



# [10주차 2강] 함수 (1)



# 학습 내용

8.1 함수 개요

8.2 함수 정의



# 학습 목표

- 8.1 함수가 무엇인지 이해한다.
- 8.2 함수 정의가 무엇인지 이해하고,  
함수를 정의하는 방법을 익힌다.





## 8.1 함수 개요

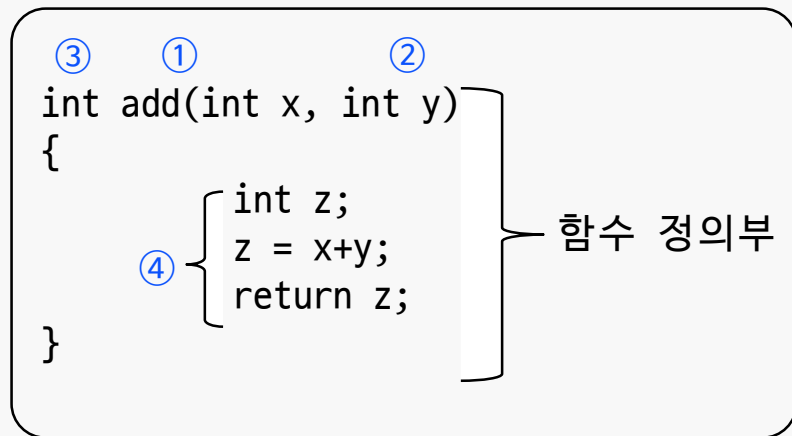
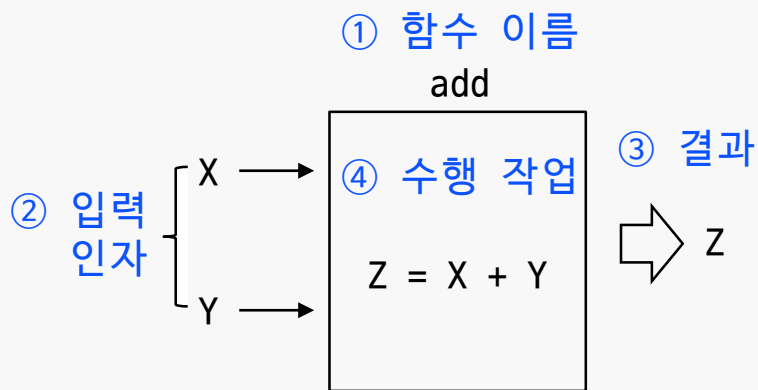
## 8.2 함수 정의





## 수학의 함수와 C언어의 함수 개략적 비교

- $\text{add}(x, y) = x + y$  : 두 수의 합을 구하는 함수
  - ✓ 함수(마술상자) `add`에 `x` 와 `y`를 입력하면 두수의 합이 결과로 나옴



- 구성요소 4가지가 전부 포함됨을 짐작할 수 있음



## C언어에서 함수

- 어떤 특정한 일을 수행하는 독립적인 단위
- 함수의 예 : printf(), scanf(), main()
- 함수는 함수 정의부와 함수 호출부로 구성
  - ✓ **함수 정의부** : 함수를 구현하는 부분(즉, 마술상자 내부 구현)  
예) main() 함수 구현
  - ✓ **함수 호출부** : 함수의 기능을 이용하는 부분(즉, 마술상자 이용)  
예) printf() 함수 호출

## 8.2 함수 정의



### 함수 정의

- 함수가 수행해야 할 기능 명세하기
- 구문

```
반환형    함수이름    (인자선언1, 인자선언2, ...)  
{  
    함수의 수행코드  
  
    return 문;           ⇒ 필요 시  
}
```

```
int add (int x, int y)  
{  
    int z;  
    z = x+y;  
  
    return z;  
}
```



### 함수 정의 요소

#### 1. 함수 이름

- ✓ 함수 이름 규칙은 변수명 규칙과 동일(식별자)
- ✓ 의미 있는 이름 사용 권장

#### 2. 함수 인자 or 매개 변수(parameter)

- ✓ 함수의 입력 데이터를 나타내는 변수들
- ✓ 인자가 다수인 경우 콤마(,)로 구분
- ✓ 인자가 없더라도 소괄호는 반드시 필요

#### 3. 반환형(return type)

- ✓ 함수는 수행된 후에 반환하는 결과의 자료형
- ✓ 아무 결과도 반환하지 않으면 자료형에 **void**를 씀

▪ 참고) 위 세 구성 요소는 함수의 형태 명세 → 함수 헤더라 칭함

```
int add (int x, int y)
{
    int z;
    z = x+y;

    return z;
}
```





### 함수 정의 요소

#### 4. 함수 몸체(body)

- ✓ 함수가 수행해야 할 작업을 { } 안에 명세
- ✓ 함수는 제어 흐름에 따라 수행:  
맨 마지막 문장까지 수행 되거나  
**return** 문을 만나면 종료

```
int add (int x, int y)
{
    int z;
    z = x+y;

    return z;
}
```



### 함수 정의 요소

- **return 문**
  - ✓ 함수를 종료하고 함수의 결과를 반환하는 역할
  - ✓ 반환형이 void 인 경우 보통 return 문 사용하지 않지만, 중간에 함수를 종료시키기 위해 쓰기도 함
- return z;      ⇨ 변수 z에 저장된 값 반환
- return 10+20;    ⇨ 10+20, 즉 30을 반환
- return;          ⇨ 아무 것도 반환하지 않음  
(반환형이 void 인 경우 사용)



### 함수 정의 예시 1

```
char next_char(char c, int num)
{
    char c1;

    . . .

    return c1;
}
```

1. 반환형 : **char**
2. 함수 이름 : next\_char
3. 함수 인자 : **char형 변수 c, int형 변수 num**

※ 반환형이 char형이기 때문에,  
return문이 없으면 컴파일 오류 발생!!

### 함수 정의 예시 2

```
void print_heading( void )
{
    printf("\n=====\\n");
    printf("    heading    ");
    printf("\n=====\\n");
}
```

1. 반환형 : 없음(void)
2. 함수 이름 : print\_heading
3. 함수 인자 : 없음(void)

※ 반환형이 void형이므로 return문 없어도 됨

※ 인자가 없는 경우, void를 명시적으로 적어도 되고, 생략해도 됨



## [예제 8.1]

다음 조건을 만족하는 함수를 정의 하시오.  
(여기서 정의한 함수는 이후 예제에서 활용됨)

### 1. max() 함수

- 함수 이름 : max
- 인자 : int형 변수 a와 b
- 반환형 : int형
- a와 b 중 큰 값을 반환



## [예제 8.1]

다음 조건을 만족하는 함수를 정의 하시오.  
(여기서 정의한 함수는 이후 예제에서 활용됨)



### 2. print\_characters() 함수

- 함수 이름 : print\_characters
- 인자 : char형 변수 c와 int형 변수 n
- 반환형 : void
- 하나의 줄에 변수 c의 문자를 n개 출력



## [예제 8.1]

다음 조건을 만족하는 함수를 정의 하시오.  
(여기서 정의한 함수는 이후 예제에서 활용됨)



### 3. divide 함수

- 함수 이름 : divide
- 인자 : int형 변수 a와 b
- 반환형: double
- a를 b로 나눈 결과를 반환. 단, 실수 연산을 해야 함
- 예를 들어  $3/2$ 의 경우 1.5를 반환



## [예제 8.1]

다음 조건을 만족하는 함수를 정의 하시오.  
(여기서 정의한 함수는 이후 예제에서 활용됨)



### 4. add3 함수

- 함수 이름 : add3
- 인자 : double형 변수 a, b, c
- 반환형 : double
- a, b, c의 합을 반환





## [예제 8.1]

다음 조건을 만족하는 함수를 정의 하시오.  
(여기서 정의한 함수는 이후 예제에서 활용됨)



### 5. atoA 함수

- 함수 이름 : atoA
- 인자 : char형 변수 ch
- 반환형 : char
- 소문자인 ch를 대문자로 변환하여 반환  
(인자 ch는 항상 소문자라고 가정)

# 학습 정리

- **함수**는 특정한 일을 하는 하나의 **독립된 단위**로, 프로그램은 함수들로 구성됨
- **함수 정의**는 크게 **헤더**와 **몸체**로 구성됨
- **함수의 헤더**는 함수의 형태를 표현하는 반환형, 함수 이름, 인자 목록으로 구성됨
- **함수의 몸체**는 함수의 동작을 구현하는 코드로 중괄호 {} 안에 명세됨