

`/* elice */`

핵심 파이썬 기초 프로그래밍

같은 듯 다른 함수와 메서드



이호준 선생님

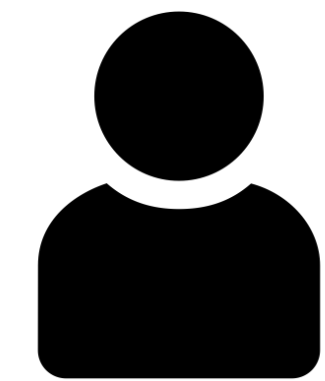
함수

프로그래밍의 기본 틀

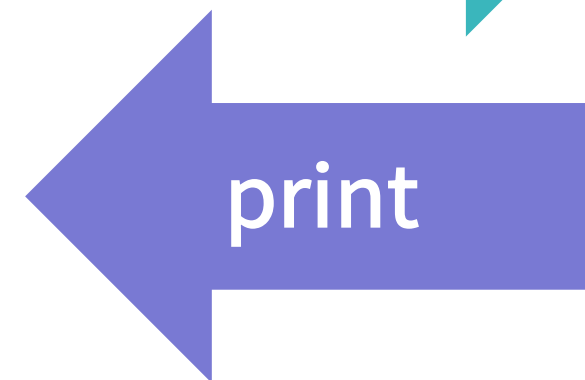
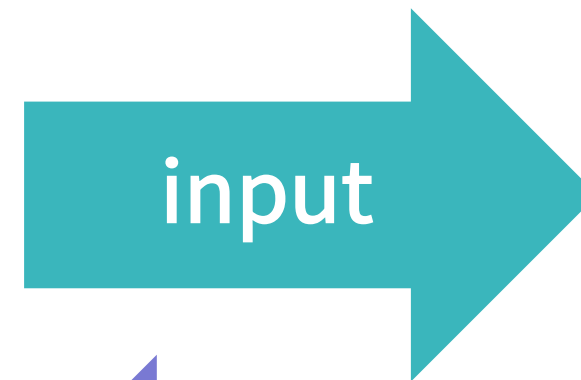
컴퓨터에게 정보를 **입력**하고.

컴퓨터가 **작업**을 하고

컴퓨터가 작업 결과를 **출력**

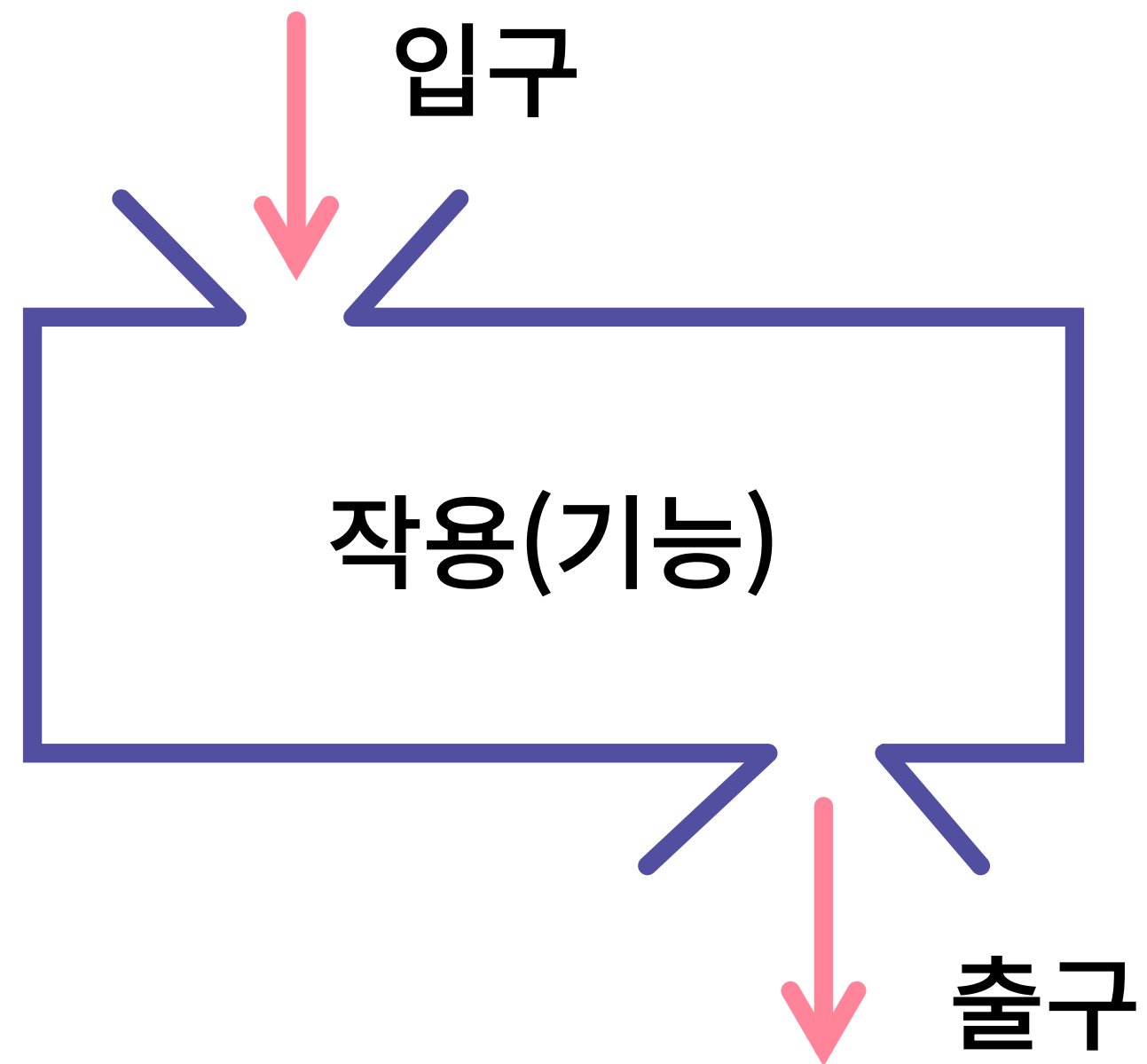


사용자



컴퓨터

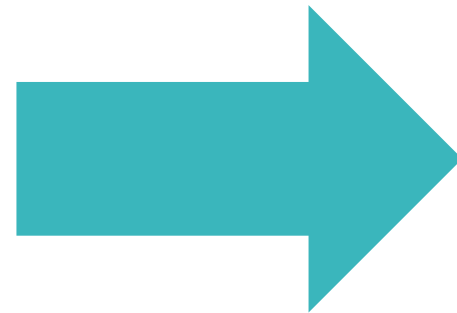
프로그래밍의 기본 틀



함수

특정 기능을 수행하는 코드(들의 모임)

라면 끓이기



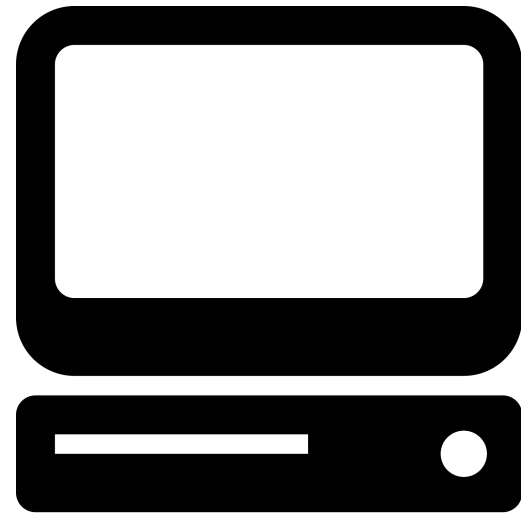
냄비를 가스레인지에 올리기

가스레인지의 불을 켜기

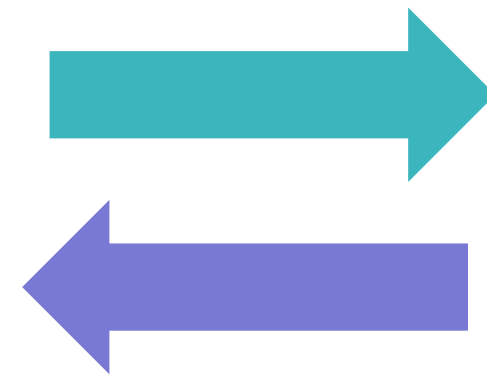
냄비에 물을 550mL 붓기

...

함수의 구조



컴퓨터



함수

함수의 구조

len()

자료를 넣으면 그 **자료의 길이**를 알려준다

int()

자료를 넣으면 **정수형으로 변환**해서 알려준다

str()

자료를 넣으면 **문자열로 변환**해서 알려준다

함수의 종류

내장 함수

파이썬 개발자들이 이미 만들어 둔 함수들
편리하게 가져다 쓰면 된다

```
print()
```

```
len()
```

```
max()
```

```
int()
```

input()과 print()

input() – 자료를 입력하는 함수

print() – 자료를 출력하는 함수

```
a = input() # Hello 입력
```

```
print(a) # Hello
```

max()와 min()

max() – 시퀀스 자료의 최대값을 구하는 함수

min() – 시퀀스 자료의 최솟값을 구하는 함수

```
print(max(1, 2, 3, 4, 5)) # 5
```

```
print(min([1, 2, 3, 4, 5])) # 1
```

sum()과 len()

sum() – 숫자 원소로 이루어진 시퀀스 자료의 합

len() – 시퀀스 자료의 길이를 구하는 함수

```
print(sum(1, 2, 3, 4, 5)) # 15
```

```
print(len("Triangle")) # 8
```

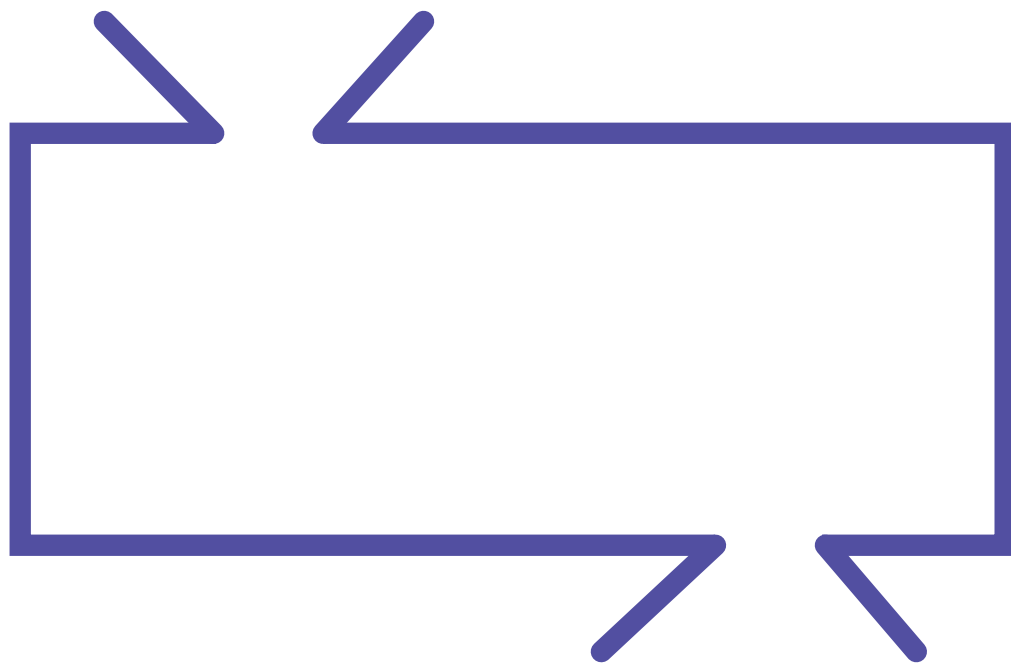
사용자 지정 함수

사용자가 여러 코드를 묶어서 **새로 만든 함수**

```
def plusDouble(a, b):  
    c = a+b  
    return 2*c  
  
print(plus(3, 4)) # 14
```

함수 만들기

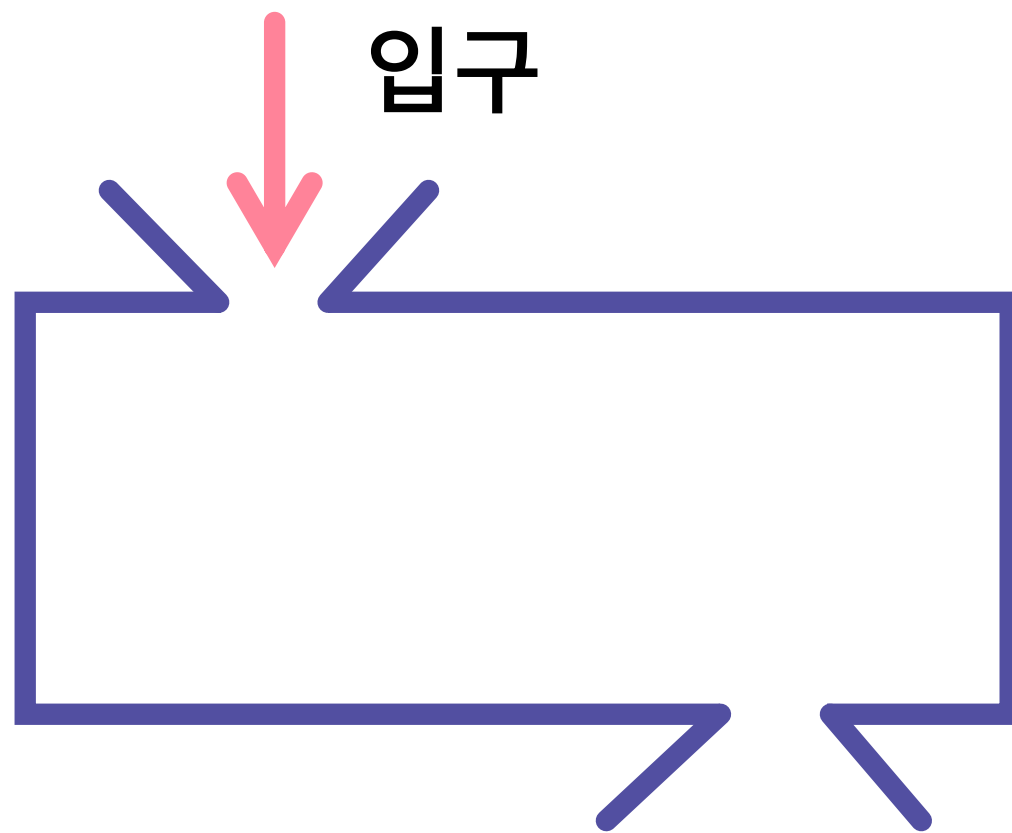
define(정의하다) 키워드를 이용해서 함수 정의



```
def 함수이름(매개변수):  
    <수행할 명령>  
    . . .  
    return 반환값
```

함수의 입력

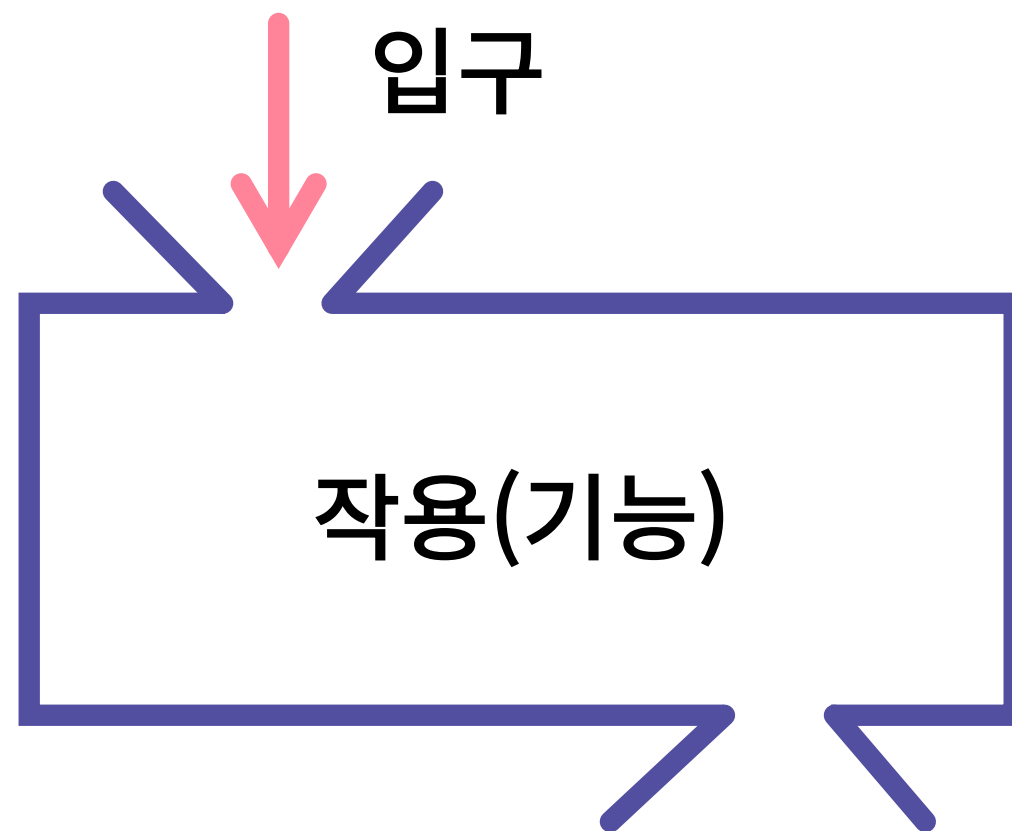
매개변수를 이용해서 함수 내부로 값을 전달



```
def 함수이름(매개변수):  
    <수행할 명령>  
    . . .  
    return 반환값
```

함수 속 명령 작성

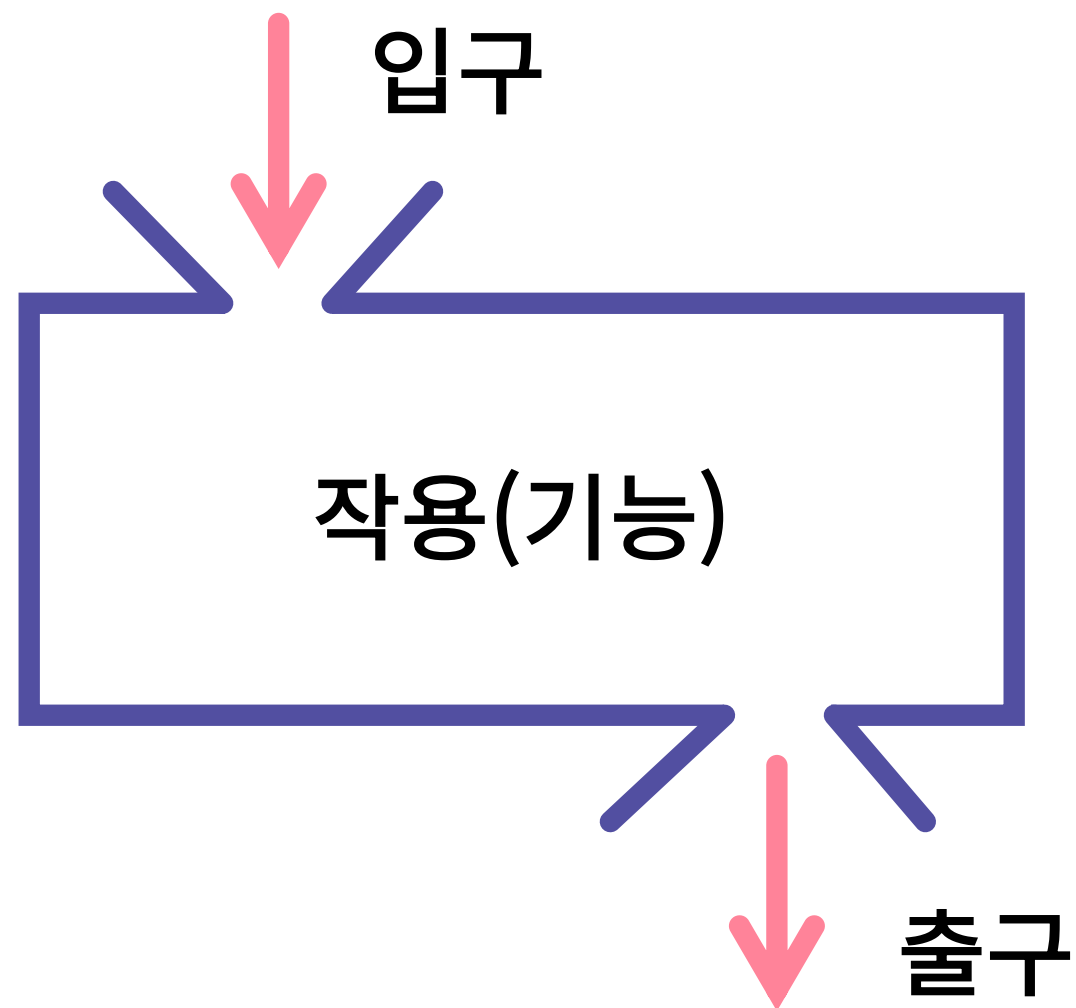
같은 들여쓰기를 통해 명령 작성



```
def 함수이름(매개변수):  
    ____<수행할 명령>  
    ____.  
    ____.  
    ____return 반환값
```


함수의 반환(출력)

return을 이용해서 함수 외부로 값을 전달



```
def 함수이름(매개변수):  
    <수행할 명령>  
    . . .  
    return 반환값
```

왜 반환이 필요할까?

함수 내부에서 일어난 일은 함수 외부에서 알 수 없다!

→ 반환을 통해 외부로 전달!

```
def plus(a, b):  
    c = a+b  
    return c
```

```
print(plus(3,4))
```

```
def plus(a, b):  
    c = a+b  
    return c
```

```
print(7)
```

Method(메서드)

메서드(Method)

특정 자료에 대해 특정 기능을 하는 코드

```
my_list = [1, 2, 3]
```

```
my_list.append(4)
```

```
my_list.count(2)
```

```
my_list.pop()
```

함수 vs 메서드

함수는 특정 기능을 한다
(매개변수를 이용해 자료를 전달해준다)

```
my_list = [1, 2, 3]
```

```
len(my_list)
```

```
sum(my_list)
```

```
min(my_list)
```

함수 vs 메서드

메서드는 **특정 자료와 연관 지어** 기능을 한다
(자료 뒤에 **.을 찍어** 사용한다)

```
my_list = [1, 2, 3]
```

```
my_list.sort()
```

```
my_list.pop()
```

```
my_list.clear()
```

/* elice */

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice