1 INTRO: C언어학개론

1-1. C언어는 왜 필요한 것일까?

1-2. C언어로 프로그램을 작성하는 방법

1-3. 기본적인 프로그래밍

@ 더 알아보기: 주석 달기

1-1. C언어는 왜 필요한 것일까?

C언어를 배우기 시작한, 혹은 이미 C언어를 배우고 복습하는 여러분들을 환영한다. 여러분들은 많은 동기를 통해 C언어를 배우기로 마음먹었고, 배우는 다양한 방법 중 하나로 이 책을 선택하였다. 하지만 본격적으로 C언어를 공부하기에 앞서 우리는 C언어라는 것이 무엇인지를 먼저 알아야 한다. 모든 것이 그러하듯이 C언어 또한 이것이 어떤 필요에 의해 생겨났는지를 먼저 이해한다면 자연스럽게 C언어란 무엇인지, 이것을 왜 배워야 하는지 알게 될 것이다. 그렇다면 c언어는 왜 필요한 것일까?

(그림: 왜 C언어 배워요? By 어린아이)

독자들은 연애를 해보았는가? 연애 미경험자라면 공감할 수 없지만, 남자의 언어와 여자의 언어는 다르다. 그렇기 때문에 서로 사귀는 사이임에도 불구하고 “대화가 안 통한다. 눈치가 없다.”라는 말이 나온다.

(그림: 여자 –여자 언어-> 남자(이해 못함))

그렇다면 서로의 대화를 위해서는 어떤 방법이 필요할까? 아래의 그림처럼 중간에서 서로의 말을 해석해주고 편하게 전달해주는 사람이 있다면 대화가 통하지 않을까?

(그림: 여자 –여자 언어-> 공통 친구 –남자 언어-> 남자(이해 잘함))

사람과 컴퓨터 사이의 관계 역시 남자친구와 여자 친구의 관계와 비슷하다. 사람의 말과는 달리 컴퓨터가 사용하는 언어는 “[[1]](#footnote-1)기계어”이기 때문에 둘 사이는 마치 여자친구와 남자친구 단 둘이 있는 경우처럼 말이 통하지 않는다.

(그림: 인간 –언어-> 컴퓨터(이해 못함))

둘 사이의 원활한 대화를 위하여 서로의 말을 해석하여 전달하는 것을 “컴파일러”라고 한다. 컴파일러의 기능은 간단하다. 마치 번역기처럼 C언어로 언어를 전달받으면 그것을 기계어로 바꾸어 컴퓨터에게 전달해준다. 그 내부의 과정을 더욱 자세히 알고 싶다면 11강을 참고하자.

(그림: C언어-> 컴파일러 -> 기계어)

그렇기에 사람은 컴퓨터에게 말을 전달하려면, C언어로 컴파일러에게 설명을 해주어야 하고, 컴파일러는 “C언어”를 번역하여 “기계어”로 만든 뒤에 컴퓨터에게 알려준다. 최종적으로 컴퓨터는 기계어를 해석, 실행하여 결과를 화면에 전송한다.

(그림: 인간 –C 언어-> 컴파일러 –기계어-> 컴퓨터)

사람의 말로 프로그램을 설계하면 컴퓨터가 이것을 이해하고 진행하는 것이 가장 최상의 방법이지만, 안타깝게도 컴퓨터는 아직 그렇게 똑똑하지 않다. 그러므로 사람은 컴퓨터와 대화가 가능한 컴파일러를 통해 말하고자 하는 바를 전해줘야 하고, 컴파일러는 C언어를 기본적으로 알아듣기에 우리는 그것을 공부해야 한다.

(그림: 컴퓨터가 사람의 말을 이해하지 못하는 장면. 컴퓨터의 말 풍선에 컴파일 에러 화면을 출력.)

즉 우리가C언어를 통해 원하는 프로그램의 설계도를 컴파일러에게 전달하면, 컴파일러는 C언어로 작성된 설계도를 기계어로 번역된 설계도로 바꿔 컴퓨터에게 전달한다. 컴퓨터는 기계어를 읽고 설계도를 기반으로 프로그램을 만들고, 실행시킨다. 이처럼 C언어를 기계어로 번역하는 과정을 “컴파일”이라고 한다.

1-2. C언어로 프로그램을 작성하는 방법

본서는 Dev C++(ver 5.9.2) 컴파일러를 기준으로 설명을 진행한다. Dev C++가 아니더라도 Visual C++ Express Edition 2015등의 여려 컴파일러가 존재하나 학습을 위하여 무료로 배포되며 설치 및 사용이 가장 간단하기 때문에 Dev C++을 기준으로 설명한다. 어떤 컴파일러든 본서를 공부하는데 상관없으므로 크게 중요하지 않다.

1. 컴파일러 실행

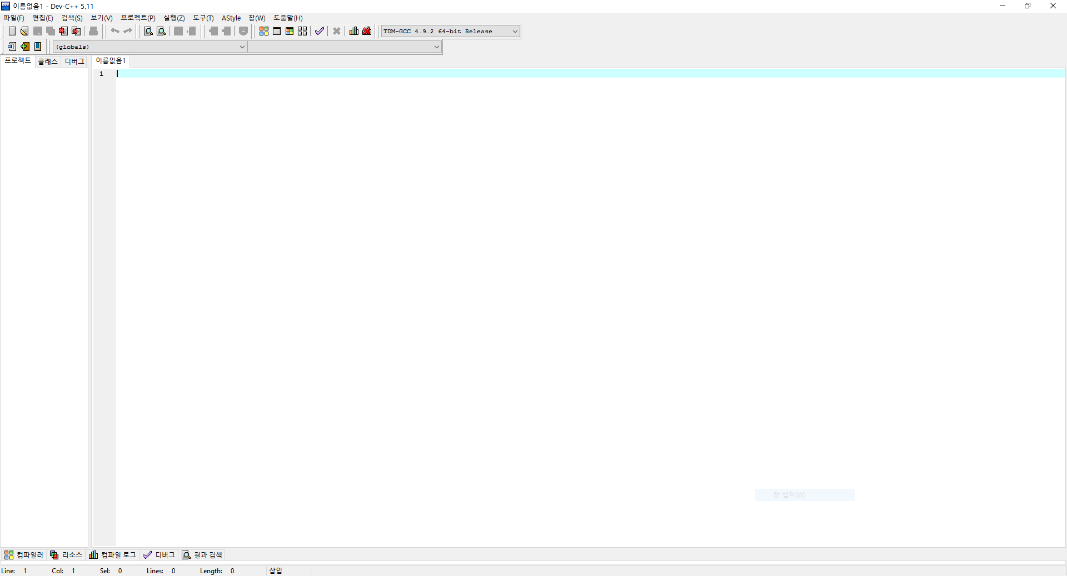
Dev-C++로 프로그램을 완성하는 과정을 통하여 C언어로 어떻게 프로그램을 완성할지, 그 과정은 어떠한지 이해를 해보자. 먼저 Dev-C++을 실행한다.

(그림: Dev 실행화면)

왼쪽 상단을 찾아보면 (New)아이콘이 보일 것이다. 혹은 상단의 File바-New-Source File, 혹은 실행화면에서 Ctrl+N을 입력하면 그림과 같은 화면이 생성된다.

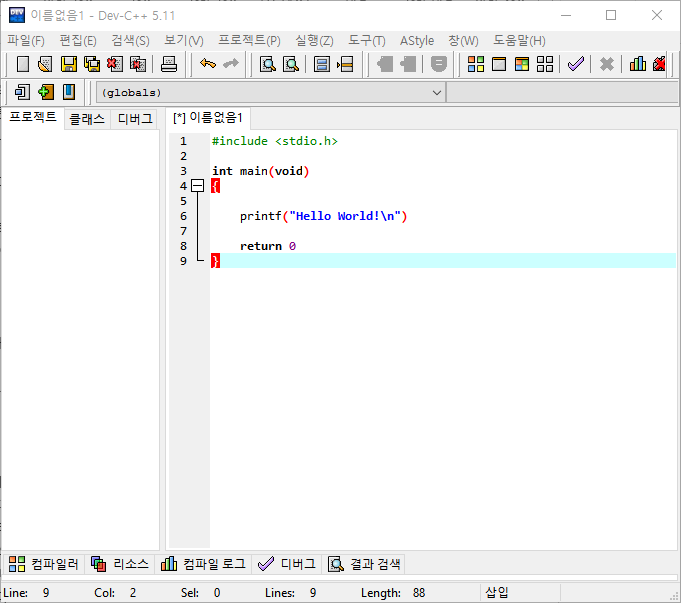
(그림: (New)아이콘)

(그림: 상단의 File바-New-Source File)



1. 프로그램 작성

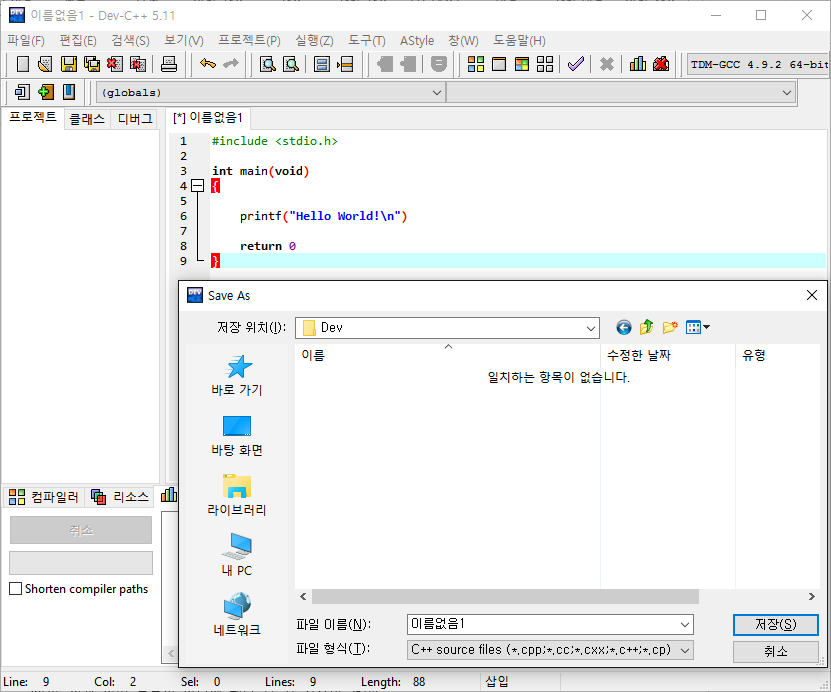
이제 프로그램을 작성할 준비가 끝났다. 아래의 그림과 똑같은, 간단한 프로그램을 작성해보자. 당신이 만든 설계도에 사용된 C언어가 어떠한 의미인지에 대한 이해는 작성 완료 직후 설명하겠다.

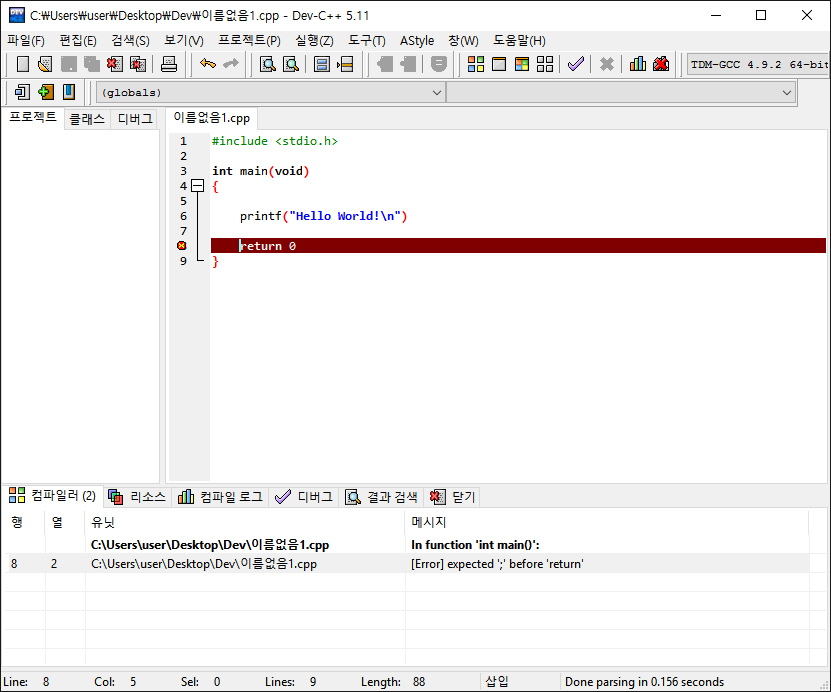


1. Compile

C언어를 컴파일러에 입력하였으니, 이젠 컴파일러가 컴파일을 할 차례이다. F9를 입력하거나 상단의 Execute 바에서 Compile을 클릭하면 컴파일이 실행된다.

(그림: 상단의 Execute 바 - Compile)

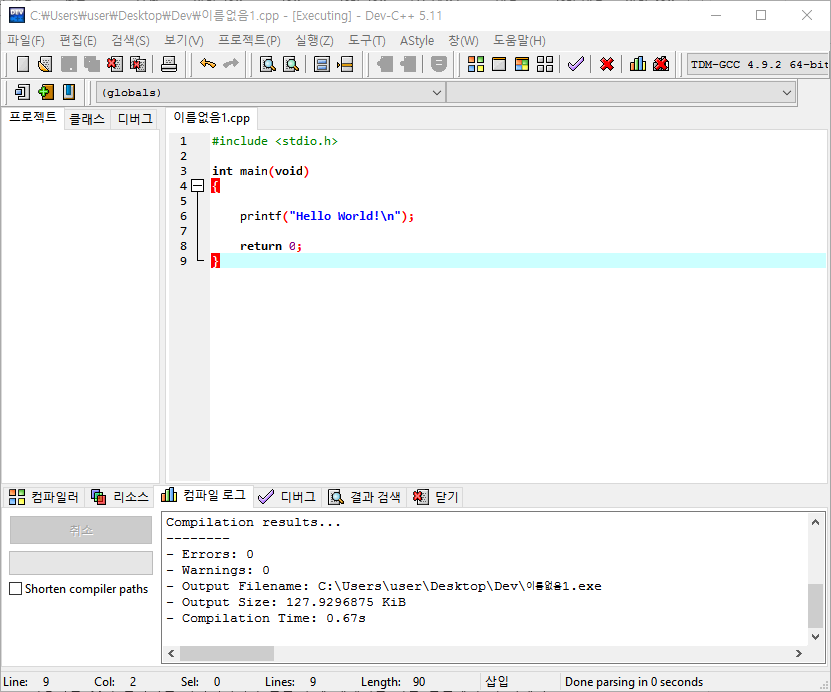


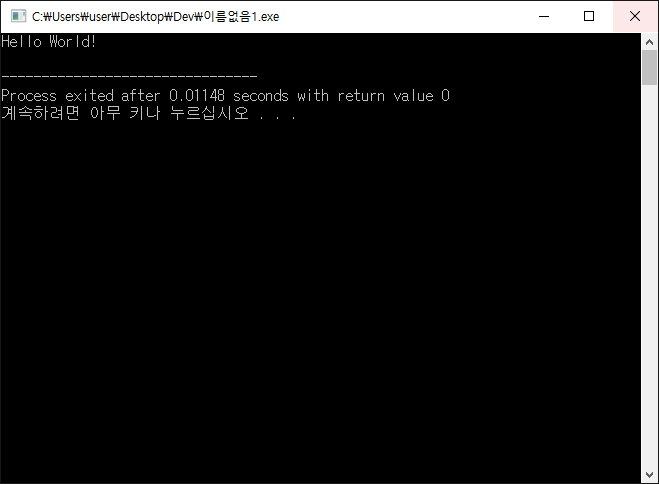


1. Errors

위와 같이 C언어의 프로그래밍이 잘못되었다면 컴파일러가 컴파일을 하는 도중에 C언어를 이해 못하여 에러가 발생된다. 아래의 그림과 같이 C언어를 고치고 다시 컴파일을 한 뒤 실행을 한다. 아까의 F9는 컴파일만 하지만, F11은 컴파일러가 컴파일을 한 기계어를 바탕으로 컴퓨터가 프로그램을 실행도 한다. F11을 입력하거나 상단의 Execute 바에서 Compile & Run을 클릭하여 실행한다.

(그림: 상단의 Execute바-Compile & Run)





1. 실행파일 생성

이렇게 가장 기본적인 프로그램을 C언어를 통하여 실행해보았다. 위의 과정을 생각하며 아래의 C프로그램 완성과정을 이해하여보자.

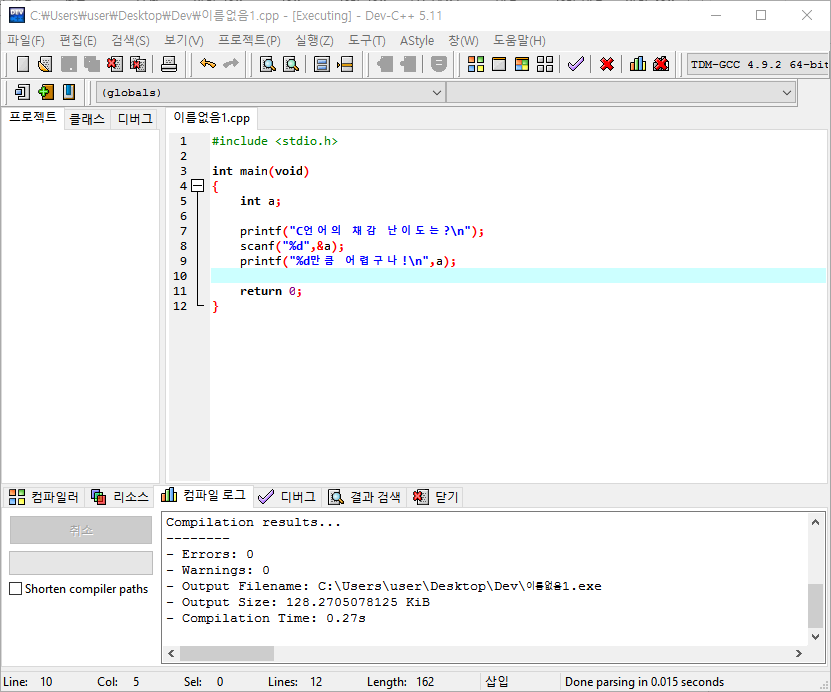
(그림: 기존 교수님 교재 앞부분 컴파일러의 프로그래밍 과정 순서도)

(아이: 아까 어떤 부분이 어디에 해당하는지 짚으며 설명)

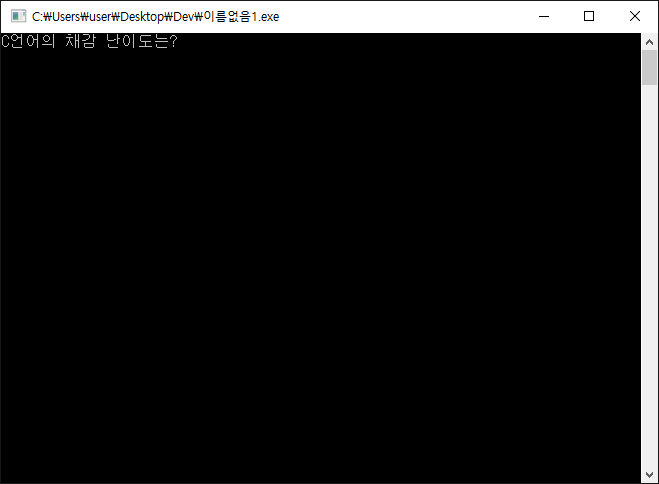
만든 프로그램의 실행파일과 목표한 프로그램이 일치한다면 그것을 완성된 프로그램이라고 부른다. 프로그램을 만드는 방법에 대하여 배웠으니 이제 우리가 프로그램의 설계도를 작성하면서 사용한 간단한 C언어를 공부해보자.

1-3. 기본적인 프로그래밍

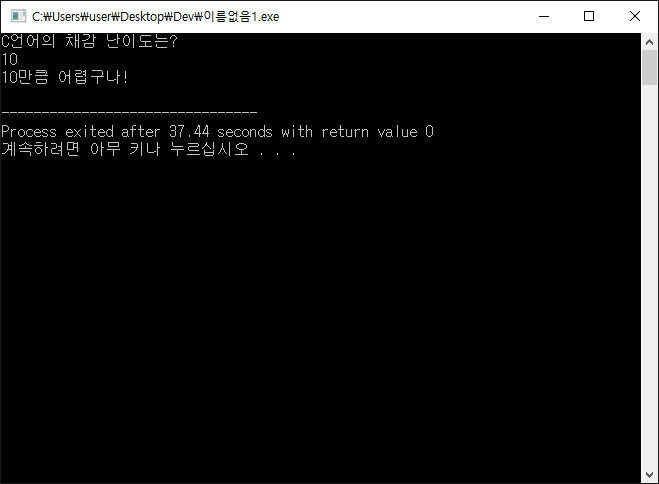
본격적으로 C언어에 대해서 배우기에 앞서 단원을 구분하지 않고, 셀 수 없이 많이 쓰이는 C언어의 기본들을 예제를 통하여 공부해보자.



우선 위와 같은 프로그램을 작성하여 실행해보자. 그러면 아래의 그림과 같은 화면이 뜬다.



이전의 예제와는 다르게 정수를 입력할 수 있다. 정수를 입력하면 프로그램이 실행결과를 출력한다. 임의의 정수를 입력하여 실행해보자.



어떻게 이런 프로그램을 컴퓨터가 실행한 것인지 이해하기 위해 우리가 컴파일러에게 C언어로 작성한 설계도의 내용이 어떤 것인지 C언어를 공부하며 하나씩 살펴보자. 프로그래밍은 처음 시작하는 사람들에게는 굉장히 생소한 개념이기 때문에, 약간의 비유를 통해서 C언어를 설명하도록 하겠다.

우리가 작성해주는 설계도를 바탕으로 컴퓨터가 하나의 가구(프로그램)를 만든다고 비유한다면, 우리는 컴파일러에게 설계도를 제공, 그들은 설계도를 완벽히 이해하여 컴퓨터에게 설계도를 가르쳐주고, 컴퓨터는 알맞은 재료로 원하는 가구를 얻는 만들어 내는 것이다. 컴퓨터는 사고하지 못하기 때문에 우리는 설계도에 아주 세세한 사항과 지시까지 모두 기입하여(의자의 오른쪽 다리는 톱을 사용해서 나무를 가로 1cm, 세로 1cm, 높이 40cm로 가공하여 사포 질을 한 후에 의자 앉는 부분의 오른쪽에 붙여야 한다) 컴퓨터를 이해시켜야 한다.

위와 같이 프로그래밍 과정을 가구를 만드는 과정에 비유하자면, 맨 위의 #include <stdio.h>는 도구상자를 의미한다. stdio라는 도구상자를 이용하여 앞으로 프로그램을 작성할 때 필요한 도구들을 이 맨 위의 도구상자에서 꺼내 사용한다. 프로그래밍을 위하여 여러 가지 도구상자가 필요할 때에는 여러 도구상자를 사용하기도 하지만 가장 기본이 되는 도구상자를 stdio이다.

(그림: stdio라는 상자에서 도구를 꺼내는 사람의 모습)

int main (void)는 가구를 만드는 주된 절차가 기입된 곳이다. 어떠한 재료를 사용할 것인지, 그 재료는 어떠한 과정을 통하여 가구를 이루게 되는 것인지가 기입된 부분이다. 정확한 이름은 “메인 함수”이고 역할은 프로그래밍을 작성하는 함수이다. 중학교 때 배운 함수, 즉 숫자를 넣으면 결과 값이 나오는 것처럼 재료를 넣으면 가구가 나오는 과정이기에 함수라고 부른다. 이에 대한 더욱 자세한 부분은 8단원에서 배우도록 하자.

(그림: 사람이 작업을 할 터전까지 메인 함수에 기입하여 컴파일러가 컴퓨터에게 가구를 만들 장소를 지정해주는 모습)

{}는 해당 함수의 내부를 의미한다. 여기에서는 메인 함수의 내부를 뜻한다. 메인 함수의 상단부분을 “선언 부”라고 하고, 선언 부에는 앞으로 사용될 모든 변수를 이곳에 미리 선언되어 있다. 컴파일러가 개정됨에 따라 변수를 자유롭게 선언하고 사용하여도 에러를 표시하지 않는 컴파일러가 많아졌으나, 선언 부에 변수를 미리 선언하고 사용하는 것이 암묵적인 약속이다. 이 선언 부에 대해서는 다음 단원에서 더 자세히 공부하자.

그런데 “변수”를 “선언한다”는 말은 어떤 의미일까? 다음의 비유를 들으면 쉽게 알 수 있다. 가구를 만들기 위해서는 우선 외부로부터 재료를 받아서 특정 부분에 보관해야 한다. 이때 보관 장소에 해당하는 것이 변수이다. 변수의 정확한 정의는 “값을 저장하는 공간”으로 이 변수에 저장된 값(재료)을 도구를 이용하여 가공하는 과정이 바로 프로그래밍이다(변수에 대해서도 다음 단원에서 더 자세히 알아보자). 재료에도 많은 종류가 있듯이, 변수에도 많은 종류가 있다. 그 종류에 대해서는 다음 단원에서 공부하고, 우선 변수를 선언한다는 의미부터 알아두자

(그림: int a; 부분)

변수를 선언한다는 의미는 “내가 이 공간에 저장할 재료의 성질은 나무이고 다른 저장 공간들과 햇갈리지 않기 위해 a라고 이름을 붙이겠다! 이번 작업에서 이 공간에는 이러한 재료를 저장한다!”라고 선언하는 것이다. 컴파일러가 설계도를 이해할 때 돕기 위해 재료들의 재질이 어떠한지, 여러 재질의 재료 중에서도 햇갈리지 않기 위해 재료에 이름을 붙이는 것 이다.

이러한 재료에 해당하는 변수를 선언할 때 재료의 재질에 해당하는 것이 “데이터 형(그림에서의 int)”이고, 장소의 이름에 해당하는 것이 “변수의 이름(그림에서 a)”의 이름이다. 변수의 이름은 어차피 이름이므로 편한대로 붙여주어도 상관없다(사실 이름에 사용할 수 없는 기호도 있다. 이에 대해서는 다음 단원에서 더 자세히 공부하자). 하지만 앞의 데이터 형은 의미하는 바가 있다. 재료의 종류를 분류하듯이 데이터 형은 변수의 성질을 의미한다. int는 정수, float은 실수, char는 문자를 의미한다. 약자의 의미를 알아보면, char는 character의 약자로 문자, int는 integer의 약어로 영어로 정수를 의미, float은 부동 소수점을 의미하니 쉽게 암기가 될 것이다. 다음 단원에서 공부하고 우선은 그러려니 하고 넘어가자.

(그림: 컴퓨터에게 나무를 가지고 재질, 이름을 지정해서 설명해주는 컴파일러)

(그림: int a;)

Dev를 활용하는 방법을 배울 때 ;를 붙이지 않아 오류가 나타났듯이, C언어를 사용하면서 웬만한 문장에는 항상 세미콜론이 붙는다. 그 이유는 다음과 같다. 컴파일러는 띄어쓰기, 들여쓰기, 엔터 등을 인식하지 못하기 때문에 문장과 다음 문장의 차이를 사람처럼 띄어쓰기, 엔터 등이 아닌 세미콜론“;”을 통해 인식한다. C언어를 배우기 시작한 초반에 세미콜론을 붙이지 않는 실수가 가장 하기 쉬운 실수이므로 기억하자.

(그림: printf(“C언어의 채감 난이도는? \n”);)

printf(“”)는 가장 기본적인 도구 중 하나이다. 도구상자인 stdio에서 나오는 도구이므로 앞에 도구상자가 있어야 사용 가능하다. Printf는 “printing하는 function이다”는 의미의 축약어로, “”안에 있는 내용을 프로그램 실행을 하면 출력한다. “”안에서 특수한 기호를 입력하면 특별한 것을 출력하는 경우가 있는데 그 대표적인 예가 \n이다. \n은 줄 바꿔 쓰기를 의미한다. 차이를 알고 싶다면 아래의 그림으로 확인하자.

(그림: \n사용한 것, \n사용 안 한 것)

scanf(“%”,&)역시 가장 기본적인 도구 중 하나이다. stdio 도구상자에 포함되어 있는 도구이므로 stdio 도구상자가 존재해야 사용 가능하다. scanf는 “scan하는 function이다”의 축약어로, 입력되는 값을 &뒤에 있는 변수에 저장해주는 도구이다. %와 &에도 의미가 있으나 뒤에 7단원에서 자세히 설명하고 우선은 암기를 하자. 앞의 %에 형식문자열을 입력하고 뒤의 &에 해당하는 변수를 입력하면 입력하는 값을 해당되는 변수에 저장한다. 형식문자열에 대한 자세한 것은 다음 단원에서 알아보자.

(그림: 아이가 왜 자꾸 다음단원에서 알려주냐고 불평하는 모습.)

여기서는 기본적인 프로그래밍을 공부할 것 이기 때문에, 다루지 않는 거란다.

(그림: printf("%"); 있는 부분)

printf 도구에도 형식문자열을 표시해주면 변수를 받아서 출력을 해준다.

(그림: return 0;)

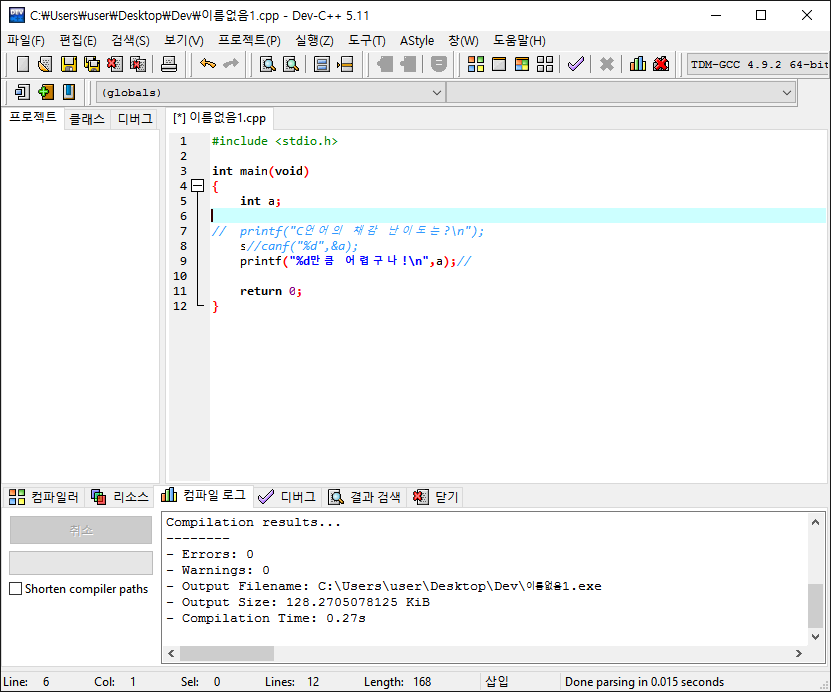
아직은 return에 대하여 자세히 설명하기에는 지식이 많이 부족하다. 함수 단원에서 자세히 설명할 테니 일단은 프로그래밍의 마침표라고 생각해두자. 마지막으로 아래의 총 정리된 그림으로 복습하자.

(그림: C언어의 채감 난이도는?.exe의 옆에 각 함수 간략하게 설명)

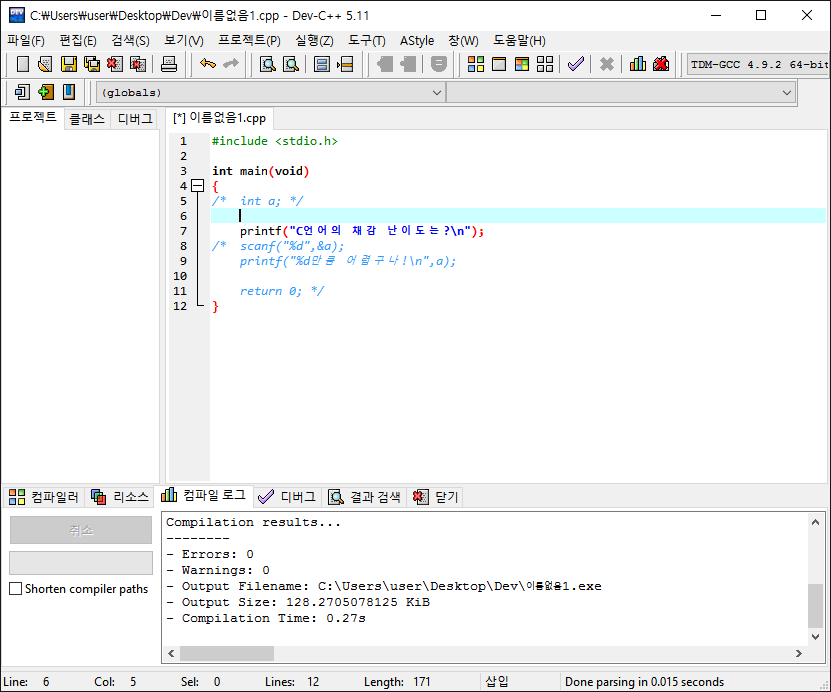
@ 더 알아보기: 주석 달기

C언어로 작성된 설계도는 사람이 컴파일러에게 번역을 의뢰하는 설계도이다. 그렇기에 그 설계도는 사람끼리도 나눠보는 경우가 잦고, 그에 따라 자신의 설계도를 설명하는 별도의 메모가 필요한 경우가 잦다. 혹은 자신이 만든 설계도에도 자신이 어떠한 생각으로 만들었는지 등 메모할 내용이 있는 경우가 잦다. 그러한 경우에는 컴파일러에 작성한 설계도에 영향을 끼치지 않으면서 메모를 하는 방법이 필요한데, 바로 이 주석 기능이 메모 기능을 한다.

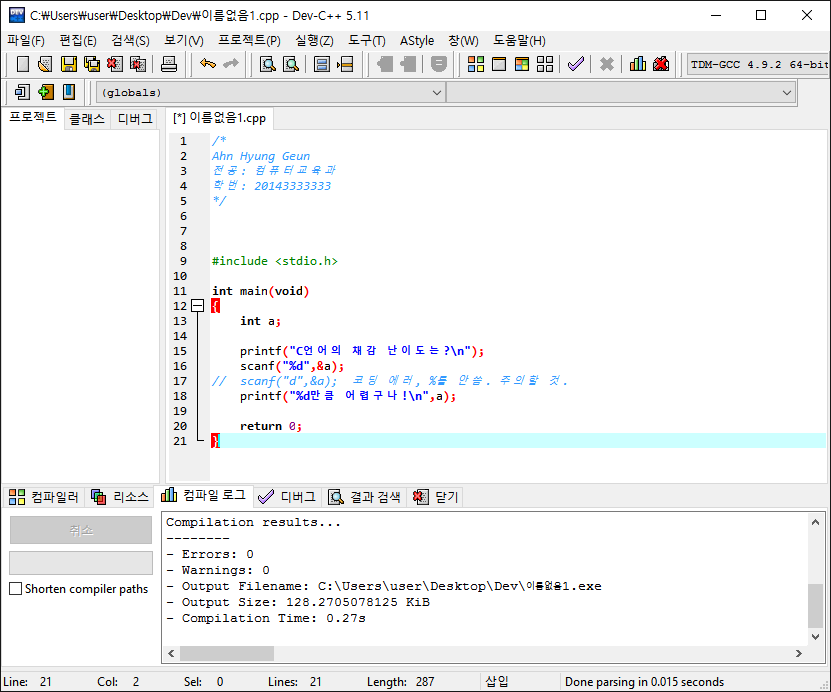
주석을 표시하는 방법은 두 가지 방법이 존재하는데, 먼저 해당 줄을 주석으로 처리하는 방법은 다음과 같다. 아주 간단하게 //를 써주면 그 행에서 //이후의 내용은 모두 주석으로 처리된다.



주석을 표시하는 다른 방법은 행만 주석으로 처리하는 방법이 아니라 해당 구간의 내용을 전부 주석으로 바꿔주는 방법이다. /\*로 해당 구간의 시작을 정해주고 \*/로 해당 구간의 끝을 지정해주면 그 내부에 속하는 모든 내용은 주석으로 처리된다. 다음의 그림을 참고하자.



이러한 주석은 위와 같은 방법으로만 사용되지않고, 자신의 개인 정보를 입력하고, 코딩 에러가 남지만 프로그램에는 영향을 끼치지 않게도 쓰이는 등 앞으로 자주 활용되니 알아두도록 하자.



1. 0과 1로만 구성되어 있는(2진법) 언어이다. [↑](#footnote-ref-1)