

9장 함수

1. 함수를 정의하고 선언하기
2. 변수의 존재기간과 접근범위 1 : 지역변수
3. 변수의 존재기간과 접근범위 1 : 전역변수, static 변수
4. 재귀함수에 대한 이해



□ 함수 호출(function call) : 정의한 함수를 실행하는 것

□ 문법

함수이름 (데이터1, 데이터2, 데이터3, ...);

함수이름 ();

매개변수에 전달될 입력 데이터

매개변수가 없는 함수인 경우 비워 둠

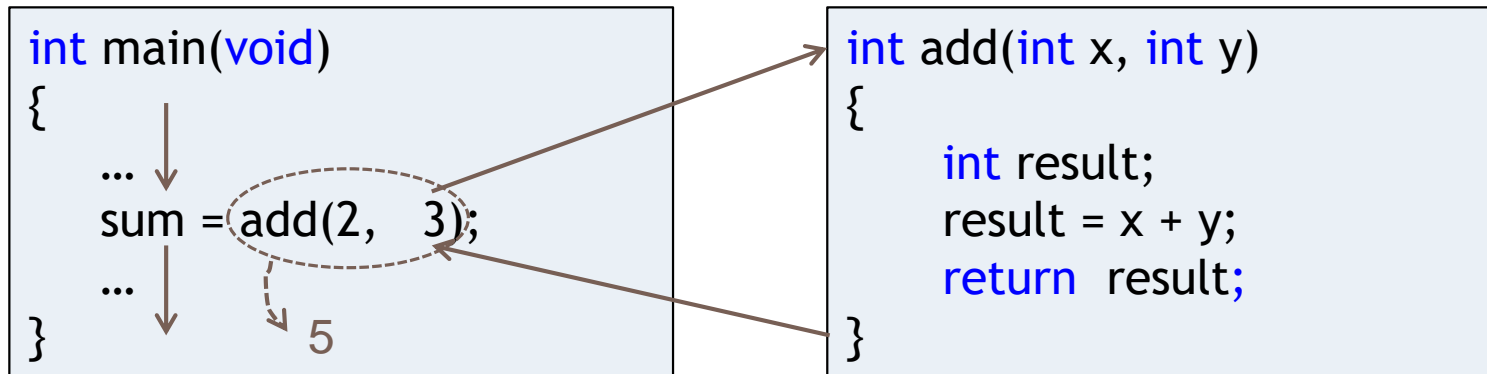
□ 예

add(2,3);

get_max(2,3);

print();

- 함수 호출 전에는 함수는 절대 실행되지 않는다.
- main 함수 안에서 함수를 호출하면 함수 안의 문장들이 순차적으로 실행된다.
- 함수 안의 문장의 실행이 끝나면 호출한 위치로 되돌아 가고 함수 호출문장은 함수의 반환값(리턴값)으로 대체된다.





인수와 매개 변수

5

- 인수(argument): 함수 호출 시 괄호 안에 적어주는 값
- 인수는 함수 호출시 매개변수에 전달된다.
- 매개변수의 자료형, 개수와 동일해야 한다.
- 매개변수가 없는 경우 호출시 인수 없이 괄호안을 비워 둠 -> print();

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

매개 변수에 저장

```
int main(void)
{
    ...
    sum = add(2, 3);
    ...
}
```

인수



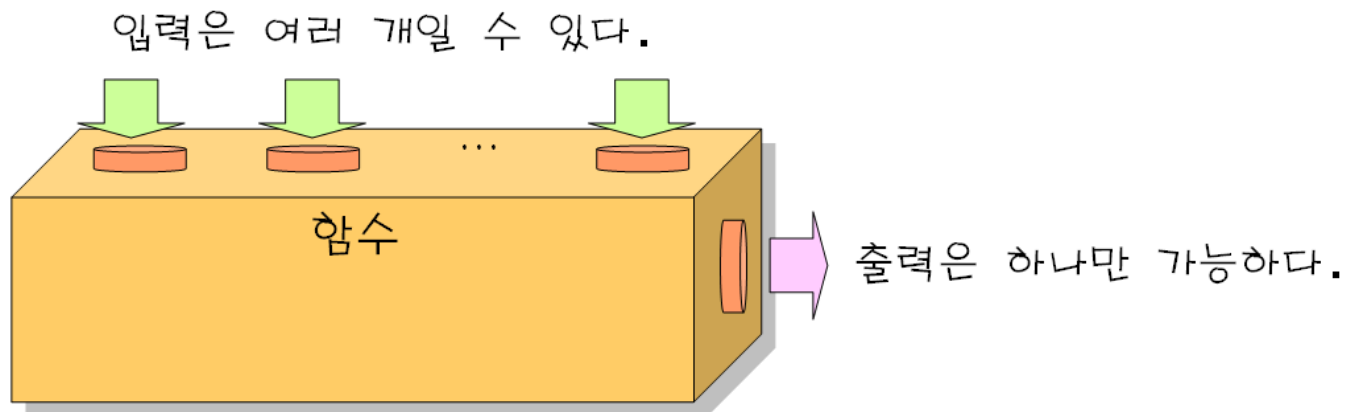
인수와 매개 변수

6

```
#include<stdio.h>
int add(int x, int y)
{
    int result;
    result = x + y;
    return result;
}
int main(void)
{
    add(2, 3);           // 인수 2, 3
    add(5, 6);           // 인수 5, 6
    add(10);             // error : 인수가 2개여야 함
    add(0.1, 0.2);       // error : 인수가 정수여야 함
    add( );              // error : 인수가 2개여야 함

    return 0;
}
```

- 반환값(return value): 호출된 함수(add)에서 호출한 곳(main)으로 전달하는 작업의 결과값
- 인수는 여러 개가 가능하나 반환값은 하나만 가능
- 반환값이 없으면 void로 선언





값을 반환하는 문장

8

- 함수에서 값을 반환하는 방법 : **return** (반환값 or 수식);
- 반환값의 자료형과 함수 헤더의 반환형이 일치해야한다.

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

- return 문장의 사용 예

```
return 0;           //상수 가능
return x;           //변수 가능
return x+y;         //수식도 가능. 수식이 오면 수식의 결과값이 리턴됨
```



가을 반환하는 문장

9

- 반환값이 없는 경우 함수의 마지막 줄에 `return;` 또는 `return`문을 사용하지 않는다.

```
void print_hello( )  
{  
    printf("Hello World!\n");  
    return;  
}
```

```
void print_hello( )  
{  
    printf("Hello World!\n");  
}
```



반환되는 과정

10

- 함수가 호출되면 함수 몸체의 문장들이 하나씩 실행되고 마지막에 return 문이 실행되면 호출한 쪽으로 리턴값이 반환(출력)된다.
- 함수를 호출한 쪽에서는 리턴값을 받기 위한 변수를 선언하고 **변수 = 함수(...);** 형식으로 작성한다.
- 함수호출문은 함수의 반환값으로 대체되고 반환값이 변수에 저장된다.

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

```
int main(void)
{
    int sum;
    ...
    sum = add(2, 3); 함수호출문
    ...
}
```

5



```
#include <stdio.h>
int add(int x, int y)
{
    int result;
    result = x + y;
    return result;
}
int main(void)
{
    int a, b, result;
    printf("두 정수를 입력하시오 : ");
    scanf("%d%d", &a, &b);
    result = add(a, b);           //result = a + b;
    printf("두수의 합 = %d \n", result);
    return 0;
}
```

두 정수를 입력하시오:10 20<엔터>
두수의 합 = 30



예제 1

12

```
#include <stdio.h>
int get_integer()           // 함수 정의
{
    int value;
    printf("정수를 입력하시오 : ");
    scanf("%d", &value);
    return value;
}
int main(void)
{
    int x, y, result;
    x = get_integer();      // 함수 호출
    y = get_integer();      // 함수 호출
    result = x + y;
    printf("두수의 합 = %d \n", result);
    return 0;
}
```

정수를 입력하시오:10<엔터>
정수를 입력하시오:20<엔터>
두수의 합 = 30



예제 2-1

13

```
#include <stdio.h>
int get_integer() // 함수 정의
{
    int value;
    printf("정수를 입력하시오 : ");
    scanf("%d", &value);
    return value;
}
int get_max(int x, int y) // 함수 정의
{
    if (x > y) return x;
    else return y;
}
```

정수를 입력하시오:10<엔터>
정수를 입력하시오:20<엔터>
두수중에서 큰수는 20입니다.



예제 2-2

14

```
int main(void)
{
    int a, b, max;
    a = get_integer();           // 함수 호출
    b = get_integer();           // 함수 호출
    max = get_max(a, b);
    printf("두수 중에서 큰 수는 %d입니다.\n", max);
    return 0;
}
```

정수를 입력하시오:10<엔터>
정수를 입력하시오:20<엔터>
두수중에서 큰수는 20입니다.



예제 3-1

15

```
#include <stdio.h>
int get_integer() // 함수정의
{
    int value;
    printf("정수를 입력하시오 : ");
    scanf("%d", &value);
    return value;
}
int power(int x, int y) // 함수정의
{
    int i;
    int result = 1;
    for (i = 0; i < y; i++) result *= x;
    return result;
}
```

정수를 입력하시오:10<엔터>
정수를 입력하시오:2<엔터>
10의 2승 = 100

// result = result * x



예제 3-2

16

```
int main(void)
{
    int a, b, result;
    a = get_integer();    // 함수호출
    b = get_integer();    // 함수호출
    result = power(a, b);
    printf("%d의 %d승 = %d \n", a, b, result);
    return 0;
}
```

정수를 입력하시오:10<엔터>
정수를 입력하시오:2<엔터>
10의 2승 = 100



실습과제 1

17

- 키보드로부터 2개의 정수를 받아 곱을 계산하는 코드를 작성하라. 단, 2개 정수의 곱을 계산하는 부분만을 함수로 작성하고 나머지 부분은 main함수에 작성하라. 아래 결과처럼 실행되도록 코드를 작성하라. 함수의 정의, 호출을 사용할 것

정수를 입력하세요:7<엔터>
정수를 입력하세요:9 <엔터>
곱은 63입니다.

- 키보드로부터 국어, 영어, 수학 점수를 입력 받아 평균값을 계산하는 프로그램을 작성하라. 단, 3과목 점수의 평균값을 구하는 부분만 함수로 만들고 나머지 부분은 main 함수에서 작성하라.
- 함수의 정의, 호출을 사용할 것.

국어점수를 입력하세요:70<엔터>
영어점수를 입력하세요:90<엔터>
수학점수를 입력하세요:80<엔터>
평균값은 70점입니다.



실습과제3

19

- 정수 2개 중에 최대값을 구하는 함수 `get_max`, 최소값을 구하는 함수 `get_min` 함수를 작성하여 아래 결과처럼 동작하는 코드를 작성하라. 데이터 입출력은 메인함수에서 `printf`, `scanf` 함수를 이용하여 작성하라. 함수의 정의, 호출을 사용할 것.

첫번째 정수를 입력하시오:20 <엔터>
두번째 정수를 입력하시오:30 <엔터>
최대값은 30입니다.
최소값은 20입니다.



과제제출방법

20

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡처할 때 글자를 알아보기 쉽게 확대해서 캡처할 것.
- 소스코드의 첫 부분은 아래처럼 제목,날짜,작성자(학번,이름)를 작성할 것

```
// *****  
//   제   목   : 정수 4개의 평균을 구하는 프로그램  
//   날   짜   : 2023년 9월10일  
//   작성자   : 15010101 홍길동  
// *****  
  
// 소스코드 작성
```