

초보자를 위한 C언어 강의

씹어먹는 C언어

이재범

v.2020.06.20

Copyright © 2019-2020 이재범

이 책은 모두의 코드에 연재된 씹어먹는 C 언어 강좌를 책으로 옮긴 것입니다. 해당 강좌는 <https://modoocode.com> 에서 볼 수 있습니다.

차례

차례	3
제 1 장 C 언어의 세계로	1
C 언어 본격 맛보기	14
주석(Comment)	19
컴퓨터에서 수를 표현하는 방법	22
제 2 장 변수 (variable)	30
제 3 장 계산 하기	40
컴퓨터가 음수를 표현하는 방법 (2의 보수)	55
제 4 장 문자 입력 받기	61
제 5 장 만약에... (조건문)	69
제 6 장 반복문 (for, while)	89
리눅스에서 C 프로그래밍 하기	108
제 7 장 switch 문	109
제 8 장 형 변환 (타입 캐스팅)	118
제 9 장 C 언어의 아파트 - 배열	128
다차원 배열	148
제 10 장 포인터	159
상수 포인터, 포인터의 덧셈 뺄셈, 배열과 포인터	171
포인터의 포인터	190

제 11 장 함수 (function)	210
포인터로 받는 인자	224
여러가지 인자들	241
생각해볼 문제에 대한 아이디어	253
제 12 장 디버깅(Debugging)	261
제 13 장 문자열 (string)	269
버퍼에 대한 이해	280
문자열 리터럴에 대한 이해	294
도서 관리 프로젝트	309
제 14 장 구조체 (struct)	334
구조체를 인자로 받는 함수	347
공용체(union)와 열거형(enum)	361
제 15 장 변수의 생존 조건과 데이터 세그먼트의 구조	375
제 16 장 여러 파일로 나누기	388
여러 전처리기 구문과 라이브러리 사용법	409
제 17 장 void 타입과 main 함수에 대한 이해	420
제 18 장 동적 메모리 할당 (dynamic memory allocation)	434
구조체의 동적 할당과 메모리 관리 함수	447
제 19 장 매크로 함수, 인라인 함수	461
제 20 장 그 밖에 키워드들	472
제 21 장 파일 입출력	487
파일 안에서 이동하기	499
도서 관리 프로그램의 완성	525
제 22 장 C 코드 최적화	536

C 언어의 세계로

안녕하세요 여러분! 씹어먹는 C 언어 강좌에 오신 것을 환영합니다.

왜 C 언어를 배워야 하는가?

아마도 이 강좌를 보고 있는 여러분들은 분명 누군가에 의해 “너는 C 언어를 배워야만 해!” 해서 들어 오셨을 것입니다. 하지만 많은 경우 도대체 왜 굳이 C 언어를 배워야 하는지 알려주지는 않았을 것입니다. 저 또한 C 언어를 처음 공부한 이유가 누가 배워야 한다고 해서 배운 것이지, 제가 필요성을 느껴서 배운 것도 아닙니다.

여러분이 프로그래밍을 시작한 계기는 분명 여러가지가 있을 것입니다. 저는 처음 프로그래밍을 배운 계기가 게임을 만들고 싶어서였습니다. 어떤 분들은 멋진 웹사이트를 만들고 싶었을 것이고, 또 어떤 분들은 안드로이드 앱을 만들고 싶어서 었을 것입니다. 그런데 이러한 것들을 하기 위해 굳이 C 언어를 알아야 할까요?

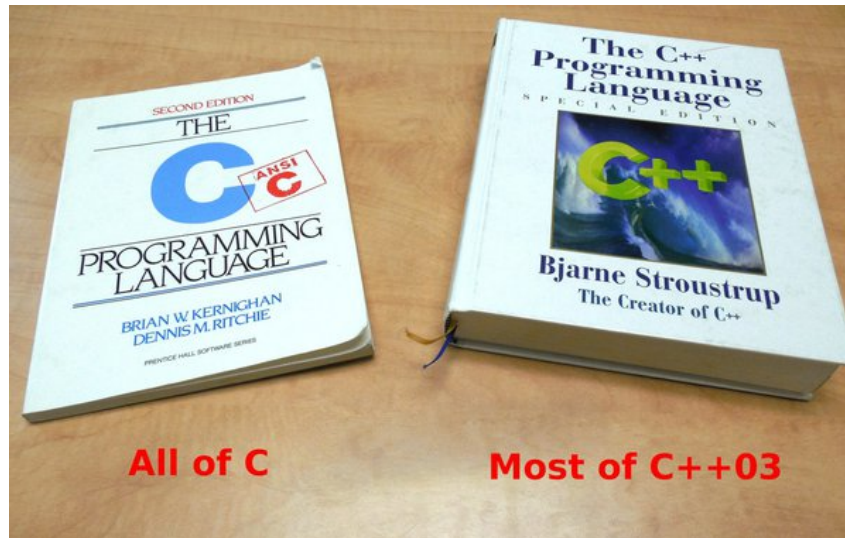
사실 아닙니다. 요즘에 게임을 만드는 가장 핫한 플랫폼인 유니티는 보통 C# 으로 프로그래밍 하고, 웹사이트의 경우 대개 파이썬, PHP, 자바스크립트 등등으로 만듭니다 (여기서 말하는 웹사이트는 백엔드). 또 안드로이드 앱은 자바 (혹은 코틀린), iOS 앱은 스위프트나 Objective-C 로 만듭니다. 여기 어디에도 C 는 쓰지 않습니다. 아마 이러한 것들을 지금 빨리 만들고 싶어서 손이 근질근질 거리시는 분들은 이 부분에서 뒤로가기를 누르셔도 좋습니다.

그렇다면 왜 사람들은 C 언어를 배우라고 할까요? 도대체 왜 학교에선 다른 언어들 놔두고 C 언어를 가르칠까요? 제가 생각하는 이유는 다음과 같습니다.

- 만약 적당히 잘하는 프로그래머가 목표라면, 굳이 컴퓨터 내부가 어떻게 돌아가는지 몰라도 괜찮습니다. 하지만 좋은 프로그래머가 되려면, 컴퓨터의 내부 원리를 아는 것이 필수적입니다. 만약에 C 언어를 배우게 된다면, 컴퓨터 내부 원리를 더 쉽게 이해할 수 있습니다.
- C 언어를 배운다면, 다른 언어를 더 쉽게 습득할 수 있습니다. 많은 언어들이 C 언어에서

파생되어서 생겨났습니다 (위에서 나열한 C#, Objective-C, 그리고 가장 유명한 C++ 까지 모두 이름에서 알 수 있듯이 C 의 영향을 많이 받은 언어들 입니다.) 따라서 이런 언어들을 배우는데 많은 도움이 됩니다.

- 이미 엄청나게 많은 코드들이 C 언어로 작성되어 있습니다. 따라서, 어느 정도 수준 이상에 도달하게 된다면 C 언어를 결코 피하실 수 없을 것입니다.



참고로 제일 최근에 나온 C++ 은 C++ 20 으로 그 내용을 다 포함한다면 저 책의 2 배는 되야 할 것입니다.

- 마지막으로, C 언어는 배워야 할 내용이 다른 언어에 비해 매우 적습니다! (C 언어의 최고의 교과서라 불리는 *The C Programming Language* 라는 책은 200여 페이지 밖에 되지 않습니다. 반면에 C++ 에서 비슷한 역할을 하는 책인 *The C++ Programming Language* 는 1300페이지가 넘습니다.) 물론, 배워야 할 내용이 적다 와 쉽다 는 차이가 있으니 유의해두시기 바랍니다 :)

이러한 연유에서 저는 여러분들이 C 언어를 배워야 하겠다라고 마음 먹기 매우 잘한 일이라 생각합니다. 특히 C 언어를 배우는데 있어서 제 강좌인 씹어먹은 C 언어를 선택한 것 역시 훌륭한 선택이라 자부합니다.

어떤 식으로 배워야 할까요?

흔히 언어를 가장 쉽게 배우는 방법은 그 말을 하는 친구를 사귀는 것이라 하였습니다. (여자/남자 친구면 더 좋아요!) 이런 식으로 배우는 것이 딱딱한 문법책 외우면서 단어만 죽어라 외우는 것 보다 더 재밌고 실력도 금방 금방 늡니다.

C 언어도 마찬가지입니다. 더군다나 우리는 C 언어를 매우 잘하는 친구를 이미 사귀고 있습니다. 바로 여러분들이 지금 사용하고 있는 컴퓨터가 있잖아요. C 언어를 가장 빨리 습득하는 방법은 강좌에서 배운 내용을 토대로 간단한 프로그램을 직접 만들어 보는 것입니다. 여기에 재미를 붙인다면 더 빠르게 배울 수 있겠지요!

저는 누구인가요?

저는 C 언어를 전문적으로 가르치는 강사는 아닙니다. 전문적인 강의를 원했던 사람은 여기서 다시 살포시 뒤로가기를 눌러도 됩니다. (아마 80% 이상이 누를 것이라 추정) 하지만 제가 유일하게 잘 할 수 있다고 자부하는 것은 어려워만 보이던 C 언어의 기초 부분을 최대한 쉽게 설명하는 것입니다. [\(저에 대한 자세한 프로파일은 여기를 참조해주세요\)](#)

앞으로 저와 함께 C 언어의 세계로 여정을 떠날 사람들은 준비가 되었습니까? 그럼 모두 스크롤을 내려주세요!

여정에 필요한 준비물

C 언어를 배우기 위해선 다음과 같은 준비물이 필요합니다.

1. 컴퓨터
2. 머리
3. 노오력
4. 컴파일러.

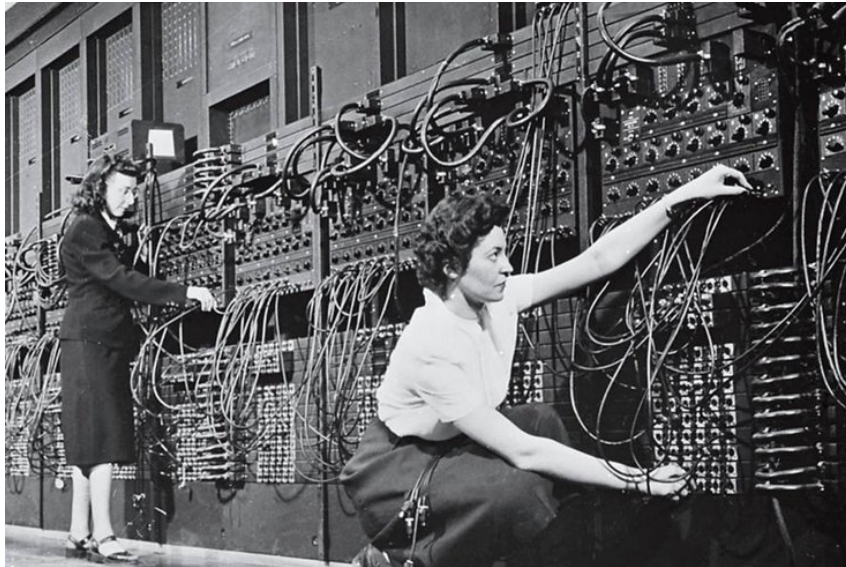
1,2,3 번은 여러분들께서 충분히 준비할 수 있다고 생각합니다.

그런데 4번, 컴파일러는 뭐지?

컴파일러?

컴파일러, 영어로는 *Compiler* 라고 씁니다. 말그대로 컴파일(Compile) 해주는 것(-r) 입니다. 그렇다면 컴파일은 도대체 무엇일까요?

아마 여러분은 컴퓨터는 0 과 1 밖에 모르는 바보(?) 라고 들었을 것입니다. 맞습니다. 컴퓨터의 두뇌라고 할 수 있는 CPU 에서는 수 많은 0 과 1 들이 이리저리 돌아다니고 있을 것입니다. 만약에 이런 컴퓨터에 명령을 내리려면 오직 0 과 1 로만 써야겠지요? 010101010111 이런식으로 말입니다.



C 언어는 노오력으로 배울 수 있지만, 이건 아마 노오오오오오력이 필요했을 것입니다!

아마 옛날에는 실제로 전선을 이리저리 연결해가며 비슷한 방식으로 명령을 내렸을 것입니다.

만약 우리가 프로그래밍을 0 과 1 로만 한다면 얼마나 끔찍할지 생각해 보세요. 만약 이런 세상이었다면 프로그래머 연봉이 평균 10억은 되었을 것입니다. 하지만 훌륭한 컴퓨터 과학자들 덕분에, 0 과 1 대신에, 그나마 사람이 알아들을 수 있는 '언어'로 프로그래밍을 할 수 있게 하였습니다. 0 과 1 로 표현되는 명령을, 사람들이 그나마 알아듣기 쉽게 표현하게 말입니다. 예를 들어서 $1 + 1$ 을 하기 위한 명령을 컴퓨터에 내린다면

```
01011101010101
```

이 되겠지만 컴퓨터 언어를 통해 간단히

```
1 + 1
```

이렇게 표현할 수 있을 것입니다. 문제는 컴퓨터가 인간의 언어를 도무지 알아 들을 수 없으니, 컴퓨터가 이해할 수 있는 0..1 들로 바꿔주는 녀석이 필요합니다. 이렇게, 사람들이 사용하는 '프로그래밍 언어'와 컴퓨터가 이해하는 '기계어' 사이 다리 역할을 수행하는 것이 바로 **컴파일러** 입니다.

좋은 소식은 쉽게 컴파일러들을 구할 수 있다는 점입니다. 유명한 컴파일러로는 무료로 배포되는 **gcc** 와 **clang** 를 들 수 있지만 사용법이 초보자들에게는 약간 복잡합니다. 대신에, 저희는 마이크로소프트에서 무료로 배포하는 **Visual C++ 2017 Community** 를 사용할 것입니다.

더 좋은 소식은 Visual Studio Community 2017 을 다운받으면 아예 위 컴파일러 뿐만이 아니라 사용자들이 프로그래밍 하기 편하게 여러가지 프로그램들이 딸려 옵니다. 땡큐죠!

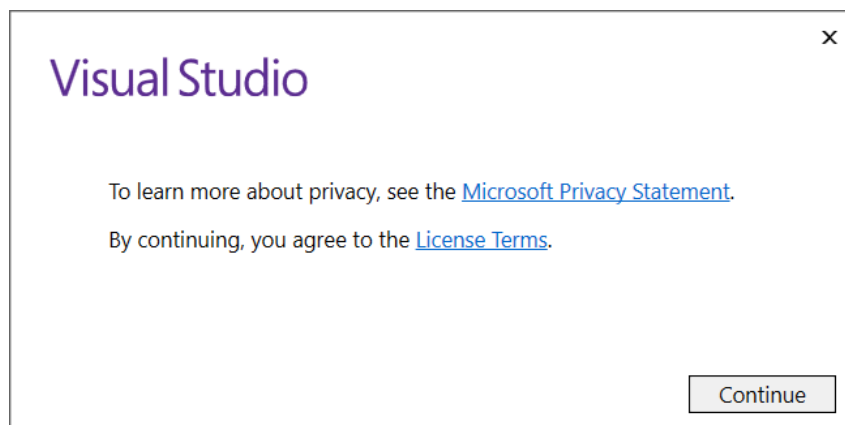
비주얼 스튜디오 커뮤니티 2017 은

<https://www.visualstudio.com/vs/community/>

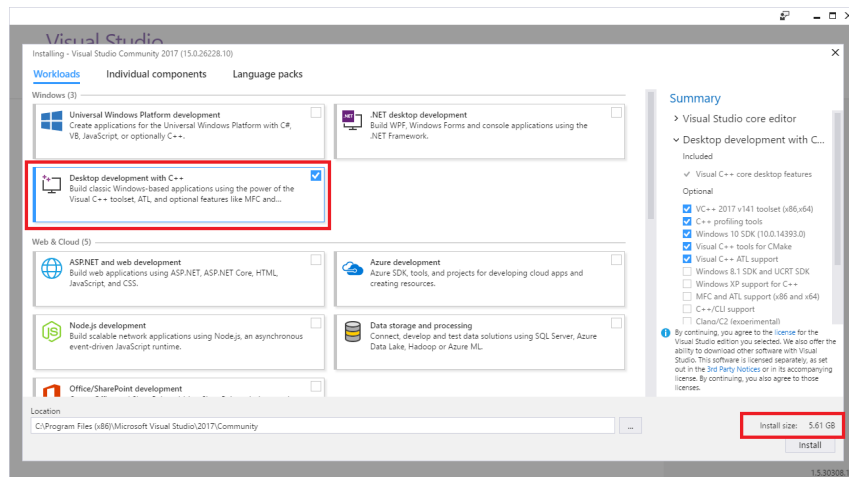
에서 받으실 수 있습니다.



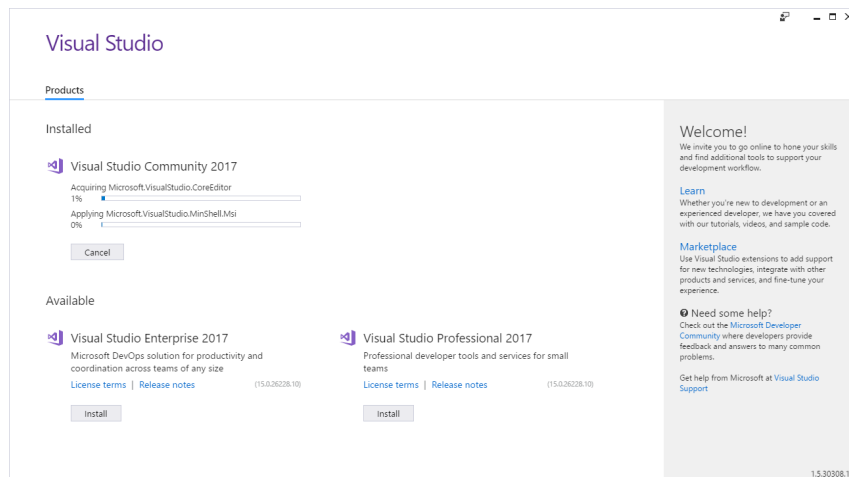
위와 같이 사이트에 들어가서 *Download VS Community 2017* 을 누르시면 됩니다. (제가 지금 미국에서 있어서 영어로 뜨지만 아마 한국에 계신 분들은 한국말로 뜰 것입니다 :)



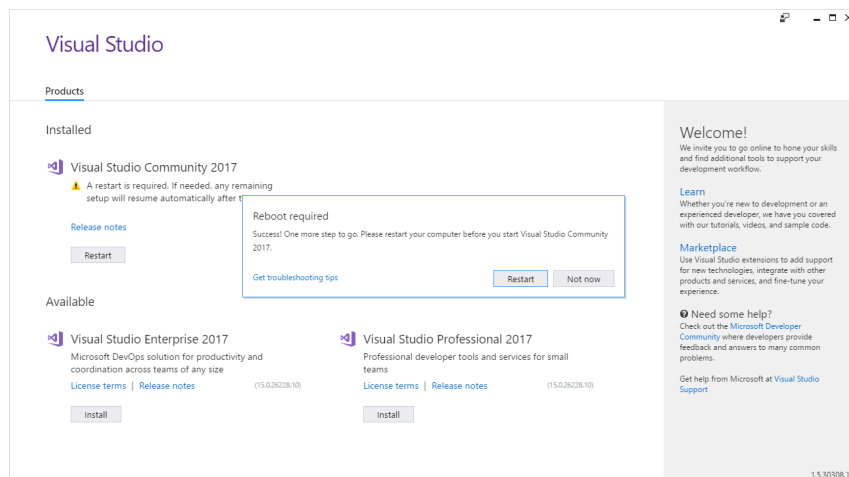
다운로드 된 실행파일을 실행하면 위와 같이 뜹니다. *Continue* 를 눌러주세요



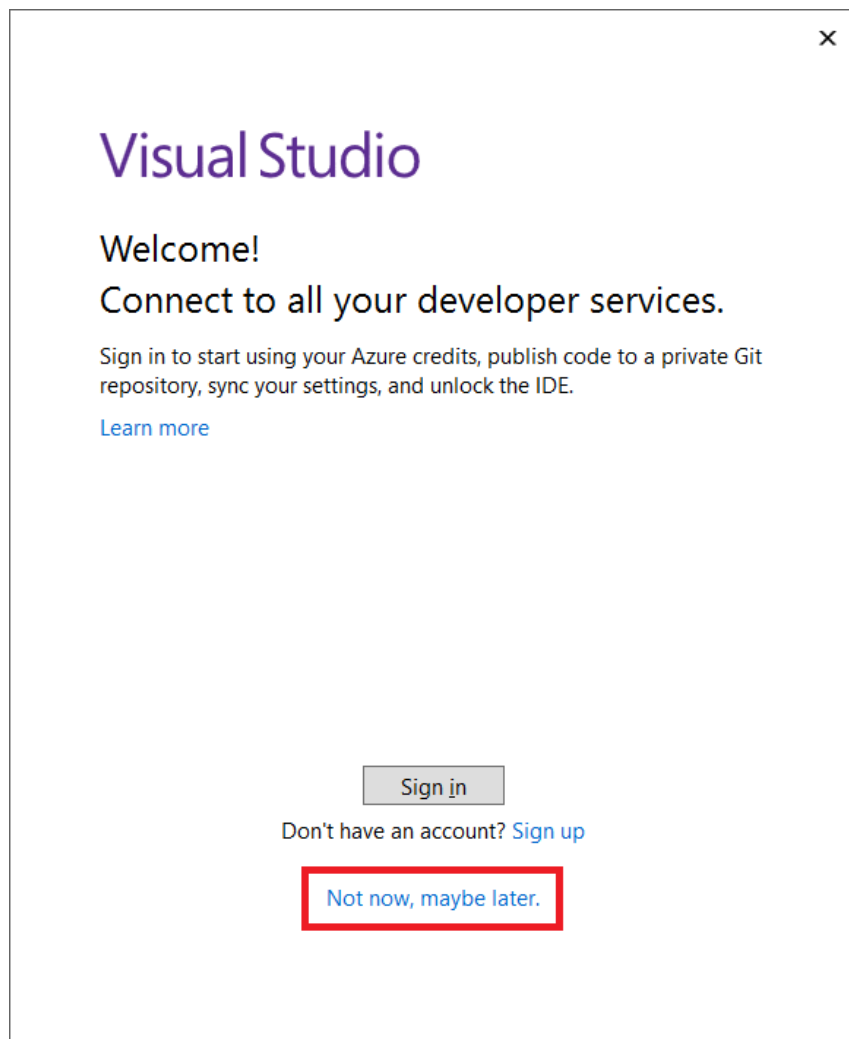
이제 어떠한 것들을 설치할 지 선택할 수 있는데, 일단 여러분은 C/C++ 을 배우는 것이 목적이므로 화면에 **Desktop development with C++** 만 선택해 주시면 됩니다. 혹시 내가 나중에 C# 이나 기타 등등으로 더 많은 코딩을 하고 싶다 그러시는 분들은 다른 것들을 선택해서 설치해주셔도 상관 없지만, 설치 용량이 엄청 커지게 됩니다. 저거 하나만 선택했는데에도 벌써 5.61 GB 네요.



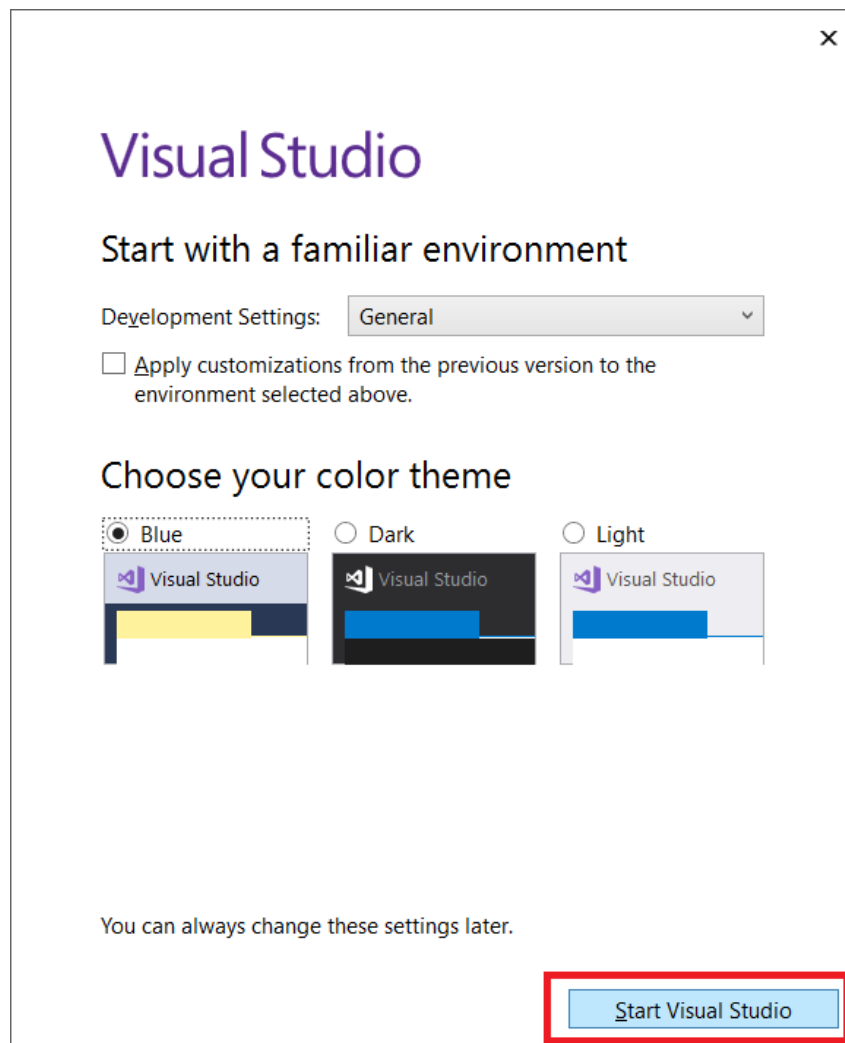
그럼 이제 알아서 필요한 것들을 다운 받아서 설치하게 됩니다. 인터넷 상황에서 따라서 30분에서 1시간 정도 기다려야 합니다.



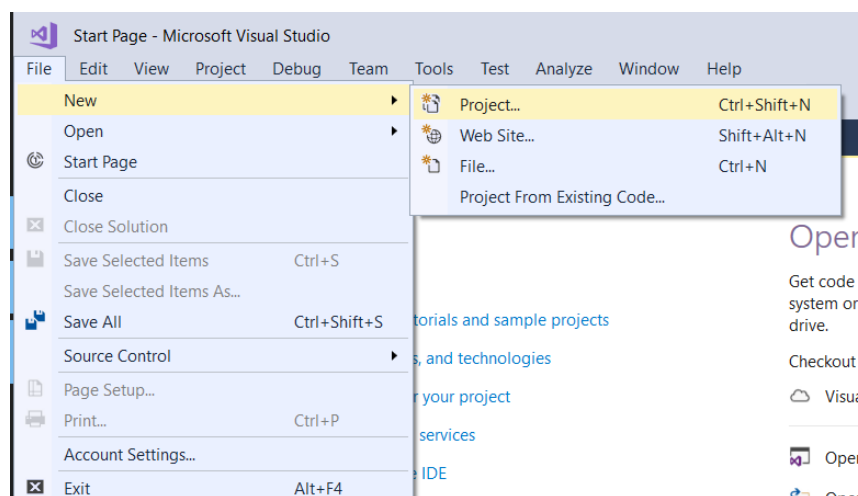
설치가 다 되었으면 컴퓨터를 재시작 해야 합니다.



뭔가 가입하라고 나오는데 그냥 나중에 한다고 하고 무시하면 됩니다.

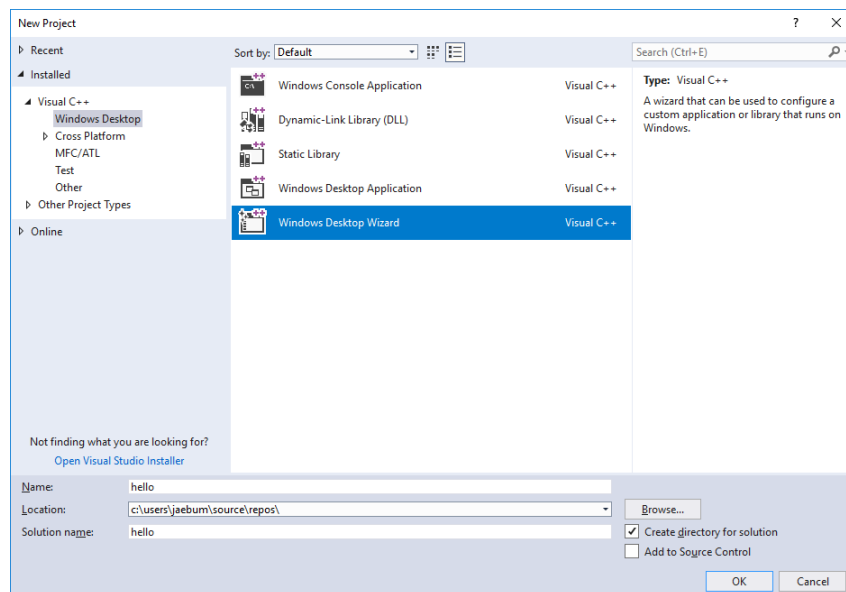


그 다음으로 어떤 테마를 고를지 정하면 됩니다. 저의 경우 그냥 디폴트인 파란색 바탕을 사용하는데, 사람들에 따라서 어두운 테마를 좋아하는 경우도 있습니다 :)

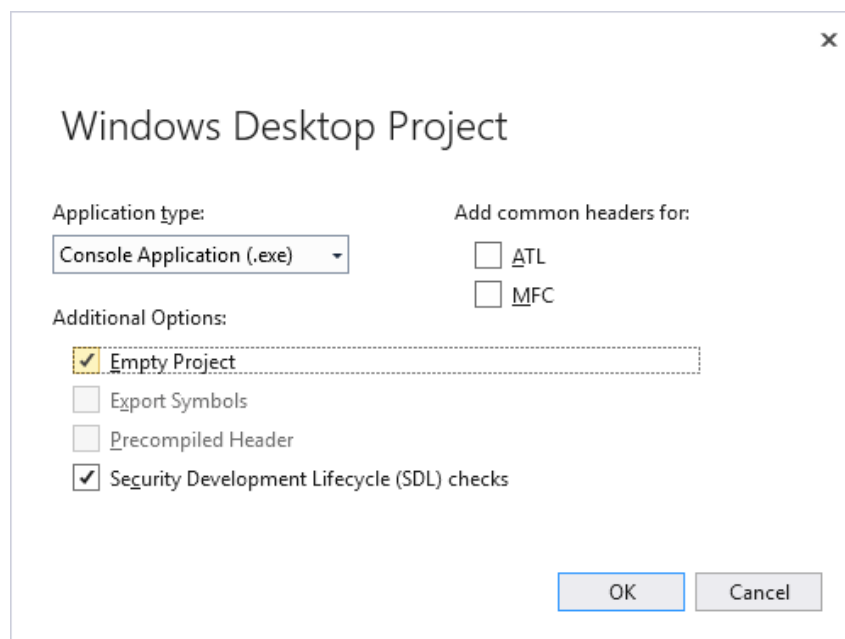


이제 메인 화면에서 새로운 프로젝트를 만들어봅시다.

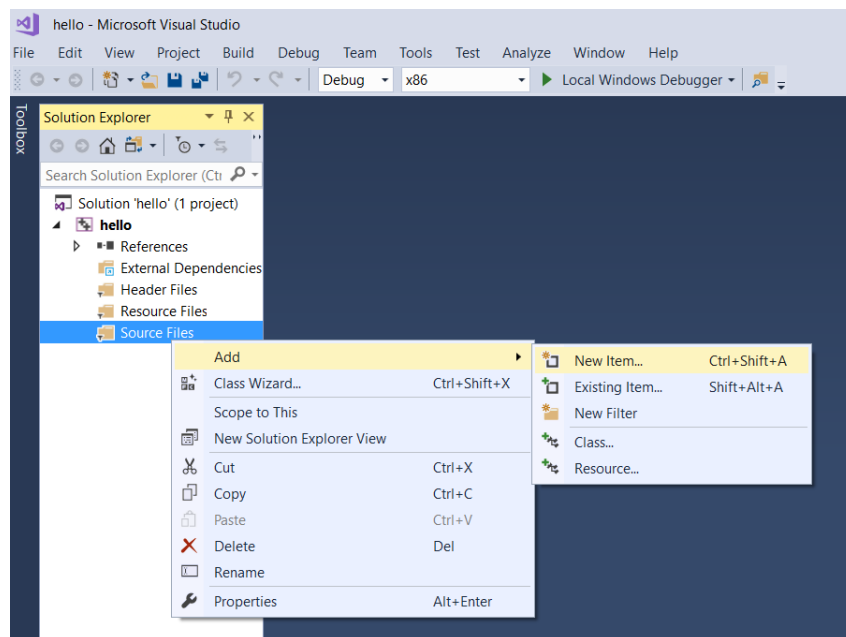
간단히 **Ctrl + Shift + N** 을 누르면 새 프로젝트를 만드는 창을 띄울 수 있습니다.



다음으로 프로젝트 선택 화면에서 **Windows Desktop Wizard (Windows 데스크톱 마법사)** 를 선택합니다.

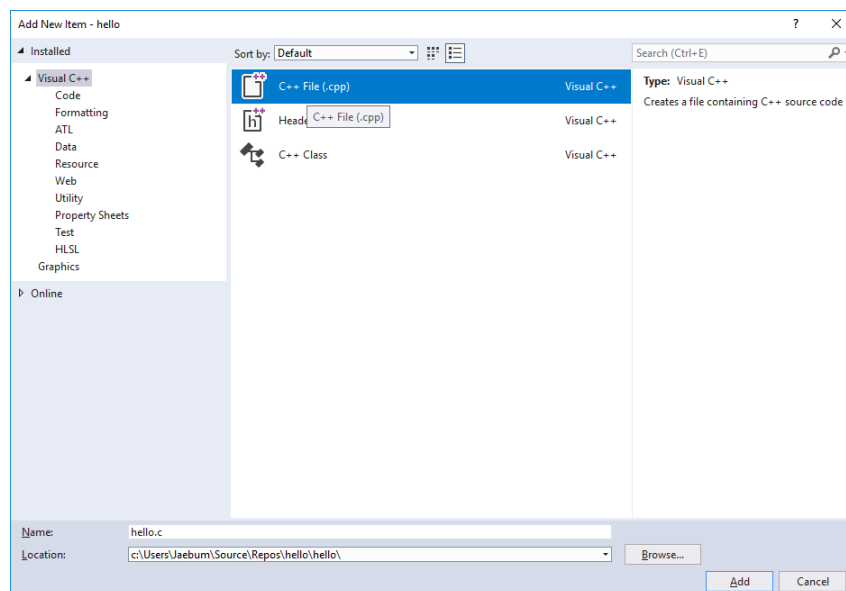


그 다음, 위와 같은 화면에서 Application type 에 콘솔 프로그램 (**Console Application**) 을 선택한 후, 아래에 빈 프로젝트 (**Empty Project**) 에 체크해줍니다. 그 후 확인을 눌러주면 프로젝트가 생성됩니다.



이제 프로젝트에 소스 파일을 추가해야 합니다.

기본적으로 왼쪽에 보시면 솔루션 탐색기에서 소스 파일을 쉽게 추가할 수 있는데, 솔루션 탐색기가 안보인다면, `Ctrl + Alt + L` 을 눌러서 띄울 수 있습니다.



위 화면에서 C++ 파일을 클릭한 뒤에, 이름에 `hello.c` 라고 적습니다.¹⁾

그럼 이제 왼쪽에 만들어진 `hello.c` 파일을 클릭합니다.

1) C++ 파일을 선택하였지만, 파일 확장자가 `.c` 이므로 비주얼 스튜디오에서 알아서 C 컴파일러로 컴파일 해줍니다.

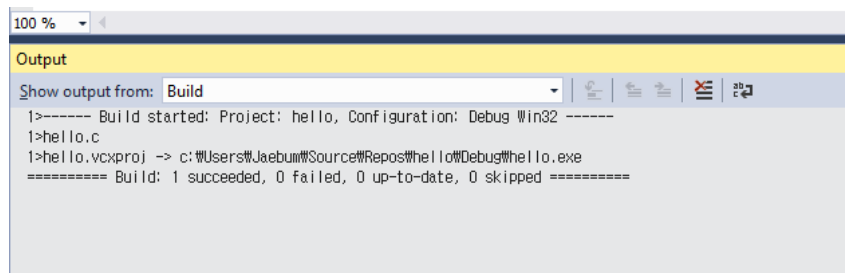
내 생애 첫 프로그램

드디어 여러분 인생 첫 프로그램을 만들 시간이 다가왔습니다. 에디터 화면에 다음과 같이 입력합니다.

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

한 가지 당부 드리고 싶은 말은, 반드시 손으로 직접 입력해 보기 바랍니다. 그냥 Ctrl - C, Ctrl - V 하는 것은 시간은 절약되지만 결국 나중에 머리에 남는 것은 없게 됩니다.

이제, 위 내용을 다 입력하였으면 F7 를 눌러 주어서, 또는 상단의 빌드 → 솔루션 빌드를 눌러서 컴파일 합니다. 만약 위 내용을 잘 써서 성공적으로 빌드 되었다면 아래 아래와 같은 화면을 보게 될 것입니다.



성공적으로 컴파일을 하였습니다

그런데, 간혹 가다 어떤 사람들은 오류가 뜨는 사람들도 있는데, 대표적으로

컴파일 오류

error C2143: 구문 오류 : ';'이(가) 'return' 앞에 없습니다

라 던지,

컴파일 오류

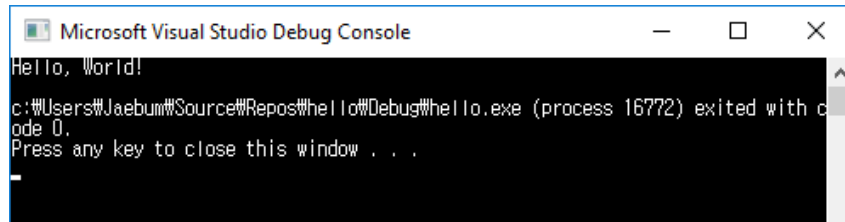
error C2001: 상수에 줄 바꿈 문자가 있습니다.

error C2143: 구문 오류 : ')'이(가) 'return' 앞에 없습니다.

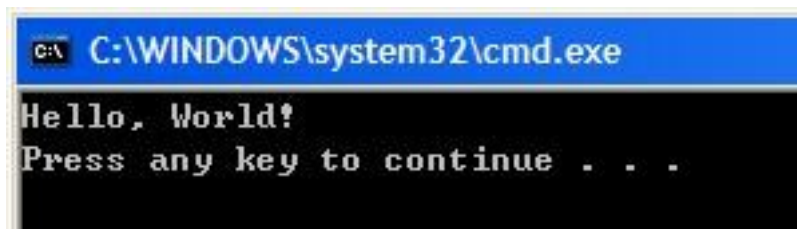
와 같은 오류를 보게 되는 사람들이 있습니다.

첫 번째의 경우, `printf("Hello, World! \n")` 다음에 세미 콜론 (;) 을 쓰지 않아서 나타나는 경우이고, 두 번째의 오류는 `printf` 안에 큰 따옴표로 제대로 닫지 않았을 경우입니다. 위에서 언급한 오류가 아니더라도, 어디선가 오류가 발생하였다면 십중 팔구 위의 소스 코드를 잘못 쳤기 때문 이므로 다시 한 번 신중히 쳐보거나, 그래도 안되면 복사해 보시기 바랍니다.

이제, 드디어 이렇게 완성된 프로그램을 실행할 시간이 다가왔습니다. `Ctrl + F5` 를 눌러서 프로그램을 실행해 봅시다.



Hello, World! 가 출력되었습니다. 밑에 나온 자질구레한 메시지는 무시해도 됩니다.



2008년 윈도우 XP 쓰던 시절의 출력 화면

만세! 위와 같이 Hello, World! 가 성공적으로 출력됨을 확인할 수 있습니다. 아래 나오는 자질구레한 메시지는 무시해도 됩니다. 그 부분은 여러분이 쓴 코드 때문에 나온 것이 아니라 비주얼 스튜디오 자체에서 추가한 메시지 입니다.

참고로 아래 사진은 2008년 강좌를 처음 제작했을 때 나온 프로그램 결과 입니다. 10년 이 넘게 흘렀는데, 모습은 비슷하지요?

아무튼 방금 여러분은 인생 첫 프로그램을 만들게 되었습니다!

다음 강좌에서는, 위 프로그램이 도대체 어떻게 동작하는 것인지에 대해 알아보도록 합시다.

뭘 배웠지?

- C 언어는 (요새 나오는 언어들에 비해) 조금 어려워보일 수 있지만, C 언어를 배움으로써 얻을 수 있는 것들이 더 많습니다! 여러분은 훌륭한 선택을 하신 것입니다!
- 컴퓨터는 0 과 1 밖에 모르기 때문에 컴퓨터에게 명령을 내리기 위해서는 **컴파일** 이라는 과정을 통해서 사용자가 입력한 코드를 변환해야 합니다. 이를 수행하는 프로그램을 **컴파일러** 라고 합니다.
- 비주얼 스튜디오를 설치하였고, 해당 IDE 의 컴파일러로 코드를 컴파일 하였습니다.
- 여러분 인생 첫 프로그램을 작성하였습니다.