

Do it! C언어 입문

20164389 멀티미디어공학과 박병주

김상엽 지음
이지스 퍼블리싱



목차

1장 : 프로그램과 C언어

2장 : C언어로 만드는 첫 번째 프로그램

1장 : 프로그램과 C언어

프로그램과 C언어

프로그래밍 기초

프로그램



☐ 어떤 목적을 위해 진행 계획을 순서대로 표현한 것임

☐ 컴퓨터가 작업할 내용을 컴퓨터가 이해할 수 있는 언어(기계어)로 만들어진 파일임

프로그래밍 언어



☐ 컴퓨터는 2진 숫자(0과 1)로 이루어진 기계어를 사용함

☐ 사람에게 친숙한 표현법을 사용하여 프로그래밍 할 수 있도록 만든 언어임

컴파일러



☐ 프로그래밍 언어로 작성한 내용을 기계어로 번역함

☐ 프로그래밍 언어는 컴파일러와 약속된 표현을 사용함



프로그래밍과 C언어

C언어의 소개

탄생

1972년 켄 톰슨과 데니스 리치에 의해 개발됨

운영체제 유닉스(UNIX) 개발에 사용함

특징

구조화된 언어

이식성이 높음

확장성이 좋고 표현법이 다양함

생산성이 높음

프로그램과 C언어

프로그램의 원리와 개념

입력문 : 사용자가 원하는 정보 입력함

출력문 : 사용자에게 정보를 표시함

상수 : 입력 대상을 한정함

연산자 : 연산 기능을 수행함

변수 : 입력한 값과 연산 결과를 기억하는 공간임

조건문 : 조건에 따른 처리가 가능하도록 제공되는 문법임

반복문 : 원하는 작업을 반복해서 수행하도록 제공하는 문법임

프로그래밍과 C언어

C언어의 문장 서술 형식

단일 서술문

논리적 최소 명령 실행 단위이며 ;(세미콜론) 으로 구분함

```
a = a + 30;
```

한 줄로 이루어진 단일 서술문

```
a = a
```

```
    + 30;
```

두 줄로 이루어진 단일 서술문

```
a = a + 30; b = b + 50;
```

한 줄에 두 개의 단일 서술문을 사용

복합 서술문

단일 서술문 여러 개를 하나로 묶은 형태

{ 중괄호를 사용하여 표현함

{ 중괄호 뒤에는 ;를 사용할 필요가 없음

```
{  
  
    a = a + 30;  
    b = b + 50;  
  
};
```

프로그래밍과 C언어

C언어의 문장 서술 기본 형식

주석문

- 소스 파일에 메모를 남기기 위해 사용함
- `/* */` 기호를 사용하며 `/*` 기호로 시작하여 `*/`로 끝남
- 컴파일러가 번역할 때 주석문은 번역하지 않음

```
/*a, b의 평균값을 계산하여 average에 저장한다*/ [아래 코드를 설명하는 주석문]  
Average = (a + b) / 2;
```

주석문 사용시 주의 사항

주석문을 중첩하여 사용하면 안된다.

```
/* 주석문은 /* 여러 줄에 걸쳐서도 */ 사용할 수 있습니다. */  
시작      무시      끝      오류발생
```


프로그래밍 언어

C언어의 문장 서술 형식

예약어

- 프로그래밍 언어의 문법에서 사용되는 단어임
- 이름 짓는 문법에서는 예약어 사용이 불가능함

main

while

char

int

for

do
while

double

auto

break

case

default

float

return

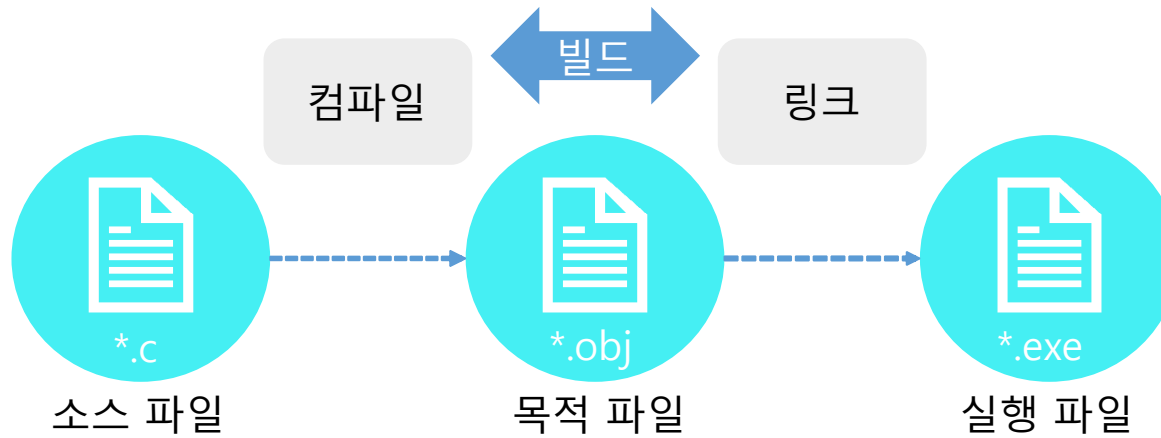
extern

goto

if

프로그램과 C언어

C프로그램 실행 파일



소스 파일 (*.c) : 프로그램을 C언어 문법을 사용해서 작성한 파일임

목적 파일 (*.obj) : 소스 파일을 컴파일러가 컴파일하면 생성되는 파일임

실행 파일 (*.exe) : 컴퓨터에서 실행할 수 있는 파일

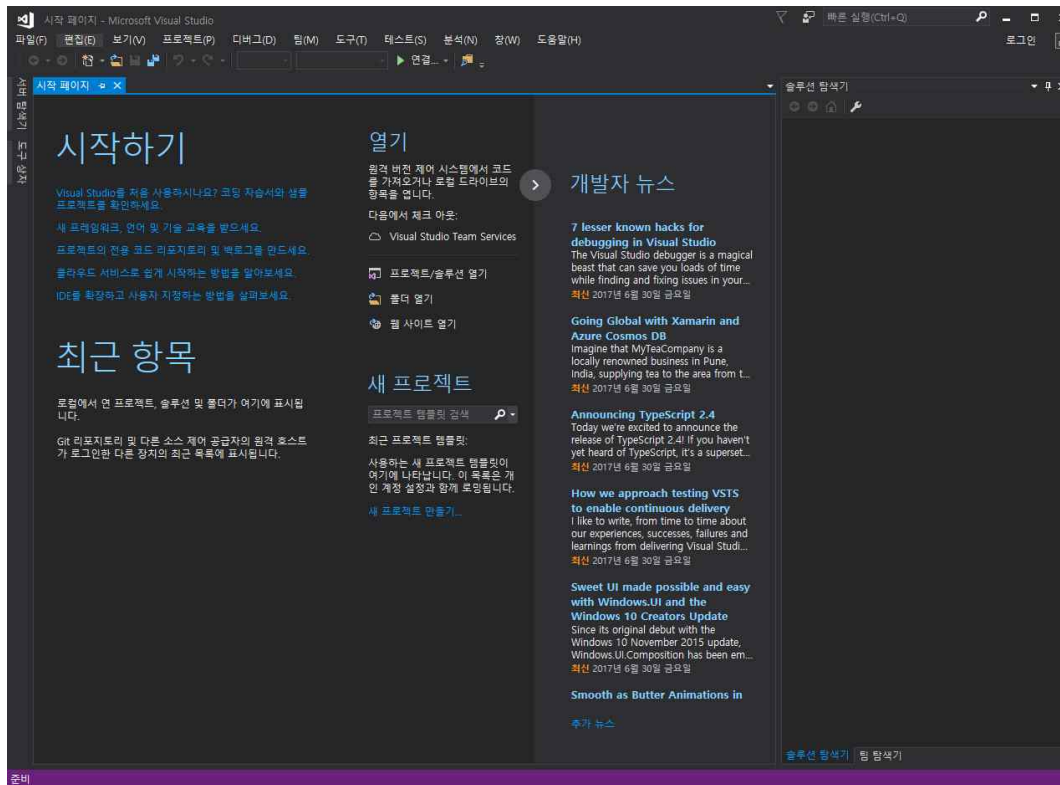
하나 이상의 목적 파일을 링커가 링크하여 만듦

2장 : C언어로 만드는 첫 번째 프로그램

C언어로 만드는 첫 번째 프로그램

C언어 개발 환경 구축하기

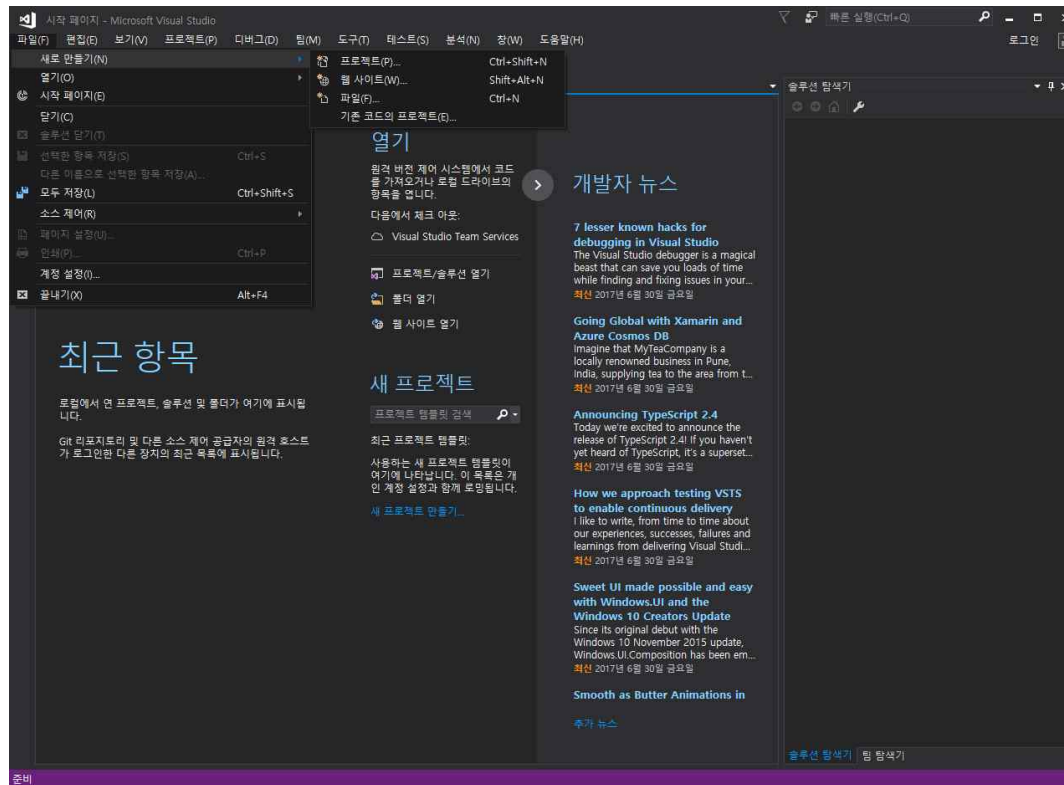
1. Visual Studio 2017 실행



C언어로 만드는 첫 번째 프로그램

C언어 개발 환경 구축하기

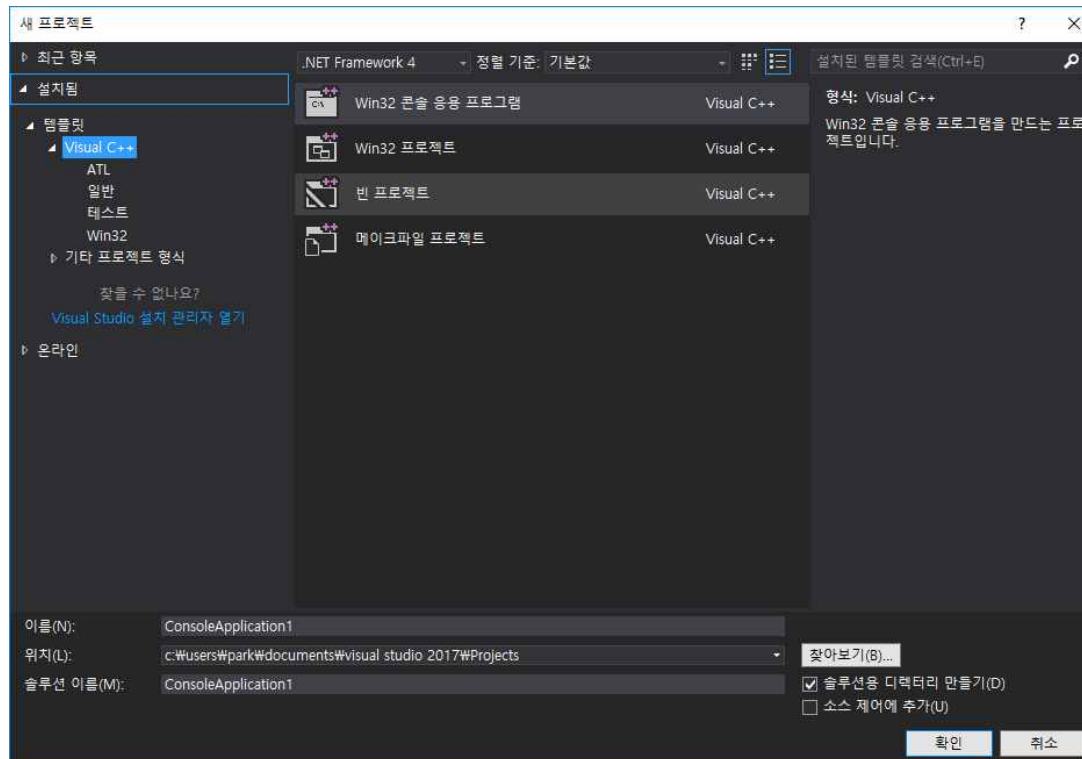
2. 파일 - 새로 만들기 - 프로젝트



C언어로 만드는 첫 번째 프로그램

C언어 개발 환경 구축하기

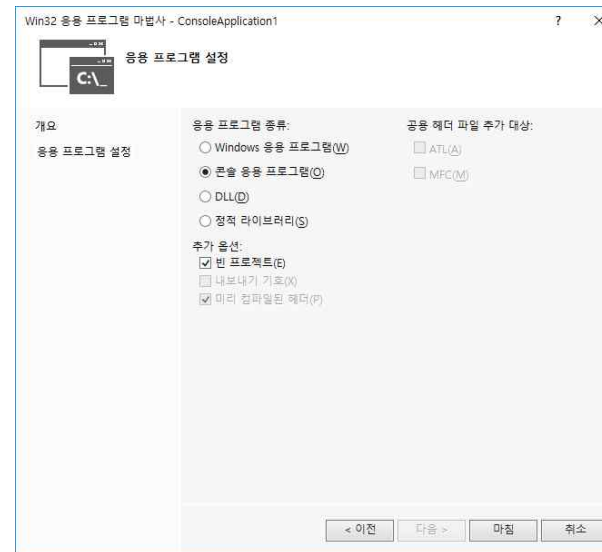
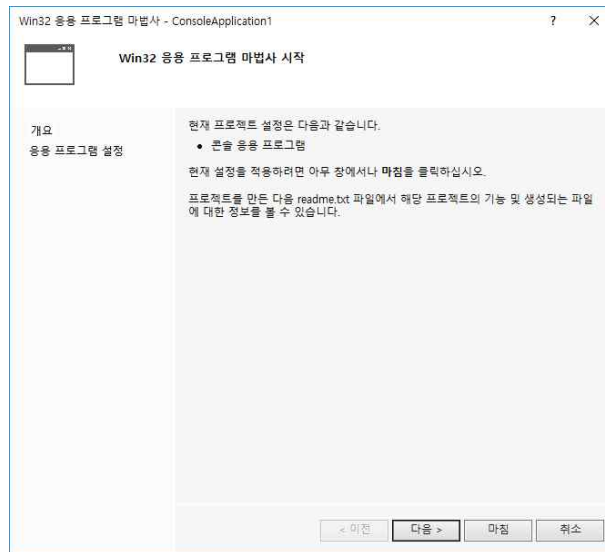
3. Win32 콘솔 응용 프로그램 선택 후 프로젝트 이름 설정 후 확인



C언어로 만드는 첫 번째 프로그램

C언어 개발 환경 구축하기

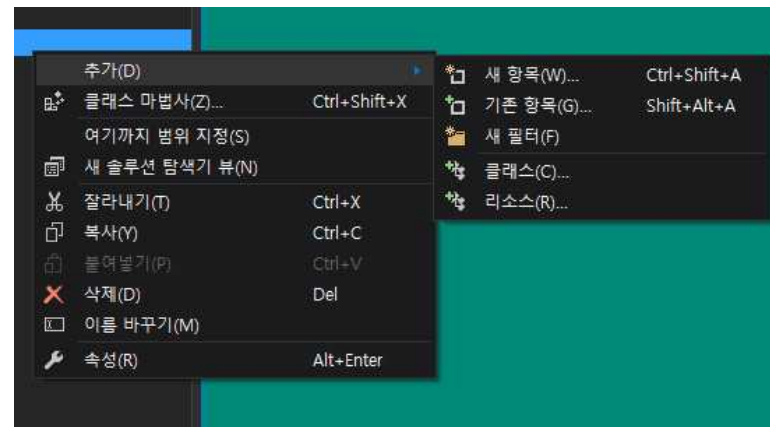
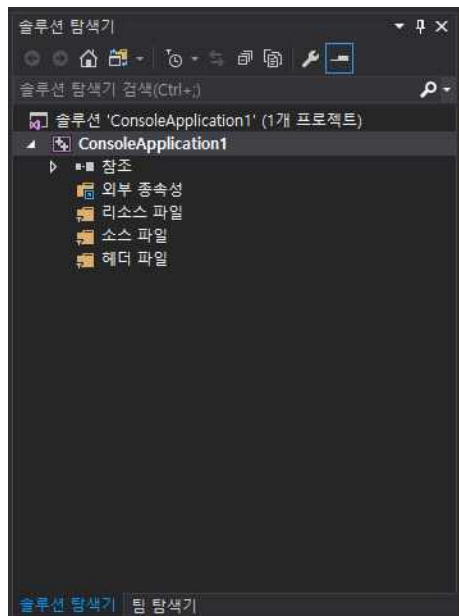
4. 콘솔 응용 프로그램, 빈 프로젝트 체크 후 마침



C언어로 만드는 첫 번째 프로그램

C언어 개발 환경 구축하기

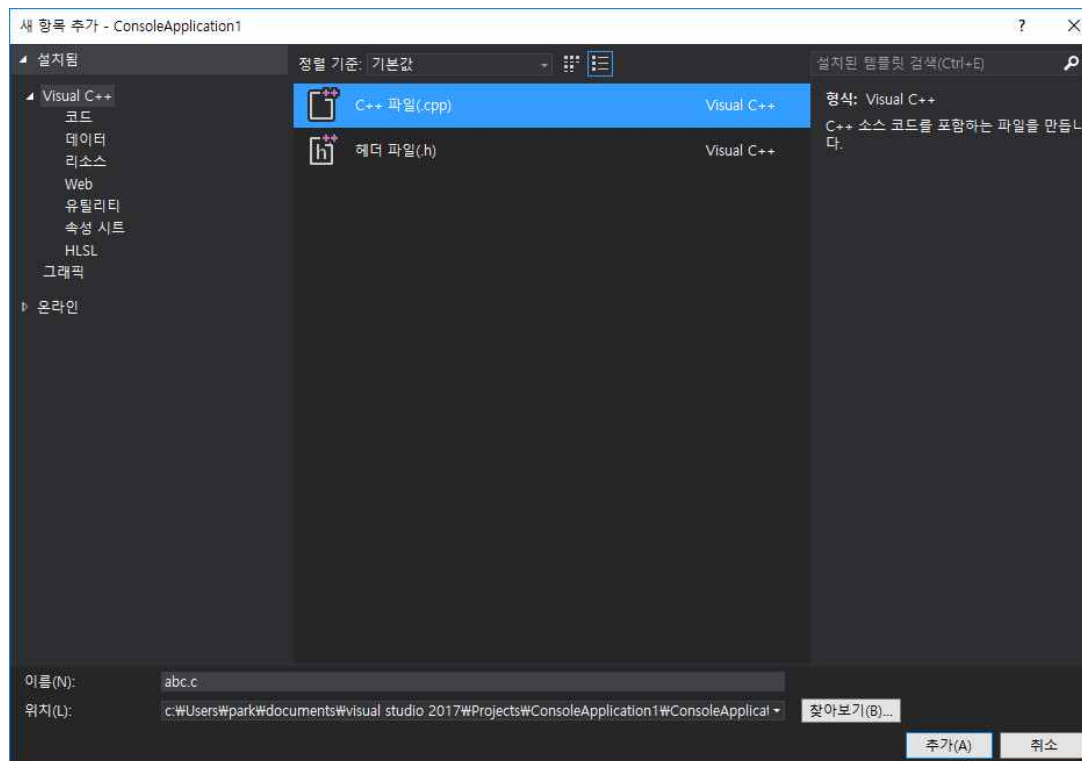
5. 소스파일 오른쪽 클릭 후 추가 - 새 항목



C언어로 만드는 첫 번째 프로그램

C언어 개발 환경 구축하기

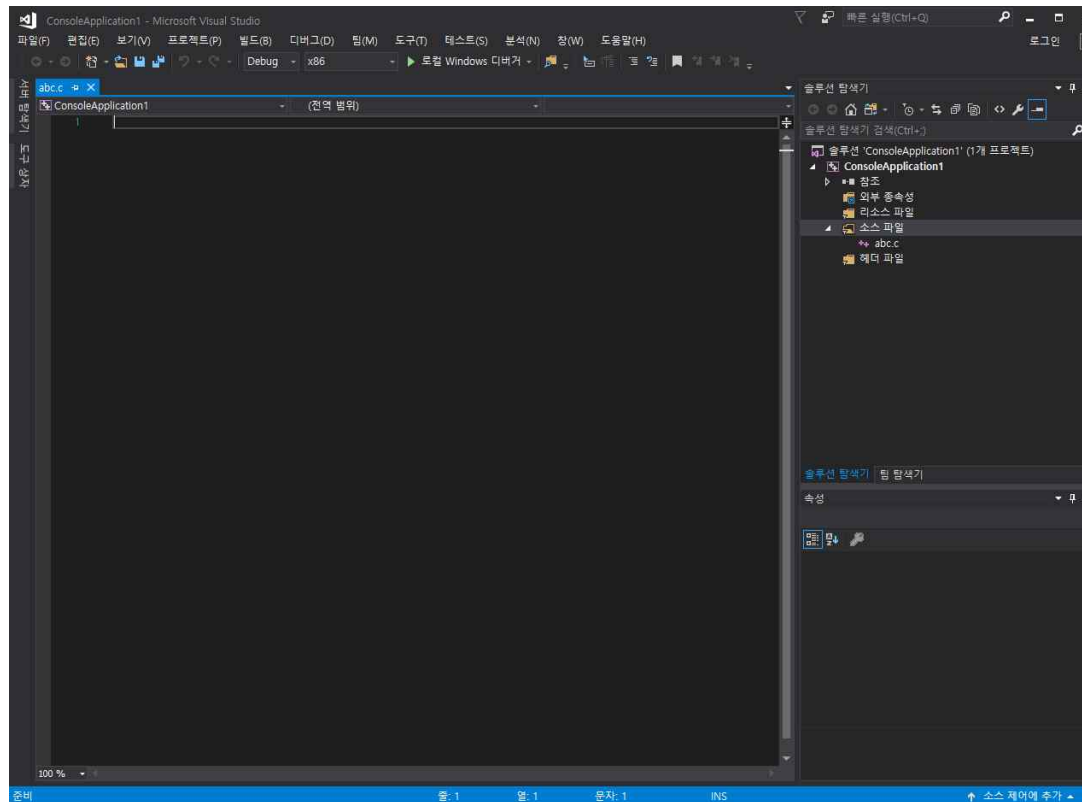
6. C++파일 선택 후 파일 이름 설정 (확장자 명을 *.c로 기입함)



C언어로 만드는 첫 번째 프로그램

C언어 개발 환경 구축하기

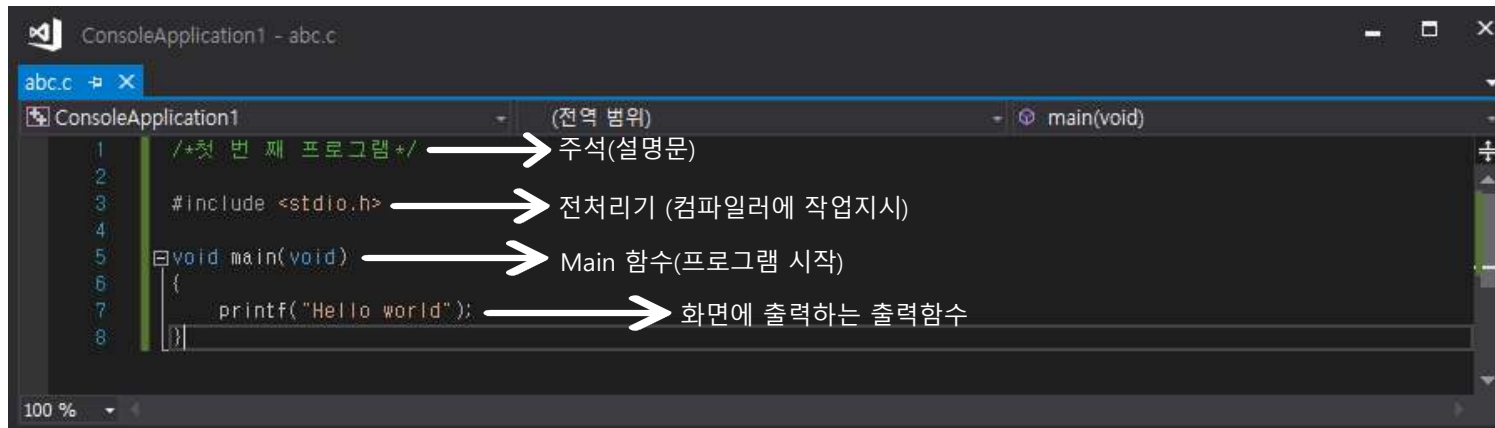
7. 소스코드 입력



C언어로 만드는 첫 번째 프로그램

내가 만든 첫 번째 프로그램

1. 예제 코드 작성하기



```
ConsoleApplication1 - abc.c
abc.c
ConsoleApplication1 (전역 범위) main(void)
1  /*첫 번째 프로그램*/
2
3  #include <stdio.h>
4
5  void main(void)
6  {
7      printf("Hello world");
8  }
```

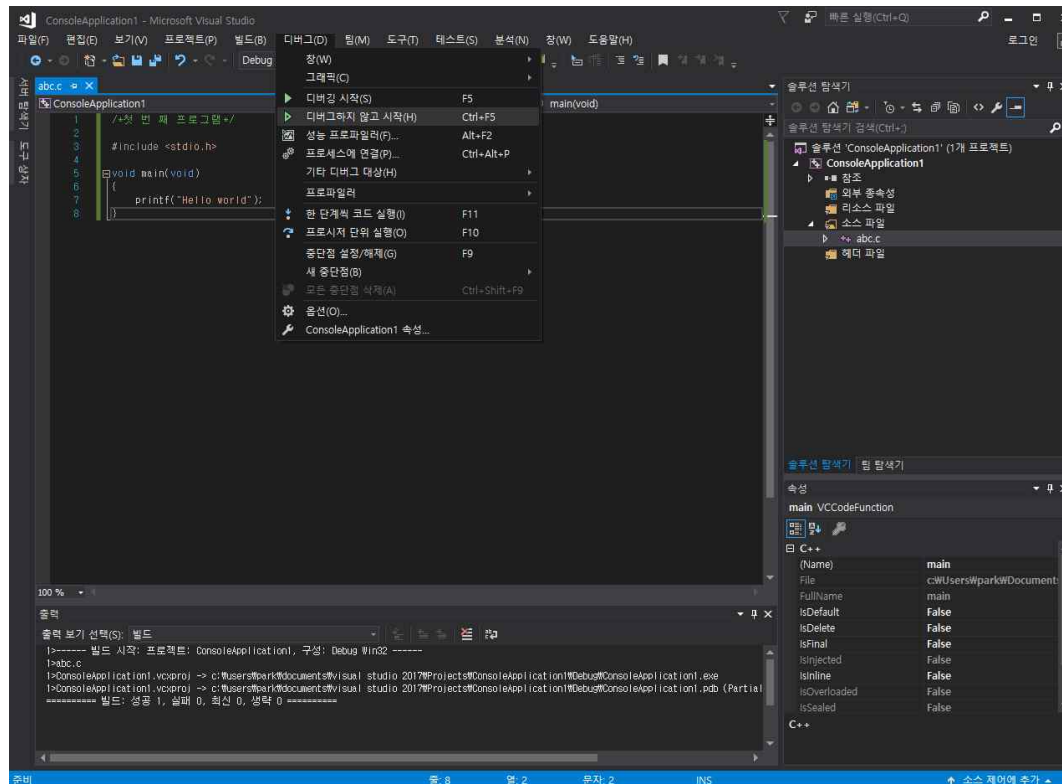
Annotations:

- /*첫 번째 프로그램*/ → 주석(설명문)
- #include <stdio.h> → 전처리기 (컴파일러에 작업지시)
- void main(void) → Main 함수(프로그램 시작)
- printf("Hello world"); → 화면에 출력하는 출력함수

C언어로 만드는 첫 번째 프로그램

내가 만든 첫 번째 프로그램

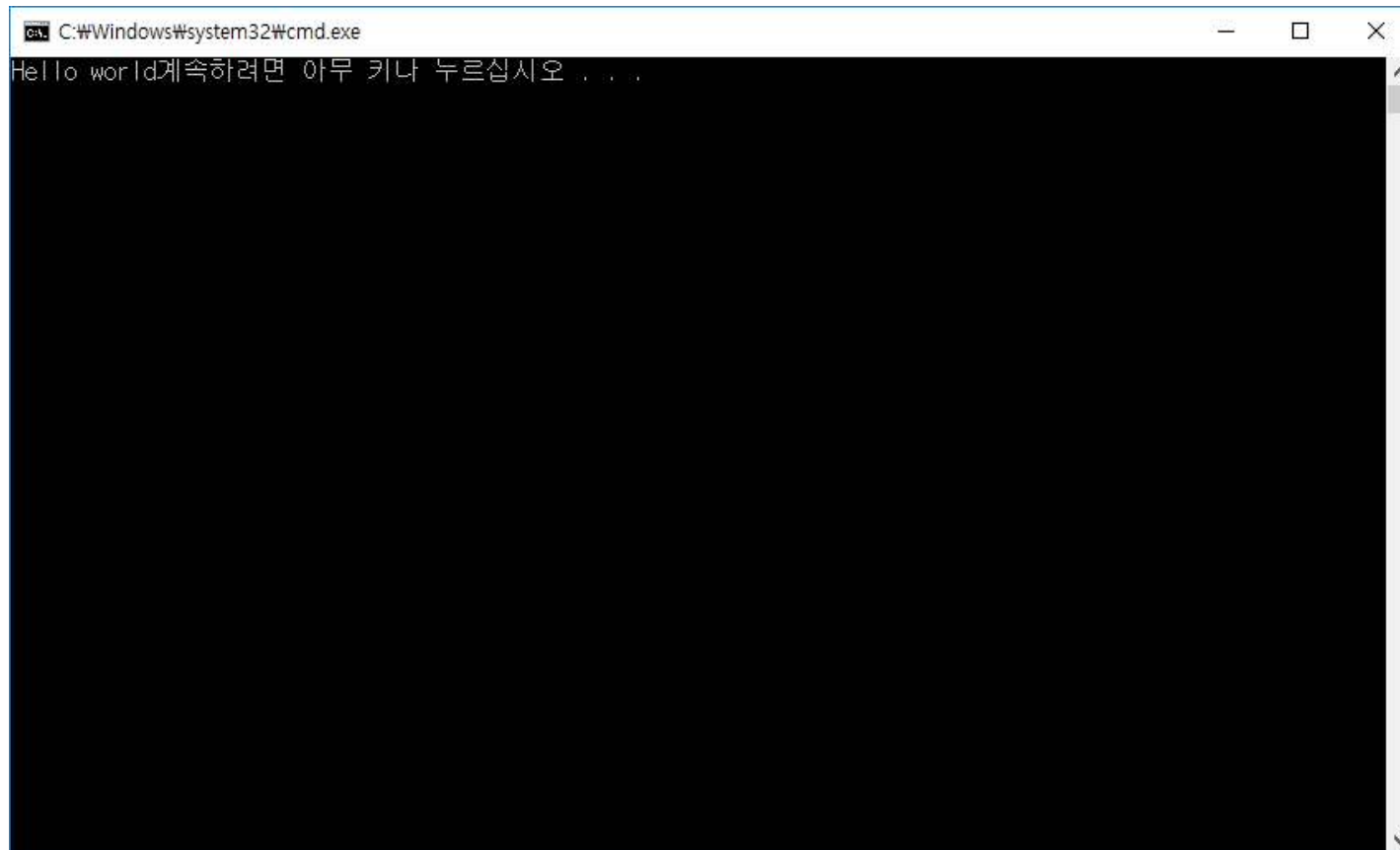
2. 디버그 - 디버그하지 않고 시작 (Ctrl + F5)



C언어로 만드는 첫 번째 프로그램

내가 만든 첫 번째 프로그램

3. 실행 화면



A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Windows\system32\cmd.exe". The command prompt shows the output of a C program: "Hello world" followed by Korean text "계속하려면 아무 키나 누르십시오 . . .". The rest of the window is black, indicating that the program has finished execution and is waiting for a key press to close.

C언어로 만드는 첫 번째 프로그램

#include <stdio.h>

- C언어에서 #으로 시작하는 문장은 전처리기임
- 컴파일러에게 소스 코드를 번역 전에 명시한 파일을 먼저 참조하라고 지시함
- printf 함수에 대한 정보가 stdio.h 파일에 기록되어 있음

main

함수

- 명령들을 일정한 작업 단위로 묶어서 코드를 작성하기 위해 제공함
- C 언어 프로그램은 사용자가 만든 함수의 집합체로 구성함

main 함수

- 프로그램 시작을 나타내므로 반드시 정의해야 함
- main 함수가 없거나, 2개 이상 선언하면 링크할 때 오류 발생
- 함수는 서술문의 그룹으로 구성되므로 { } (중괄호)를 사용하여 영역 표시

감사합니다.

