

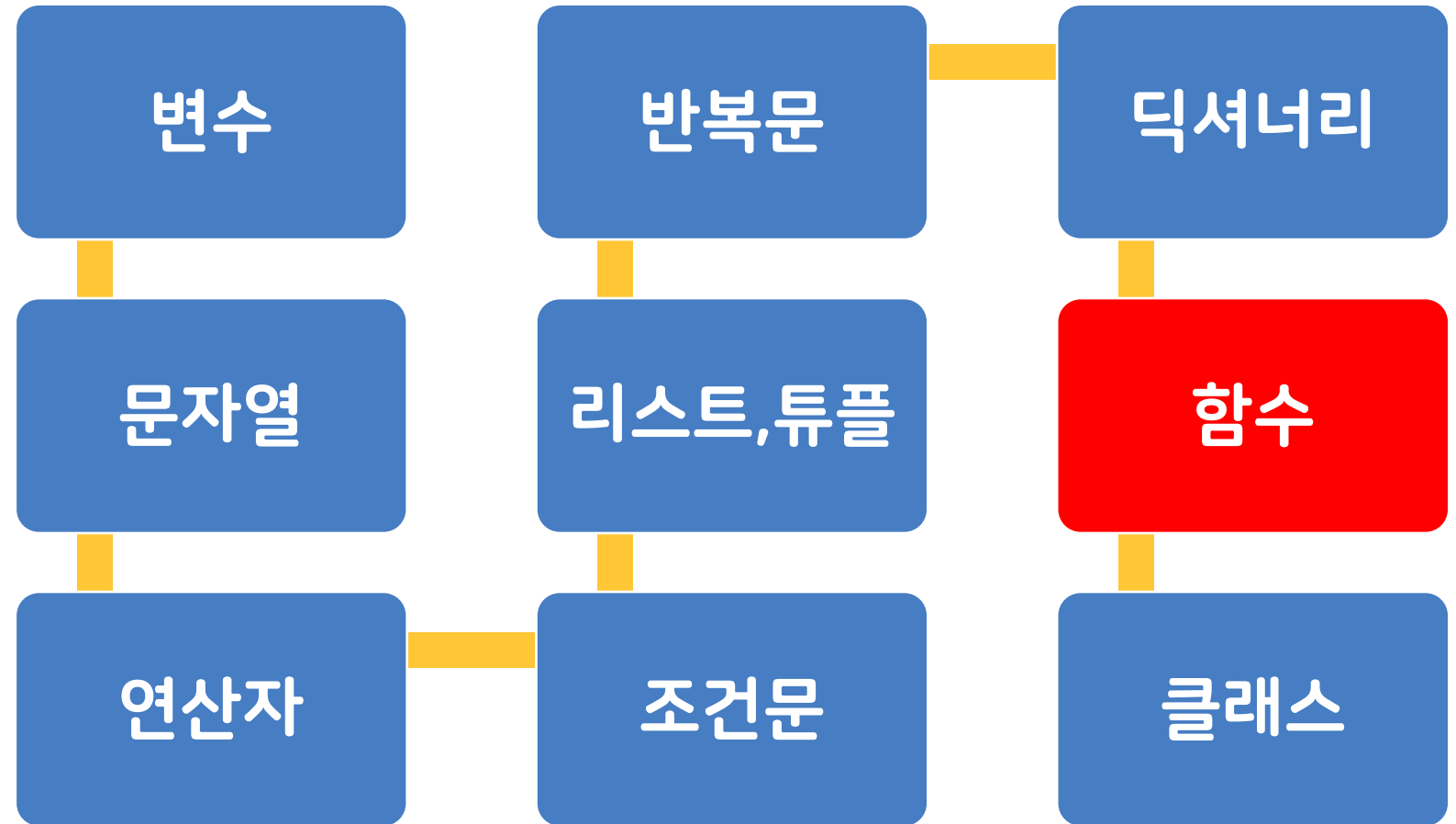


스마트인재개발원
Smart Human Resources Development

최성우 연구원



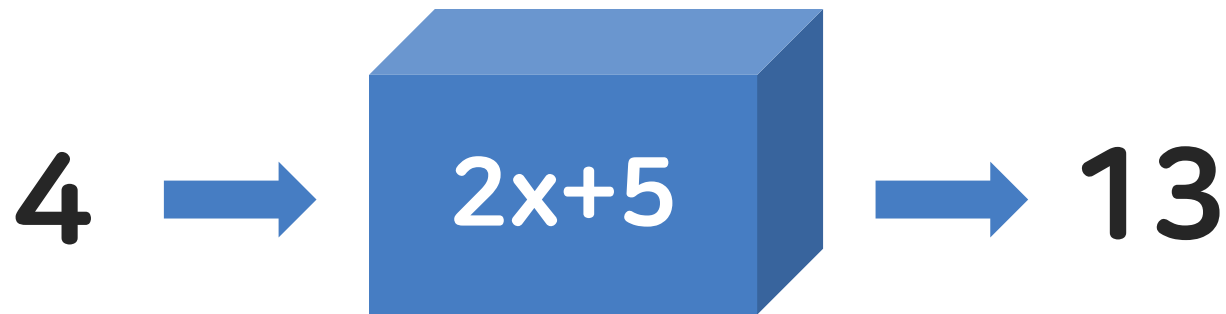
수업 진행방향

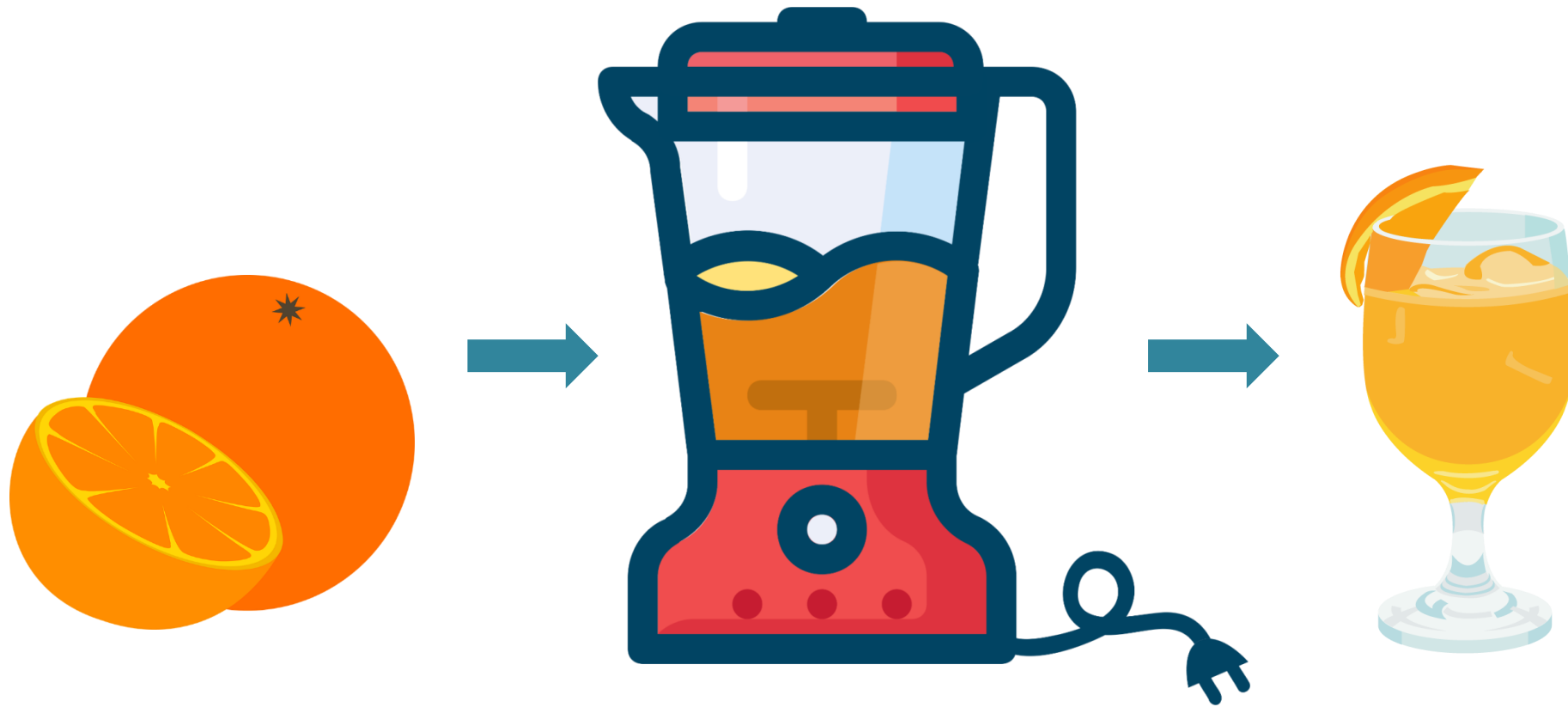




학습목표

- 함수에 대해 알 수 있다.
- 함수를 정의하고 활용 할 수 있다.





함수란?

- 하나의 특별한 목적의 작업을 수행하기 위해 독립적으로 설계된 코드의 집합

함수를 사용하는 이유

- 반복적인 프로그래밍을 피할 수 있어 **효율**이 높아진다.
- **모듈화**로 인해 전체적인 코드의 **가독성**이 좋아진다.
 - ※ 모듈화 : 프로그래밍 언어로 프로그램을 제작시, 최적화 및 관리에 용이하게 기능 단위로 분할하는 것
- 프로그램에 문제가 발생하거나 기능 변경이 필요할 때에도 손쉽게 **유지보수**가 가능하다.

define : 정의를 내리다

def

함수명(매개 변수):

(colon, 콜론)

실행문장

return 반환 변수

들여쓰기 (Tab, Space*4)

함수호출

- 함수명(인수1, 인수2)

```
def 함수명(매개변수):  
    실행문장  
    return 반환변수
```

```
def number_sum(num1, num2):  
    result = num1 + num2  
    return result
```

```
number_sum(3, 10)
```

13

```
number = number_sum(3, 10)  
number
```

13

두 수를 입력 받아서 차이를 return하는 함수를 정의하시오.

(함수명 : number_minus)

```
num1 = int(input("첫 번째 정수 입력 >> "))  
num2 = int(input("두 번째 정수 입력 >> "))  
result = number_minus(num1, num2)  
result
```

첫 번째 정수 입력 >> 10

두 번째 정수 입력 >> 3

7

문자열을 입력 받아 'ㅋ'을 제거하고 return하는 함수를 정의하시오.

```
s = input("문자열 입력 >> ")  
result = s_replace(s)  
result
```

문자열 입력 >> ㅋ을 모두 지워주세요ㅋㅋㅋㅋㅋㅋ

'을 모두 지워주세요'

두 수를 입력 받아서 원하는 연산을 수행하는 함수를 정의하시오.

```
num1 = int(input("첫 번째 정수 입력 >> "))  
num2 = int(input("두 번째 정수 입력 >> "))  
op = input("연산자 입력(+, -) >> ")  
result = cal(num1, num2, op)  
print("결과 : {}".format(result))
```

```
첫 번째 정수 입력 >> 5  
두 번째 정수 입력 >> 3  
연산자 입력(+, -) >> +  
결과 : 8
```

```
첫 번째 정수 입력 >> 5  
두 번째 정수 입력 >> 3  
연산자 입력(+, -) >> -  
결과 : 2
```

독스트링(docstring)

- 함수의 설명을 작성 (Shift + <Tab>)

```
def cal(num1, num2, op):  
    """덧셈과 뺄셈을 계산하는 함수"""  
    if op=='+':  
        return num1+num2  
    else:  
        return num1-num2
```

cal()

Signature: cal(num1, num2, op)

Docstring: 덧셈과 뺄셈을 계산하는 함수

약수를 구하는 함수를 정의하시오.

```
divisor(10)
```

```
1 2 5 10
```

```
divisor(32)
```

```
1 2 4 8 16 32
```

```
divisor(100)
```

```
1 2 4 5 10 20 25 50 100
```

가변 매개변수(variable parameters)

- 함수 호출 시 몇 개의 인수가 입력될지 알 수 없다면, 사용자가 직접 입력하는 매개변수의 개수를 정할 수 있도록 선언

```
def 함수명(*매개변수):  
    실행문장  
    return 반환변수
```

가변 매개변수(variable parameters)

- 전달된 모든 인수는 튜플(tuple)의 형태로 저장

```
def add(*args):  
    print(args)
```

args → arguments
인수들 이라는 뜻

```
add(1,2,3)
```

```
(1, 2, 3)
```

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

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

가변 매개변수를 활용해 모든 숫자를 더해서 반환하는 함수를 작성하시오.(sum() 이용)

```
def add(*args):  
    ?
```

```
add(1,2,3)
```

6

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

55

데이터 전처리로 텍스트의 "ㅋ"을 모두 제거하고 싶을 때

- 데이터는 review_naver, review_google 에 대입되어 있음

```
review_naver = ["월 스미스가 하드캐리ㅋㅋㅋㅋㅋㅋ",  
                "자스민 너무 멋지고 팬 엄청나게 생길듯ㅋㅋ",  
                "기대보다 더욱 재미있는 영화였다^^"  
                ]  
review_google = ["색감도 노래도 너무 화려하고 재밌었어요ㅋㅋ",  
                 "오늘부터 디즈니 팬입니다ㅋㅋ",  
                 "디즈니의 새로운 해석도 놀랍고, 월스미스도 신의한수!"  
                 ]
```

```
1 review_naver.replace("ㅋ", "")
```

```
-----  
AttributeError                                Traceback (most recent call last)  
<ipython-input-3-289d5f4288b1> in <module>  
----> 1 review_naver.replace("ㅋ", "")  
  
AttributeError: 'list' object has no attribute 'replace'
```

데이터 전처리로 텍스트의 "ㅋ"을 모두 제거하고 싶을 때

- for문을 활용하여 데이터에 하나하나 접근해야 함

```
1 for i in review_naver :  
2     print(i.replace("ㅋ", ""))  
3 for i in review_google :  
4     print(i.replace("ㅋ", ""))
```

월 스미스가 하드캐리
쟈스민 너무 멋지고 팬 엄청나게 생길듯
기대보다 더욱 재밌는 영화였다^^
색감도 노래도 너무 화려하고 재밌었어요
오늘부터 디즈니 팬입니다
디즈니의 새로운 해석도 놀랍고, 월스미스도 신의한수!

데이터 전처리로 텍스트의 “ㅋ”을 모두 제거하고 싶을 때

- “ㅋ”을 제거하는 함수를 활용

```
1 def string_replace(num_list) :  
2     for i in num_list :  
3         print(i.replace("ㅋ", ""))
```

```
1 string_replace(review_naver)  
2 string_replace(review_google)
```

월 스미스가 하드캐리
쟈스민 너무 멋지고 팬 엄청나게 생길듯
기대보다 더욱 재밌는 영화였다^^
색감도 노래도 너무 화려하고 재밌었어요
오늘부터 디즈니 팬입니다
디즈니의 새로운 해석도 놀랍고, 월스미스도 신의한수!



다음시간에는?