mid]  
    else:  
        low = mid - 1  
        high = mid  
        return (x[low] + x[high])/2
```



3.3 체질량지수 리스트 계산 함수

체질량지수 계산 함수

```
def bmi(height, weight):  
    return weight / (height/100 * height/100)
```

키와 몸무게 리스트를 받아 체질량지수 리스트를 계산하는 함수

```
def bmlist(height_weight_list) :  
  
    list = []  
  
    for h, w in height_weight_list:  
        list.append(bmi(h,w))  
  
    return list
```

실행 결과

```
>>> list1 = [(160,52), (162,65), (170,60), (157,50), (165,48)]  
>>> result = bmlist(list1)  
>>> print(result)  
[20.3125, 24.767565919829295, 20.761245674740483, 20.28479857195018,  
17.63085399449036]
```

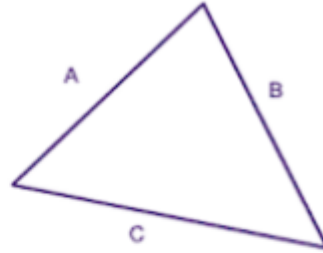
함수를 선언한 뒤,
이처럼 함수 실행까지 해야 함



3.4 삼각형 면적 계산 함수

Heron's Formula

$$S = \frac{A+B+C}{2}$$



$$\text{Area} = \sqrt{S(S-A)(S-B)(S-C)}$$

```
import math
```



라이브러리를 가져와서 함수 사용

```
# 헤론의 공식을 이용한 삼각형 면적 계산 함수
```

```
def heron(A, B, C):
```

```
    S = (A + B + C)/2
```

```
    area = math.sqrt(S * (S-A) * (S-B) * (S-C))
```

```
    return area
```

실행 결과

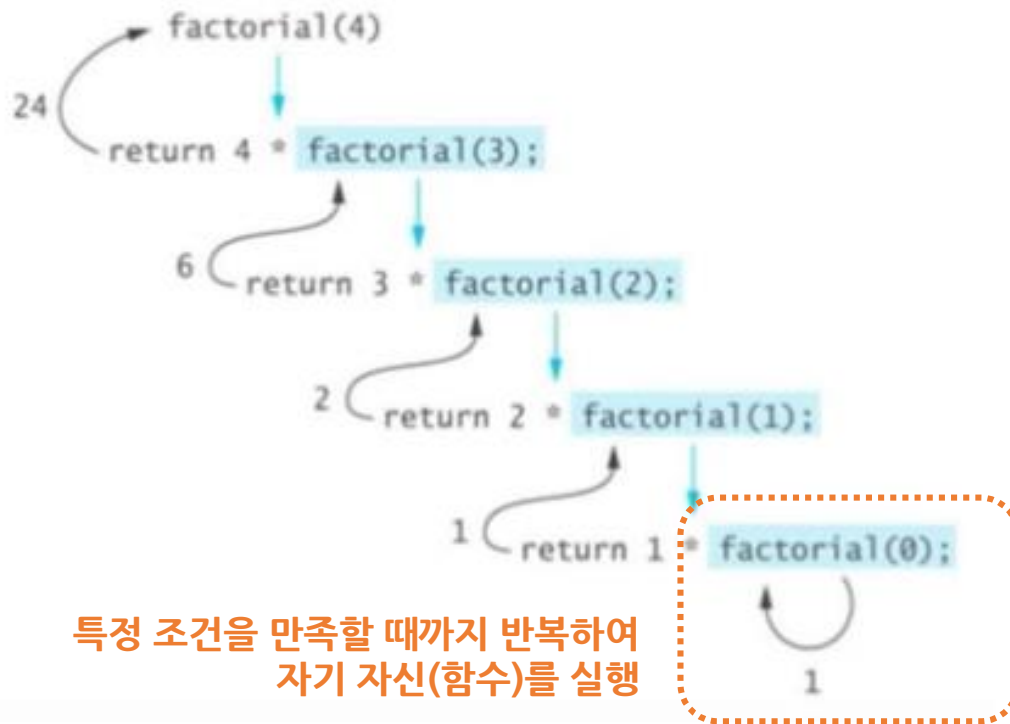
```
>>> heron(3,4,5)
```

```
6.0
```



4.1 재귀 함수

자기 자신을 사용하여 정의하는 함수



4.2 리스트에서 값 검색 함수

리스트에서 순차적으로 검색한다.

- 첫 번째 원소와 비교하여 찾으면 True 리턴
- 그렇지 않을 경우, 나머지 원소에 대하여 재귀 호출

재귀를 이용한 값 검색 함수

```
def search1(lst, key):  
    if not lst:                # 빈 리스트인 경우  
        return False  
    elif lst[0] == key:        # 찾고자 하는 원소를 찾은 경우  
        return True  
    else:                      # 리스트 나머지 부분에 대해서 검색  
        return search1(lst[1:], key)
```

실행결과

```
>>> search1([35, 28, 30, 29, 33, 31, 30], 30)  
True
```



4.3 리스트 내의 값의 개수 계산 함수

리스트에서 순차적으로 값을 검색하여 개수를 계산한다.

- 첫 번째 원소와 비교하고 나머지 원소에 대해서는 재귀 호출
- 총 개수 = 재귀 호출에서 찾은 개수(n) + 1 혹은 0

리스트 내의 값의 위치 검색 함수

```
def search2(lst, key):
```

```
    if not lst:
```

```
        return False
```

```
    elif lst[0] == key:
```

```
        n = search2(lst[1:], key)
```

```
        return n+1
```

```
    else:
```

```
        n = search2(lst[1:], key)
```

```
        return n
```

실행결과

```
>>> search2([35, 28, 30, 29, 33, 31, 30], 30)
```

```
2
```



4.4 리스트 내의 값의 위치 검색 함수

리스트에서 순차적으로 값을 검색하여 위치를 계산한다.

- 첫 번째 원소와 비교하고, 나머지 원소에 대해서 재귀 호출
- 총 개수 = 재귀 호출에서 찾은 개수(n) + 1 혹은 0

리스트 내의 값의 위치 검색 함수

```
def search2(lst, key):
```

```
    if not lst:
```

```
        return False
```

```
    elif lst[0] == key:
```

```
        return 0
```

```
    else:
```

```
        n = search2(lst[1:], key)
```

```
        if n is False:
```

```
            return False
```

```
        else:
```

```
            return n+1
```

실행결과

```
>>> search2([35, 28, 30, 29, 33, 31, 30], 30)
```

```
2
```



과제 공지

소프트웨어의 이해



과제 제출

- 과제 제출 기한
 - 실습 **다음주 수요일(5월 3일) 오후 11시 00분**까지
- 제출 장소
 - **Snowboard** 해당 실습 과제 제출 페이지에 업로드

★ 표절 검사 및 기한 내 제출 필수!

※ 기한 내 제출은 만점을 기준으로 성적이 반영되나, 기간 외 제출은 점수 조정이 들어갑니다.
기간 이후 1주일 : 50점 만점 기준. / 그 이후 제출 : 5점 이하.

#1. 근로소득세 계산 프로그램2

파일 이름 : tax.py

- Lab03-1 에서 작성한 코드를 활용하여 근로소득세를 계산하는 함수를 작성하시오.
- 이 함수는 과세 표준 금액을 매개변수로 받아 근로소득세를 계산하여 반환한다.

- 함수 이름 : tax
- 매개변수 값 입력 : 0 이상의 정수
- 반환 값 : 계산된 근로소득세 (정수)
- 계산 결과는 만원 단위의 정수로 계산

```
===== RESTART:
=====
>>> tax(5500)
798
>>> tax(10000)
2010
>>> tax(50000)
17460
>>> |
```

과세 표준 금액	세율
1,200만원 이하	과세 표준 금액의 6%
1,200만원 초과~4,600만원 이하	72만원 + 1,200 만원 초과 금액의 15%
4,600만원 초과~8,800만원 이하	582 만원 + 4,600 만원 초과 금액의 24%
8,800만원 초과~3억원	1,590 만원 + 8,800 만원 초과 금액의 35%
3억원 초과	9,010 만원 + 3억원 초과 금액의 38%

#1. searchn 함수

파일 이름 : searchn.py

- 리스트 내에서 key 값의 위치 (index) 를 모두 찾아 리스트로 반환하는 재귀 함수 searchn 을 작성하시오.
(단, 반드시 재귀 함수로 구현할 것)

- 함수 이름 : searchn(list, key)
- 매개변수 : ① 임의의 정수 값이 1개 이상 들어있는 리스트
② 찾고자 하는 key 값
- 반환 값 : ① 리스트 에서 모든
② key 의 위치(index)값을 저장한 리스트

```
===== RESTART: D:\뽕소프트웨어
>>> searchn([35,28,30,29,33,31,30],30)
[2, 6]
>>>

===== RESTART: D:\뽕소프트웨어
>>> searchn([2,4,6,8,0,2,4,6,8,0],0)
[4, 9]
>>>
```

과제 제출 주의사항

소프트웨어의 이해



과제 제출 주의사항

- 과제 제출 기한
 - 실습 다음주 수요일 오후 11시 00분까지
- 제출 장소
 - **Snowboard** 해당 실습 과제 제출 페이지에 업로드




★ 표절검사!

과제 제출

- 과제 파일 형식

- 소스파일(.py)과 보고서(.pdf)를 한 폴더에 넣고 압축(.zip)하여 제출
- 파일명 : **Lab과제번호_학번_이름.zip**
ex) Lab01_2201234_김눈송.zip

*** 반드시 압축 파일 이름과 내부 파일 이름을 지켜 주시기 바랍니다***

 Lab01_2201234_김눈송	2022-03-02 오후 9:00	Microsoft Edge P...
 octagon	2022-03-02 오후 8:40	Python File
 triangle	2022-03-02 오후 8:52	Python File

- 소스파일(.py)

- 파일명 : 매 실습마다 제공하는 실습 자료 이름
ex) triangle.py

- 보고서(.pdf)

- 파일명 : Lab과제번호_학번_이름 .pdf
ex) “Lab01_2201234_김눈송.pdf”
- 보고서 포함 항목 : 문제-기능별 실행화면 캡처, 소스코드 텍스트

과제 질문 주의사항



조교 이메일
nayeonjo@sookmyung.ac.kr

1. 질문이 생길 경우, **Q&A 게시판에 글 작성**
2. Q&A 게시판 질문 시, **반드시 설명과 함께 질문**
3. **과목, 분반, 전공, 이름, 학번 작성 필수**
4. 주말에는 메일 답장이 없을 수 있음
5. 실습 제출 마감 당일에는 답장이 늦을 수 있음

Office Hour

메일로 약속 시간을 미리 정하고 방문해주세요.

메일 예시

컴퓨터학과 전공 2301234 소프트웨어의 이해 3분반
김눈송 입니다.

n번 과제의 코드 4번째 줄에서 에러가 발생해요.
or
1번 과제에서 테스트 파일 업로드 과정에서 문제가
발생했어요.

코드를 첨부해서 메일 보내드려요.