

I 함수

- ▶ 함수의 정의
- ▶ 함수의 호출
- ▶ 함수의 선언

I 지역 변수와 전역 변수

- ▶ 지역 변수
- ▶ 전역 변수
- ▶ 변수의 영역 규칙

I 함수의 인자 전달 방법

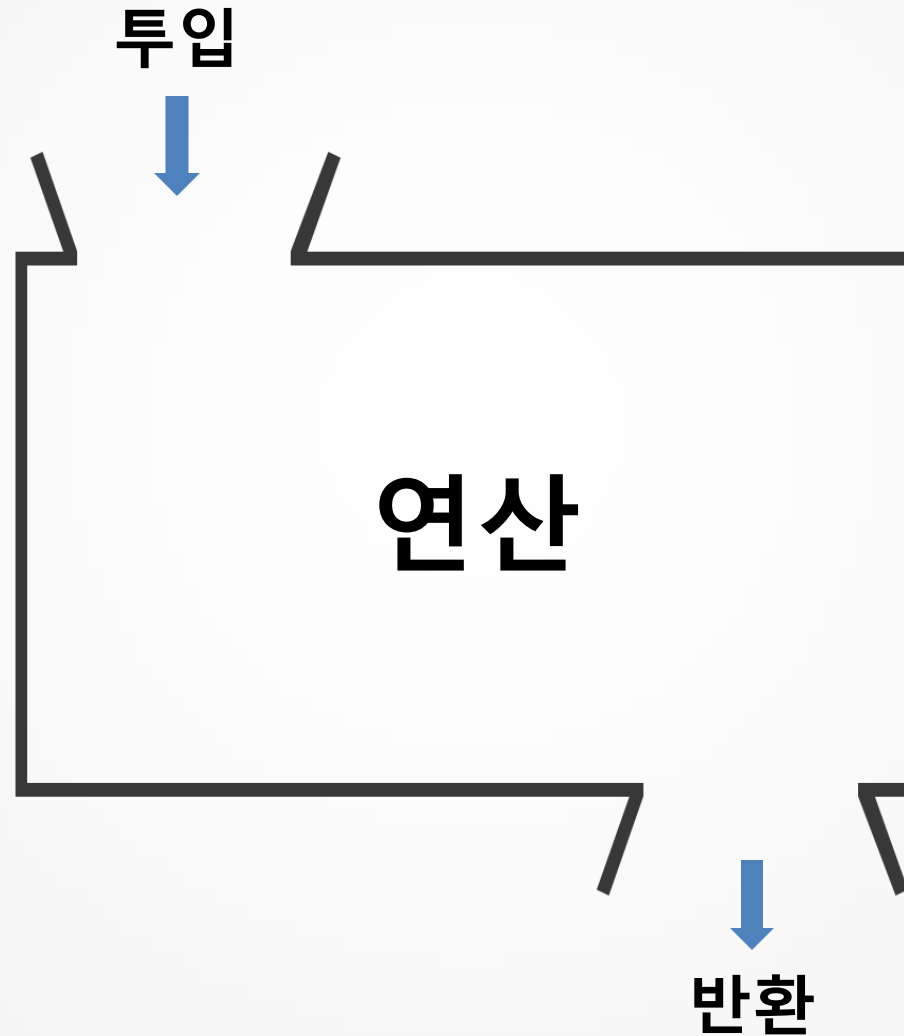
- ▶ 값에 의한 전달
- ▶ 포인터에 의한 전달(현재 강의 범위는 아님)

Artificial Intelligence Laboratory

함수

함수란 무엇인가?

- 함수는 마법 상자!



함수의 정의

함수의 기본 형식

형식

```
리턴형 함수명 (데이터형 매개변수명 [, 데이터형 매개변수명, ..])  
{  
    문장;  
}
```

예제

리턴형

함수명

데이터형

매개변수명

int

GetSum

(int

num)

I 함수가 반환할 값의 자료형

- void는 반환값이 없다는 의미
- 리턴형 생략 = int로 간주

int GetFactorial (int num);

- int 자료형의 데이터를 반환한다는 의미

함수의 이름

- 함수의 이름도 식별자의 규칙에 따라서 만들어야 함
 - ▶ 식별자란 변수, 함수, 자료형 등의 이름을 일컫는 말이다
- 동일한 이름은 오직 하나만!
 - ▶ 함수는 이름으로 구분한다
 - ▶ 기왕이면 알기 쉬운 이름이 좋다

int **GetFactorial** (int num);

int **GetSum** (int num);

float **GetLength** (void);

void **print_area** (int x, int y);

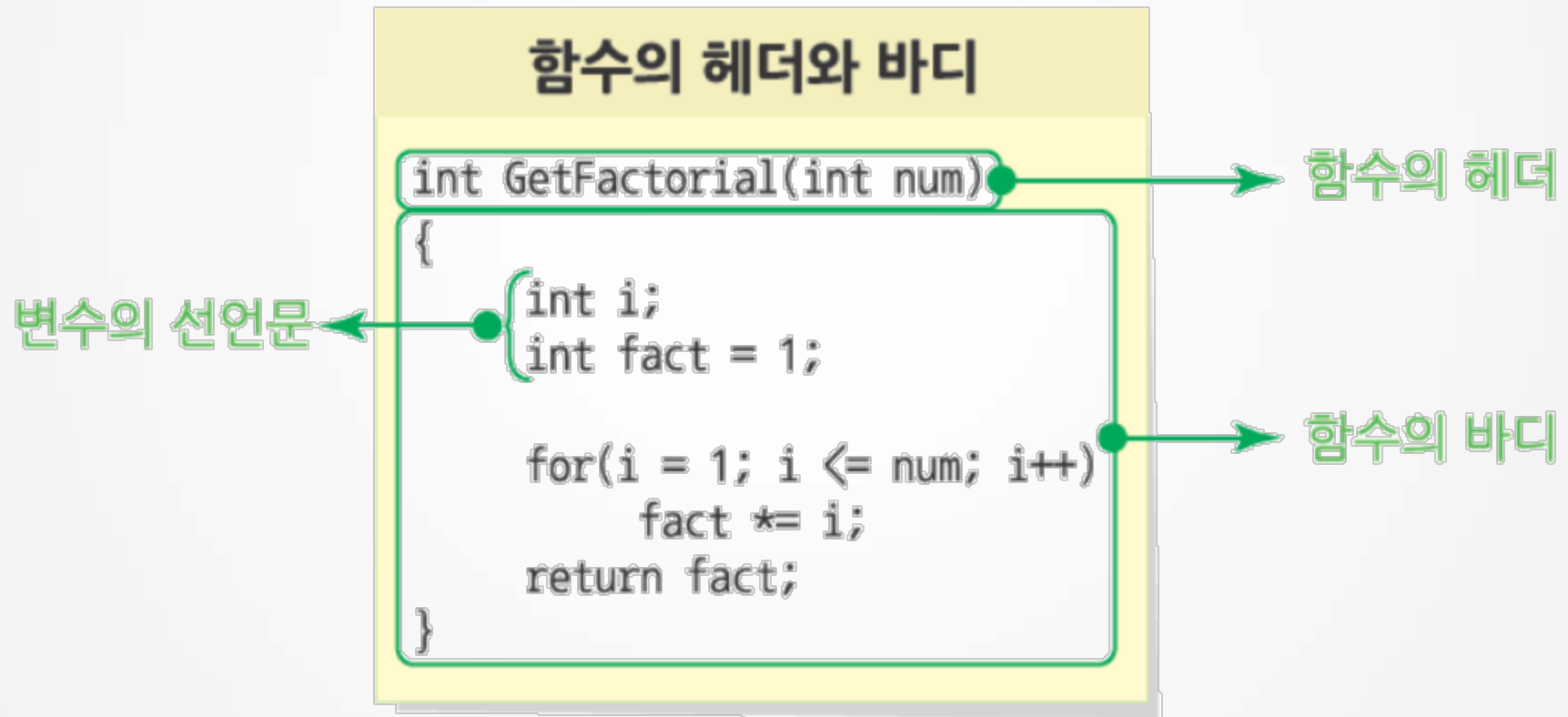
함수의 매개변수

- 함수로 값을 넘겨주기 위한 변수
 - 소괄호 내부에 선언의 형태로 적는다
 - 개수 제한은 없다
- 함수가 매개변수를 갖지 않을 때는 () 안에 void라고 적음

```
int GetFactorial (int num);  
float GetSum (float a, float b, float c);  
int addToList (void);
```

함수의 헤더와 바디

- 함수의 정의는 헤더와 바디로 구성
 - 헤더 = 함수의 리턴형, 함수의 이름, 매개변수
 - 바디 = {} 안에 처리할 문장을 나열한 부분



함수 호출

- 함수를 사용하는 것을 '호출'이라고 한다
 - ▶ 함수이름(); 의 형태로 호출한다

함수의 호출 시 진행 과정

인자 없는 함수의 호출 과정

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

int main(void)
{
    ...
    PrintHello();
    ...
    return 0;
}
```

① 함수 정의로 이동합니다.

② 함수의 코드를 수행합니다.

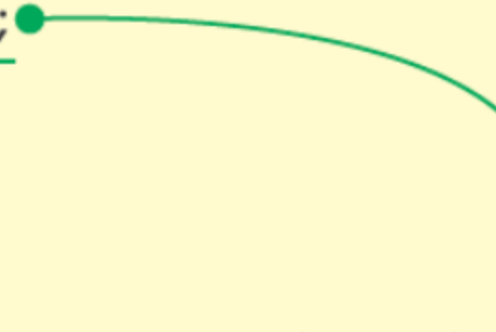
③ 함수를 호출한 곳으로 되돌아 옵니다.

④ 그 다음 코드를 계속 수행합니다.

함수의 리턴 값이 전달되는 과정

함수의 리턴 값이 전달되는 과정

```
int GetFactorial ( int num )  
{  
    ...  
    return fact;  
}  
  
int main(void)  
{  
    ...  
    result = GetFactorial ( 10 );  
    ...  
  
    return 0;  
}
```



return 문이 리턴하는 값이
GetFactorial(10)의 값이 됩니다.

함수의 호출 - 인자를 갖는 함수

인자를 갖는 함수의 호출 과정

①함수의 인자를 전달합니다.

```
void PrintSumAndProduct (int a, int b)
{
    printf("%d + %d = %d\n", a, b, a + b);
    printf("%d * %d = %d\n", a, b, a * b);
}
```

③함수의 코드를
수행합니다.

```
int main(void)
{
    ...
```

④함수를 호출한 곳
으로 되돌아 옵니다.

②함수 정의로
이동합니다.

```
PrintSumAndProduct (10, 20);
    ...
```

⑤그 다음 코드를
계속 수행합니다.

```
return 0;
}
```

함수의 매개변수와 인자

I 매개변수

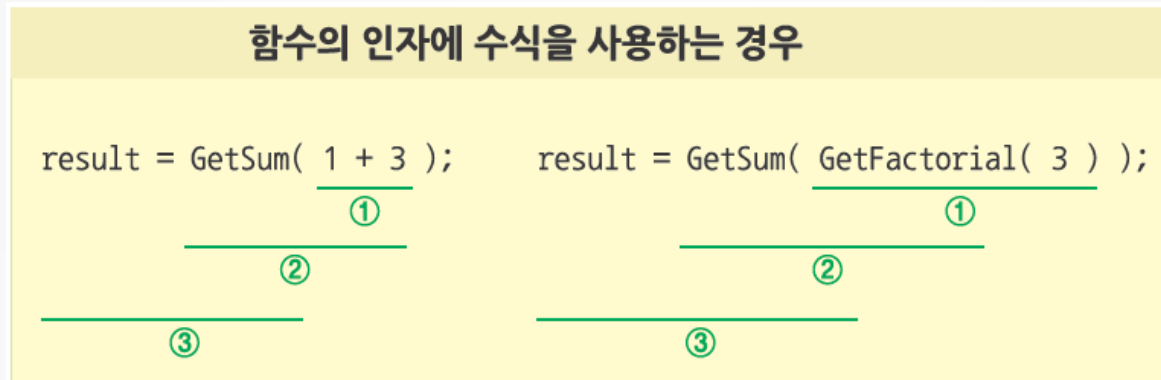
- ▶ 함수에 필요한 값을 넘겨주기 위한 변수

I 인자

- ▶ 함수를 호출할 때 직접 넘겨주는 값
- ▶ 함수 호출 시 넘겨준 인자가 함수의 매개변수로 전달됨

함수 호출 시 주의사항

- 함수의 인자로 함수나 수식이 주어진 경우
 - 함수와 수식부터 다 처리한 다음 함수를 호출함



- 함수 호출 시 매개변수의 개수 = 인자의 개수
- 함수의 리턴 값을 사용할지 말지는 자유
- 함수를 호출할 때는 반드시 함수 호출보다 앞 쪽에 함수를 정의해주어야 함

함수의 선언

- 함수의 정의된 위치에 관계 없이 함수를 호출할 수 있게 함
 - 함수의 리턴형, 이름, 매개변수에 대한 정보만 미리 알려주는 것
 - 함수의 원형(prototype)이라고도 한다

형식

리턴형 함수명 (데이터형 매개변수명 [...]);

예제

void
void
int
int
float

리턴형

PrintHello
PrintSumAndProduct
GetFactorial
GetSum
GetMax

함수명

(void);
(int a, int b);
(int num);
(int num);
(float a, float b, float c);

매개변수 목록

함수의 정의(definition)

- ▶ 함수의 리턴형, 이름, 매개변수를 기술
- ▶ { } 안에 실제로 함수가 처리할 내용을 기술

함수의 호출(call)

- ▶ 앞에서 선언되거나 정의된 함수를 이용
- ▶ 인자를 넘겨주고 리턴 값을 받아올 수 있음

함수의 선언(declaration)

- ▶ 함수의 내용을 알려주지는 않지만 함수 호출에 필요한 리턴형, 함수의 이름, 매개변수 정보를 미리 알려줌

함수의 선언만 생략 가능

함수의 선언과 정의, 호출

```
int GetFactorial ( int );
```

함수의 선언

```
int main(void)
{
    ...
    result = GetFactorial ( 10 );
    ...

    return 0;
}
```

함수의 호출

```
int GetFactorial ( int num )
{
    int i;
    int fact = 1;
    for(i = 1 ; i <= num ; i++)
        fact *= i;
    return fact;
}
```

함수의 정의

실습 1. Absolute.c

- 정수 값을 인자로 받아서 절대값을 구하는 Absolute 함수를 정의하고, 실제 사용 예를 보이시오.
- 소스파일명: Absolute.c

Artificial Intelligence Laboratory

지역 변수, 전역 변수

지역 변수와 전역 변수

지역 변수

- 함수 안에 선언된 변수
- 해당 함수 안에서만 사용 가능

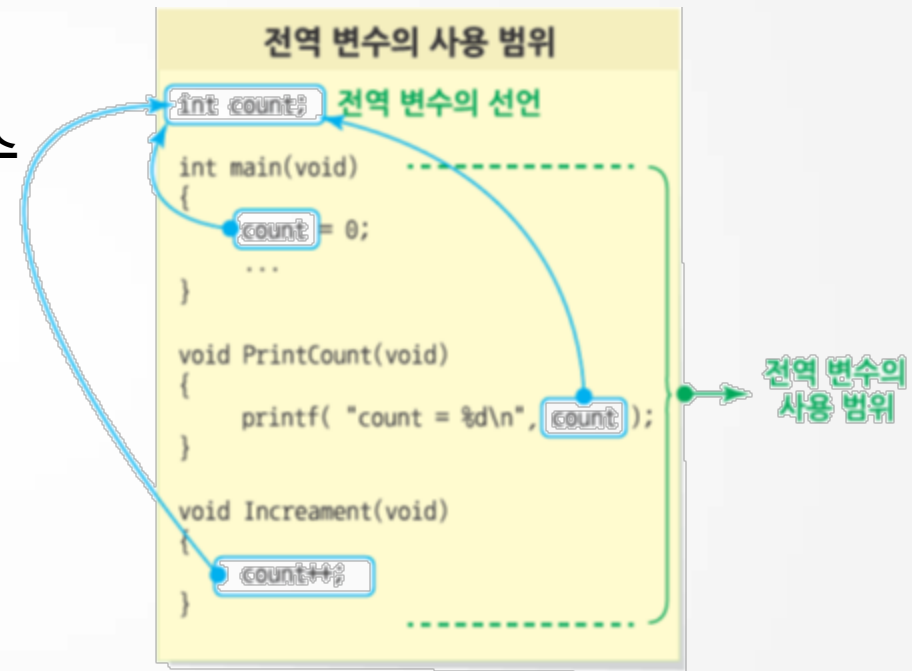
전역 변수

- 함수 밖에 선언된 변수
- 여러 함수에서 사용 가능

특징	지역 변수	전역 변수
사용 범위	함수 안	프로그램 전체
생존 기간	함수 안에서 변수 선언문을 만나면 생성되고 함수가 리턴할 때 해제됨	프로그램이 시작할 때 생성되고, 프로그램이 끝날 때 해제됨
초기화하지 않은 경우	쓰레기 값	0으로 자동 초기화

전역 변수

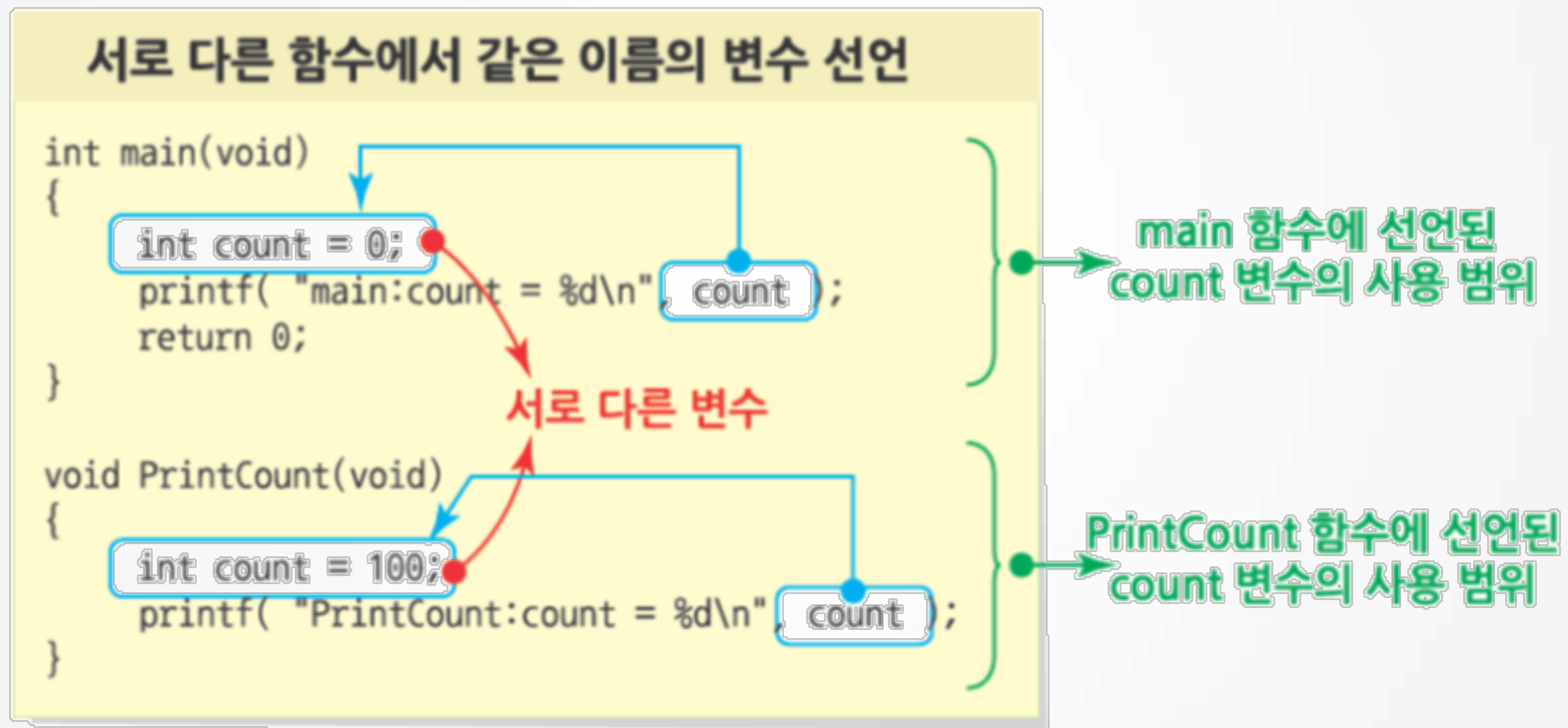
- main()함수 바깥에 선언
- 프로그램이 시작될 때 한 번만 생성
- 프로그램 전체에서 사용될 수 있는 변수
 - 프로그램이 종료될 때 해제됨



지역 변수

■ 각 함수 내에서 선언된 변수는 이름이 같아도 다른 변수로 취급

- 함수는 하나의 독립된 존재이다
- 지역 변수는 함수별로 각각 독립적으로 존재한다



같은 함수를 여러 번 호출하는 경우

```
01: /* Ex06_09.c */
02: #include <stdio.h>
03:
04: void TestLocal(void);
05:
06: int main(void)
07: {
08:     TestLocal( );
09:     TestLocal( );
10:
11:     return 0;
12: }
13:
14: void TestLocal(void)
15: {
16:     int num = 0;
17:
18:     printf("num = %d\n", num++);
19: }
```

변수의 영역 규칙 - 블록 범위

- 지역 변수는 블록의 범위(scope)에도 영향을 받는다
 - 우선적으로 같은 블록을 찾는다
 - 찾지 못하면 그보다 상위 블록에서 해당 이름의 변수를 찾는다
 - 계속 찾지 못한다면 결국 해당 이름의 전역 변수가 있는지 찾는다

```
int main(void)
{
    int num = 10;
    if( 1 )
    {
        int num = 20;
        ...
    }
    while( 1 )
    {
        int num = 30;
    }
}
```

블록 안에서는 같은 이름의 변수를 재선언할 수 있다.

블록 안에서는 같은 이름의 변수를 재선언할 수 있다.

변수의 영역 규칙

- 전역 변수와 지역 변수의 이름이 같아지는 것을 막기 위해서 전역 변수 이름에는 g_를 접두사로 붙이기도 함

```
int g_num; ○ 전역 변수라는 것을 명확히 알 수 있다.  
int main(void)  
{  
    ...  
}
```

변수의 영역 규칙

```
int num = 10; 프로그램 전체 범위  
  
int main(void) main 함수 범위  
{  
    int num = 20;  
  
    while( 1 ) while 블록 범위  
    {  
        int num = 30;  
        printf( "num = %d\n", num ++ );  
        if ( num > 25 )  
            break;  
    }  
  
    printf( "num = %d\n", num );  
    Test();  
    return 0;  
}  
  
void Test(void)  
{  
    printf( "num = %d\n", num );  
}
```


Artificial Intelligence Laboratory

인자 전달 방법

함수의 인자 전달 방법

I 값에 의한 전달

- ▶ 함수를 호출할 때 넘겨주는 인자의 값을 함수 정의에 있는 매개변수로 복사해서 전달하는 방식
- ▶ 복사에 의한 전달
- ▶ 함수 안에서 매개변수의 값을 변경해도 함수를 호출한 곳에 있는 인자의 값은 바뀌지 않음

값에 의한 전달

```
int GetFactorial ( int num );

int main(void)
{
    ...
    result = GetFactorial ( 5 );
    ...

    return 0;
}

int GetFactorial ( int num )
{
    ...
    return fact;
}
```

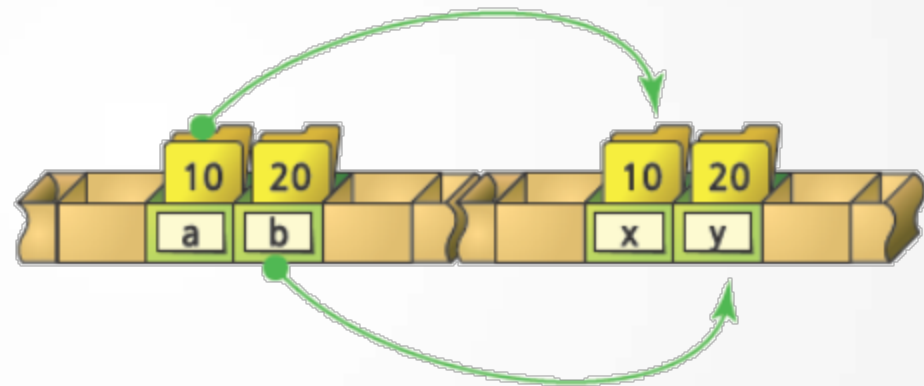
int num = 5;

함수 호출시
내부적으로
수행되는 문장

함수의 인자 전달 방법 - 예제

```
int main( void )  
{  
    ...  
    Swap( a, b );  
    ...  
}  
  
int x = a; int y = b;  
  
void Swap( int x, int y )  
{  
    ...  
}
```

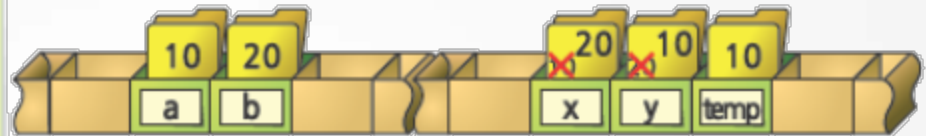
① 함수 호출시 값에 의해 인자를 전달합니다.



함수의 인자 전달 방법 - 예제

```
void Swap( int x , int y )  
{  
    int temp;  
    temp = x;  
    x = y;  
    y = temp;  
}
```

② Swap 함수의 코드를 수행합니다.



x와 y의 값을
맞교환합니다.

```
int main( void )  
{  
    ...  
    Swap( a , b );  
    ...  
}
```

③ Swap 함수를 리턴합니다.



a, b는 변경
되지 않습니다.

함수 리턴시
x, y, temp는 모두
해제됩니다.

핵심 요약 (함수)

함수의 정의

- ▶ 함수의 리턴형, 이름, 매개변수 작성
- ▶ {} 안에는 실제로 함수가 처리할 내용을 기술

함수의 호출

- ▶ 함수이름();
- ▶ ()안에 인자를 씀

함수의 선언

- ▶ 리턴형, 함수 이름, 매개변수 정보를 미리 알려주는 문장

함수의 선언과 정의, 호출

```
int GetFactorial ( int );
```

함수의 선언

```
int main(void)
{
    ...
    result = GetFactorial ( 10 );
    ...

    return 0;
}
```

함수의 호출

```
int GetFactorial ( int num )
{
    int i;
    int fact = 1;
    for(i = 1 ; i <= num ; i++)
        fact *= i;
    return fact;
}
```

함수의 정의

핵심 요약 (지역, 전역 변수)

지역 변수

- ▶ 함수 안에서 선언. 함수 안에서만 사용 가능. 함수 종료시 해제.
- ▶ 초기화하지 않으면 쓰레기 값을 가짐

전역 변수

- ▶ 함수 밖에서 선언. 프로그램 전체에서 사용
- ▶ 초기화하지 않아도 0으로 초기화됨

영역 규칙

- ▶ 가장 가까운 블록에 선언된 변수가 항상 우선적으로 사용

실습 2. IsPrime.c

- 정수 값을 인자로 받아서 소수인지 판단하는 IsPrime 함수를 정의하고, IsPrime 함수를 이용해서 1부터 100 사이의 소수를 모두 구해서 출력하는 프로그램을 작성하시오.

- 파일명: IsPrime.c

- 참고

- 소수? 1과 자기 자신만으로 나누어 떨어지는 1보다 큰 양의 정수
- 1부터 100사이의 소수

1부터 100사이의 소수

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

©doopedia.co.kr