

■ 함수란?

- 프로그램 내에서 특정한 작업을 수행하기 위해서 독립적으로 만들어진 프로그램의 단위
- C언어로 프로그램을 작성한다는 것은 여러 가지 함수를 만들거나 사용해 큰 흐름을 완성한다는 의미
- C언어는 함수의 집합으로 구성되어 있고 프로그램을 실행하면 처음으로 실행되는 함수가 바로 지금까지 작성해 온 main() 함수이다.
- 연관된 부분끼리 함수라는 단위로 묶음으로써 긴 소스코드들을 효율적으로 관리할 수 있음.

■ 함수는 두 가지로 나뉜다!

- 하나는 표준라이브러리함수 - printf(), scanf()와 같이 c언어에 내장되어 있는 함수
- 다른 하나는 사용자 정의함수 - 사용자가 직접 만들어서 사용하는 함수

■ 반복되는 일을 단순히 묶음(가장 간단한 함수)

- 함수 예제(1)

```
#include <stdio.h>
int main()
{
    printf(" *\n");
    printf(" ***\n");
    printf("*****\n");

    printf(" *\n");
    printf(" ***\n");
    printf("*****\n");
    return 0;
}
```

실행결과

```
*
***
*****
*
***
*****
```

- 동일한 일을 반복하는 부분은?
- 위와 동일한 일을 수행하는 함수를 만들어보자!
- 함수 예제(2)

```
#include <stdio.h>
// 함수 선언 : 반환자료형, 함수이름, 매개변수 등을 기술하여 함수의 외형적 특징을 기술
void printStar();

int main(void)
{
    printStar(); // 함수 호출
    printStar(); // 함수 호출
    return 0;
}

// 함수 정의
void printStar()
{
    printf(" *\n");
    printf(" ***\n");
    printf("*****\n");
}
```

실행결과

```
*
***
*****
*
***
*****
```

함수 정의의 전체적인 구조

```
반환형 함수이름( 매개변수 )
{
    함수 몸체
}
```

■ 값을 전달받아 동작하는 함수(매개변수, parameter)

- 아래와 같이 출력할 삼각형의 개수를 파라미터로 전달할수 있게 수정함.
- 함수 예제(3)

```
#include <stdio.h>
// 함수 선언
void printStar(int n);
int main()
{
    printStar(2); // 함수 호출
    return 0;
}
// 함수 정의
void printStar(int n)
{
    int i; // 지역변수
    for (i = 1; i <= n; i++) {
        printf(" *\n");
        printf(" ***\n");
        printf("*****\n");
    }
}
```

실행결과

```
*
***
*****
*
***
*****
```

함수 정의의 전체적인 구조

```
반환형 함수이름( 매개변수 )
{
    함수 몸체
}
```

- 함수의 매개변수는 호출하는 함수(caller)가 호출당하는 함수(callee)에게 값을 '복사'해서 넘겨줌.
- call-by-value 전달
- caller가 전달하는 값을 '실 매개 변수', callee가 전달받는 값을 '형식 매개 변수' 라고 함.
- main 함수에서 printStar(2); 호출시 매개변수 숫자 2가 실 매개 변수 임.
- void printStar(int n) 함수 정의 부분에서 int n 변수가 형식 매개변수임.
- 함수 예제(4)

```
#include <stdio.h>
// 함수 선언
int Add(int n1, int n2);
int main()
{
    int x, y, sum;
    printf("정수 두 개를 입력하세요~!(예 1 2) : ");
    scanf("%d %d", &x, &y);

    sum = Add(x, y); // 함수 호출
    printf("%d + %d = %d", x, y, sum);
    return 0;
}
// 함수 정의
int Add(int n1, int n2)
{
    int n3 = n1 + n2;
    return n3;
}
```

형식 매개변수는?
실 매개변수는?

- 함수 예제(5)

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
// 함수 선언
void change(int n1, int n2);
int main()
{
    int x, y, sum;
    printf("정수 두 개를 입력하세요~!(예 1 2) : ");
    scanf("%d %d", &x, &y);
    printf("함수 호출 전: %d %d\n", x, y);
    change(x, y); // 함수 호출
    printf("함수 호출 후: %d %d\n", x, y);
    return 0;
}
// 함수 정의
void change(int n1, int n2)
{
    int temp;
    temp = n1;
    n1 = n2;
    n2 = temp;
    printf("함수 내부 : %d %d\n", n1, n2);
}
```

실행결과

- 함수의 매개변수는 호출하는 함수(caller)가 호출당하는 함수(callee)에게 값을 '복사'해서 넘겨줌.
- 이를 값에 의한 호출(call-by-value)이라고 함
- 이와 다르게 참조에 의한 호출(call-by-reference)이 있는데 이는 복사가 아닌 변수 자체를 전달
- 참조에 의한 호출은 포인터 개념을 배운 뒤 설명할 것임

■ 수행 후 값을 반환(return)하는 함수

- 함수의 반환 자료형은 호출하는 함수가 호출당하는 함수를 호출해서 얻을 수 있는 정보의 형식
- 함수가 반환한 자료는 연산의 '임시결과'처럼 즉시 활용하거나 저장하지 않으면 유실됨.
- 반환자료형을 즉시 활용하는 방법

- ① 자료형이 일치하는 변수에 대입하여 정보를 보관
- ② 피연산자로 다른 연산에 참여
- ③ 다른 함수를 호출하는데 인수로 사용

● 함수 예제(5)

```
#include <stdio.h>
// 함수 선언
int GetMax(int a , int b, int c);

int main()
{
    int nResult = 0;

    // 함수가 반환한 값을 %d 형식으로 출력한다.
    printf("MAX : %d\n", GetMax(1, 2, 3));

    // 함수가 반환한 값에 * 2 연산을 수행하고 %d 형식으로 출력한다.
    printf("MAX : %d\n", GetMax(2, 3, 1) * 2);

    // 함수가 반환한 값을 nResult 변수에 저장한 후
    // nResult에 저장된 값을 %d 형식으로 출력한다.
    printf("MAX : %d\n", nResult = GetMax(3, 1, 2));

    return 0;
}

// 함수 정의
int GetMax(int a, int b, int c)
{
    int nMax = a;
    if (b > nMax) nMax = b;
    if (c > nMax) nMax = c;

    return nMax;
}
```

● 함수 예제(6) - 반환값과 매개변수의 유무에 따라서 구분할 수 있는 함수의 유형

```
#include <stdio.h>
// 함수 선언
int Add(int num1, int num2); // 매개변수 0, 반환값 0
void ShowAddResult(int num); // 매개변수 0, 반환값 X
int InputNum();             // 매개변수 X, 반환값 0
void HowToUseThisProg();    // 매개변수 X, 반환값 X

int main()
{
    int result, num1, num2;
    HowToUseThisProg();
    num1 = InputNum();
    num2 = InputNum();
    result = Add(num1, num2);
    ShowAddResult(result);
    return 0;
}

// 함수 정의
int Add(int num1, int num2)
{
    return num1 + num2;
}

void ShowAddResult(int num)
{
    printf("덧셈결과 출력 : %d\n", num);
}

int InputNum()
{
    int num;
    scanf("%d", &num);
    return num;
}

void HowToUseThisProg()
{
    printf("두 개의 정수를 입력하시면 덧셈결과가 출력됩니다.\n");
    printf("자! 그럼 두개의 정수를 입력하세요.\n");
}
```

실행결과

두 개의 정수를 입력하시면 덧셈결과가 출력됩니다.
자! 그럼 두개의 정수를 입력하세요.
12 24
덧셈결과 출력 : 36

● 함수 예제(7) - 하나의 함수 내에 둘 이상의 return 문이 존재하는 경우

```
#include <stdio.h>
// 함수 선언
int NumCompare(int num1, int num2);
int main()
{
    printf("3과 4중에 큰 수는 %d이다\n", NumCompare(3, 4));
    printf("7과 2중에 큰 수는 %d이다\n", NumCompare(7, 2));
    return 0;
}
// 함수 정의
int NumCompare(int num1, int num2)
{
    if (num1 > num2)
        return num1; // return 문이 실행되면 값을 반환하면서 함수를 빠져나간다!
    else
        return num2;
}
```

실행결과

● 함수 예제(8) - 호출된 함수내에서 또 다른 함수를 호출한다!

```
#include <stdio.h>
// 함수 선언
int Absocompare(int num1, int num2); // 절대값이 큰 수 반환
int GetAbsoValue(int num);          // 매개변수의 절대값을 반환
int main()
{
    int num1, num2;
    printf("두 개의 정수 입력: ");
    scanf("%d %d", &num1, &num2);
    printf("%d와 %d 중 절대값이 큰 정수 : %d\n", num1, num2, Absocompare(num1, num2));
    return 0;
}
// 함수 정의
int Absocompare(int num1, int num2)
{
    if (GetAbsoValue(num1) > GetAbsoValue(num2))
        return num1;
    else
        return num2;
}
int GetAbsoValue(int num)
{
    if (num < 0)
        return num * (-1);
    else return num;
}
```

실행결과

- 함수 작성 연습(1)

```
// 세 개의 정수를 인자로 전달받아서 그중 가장 큰 수를 반환하는 함수와
// 가장 작은 수를 반환하는 함수를 정의해보자!
#include <stdio.h>
// 함수 선언
int GetMax(int num1, int num2, int num3); // 가장 큰 수 반환하는 함수
int GetMin(int num1, int num2, int num3); // 가장 작은 수 반환하는 함수
int main()
{
    int num1, num2, num3;
    printf("세 개의 정수 입력 : ");
    scanf("%d %d %d", &num1, &num2, &num3);
    printf("가장 큰 수 : %d\n", GetMax(num1, num2, num3));
    printf("가장 작은 수 : %d\n", GetMin(num1, num2, num3));
    return 0;
}
int GetMax(int num1, int num2, int num3)
{
    // 작성해보세요~!
}
int GetMin(int num1, int num2, int num3)
{
    // 작성해보세요~!
}
```

- 함수 작성 연습(2)

섭씨(Celsius)온도를 입력하면 화씨(Fahrenheit)온도를 반환하는 CelToFah라는 이름의 함수와 그 반대로 화씨 온도를 입력하면 섭씨 온도를 반환하는 FahToCel라는 이름의 함수를 정의하고 이 두 함수를 호출하는 예제를 완성해보자. 참고로 섭씨와 화씨간 온도변환의 공식은 다음과 같다.

$$\text{Fah} = 1.8 * \text{Cel} + 32$$

예시) 1. 섭씨를 화씨로 2. 화씨를 섭씨 : 선택 >> 1 섭씨 입력 : 36.8 변환된 화씨 : 98.240000	1. 섭씨를 화씨로 2. 화씨를 섭씨 : 선택 >> 2 화씨 입력 : 86.56 변환된 섭씨 : 30.311111
1. 섭씨를 화씨로 2. 화씨를 섭씨 : 선택 >> 3 입력 오류	

● 함수 작성 연습(3)

$(A+B)\%C$ 는 $((A\%C) + (B\%C))\%C$ 와 같을까?

$(A\times B)\%C$ 는 $((A\%C) \times (B\%C))\%C$ 와 같을까?

세 수 A, B, C가 주어졌을 때, 위의 네 가지 값을 구하는 함수를 작성하시오.

```
#include <stdio.h>
int Fun1(int A, int B, int C); // (A+B)%C 값 반환
int Fun2(int A, int B, int C); // ((A%C) + (B%C))%C 값 반환
int Fun3(int A, int B, int C); // (A×B)%C 값 반환
int Fun4(int A, int B, int C); // ((A%C) × (B%C))%C 값 반환
int main()
{
    int A, B, C;
    scanf("%d %d %d", &A, &B, &C);

    printf("%d\n", Fun1(A, B, C));
    printf("%d\n", Fun2(A, B, C));
    printf("%d\n", Fun3(A, B, C));
    printf("%d\n", Fun4(A, B, C));

    return 0;
}
```

<https://www.acmicpc.net/problem/10430> 에 제출해볼 것!

● 함수 작성 연습(4)

두 개의 자연수를 입력받아 최대 공약수와 최소 공배수를 구하는 함수를 작성하시오.

```
#include <stdio.h>
int GCD(int A, int B); // A 와 B의 최대 공약수 반환
// 두 수 A와 B의 최대 공약수는 A와 B의 공통된 약수 중에서 가장 큰 수이다.
int LCM(int A, int B); // A 와 B의 최소 공배수 반환
// 두 수 A와 B의 최소 공배수는 A와 B의 공통된 배수 중에서 가장 작은 수이다.
int main()
{
    int A, B;
    scanf("%d %d", &A, &B);

    printf("%d\n", GCD(A, B));
    printf("%d\n", LCM(A, B));

    return 0;
}
```

- <https://www.acmicpc.net/problem/2609> 에 제출해볼 것!

- 함수 작성 연습(5)

두 개의 자연수를 입력받아 최소 공배수를 구하는 함수를 작성하시오.

```
#include <stdio.h>
int LCM(int A, int B); // A 와 B의 최소 공배수 반환
// 두 수 A와 B의 최소 공배수는 A와 B의 공통된 배수 중에서 가장 작은 수이다.
int main()
{
    int A, B, TC;
    scanf("%d", &TC);
    while (TC--)
    {
        scanf("%d %d", &A, &B);
        printf("%d\n", LCM(A, B));
    }
    return 0;
}
```

- <https://www.acmicpc.net/problem/1934> 에 제출해볼 것!

- 함수 작성 연습(6)

주어진 수 N이 소수인지 아닌지를 출력하는 함수를 작성하시오.

```
#include <stdio.h>
int isPrime(int N); // N이 소수이면 1, 아니면 0을 반환하는 함수
int main()
{
    int N, TC;
    scanf("%d", &TC);
    while (TC--)
    {
        scanf("%d", &N);
        printf("%d\n", isPrime(N));
    }

    return 0;
}
```

위의 소스를 기반으로 <https://www.acmicpc.net/problem/1978> 을 풀어볼 것!