



C프로그래밍

Lecture 02. 프로그램의 기본 구성

동덕여자대학교
데이터사이언스 전공
권 범

목차

- ❖ 01. “Hello, world!” 들여다보기
- ❖ 02. 주석이 들어가야 완성된 프로그램
- ❖ 03. printf 함수의 기본적인 이해
- ❖ 04. 연습 문제

01. “Hello, world!” 들여다보기

- 02. 주석이 들어가야 완성된 프로그램
- 03. printf 함수의 기본적인 이해
- 04. 연습 문제

01. “Hello, world!” 들여다보기

❖ Hello.c 소스 코드

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

각 문장들은 무엇을 의미하는 것일까요?

“ 소스 코드 ”
(Source Code)

프로그램 제작에 사용되는 설계파일을 의미합니다.

01. "Hello, world!" 들여다보기

❖ C언어의 기본 단위: 함수 (1/5)

- C언어로 구현되어 있는 프로그램은 보통 여러 개의 함수로 구성됩니다.
- 이 함수들의 실행 순서를 정의하는 것이 C언어로 프로그램을 작성하는 것입니다.

C언어는 함수에서 시작해서 함수로 끝난다

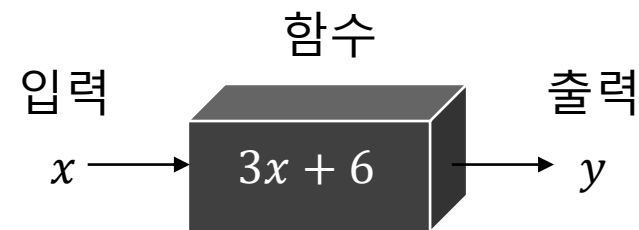
01. "Hello, world!" 들여다보기

❖ C언어의 기본 단위: 함수 (2/5)

- 함수의 특성 → 수학적으로 함수에는 입력과 출력이 존재
- C언어의 함수에도 입력과 출력이 존재

“ 함수 ”
(Function)

적절한 입력과 그에 따른 출력이 존재하는 것을
함수라고 합니다.



01. "Hello, world!" 들여다보기

❖ C언어의 기본 단위: 함수 (3/5)

C언어의 함수와 관련된 용어 정리

- ✓ 함수의 정의: 만들어진 함수, 실행이 가능한 함수를 일컬음
- ✓ 함수의 호출: 함수의 실행을 명령하는 행위
- ✓ 인자의 전달: 함수의 실행을 명령할 때, 전달하는 입력 값



01. "Hello, world!" 들여다보기

❖ C언어의 기본 단위: 함수 (4/5)

- Hello.c에서 함수는 어디에 있나요?

C언어는 함수에서 시작해서 함수로 끝난다

프로그램의 시작

- ✓ 첫 번째 함수가 호출이 되면서 프로그램은 시작이 됩니다.

제일 먼저 호출되는 함수는?

- ✓ main이라는 이름의 함수!
- ✓ 따라서 C언어로 구현된 모든 프로그램은 시작점에 해당하는 main이라는 이름의 함수를 반드시 정의해야 합니다.
- ✓ 즉, main이라는 이름의 함수가 자동으로 호출이 되면서 프로그램은 실행됩니다.

01. "Hello, world!" 들여다보기

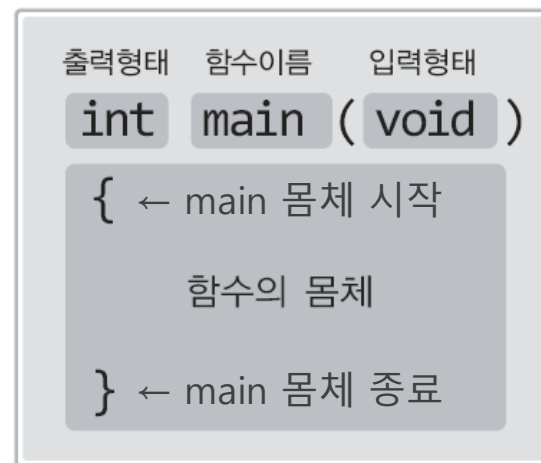
❖ C언어의 기본 단위: 함수 (5/5)

함수의 기능

- ✓ 함수의 기능은 중괄호 안에 표현이 되며, 중괄호 안에 표현된 함수의 기능을 가리켜 함수의 몸체(Body)라고 합니다.

C언어의 함수에 표시가 되는 세 가지

- ✓ 함수의 이름: 함수를 호출할 때 사용하게 되는 이름
- ✓ 출력 형태: 실행의 결과! 일반적으로 반환형(Return Type)이라고 합니다.
- ✓ 입력 형태: 함수를 호출할 때 전달하는 입력 값의 형태



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

↓
순차적으로 실행됩니다.

01. "Hello, world!" 들여다보기

❖ 연산을 수행하는 모든 문장들의 끝: 세미콜론(;) (1/2)

- 함수 내에 존재하는 문장의 끝에는 세미콜론 문자(;)를 붙여줍니다.

✓ C언어에서 세미콜론은 문장의 끝을 표현하기 위한 문자입니다.

- 열 줄에 표현된 코드는 열 개의 문장인가요?

- ✓ 하나의 문장이 둘 이상의 줄에 표시될 수도 있고, 한 줄에 둘 이상의 문장이 표시될 수도 있습니다.
- ✓ 즉, 줄 바꿈은 문장의 바꿈을 뜻하는 것이 아닙니다.

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

01. "Hello, world!" 들여다보기

❖ 연산을 수행하는 모든 문장들의 끝: 세미콜론(;) (2/2)

- 다음 세 main 함수는 모두 동일한 프로그램입니다.
- 줄 바꿈의 차이가 프로그램의 차이로 이어지지 않습니다.

한 줄에 하나의 문장을 표시하는 것이 가장 일반적이고, 보기도 좋습니다.

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



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



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



01. “Hello, world!” 들여다보기

❖ Hello.c 소스 코드 분석 (1/11)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

6번 줄(Line)에서 printf라는 이름으로 정의되어 있는 함수의 실행을 요구하고 있습니다.

printf 함수는 입력받은 인자 값을 모니터에 출력하는 기능을 수행합니다.

01. "Hello, world!" 들여다보기

❖ Hello.c 소스 코드 분석 (2/11)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

- ✓ printf에서 f는 formatted(형식화된)를 의미합니다.
- ✓ 'wn'에서 n은 new line(줄 바꿈)을 의미합니다.
- ✓ 즉, 문자열 뒤에 'wn'이 삽입되면 줄을 바꾸라는 의미입니다.
- ✓ 'wn'과 같이 백슬래시(\) 뒤에 한 문자가 오는 문자 조합을 이스케이프 시퀀스(Escape Sequence)라고 합니다.
- ✓ 이스케이프 시퀀스는 컴퓨터를 제어하는데 사용되는 특수 문자입니다.

01. "Hello, world!" 들여다보기

❖ Hello.c 소스 코드 분석 (3/11)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

가만히 생각해 보면,
우리는 printf 함수를 만든 적이 없습니다.

printf 함수는 누가 만들어 놓은 것이고,
어디에 존재하는 함수인 것일까요?

01. "Hello, world!" 들여다보기

❖ Hello.c 소스 코드 분석 (4/11)

“표준 라이브러리” (Standard Library)

기본적으로 제공되는 함수들을 가리켜
표준 라이브러리라고 합니다.

- ✓ C언어에서는 수많은 표준 라이브러리를 제공하고 있습니다.
- ✓ 우리는 프로그램을 구현할 때 처음부터 끝까지 스스로 구현해야 할 필요가 없습니다.
- ✓ 필요한 함수가 만들어져 있다면, 가져다 사용하면 됩니다.

01. "Hello, world!" 들여다보기

❖ Hello.c 소스 코드 분석 (5/11)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

확장자가 .h로 끝나는 파일을
헤더 파일(Header File)이라고 부릅니다.

표준 라이브러리 함수를 사용하기 위해서는
해당 함수의 선언을 포함하는 헤더 파일을 포함(include)해야 합니다.

stdio.h라는 헤더 파일 내에 printf 함수가 선언되어 있습니다.

01. “Hello, world!” 들여다보기

❖ Hello.c 소스 코드 분석 (6/11)

“stdio.h 헤더 파일”

stdio은 Standard Input/Ouptut Library
(표준 입출력 라이브러리)의 약어입니다.

- ✓ C언어의 표준 라이브러리 중 하나입니다.
- ✓ 매크로 정의, 상수(Constant), 여러 입출력 함수 등이 포함되어 있습니다.
- ✓ printf 함수도 stdio.h 헤더 파일에 선언되어 있습니다.

01. "Hello, world!" 들여다보기

❖ Hello.c 소스 코드 분석 (7/11)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

경우에 따라서는 하나의 프로그램 내에 여러 개의 헤더 파일이 포함되기도 합니다.

(주의) 헤더 파일의 포함을 알리는 선언이 프로그램 작성 시 제일 먼저 등장해야 합니다.

01. “Hello, world!” 들여다보기

❖ Hello.c 소스 코드 분석 (8/11)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

return 문은 두 가지 의미를 갖습니다.

- ① 함수를 빠져 나온다는 의미입니다.
- ② 함수를 호출 한 영역으로 값을 반환(Return)한다는 의미입니다.

01. “Hello, world!” 들여다보기

❖ Hello.c 소스 코드 분석 (9/11)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

- ✓ 7번 줄(Line)은 main 함수를 빠져 나오면서 main 함수를 호출한 영역으로 0을 반환하라는 의미가 됩니다.
- ✓ main 함수의 실행을 끝내게 되면, 프로그램은 종료하게 됩니다.

01. "Hello, world!" 들여다보기

❖ Hello.c 소스 코드 분석 (10/11)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

**main 함수의 마지막에서
0을 반환하는 이유가 무엇일까요?**

- ✓ main 함수는 운영체제에 의해서 호출되는 함수입니다.
- ✓ 따라서, 반환된 값은 운영체제로 넘어가게 됩니다.
- ✓ 운영체제는 main 함수가 최종적으로 반환해 주는 값을 통해서 프로그램이 잘 종료되었는지, 아니면 문제가 발생했는지 파악하게 됩니다.

01. “Hello, world!” 들여다보기

❖ Hello.c 소스 코드 분석 (11/11)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

이제 위 Hello.c 소스 코드가
어느정도 이해가 되시지요?



02. 주석이 들어가야 완성된 프로그램

- 01. "Hello, world!" 들여다보기
- 03. printf 함수의 기본적인 이해
- 04. 연습 문제


02. 주석이 들어가야 완성된 프로그램

❖ 주석(Comment)이란?

- ✓ 주석은 소스 코드에 삽입된 메모를 뜻합니다.
- ✓ 주석은 컴파일의 대상에서 제외가 되기 때문에 주석의 유무는 컴파일 및 실행 결과에 영향을 주지 않습니다.

❖ 주석의 필요성

소스 코드의 분석은 글을 읽는 것 만큼 간단하지 않습니다.



소스 코드를 분석해야 하는 다른 사람을 위해서,
그리고 소스 코드를 작성한 작성자 스스로를 위해서라도
주석을 작성해 놓을 필요가 있습니다.

02. 주석이 들어가야 완성된 프로그램

❖ 주석을 작성하는 방식 (1/2)

① 블록 단위 주석

```
/* 주석 처리 된 문장 */
```

← 한 행(Row)의 주석 처리

```
/*
```

```
    주석 처리 된 문장1
```

```
    주석 처리 된 문장2
```

```
    주석 처리 된 문장3
```

```
*/
```

← 여러 행의 주석 처리

02. 주석이 들어가야 완성된 프로그램

❖ 주석을 작성하는 방식 (2/2)

② 행 단위 주석

```
// 주석 처리 된 문장1  
// 주석 처리 된 문장2  
// 주석 처리 된 문장3
```

← 한 행 단위로의 주석 처리

주석을 작성하는 방식은
프로젝트 별로 팀원과 상의하여 결정하게 됩니다.

02. 주석이 들어가야 완성된 프로그램

❖ Visual Studio에서 주석 단축키

- 주석 설정: Ctrl + K + C
 - 주석 해제: Ctrl + K + U
- ✓ 커서가 위치한 줄에서 단축키를 누르면 해당 줄이 주석 처리됩니다.
 - ✓ 주석 처리하고자 하는 영역을 드래그(Drag)해서 블록 처리한 다음 단축키를 누르면 해당 영역이 주석 처리됩니다.
 - ✓ 주석 해제도 주석 설정과 동일한 방법으로 수행됩니다.

02. 주석이 들어가야 완성된 프로그램

❖ 주석 처리의 예

```
/*
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
```

과도하게 처리된 주석(주석도 과하면 좋지 않습니다)!
주석 작성 방법을 소개하기 위한 예제일 뿐입니다.

02. 주석이 들어가야 완성된 프로그램

❖ 주석 처리에 있어서 주의할 점

```
1  /*
2      주석 처리된 문장1
3      /* 단일 행 주석 처리 */
4      주석 처리된 문장2
5  */
```

잘못 작성된 주석(컴파일 시 오류 발생)

```
1  /*
2      주석 처리된 문장1
3      // 단일 행 주석 처리
4      주석 처리된 문장2
5  */
```

잘 작성된 주석(컴파일 시 오류 발생하지 않음)

주석을 작성하다 보면 주석이 중첩되는 경우가 발생하기도 합니다.
(주의) 블록 단위 주석은 중첩되는 형태로 작성할 수 없습니다.

03. printf 함수의 기본적인 이해

- 01. "Hello, world!" 들여다보기
- 02. 주석이 들어가야 완성된 프로그램
- 04. 연습 문제

03. printf 함수의 기본적인 이해

❖ printf 함수를 이용한 정수(Integer) 출력 (1/5)

```
1  /* Hello.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello, world!\n");
7      return 0;
8  }
```

지금까지 문자열을 출력하기 위해서
printf 함수를 사용했습니다.

지금부터는 printf 함수를 사용하여
정수(Integer)를 출력하는 방법을 살펴보겠습니다.

03. printf 함수의 기본적인 이해

❖ printf 함수를 이용한 정수(Integer) 출력 (2/5)

```
1  /* printf1.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello Everybody\n");
7      printf("%d\n", 1234);
8      printf("%d %d\n", 10, 20);
9      return 0;
10 }
```

Hello Everybody
1234
10 20

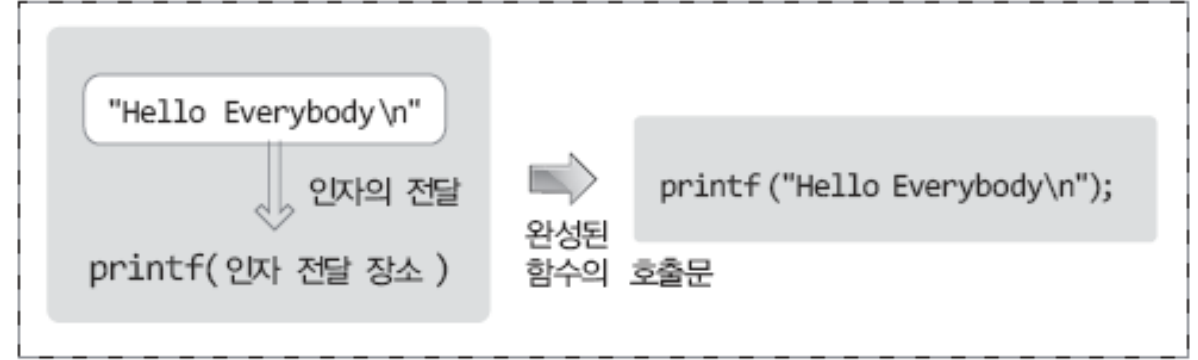
출력 결과를 보면 문자열과 정수 모두 제대로 출력되고 있음을 확인 할 수 있습니다.

지금부터 6~8번 줄(Line)에 작성된 내용을 살펴보겠습니다.

03. printf 함수의 기본적인 이해

❖ printf 함수를 이용한 정수(Integer) 출력 (3/5)

```
1  /* printf1.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello Everybody\n");
7      printf("%d\n", 1234);
8      printf("%d %d\n", 10, 20);
9      return 0;
10 }
```



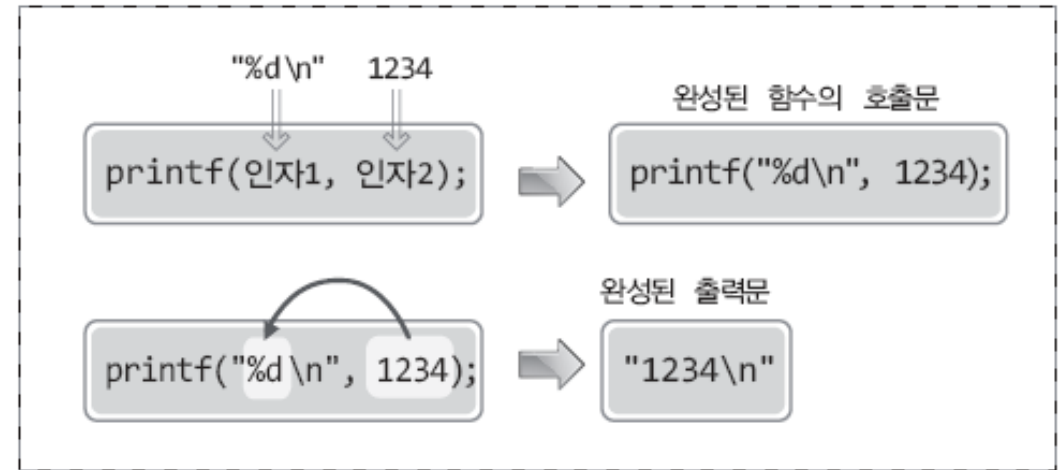
printf 함수는 전달된 문자열을 출력하는 기능을 갖고 있습니다.

03. printf 함수의 기본적인 이해

❖ printf 함수를 이용한 정수(Integer) 출력 (4/5)

```
1  /* printf1.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello Everybody\n");
7      printf("%d\n", 1234);
8      printf("%d %d\n", 10, 20);
9      return 0;
10 }
```

7번 줄을 보면 호출되는 printf 함수에 두 개의 인자가 전달되고 있습니다.



인자1: %d\n

- ✓ 문자열에 삽입된 %d를 가리켜 서식 문자라고 합니다.
- ✓ 서식 문자는 출력의 형태를 지정하는 용도로 사용됩니다.
- ✓ %d는 부호가 있는 10진수 정수의 형태로 출력하라는 의미를 갖습니다(d ← decimal, 10진법의).

인자2: 1234

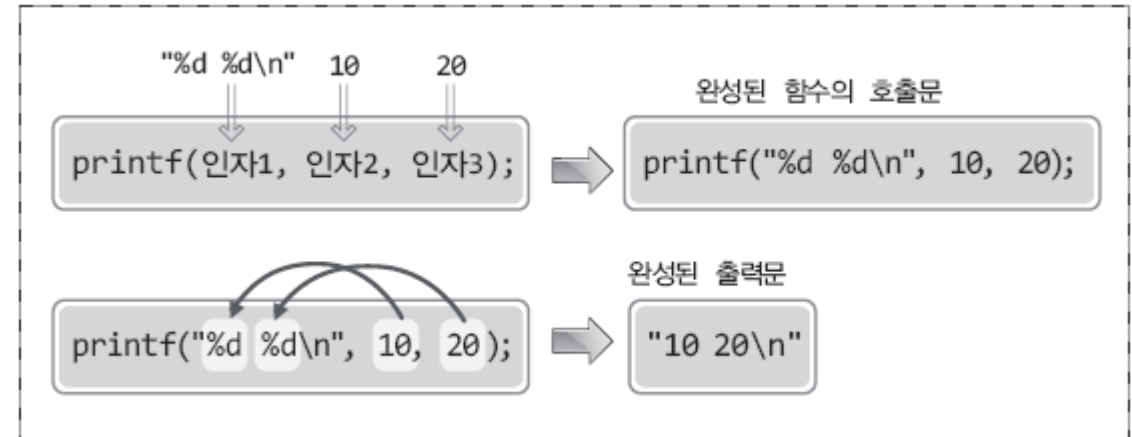
- ✓ %d를 작성하여 출력의 형태를 지정해 주고, 출력하고자 하는 대상을 두 번째 인자로 전달하면 됩니다.
- ✓ 인자1과 인자2 사이의 구분은 콤마(,)로 구분합니다.

03. printf 함수의 기본적인 이해

❖ printf 함수를 이용한 정수(Integer) 출력 (5/5)

```
1  /* printf1.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hello Everybody\n");
7      printf("%d\n", 1234);
8      printf("%d %d\n", 10, 20);
9      return 0;
10 }
```

8번 줄을 보면 호출되는 printf 함수에 서식 문자 %d가 두 개 존재합니다.



8번 줄

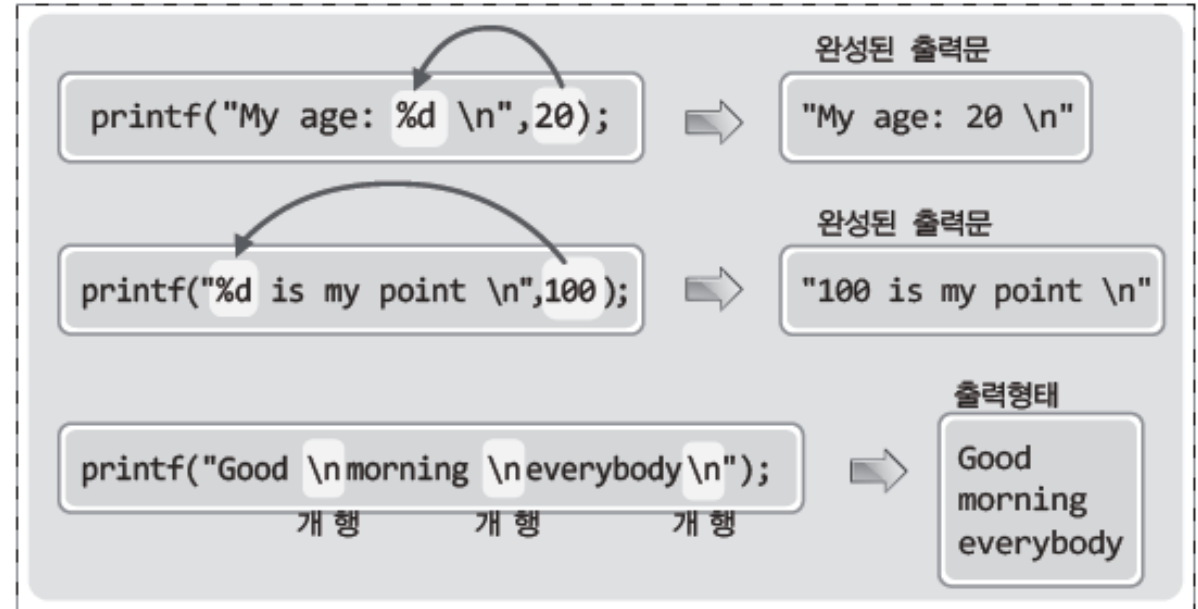
- ✓ 서식 문자가 두 개 존재하고, 실제 출력할 대상이 되는 인자가 두 번째, 세 번째 인자로 각각 전달되고 있습니다.

03. printf 함수의 기본적인 이해

❖ printf 함수를 이용한 다양한 출력 형태

```
1  /* printf2.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("My age: %d\n", 20);
7      printf("%d is my point\n", 100);
8      printf("Good \nmorning \neverybody");
9      return 0;
10 }
```

My age: 20
100 is my point
Good
morning
everybody



이후에는 보다 다양한 서식 문자를 공부하게 됩니다.
그리고 다양한 형태로 출력의 형태를 조합할 수 있게 됩니다.

04. 연습 문제

- 01. "Hello, world!" 들여다보기
- 02. 주석이 들어가야 완성된 프로그램
- 03. printf 함수의 기본적인 이해

04. 연습 문제

❖ 연습 문제 1.

- 다음과 같은 형태로 본인의 이름을 출력하는 프로그램을 작성하세요.
(단, printf 함수는 1번만 호출하는 형태로 구성하세요.)

04. 연습 문제

❖ 연습 문제 1. 정답 및 해설

```
1  /* example1.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("홍\n홍길\n홍길동\n");
7
8      return 0;
9  }
```

홍
홍길
홍길동

04. 연습 문제

❖ 연습 문제 2.

- 본인의 이름, 주소, 전화번호를 출력하는 프로그램을 작성하세요.
(단, printf 함수는 3번만 호출하는 형태로 구성하세요.)

이름: 홍길동 주소: 서울시 구로구 전화번호: 02-1234-5678
--

04. 연습 문제

❖ 연습 문제 2. 정답 및 해설

```
1  /* example2.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("이름: 홍길동\n");
7      printf("주소: 서울시 구로구\n");
8      printf("전화번호: 02-1234-5678\n");
9
10     return 0;
11 }
```

이름: 홍길동
주소: 서울시 구로구
전화번호: 02-1234-5678

04. 연습 문제

❖ 연습 문제 3.

- 다음과 같은 출력 결과를 보일 수 있도록 프로그램을 작성하세요.
(단, 등장하는 숫자는 문자열 속에 직접 넣지 말고, 서식 문자 %d를 이용하세요.)

저의 이름은 홍길동입니다.
저의 나이는 21살이고요.
제가 사는 번지수는 111-222번지입니다.

04. 연습 문제

❖ 연습 문제 3. 정답 및 해설

```
1  /* example3.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("저의 이름은 홍길동입니다.\n");
7      printf("저의 나이는 %d살이고요.\n", 21);
8      printf("제가 사는 번지수는 %d-%d번지입니다.\n", 111, 222);
9
10     return 0;
11 }
```

저의 이름은 홍길동입니다.
저의 나이는 21살이고요.
제가 사는 번지수는 111-222번지입니다.

04. 연습 문제

❖ 연습 문제 4.

- 다음과 같은 출력 결과를 보일 수 있도록 프로그램을 작성하세요.
(단, 숫자는 문자열 속에 직접 넣지 말고, 서식 문자 %d를 이용하세요.)


2 * 3 = 6

2 * 4 = 8

04. 연습 문제

❖ 연습 문제 4. 정답 및 해설

```
1  /* example4.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("%d * %d = %d\n", 2, 3, 6);
7      printf("%d * %d = %d\n", 2, 4, 8);
8
9      return 0;
10 }
```



```
2 * 3 = 6
2 * 4 = 8
```

- ❖ 01. “Hello, world!” 들여다보기
- ❖ 02. 주석이 들어가야 완성된 프로그램
- ❖ 03. printf 함수의 기본적인 이해
- ❖ 04. 연습 문제

THANK YOU!

Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: bkwon@dongduk.ac.kr