



[1주차 2강] C 프로그래밍 시작하기(2)



학습 내용

1.3 첫 프로그램 분석

1.4 출력 기초

1.5 입력 기초



학습 목표

1.3 C프로그램의 기본적인 형태를 이해한다.

1.4 출력의 기초에 대해 학습한다.

1.5 입력의 기초에 대해 학습한다.





나의 첫 C 프로그램 (hello.c)

표준 입출력(standard input output)을 포함(include)하라

main 함수
(프로그램 시작 지점)

main 함수 시작
(여는 중괄호)

main 함수의 끝
(닫는 중괄호)

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
printf("Hello, World!!");
```

```
return 0;
```

```
}
```

화면에 "... " 부분을
출력(print)하라

일단은 그대로 쓰자

프로그램의 각 줄을 하나씩 지워 보고 빌드&실행하면서
어떤 결과가 나오는 지 확인해보자.


 **들여쓰기와 주석** : 가독성을 위한 수단으로 문법적 의미는 없음

- **들여쓰기** : 라인의 포함 여부를 쉽게 파악하기 위함
- **주석** : 코드에 추가된 프로그램에 대한 설명

```
/* 나의 첫 C 프로그램
Hello world program */

#include <stdio.h>           // 헤더 파일 포함

int main( )                 // main 함수 (프로그램 시작 지점)
{                             // main 함수의 시작
    .....▶ printf("Hello, world!!"); // 화면에 "...의 내용을 출력
    .....▶ return 0;             // 0을 반환
}                             // main 함수의 끝
```

 **들여쓰기와 주석은 선택이 아니라 필수!!** 



들여쓰기

- ☑ 보통 [tab] 키나 [space] 키를 사용
 - 아래 코드에서 붉은 점이 표시된 두 라인이 main() 함수에 속해 있음을 **알아보기 쉽게 하기 위해** 들여쓰기 했음
 - 들여쓰기는 문법적으로 아무런 의미를 가지지 않음
 - ✓ 두 라인이 main() 함수에 속해있다는 것을 나타내는 문법적 장치는 **중괄호 쌍**

실습하기



int main()이후의 내용을 한 줄에 작성해보고,
컴파일 & 실행시켜 보자.



```
#include <stdio.h>
int main( )
{
    ●...▶ printf("Hello, world!!");
    ●...▶ return 0;
}
```

주석

- ❑ `/* ... */`: `/*`과 `*/` 사이의 내용을 모두 주석으로 처리
- ❑ `//`: `//` 이 후부터 해당 라인 끝까지의 내용을 주석 처리

```
/* 나의 첫 C프로그램  
Hello world program */  
  
#include <stdio.h>           // 헤더 파일 포함  
  
int main( )                  // main 함수 (프로그램 시작 지점)  
{                             // main 함수의 시작  
    printf("Hello, world!!"); // 화면에 "...의 내용을 출력  
    return 0;                 // 0을 반환  
}                             // main 함수의 끝
```




오류의 종류와 디버깅(1)

- **버그(bug)** : 프로그램에 존재하는 오류
- **디버깅(debugging)** : 오류를 고치는 행위

- **컴파일 오류** : 문법적 오류
컴파일러가 문법을 검사하여 오류가 있으면 알려줌
- **컴파일 경고** : 오류는 아니지만, 오류의 가능성이 있는 부분

실습하기



다음의 코드를 컴파일해보고,
컴파일 오류 내용을 확인해보자.



```
#include <stdio.h>
int main( ) {
    printf("Hello, World!!")
    retrn 0;
}
```

⇒ 세미콜론 누락
⇒ return 철자 틀림



오류의 종류와 디버깅(2)

- 런타임 오류(runtime error) 또는 실행 오류
: 프로그램의 결과가 의도와 다르거나 비정상적으로 종료되는 경우

```
#include <stdio.h>
int main( ) {
    printf("Jello, World!!") ⇒ H를 J로 잘못 타이핑
    return 0;
}
```

- ✓ 위에는 매우 단순한 오류이나,
일반적으로 런타임 오류는 찾기도 어렵고 해결하기도 어려움
- ✓ 보통 '디버깅'이라 함은 런타임 오류를 해결하는 것을 지칭
- 대부분의 개발도구에는 런타임 오류 해결에 필요한 부가 기능 제공
→ VS의 디버깅 메뉴 (간단한 사용법은 교재 부록 참조)



화면에 내용 출력하기

- printf()함수는 큰 따옴표의 시작과 끝 사이에 있는 내용 출력
- 큰 따옴표 밖의 공백은 출력에 전혀 영향 없고, 내부 공백만 출력에 영향을 줌

소스 코드

```
printf( "Hello,   world!!" );
```



실행 화면

```
Hello,   world!!
```

```
printf( "   I'm a student." );
```



```
I'm a student.
```

```
printf(   "100"   ) ;
```



```
100
```

```
printf(   "!@# $"   );
```



```
!@# $
```



printf 함수를 여러 번 사용하면?

☑ C 언어의 문장은 기본적으로 위에서부터 순차적으로 수행

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    printf("한국대학교");
```

```
    printf("신입생");
```

```
    printf("홍길동");
```

```
    return 0;
```

```
}
```

순차
실행

실행 화면

한국대학교신입생홍길동

→ 심화 1: 위 내용을 띄어 쓰기를 넣어서 출력해보자.

→ 심화 2: 위 내용을 printf() 함수 하나만 사용해서 출력해보자.

앞의 결과를 오른쪽과 같이 3개의 줄에 출력하려면?

☑ 개행 문자 (\n)사용

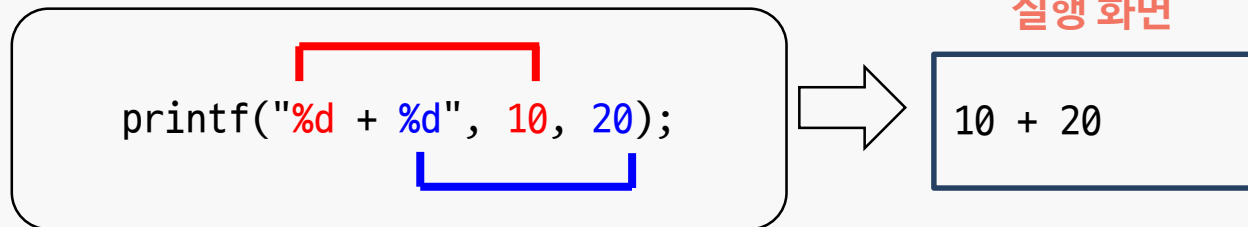
*** 참고 :** 역슬래시 기호(\)는 한글 폰트에서는 원화 기호(₩)로 보임

실행 화면

```
printf("한국대학교\n");
```

```
한국대학교  
신입생  
홍길동
```

정수 값 출력



* %d 는 정수형 값을 출력하는 서식

- '%d'는 해당 위치에 정수를 출력하라는 의미 (서식 지정자)
- 출력할 값은 두 번째 큰 따옴표 뒤에 명시
- 하나의 printf 함수에 여러 개의 서식이 삽입될 수 있으며, 문장 앞쪽의 서식부터 차례로 대응

실습하기



[예제 1.1] 다음 프로그램의 결과를 예측해보고, 프로그램을 작성하여 결과를 확인해 보자.



```
int main()  
{  
    printf("100");  
    printf("\n");  
    printf("%d", 100);  
    printf("\n");  
    return 0;  
}
```

```
int main( )  
{  
    printf("100+200");  
    printf("\n");  
    printf("%d", 100+200);  
    printf("\n");  
    return 0;  
}
```




변수에 저장된 값 출력

- ❑ 변수에 저장된 값도 동일한 방식으로 화면에 출력
- ❑ 변수에 대한 자세한 내용은 2장에서 학습

```
int main()
{
    int c;        // 변수 c 선언

    c = 10+20;    // c 에 10+20 의 결과, 즉 30 저장

    printf("c=10+20 출력:"); // "... " 내용 출력
    printf("%d", c);        // 변수 c에 저장된 값 출력
    return 0;
}
```



키보드를 이용하여 값 입력 받기

☑ scanf() 함수는 큰 따옴표 "..." 안에 있는 서식 지정자에 따라 값을 입력 받음

```
int main()
{
    int a;                // 변수 a 선언

    printf("정수를 입력하시오: "); // 안내문 출력
    scanf("%d", &a);        // 정수 값 입력 받아
                           // 변수 a에 저장
    printf("입력된 값: %d", a); // 변수 a의 값 출력

    return 0;
}
```



앞의 프로그램을 작성하여 실행하면?

- ❑ 화면에 "정수를 입력하시오: "를 출력한 후,
- ❑ 프로그램은 키보드로 값이 입력되길 기다림
- ❑ 아무 정수나 입력하고 [Enter] 키를 치면, 나머지 부분이 실행됨
실행 화면

정수를 입력하시오: 15
입력된 값: 15



참고

- VS에서 scanf_s() 함수를 사용하라고 하면서 컴파일 오류가 발생하는 경우 아래 문장을 소스코드에 추가

```
#pragma warning(disable:4996)
```



scanf() 함수 사용법

```
scanf("%d", &a);
```

변수 앞에 &를 반드시 붙이자

- 키보드로 입력된 정수 값을 변수 a에 저장
- %d: 입력되는 값을 정수로 해석하라는 의미의 서식 지정자
- printf 함수에서와 달리, 변수 a앞에 &기호를 붙여야 한다 (이유는 나중에..).
- 여러 개의 정수 입력 받기

```
scanf("%d%d", &a, &b);
```

[예제 1.2] 변수와 printf, scanf를 여러 번 사용

```
int main()
{
    int id;                // 변수 선언
    int age;

    printf("학번 입력: ");
    scanf("%d", &id);
    printf("입력 학번: %d\n", id);

    printf("나이 입력: ");
    scanf("%d", &age);
    printf("입력 나이: %d\n", age); // 입력된 값 출력

    return 0;
}
```

실행 화면

```
학번 입력: 1601111
입력 학번: 1601111
나이 입력: 20
입력 나이: 20
```

실습하기



[예제 1.3] 학번과 나이를 scanf 함수를 한번만 사용하여 입력 받아 다음과 같은 실행 결과가 나오도록 프로그램을 수정하시오.

실행 화면



```
학번과 나이 입력: 1601111 20  
입력 학번과 나이: 1601111 20
```

평가 하기

Q1

Q2

Q3

Q1

모든 C 프로그램에서 () 함수는 반드시 하나 존재해야 하고, C 언어의 문장은 ()으로 끝난다.

평가 하기

Q1

Q2

Q3

Q1

모든 C 프로그램에서 () 함수는 반드시 하나 존재해야 하고, C 언어의 문장은 ()으로 끝난다.

정 답**main, 세미콜론****해 설**

main() 함수는 프로그램의 시작 지점을 나타내는 함수로 모든 C 프로그램에는 main() 함수가 오직 하나 존재해야 한다. 세미콜론은 모든 문장의 끝에 사용되는데, 문장 끝에 세미콜론이 없으면 컴파일 오류가 발생한다.

평가 하기

Q1

Q2

Q3

Q2

프로그램에 존재하는 문법적 오류를
() 오류라고 하고, 논리에 맞지 않아 발생하는
오류를 () 오류라고 한다.

평가 하기

Q1

Q2

Q3

Q2

프로그램에 존재하는 문법적 오류를
() 오류라고 하고, 논리에 맞지 않아 발생하는
오류를 () 오류라고 한다.

정 답

컴파일, 런타임

해 설

컴파일 오류(compile error)는 문법에 맞지 않게 코드를
작성하여 발생하는 오류로, 컴파일러가 알려준다. 런타임
오류는 프로그램을 실행시킨 결과가 의도와 다르거나
프로그램이 비정상적으로 종료되는 경우로, 런타임 오류가
존재해도 정상적으로 컴파일되고 실행파일이 만들어 진다.

평가 하기

Q1

Q2

Q3

Q3

출력을 위한 printf 함수에 대한 다음 서술 중 올바른 것을 모두 고르시오.

- ① 출력 내용은 작은 따옴표 안에 작성한다.
- ② 따옴표 밖의 공백은 출력에 영향을 주지 않는다.
- ③ 여러 줄에 내용을 출력하려면 printf 함수를 여러 개 사용해야 한다.
- ④ 정수를 출력하기 위해서는 서식 지정자 %d를 사용한다.
- ⑤ 여러 개의 정수를 출력하려면 printf 함수를 여러 개 사용해야 한다.

평가하기

Q1

Q2

Q3

Q3

출력을 위한 printf 함수에 대한 다음 서술 중 올바른 것을 모두 고르시오.

- ① 출력 내용은 작은 따옴표 안에 작성한다.
- ② 따옴표 밖의 공백은 출력에 영향을 주지 않는다.
- ③ 여러 줄에 내용을 출력하려면 printf 함수를 여러 개 사용해야 한다.
- ④ 정수를 출력하기 위해서는 서식 지정자 %d를 사용한다.
- ⑤ 여러 개의 정수를 출력하려면 printf 함수를 여러 개 사용해야 한다.

정답

②, ④

해설

① 출력 내용은 "큰" 따옴표 안에 작성한다. ③ 여러 줄에 내용을 출력하기 위해서는 개행 문자(\n)을 사용한다. ⑤ 여러 개의 정수는 서식 지정자 %d를 여러 개 사용하여 출력할 수 있다.

학습 정리

- **들여쓰기**와 **주석**은 코드의 가독성을 높이기 위한 수단으로 프로그램의 의미에 아무 영향을 미치지 못하지만, 프로그래밍에서 필수적인 요소임
- **C 언어 코드**는 기본적으로 위에서부터 순차적으로 수행됨
- **printf()**는 값을 화면에 출력하고, **scanf()**는 사용자로부터 값을 입력 받는 함수로, 서식 지정자라 부르는 기호를 사용함
- **개행 문자 \n**은 줄 바꿈을 의미하는 특수 문자임