

프로그래밍을 통한 논리적 사유 연습



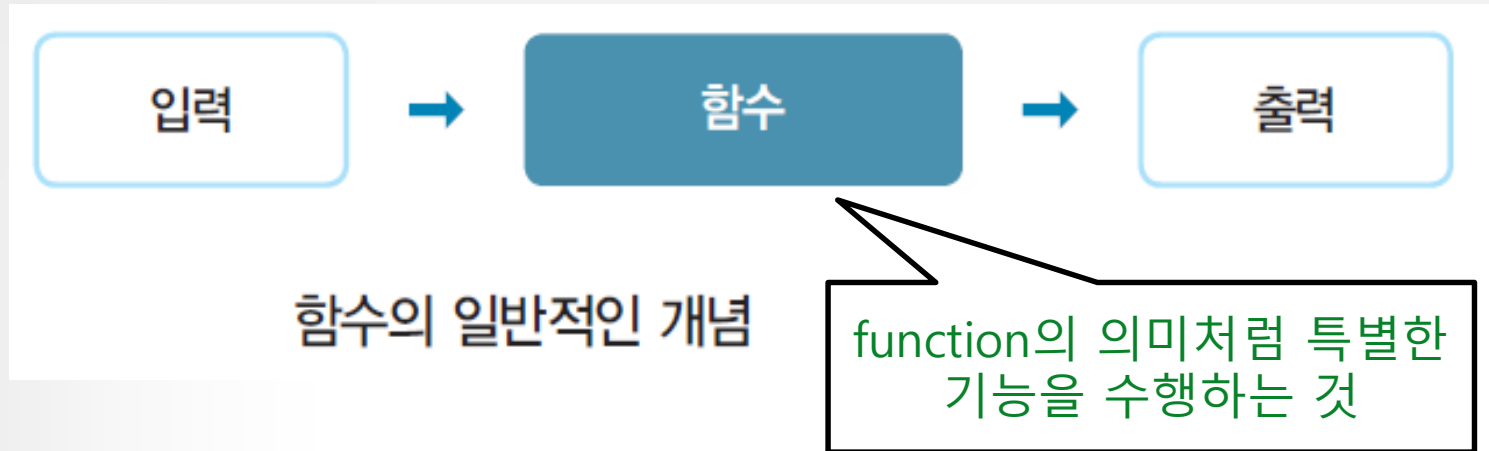


11장. 함수

(정의, 선언 등)

함수의 정의와 기능

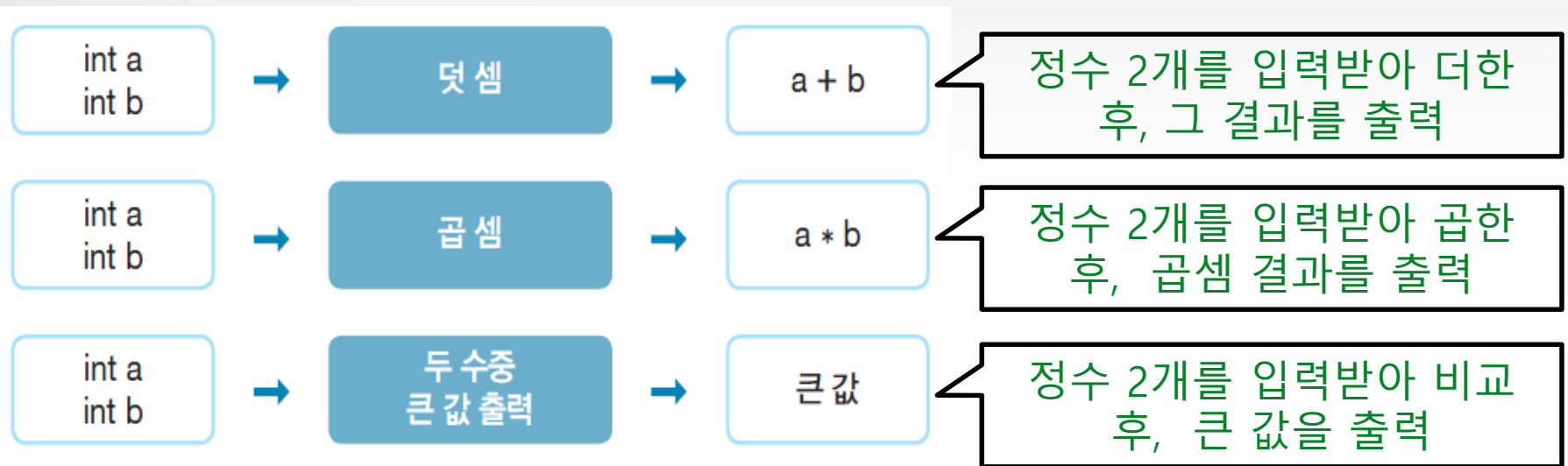
- 함수의 일반적인 개념



- 사용자 정의 함수 : 사용자가 자신에 필요한 함수를 직접 만든 것

함수의 정의와 기능

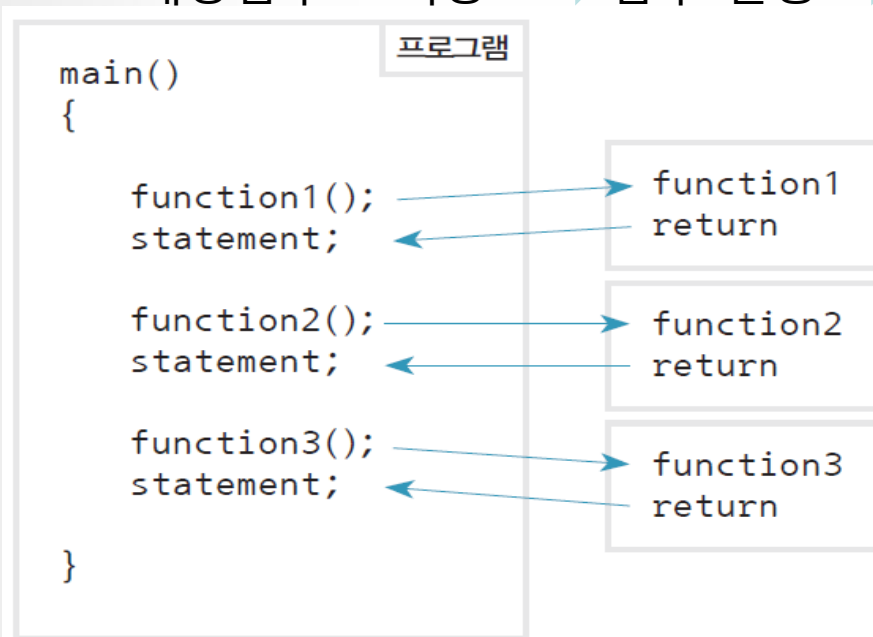
- 여러 가지 기능을 수행하는 함수



여러 가지 기능을 수행하는 함수

함수 사용의 장점

- 함수 : 특정한 기능을 수행하고 그 수행 결과로 값을 반환하는 부 프로그램
- 프로그램 실행 흐름
 - 프로그램 실행 중에 함수명과 동일한 부분을 만나면,
해당함수로 이동 ➡ 함수 실행 ➡ main() 함수로 다시 돌아옴.



프로그램의 실행 흐름

함수 사용의 장점

- 반복문 VS 함수
 - 1~10까지 더하기

```
int i, sum = 0;
for(i=1; i<11; i++)
    sum += i;
```

VS

```
int sum = 0;
sum = Summation(1, 10);
```



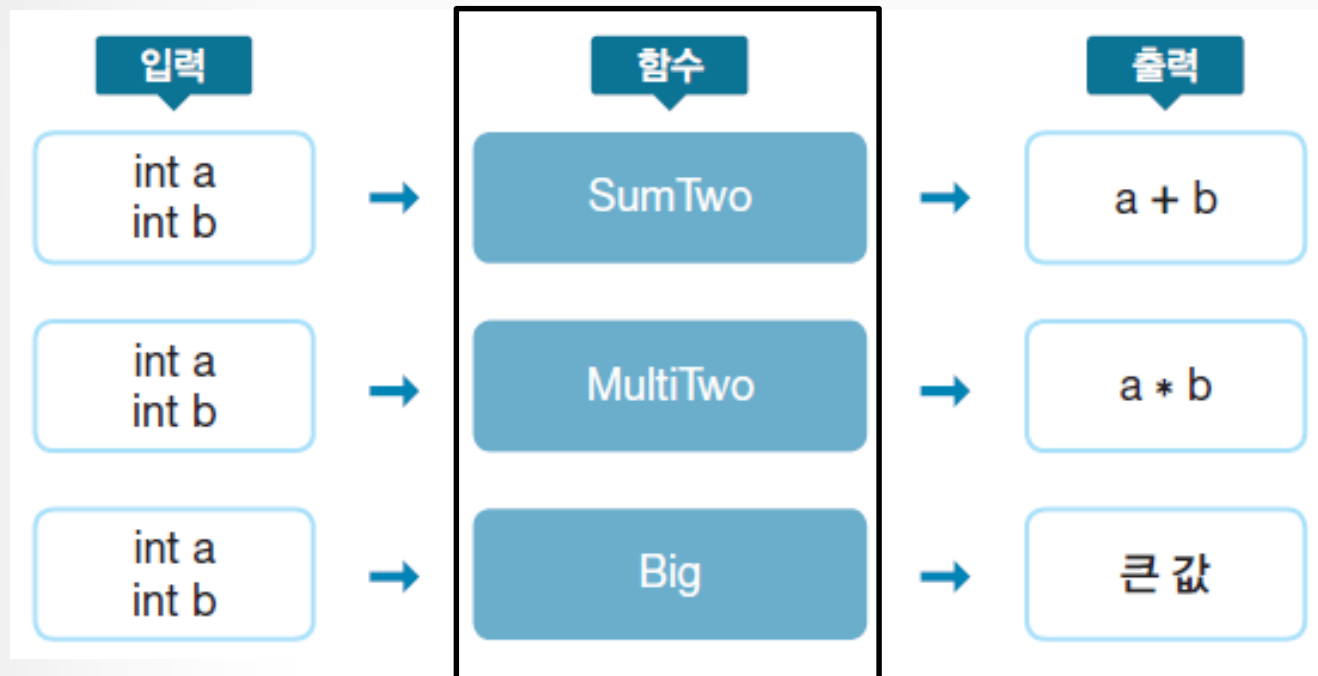
함수의 종류



- 시스템 라이브러리 함수 :
사용자가 바로 이용할 수 있도록, 자주 사용하는 함수를
미리 만들어 놓은 것
- 사용자 정의 함수 : 사용자 자신이 원하는 기능을 수행
하도록 직접 만든 함수

함수의 정의

- 시스템 라이브러리 함수에도 `printf()`, `scanf()` 같이 함수명이 있듯이 사용자 정의 함수에도 이름이 필요함



함수의 정의

- 함수의 정의 형태

[함수 정의]

```
③ 반환되는 자료형  ① 함수명 (② 입력)
{
    ④ 기능(하는 일);
    ... ..
    ③ return 반환값;
}
```

[함수 예]

```
int SumTwo (int a, int b)
{
    int result;
    result = a + b;
    return result;
}
```

- 함수 명 : 식별 자 작성 규칙에 맞게 작성
- 입력 : 함수에 어떤 인자가 입력되는지 나타냄
- 반환 값 : 함수가 입력을 받아 기능을 수행한 후에 반환되는 결과값
- 기능 : 함수를 통해서 사용자가 구현하려는 내용

함수의 정의

- MultiTwo() 함수 : 정수(a, b) 2개를 입력받아 곱함

```
int MutiTwo(int a, int b)
{
    int result;
    result = a * b;
    return result;
}
```

반환값으로 int형
변수 result를
사용하지 않고
입력된 변수 2개를
곱하는 간단한
수식이 와도 됨.

```
int MutiTwo(int a, int b)
{
    return a * b;
}
```

- Big() 함수는 정수(a, b) 2개를 입력 받아 두 정수 중에서 큰 수를 출력한다

```
int Big(int a, int b)
{
    if (a > b) return a;
    else return b;
}
```

함수의 선언

- 함수의 선언 : 컴파일러에 함수명과 반환값 그리고 함수의 매개변수를 알려줌
 - 함수 원형(prototype)이라고도 함.
 - 함수를 정의하기 전에 선언함

[선언형태]

반환되는 자료형 함수명(매개변수들);

[선언 예]

```
int SumTwo(int a, int b);
```

구체적인 인자 명 대신, 다음과 같이 자료형과 개수만 나타내도 된다.

```
int Sum (int, int);
```