

# 씨앗 beginner

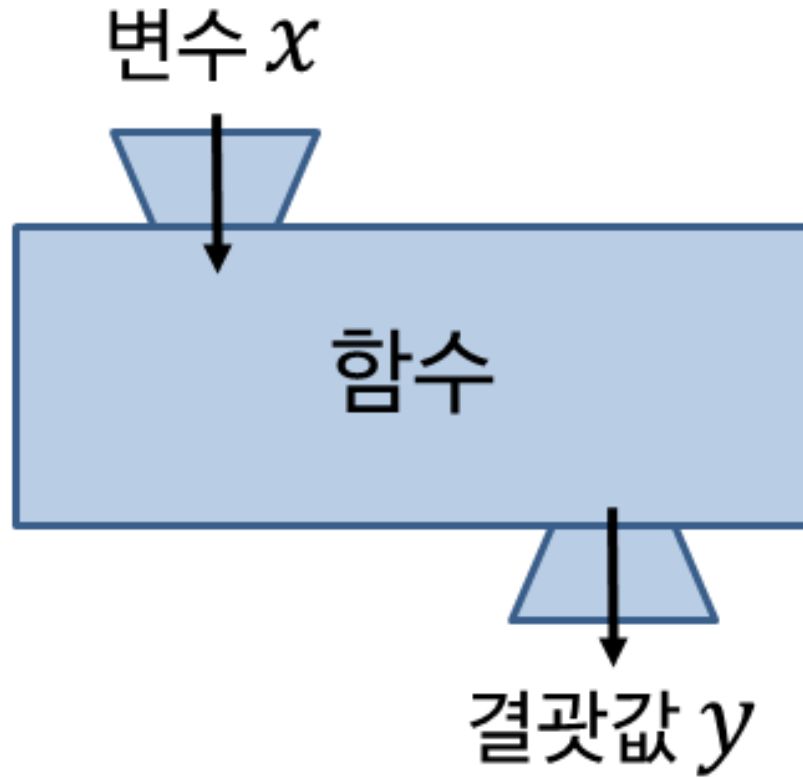
Week 3  
Date : 23.03.30  
Made by 은재민

# 오늘의 진도

1. 함수란 무엇인가
2. 함수를 직접 만들어보자.
3. 재귀 recursion
4. 재귀함수를 사용해보자.
5. 예제

# 함수 function 는 무엇인가?

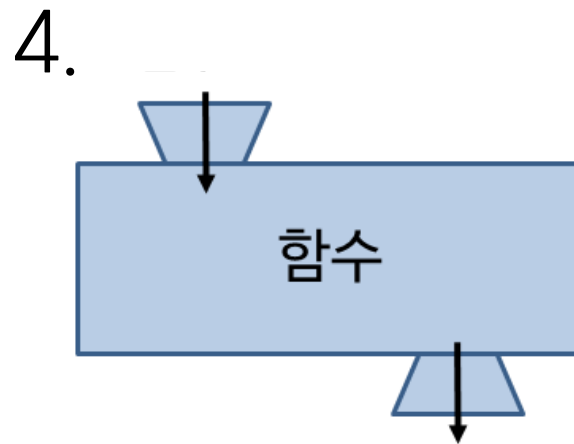
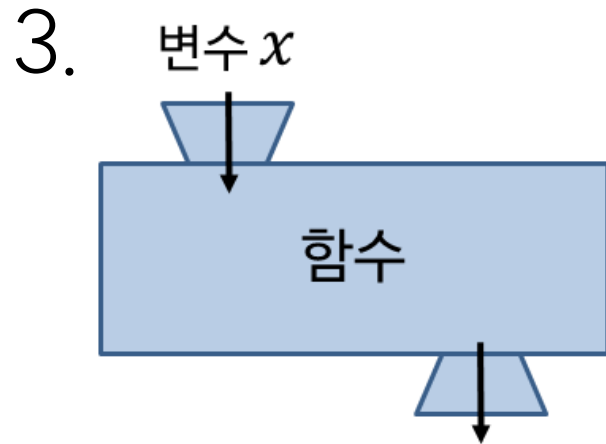
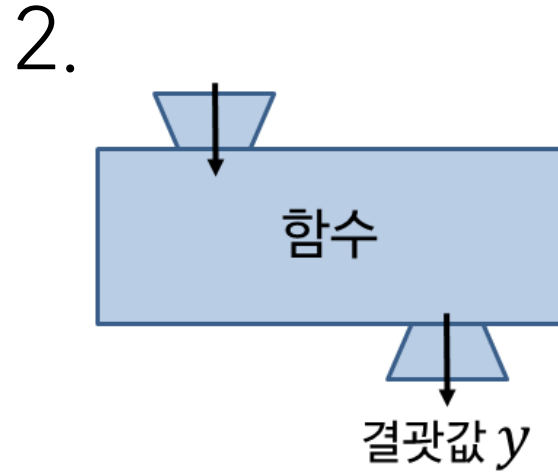
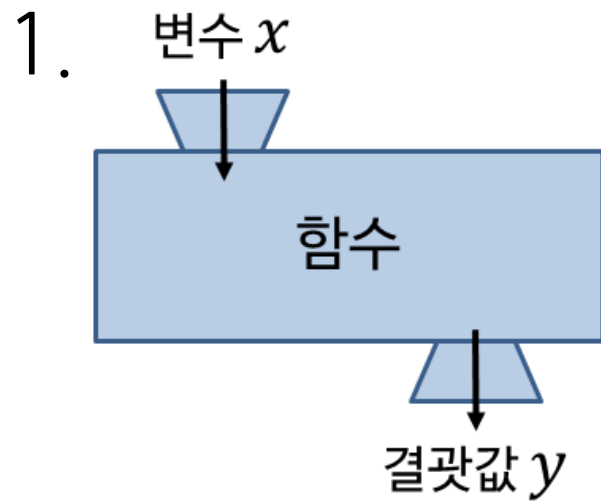
1. 입력을 받고
2. 정해진 행동 수행
3. 결과



구기틀을 생각하면 쉬움!



# 함수의 종류(입력과 출력, 없을 수도?)



함수를 직접 만들어보자.

```
def function(par):  
    action1..  
    action2..  
    return result
```

<pre>def : definite, 정의하다 par : parameter, 인자 return : 반환하다 result : 결과</pre>
---

# 함수를 사용해야 하는 이유?

1. 똑같은 행동을 반복해서 수행 할 때.
2. 코드의 가독성 증가
3. 메모리 효율성 증가
4. 만든 함수를 여러 프로그램에 다양하게 활용가능!
5. 재귀적 호출

예제 1. 계산기 프로그램!

```
def calculation( num1, operator, num2) :
```

예제 2. 리스트에 있어?

```
def in(array, num) :
```

예제 3. 계속 말하기

```
def repeatSentence(time, str) :
```

# 재귀 호출 recursive call 은 무엇인가

함수가 자기 자신을 다시 호출하는 것.

ex)

```
def function():  
    ...  
    return function()
```



# 재귀와 반복

모든 재귀호출 = 반복문으로 구현가능

모든 반복문 = 재귀호출로 구현 가능

그럼 재귀는 왜 써야함?

# 재귀는 언제 사용할까?

1. 점화식이나 분할정복의 **가독성**을 위해서
2. **변수의 개수를 줄인다.**
3. 알고리즘을 구현할 때, 반복문을 일일이 작성하기보다  
재귀함수를 사용하는 것이 **더 직관적이고 이해하기 쉬운 때가 많음**

# 재귀 사용의 주의할 점

1. 종료 조건이 없으면 오류가 발생(무한 함수 호출)  
→ RecursionError(최대 재귀깊이 초과 오류)
2. 비효율적(시간)  
→ 반복문이상의 시간이 필요

<https://www.acmicpc.net/problem/2747>

# 예제

- <https://www.acmicpc.net/problem/10872>
- <https://www.acmicpc.net/problem/10870>

SELF :

$$A_0 = 0, A_1 = 1$$
$$A_i = A_{i-2} \times A_{i-1} + A_{i-1}$$

$$A_n = ???$$

Lv. 변태

<https://www.acmicpc.net/problem/11729>