

## 4.4

## 함수와 리턴

## 개요

**함수**는 특정 목적을 위해 만들어진 재사용 가능한 코드입니다. 함수는 입력값과 출력값을 가지며 프로그램 내의 어디에서든 재사용될 수 있습니다. 프로그램을 여러 개의 함수로 만들면 코드를 조직화하고 간단하게 만들 수 있습니다. 이것이 **추상화**의 한 예입니다. 함수를 한번 작성하면 그 함수가 구체적으로 어떻게 만들어졌는지 몰라도 함수를 사용할 수 있습니다. 그리고 함수를 만들지 않은 사람도 저수준의 상세 내용을 신경 쓰지 않고 함수를 사용(호출)할 수 있습니다.

## 핵심개념

- \* 함수
- \* 추상화
- \* 매개변수
- \* 인자
- \* return
- \* 지역변수
- \* 전역변수

## 함수의 정의

```

1  #include <stdio.h>
2  반환형 함수형 매개변수
3  void sayHi(void)
4  {
5      printf("Hi!\n");
6  }
7
8  int main(void)
9  {
10     sayHi();
11     sayHi();
12 }
```

▲ &lt;코드 1&gt;

여러분이 C에서 작성하는 모든 프로그램은 main이라는 함수를 가지고 있습니다. 물론 C의 프로그램은 더 많은 함수를 가질 수 있습니다. <코드 1>은 sayHi라는 **함수를 정의**한 것입니다.

(3행)함수의 첫 행은 세 부분으로 나누어져 있습니다. 첫째는 **반환 자료형**인데, 함수를 호출한 쪽에 **출력값으로 반환해줄 값의 자료형**입니다. 반환할 값이 없을 경우 반환할 값이 비어있다는 의미로 void를 씁니다. 둘째는 **함수의 이름**입니다. 공백이 있으면 안되고, C에서 이미 사용하는 키워드는 쓸 수 없습니다. 셋째는, 괄호 안에 인자라고도 알려진 함수의 **매개변수(parameter)**를 넣어줍니다. 이것은 함수의 입력값이며 입력값이 없으면 void를 씁니다. 이 첫 행 뒤에는 중괄호로 묶인 함수 코드가 작성됩니다.

sayHi 함수는 “HiWn”를 화면에 출력하는 함수입니다. main 함수 안에서 sayHi 함수가 두 번 호출되고 있습니다. (10, 11행) sayHi 함수는 매개변수를 받지 않기 때문에, 함수를 호출할 시에 괄호 안에 **인자(argument)**를 작성하지 않습니다. 실행 결과로 “HiWn”가 화면에 두 번 출력됩니다.

## 매개변수와 반환값

```

1  #include <stdio.h>
2
3  int square(int x)
4  {
5      return x * x;
6  }
7
8  int main(void)
9  {
10     printf("%d\n", square(2));
11     printf("%d\n", square(4));
12     printf("%d\n", square(8));
13 }
```

▲ &lt;코드 2&gt;

<코드 2>는 매개변수와 반환값을 가지는 square 함수를 보여줍니다. (3행)square 함수는 x라 불리는 하나의 정수를 매개변수로 받습니다. 또한 함수를 호출된 곳으로 정수를 반환합니다. (5행)과 같이 함수의 반환값을 return이라는 단어로 명시합니다. 여기에서 square 함수는 입력값 x를 자신과 곱한 후 값을 반환합니다. **return**이 쓰여진 행이 보통 함수의 마지막 행입니다.

이제 숫자를 제공하고 싶을 때는 프로그램 어디서나 square 함수를 사용할 수 있습니다. 왼쪽의 main 함수에서 square 함수는 세 번 호출됐는데, 매번 호출될 때마다 함수가 실행되고, 매개변수에 대한 제공 값이 함수를 호출한 곳으로 반환됩니다. printf(“%dWn”, square(2))는 printf(“%dWn”, 4)라고 쓰는 것과 동일한 결과가 출력됩니다.

## 범위(Scope)

함수 안에서 변수를 정의하거나 함수의 매개변수로 전달된 것들은 그 함수 안(지역)에서만 유효합니다.

이러한 변수들을 **지역변수**라고 부릅니다. 지역변수는 그 함수 안에서만 사용될 수 있고, 다른 곳에서는 존재하지 않습니다.

〈코드 2〉의 main 함수에서 x 변수를 참조하려고 한다면, 컴파일러는 오류를 만들어낼 것입니다. x라는 변수는 main함수의 범위(Scope) 안에 존재하지 않고, 오직 square 함수의 범위 안에서만 존재하기 때문입니다. 같은 논리로, main 함수에 정의된 어떤 변수도 square 함수 내에서 접근할 수 없습니다.