

프로그램의 기본 구성

- Hello world! -

성공회대학교 IT융합자율학부
소프트웨어공학전공
홍 성 준

C언어의 기본 단위인 '함수'란 무엇일까

◎ C언어의 기본 단위는 함수 (function)

- C프로그래밍은 함수를 만들고, 만든 함수의 실행 순서를 결정하는 것

◎ 수학에서의 함수

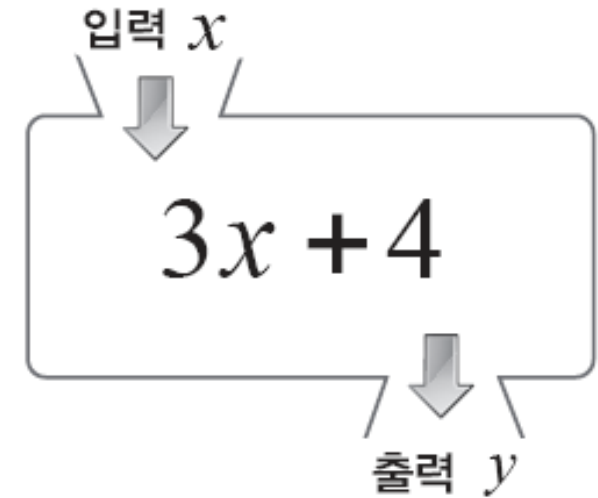
- 함수는 입력과 출력이 존재

◎ C언어의 함수

- C언어의 함수에서도 입력(인자)과 출력(반환값)이 존재

◎ C언어 함수와 관련된 용어

- 함수의 정의 : 실행 가능한 함수를 만드는 것
- 함수의 호출 : 함수 실행을 명령하는 것
- 인자의 전달 : 함수를 호출 시 함수의 입력(인자)을 전달하는 것





Hello world!

© Hello.c

```
#include <stdio.h>
int main(void)
{
    printf("Hello world!\n");
    return 0;
}
```

Hello.c 은 어떻게 실행이 될까?

◎ C 프로그램의 시작

- main() 함수부터 호출되면서 프로그램이 시작

◎ main() 함수란?

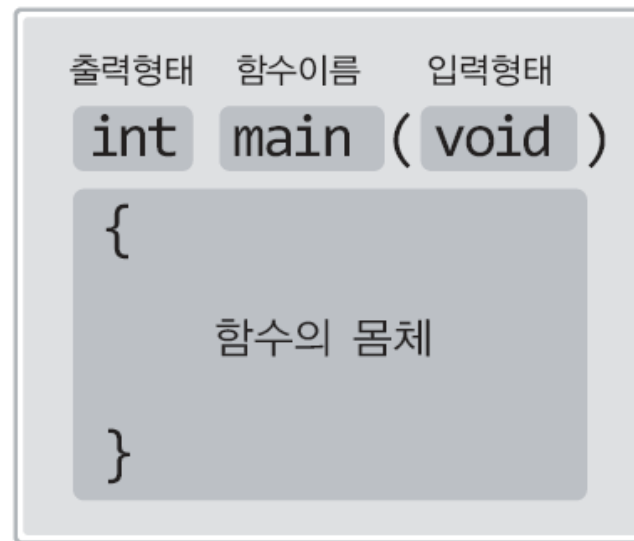
- C 프로그램에서 처음으로 호출되는 함수
- C 언어로 구현된 모든 프로그램은 시작을 알리는 main() 함수를 반드시 정의 해야 함

◎ 함수의 정의

- 함수의 이름 (name)
- 입력 형태 (input type)
- 출력 형태 (output type)
- 함수의 몸체 (body)

```
int main(void)
{
    printf("Hello world! \n");
    return 0;
}
```

↓
순차적으로 실행



```
#include <stdio.h> 헤더파일 선언문
int main(void)
{
    함수 printf()의 호출문
    printf("Hello world! \n");
    return 0;
}
```

◎ 표준 함수

- 기본적으로 제공되는 함수

◎ 표준 라이브러리

- 표준 함수들의 모임

◎ #include <stdio.h>

- stdio.h 파일 내용을 여기다 붙여라
- stdio.h 는 printf() 함수가 선언되어 있는 표준 입출력 라이브러리의 헤더 파일
- 헤더 파일의 선언은 main 함수 앞에 해야함

◎ printf("Hello world! \n");

- printf() 함수를 "Hello world! \n"를 인자(입력)로 하여 호출

◎ return 0;

- 현재 실행 중인 함수의 종료를 의미
- main() 함수의 결과로 운영체제에 0을 반환

C언어의 기본 문법 (1/3)

◎ 세미콜론

- C언어는 구문이 끝날 때 ; (세미콜론)을 붙임
- 세미콜론을 붙여야 할 때 세미콜론이 없으면 구문 오류(syntax error)가 발생
- 세미콜론을 붙여야 할 때와 붙이지 않을 때가 경우가 있음

◎ 주석 (comment)

- 소스 코드에 삽입된 메모로 컴파일러가 무시하므로 프로그램 실행에 영향을 주지 않음
- 코드에 대한 자세한 설명을 하거나 특정 코드를 임시로 컴파일 되지 않도록 하기 위해 사용

◎ 주석의 종류

- 한 줄 단위 주석 : //
- 블록 단위 주석 : 시작 /*, 끝 */

한 행 단위로의 주석처리

```
// 주석처리 된 문장1
// 주석처리 된 문장2
// 주석처리 된 문장3
```

```
/* 주석처리 된 문장 */
```

한 행의 주석처리

```
/*
    주석처리 된 문장1
    주석처리 된 문장2
    주석처리 된 문장3
*/
```

여러 행의 주석처리



주석 처리의 예

```
// Hello, world! 출력  
printf("Hello, world!\n");
```

```
//printf("Hello, world!\n");
```

```
int a = 1 + 2; // 더하기  
  
printf("Hello, world!\n");//printf("1234567890");
```

```
/*  
printf("Hello, world!\n");  
printf("1234567890");  
*/
```

```
printf("Hello" /* 안녕하세요 */);
```



주석 처리시 주의할 점

```
1.  /*
2.     주석처리 된 문장1
3.     /* 단일 행 주석처리 */
4.     주석처리 된 문장2
5.  */
```

잘못 달린 주석(컴파일 시 오류 발생)

```
1.  /*
2.     주석처리 된 문장1
3.     // 단일 행 주석처리
4.     주석처리 된 문장2
5.  */
```

잘 달린 주석(컴파일 시 오류 발생하지 않음)

주석 처리의 예

© HelloComment.c

```
/*
제 목: Hello world 출력하기
기 능: 문자열의 출력
파일이름: HelloComment.c
수정날짜: 2014. 07. 15
작성자: 윤성우
*/
#include <stdio.h>    // 헤더파일 선언

int main(void)    // main 함수의 시작
{
    /*
    이 함수 내에서는 하나의 문자열을 출력한다.
    문자열은 모니터로 출력된다.
    */
    printf("Hello world! \n");    // 문자열의 출력
    return 0;    // 0의 반환
}    // main 함수의 끝
```

C언어의 기본 문법 몇 가지 (2/3)

◎ 중괄호 { }

- C언어에서 중괄호는 코드의 범위를 나타냄
- 중괄호의 위치 스타일은 정해져 있지 않음

```
int main()  
{ // 줄의 시작 부분에서 중괄호 사용(1번 스타일)  
  
}
```

```
int main() { // 줄의 맨 마지막에서 중괄호 사용(2번 스타일)  
  
}
```

C언어의 기본 문법 몇 가지 (3/3)

◎ 들여쓰기

- C 프로그래밍을 하다 보면 키워드에 따라 들여쓰기 하는 경우가 있음
- 들여쓰기 하는 방법으로 공백(space) 2칸, 4칸, 탭(tab) 등이 있음
- 소스 코드의 가독성을 위해 반드시 필요함

```
int main()
{
    공백 2칸  printf("Hello, world!\n");

    공백 4칸  printf("Hello, world!\n");

    탭 1칸   printf("Hello, world!\n");

    return 0;
}
```

printf() 함수를 이용한 정수 출력과 서식 문자

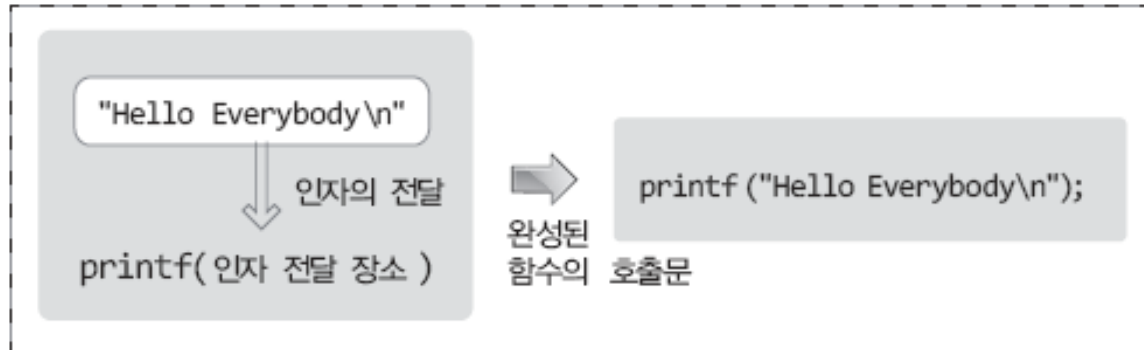
© PrintfOne.c

```
int main(void)
{
    printf("Hello Everybody\n");
    printf("%d\n", 1234);
    printf("%d %d\n", 10, 20);
    return 0;
}
```

실행결과

```
Hello Everybody
1234
10 20
```

- printf 함수는 첫 번째 인자로 전달된 문자열을 출력.



printf() 함수를 이용한 정수 출력과 서식 문자

© PrintfOne.c

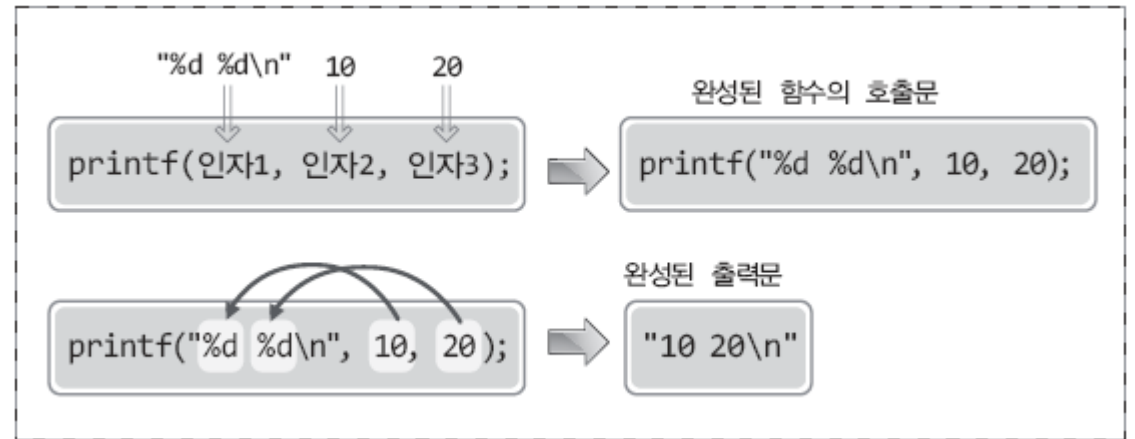
```
int main(void)
{
    printf("Hello Everybody\n");
    printf("%d\n", 1234);
    printf("%d %d\n", 10, 20);
    return 0;
}
```

실행결과

```
Hello Everybody
1234
10 20
```

- 서식문자 (format string)

- 출력의 대상의 출력하고자 하는 형태를 지정하는 용도로 사용하는 문자
- 출력의 대상은 첫번째 문자열 뒤에 이어서 표시하고, 콤마(,)로 구분함
- 서식문자가 두 개면 출력 대상도 2개여야 함
- 정수형 서식문자 '%d', 실수형 서식문자 '%f', 문자형 서식문자 '%c', 문자열 서식문자 '%s'





서식 문자와 개행 문자 사용하기

© PrintfTwo.c

```
int main(void)
{
    printf("My age: %d \n", 20);
    printf("%d is my point \n", 100);
    printf("Good \nmorning \neverybody\n");
    return 0;
}
```

실행결과

My age: 20
100 is my point
Good
morning
everybody

