

프로그래밍	9. 함수의 정의와 사용 : 값 반환편	일자: 2024. . .
학습 목표	<input type="checkbox"/> 사용자 정의 함수의 사용 (2) <input type="checkbox"/> 변수의 유효 범위 - 지역 변수	하이다:

1. 사용자 정의 함수의 사용

1) 값을 반환하는 함수의 정의

```
int1) calcComb2)(int n, int m)3) // 함수의 헤더 (반환타입 + 함수명 + 매개변수)
{
    함수의 바디4) // "return 정수값;" 필요
}
```

- ① 반환 타입(return type): 함수는 최대 1개까지의 값을 반환할 수 있는데, 이때 반환되는 데이터의 타입을 명시한다.
- ② 함수 이름: 함수를 구분하고 ____하기 위한 이름이다.
- ③ 매개변수 목록(parameters): 함수 호출 시에 전달된 인수의 값을 저장할 변수의 타입과 이름을 순서대로 명시한 것이다. 함수의 매개변수가 없을 때는 괄호 안에 void를 적거나, 괄호만 두고 내용을 생략하여 나타낸다.
- ④ 함수 몸체(body): 함수의 고유 기능을 수행할 코드들의 집합이다. 중괄호로 묶인다. 반환값이 있다면, **반드시 함수 바디에서 return문을 사용하여 값을 전달해주어야 한다.**

2) 함수의 원형 선언

- 함수의 원형 선언은 이전에 배운 것과 동일하다.

3) 값을 반환하는 함수의 호출

```
int result3) = calcComb(10, 5);
printf("%d", calcComb(n, m))3);
```

- ① 함수 이름을 불러 어떤 함수를 호출할 것인지 지정한다.
- ② 인수(인자): 함수의 매개변수로 전달해 줄 값을 적는다. 콤마 연산자를 이용해 여러 개의 인수를 전달하는 것이 가능하다. 함수로 전달할 인수가 없을 때는 괄호만 두고 내용을 생략하여 나타낸다.
- ③ 함수에서 값의 return이 일어나면 그 반환값은 함수의 호출부로 전달되므로, **호출부에서는 값을 저장하거나 바로 사용하여야 한다.** 그렇지 않으면 반환값은 날아간다.

2. 변수의 유효 범위

- C언어에서는 변수의 선언 위치에 따라 해당 변수의 유효 범위, 메모리 반환 시기, 초기화 여부, 저장되는 장소 등이 다르다.

1) 지역 변수

- ① 지역 변수는 지금껏 변수를 선언했던 것과 같이 '블록(block; _____)' 내에서 선언한 변수를 말한다. 함수의 매개변수 또한 함수 내에서 정의되는 지역 변수로 볼 수 있다.
- ② 지역 변수는 선언된 블록 내에서만 유효하며, 블록이 종료되면 메모리에서 사라진다.
- ③ 지역 변수가 선언되면 메모리상의 스택(stack) 영역에 저장되며, 직접 초기화하지 않으면 쓰레기값으로 초기화된다.
- ④ 선언되는 블록이 각기 다른 경우, 같은 이름의 지역 변수가 여럿 존재할 수 있다. 다만 한 함수 내에서 동명의 지역 변수를 여럿 선언하는 것은 혼란을 야기할 수 있으니 주의한다.

3. 개념 확인 문제

1) 아래 코드를 읽고 실행 결과를 예상해보아라. 각 변수의 유효 범위를 표시해보아라.

```
#include <stdio.h>

void local() { // 정의
    int var = 20;

    printf("local() 함수 내의 지역 변수 var의 값은 %d입니다.\n", var);
}

int main(void) {
    int i = 5;
    int var = 10;

    printf("main() 함수 내의 지역 변수 var의 값은 %d입니다.\n", var);

    if (i < 10) {
        local();

        int var = 30;

        printf("if 문 내의 지역 변수 var의 값은 %d입니다.\n", var);
    }

    printf("현재 지역 변수 var의 값은 %d입니다.\n", var);

    return 0;
}
```

2) 아래 코드를 읽고 실행 결과를 예상해보아라.

<pre>#include <stdio.h> int getBig(int a, int b); // 원형 int main(void) { int n1 = 10, n2 = 20; printf("큰 수는 %d\n", getBig(n1, n2)); return 0; }</pre>	<pre>int getBig(int a, int b) { // 정의 int big; if (a < b) big = b; else big = a; printf("큰 수 계산됨\n"); return big; }</pre>
---	---

- 함수의 인수로 함수가 전달되면, 인수로 전달된 함수가 먼저 실행된 후에 본 함수가 실행 된다!

3) 아래 코드를 읽고 실행 결과를 예상해보아라.

```
#include <stdio.h>

int getBig(int a, int b);

int main(void) {
    int n1 = 10, n2 = 20;
    int big;

    getBig(n1, n2);

    printf("%d", big);

    return 0;
}
```

```
int getBig(int a, int b) {
    int big;

    if (a < b)
        big = b;
    else
        big = a;

    return big;
}
```

4) 아래 코드를 읽고 실행 결과를 예상해보아라.

```
#include <stdio.h>

void f(int a, int b);

int main(void) {
    int n = 10, m = 10;

    printf("함수 실행 전\n");
    printf("n: %d, m: %d\n", n, m);

    f(n, m);

    printf("함수 실행 후\n");
    printf("n: %d, m: %d\n", n, m);

    return 0;
}
```

```
void f(int a, int b) {
    a++;
    b--;

    printf("함수 실행\n");
    printf("a: %d, b: %d\n", a, b);

    return;
}
```

4. 실습

1) LAB9_1 (long long int형 값을 반환하는 함수)

LAB8_6의 함수를 아래와 같이 long long int형 값을 반환하도록 수정하여라.

```
long long int calcFibonacci(int n) // (n + 1)번째 피보나치 값을 계산해 반환한다
```

```
몇 개의 피보나치 수열 값을 출력할까요? (3보다 큰 정수): 13
1 1 2 3 5 8 13 21 34 55 89 144 233
```

2) LAB9_2 (int형 값을 반환하는 함수)

LAB8_7의 함수를 아래와 같이 int형 값을 반환하도록 수정하여라. 또 -1이 입력되기 전까지는 계속 반복하여 실행되도록 하라.

```
int getCycleNumber(int n) // 수열을 출력하고 사이클 길이를 반환한다
```

```
Enter a number(-1 for exit): 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
길이는 16

Enter a number(-1 for exit): 7
7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
길이는 17

Enter a number(-1 for exit): -1
```

3) LAB9_3 (char형 값을 반환하는 함수)

두 정수 x와 y를 입력받고 x의 y승에 대한 사용자의 답 또한 입력받는다. 그리고 사용자의 답의 정답 여부를 출력한다 단, x의 y승을 계산하고 사용자의 답과 비교해 정답 여부에 관한 문자를 반환하는 함수를 정의하라.

Enter the number X: 3	Enter the number X: 10
Enter the number Y: 5	Enter the number Y: 2
3 ^ 5 = 243	10 ^ 2 = 1
0	X

4) LAB9_4 (int형 값을 반환하는 함수)

n이 소수이면 1을, 아니면 0을 반환하는 함수를 작성하여 -1이 입력받기까지 반복하여 실행 하라.

```
Enter a number(-1 for exit): 51
소수가 아닙니다.
Enter a number(-1 for exit): 11
소수입니다.
Enter a number(-1 for exit): -1
```

5) LAB9_5 (타입이 다른 매개변수를 여러 개 가지는 void 함수)

입력받은 n에 대하여, 2개 소수의 합으로 표현 가능한지 그 여부를 판별하는 프로그램을 작성하라. 단, LAB9_4에서 정의한 함수를 수정하지 않고 복사하여 사용한다.

```
22
3 + 19 = 22
5 + 17 = 22
11 + 11 = 22
2
Impossible.
```