# 3장 헬로 월드



프로젝트는 여러 개의 C 언어 코드 파일로 이루어집니다.각 파일은 크게 소스 파일과 헤더 파일로 나누어 집니다. 보통 관련있는 파일들을 같은 경로에 저장합니다. 하지만 실제 경로와 상관없이 파일의 목적에 따라 논리적인 그룹으로 묶기도 합니다. 솔루션 탐색기에서는 논리적 폴더를 생성하여 언제든 관련있는 파일들을 하나의 그룹으로 관리할 수 있습니다.

—-------------------------------

혹시나 나중을 위해 남겨놓은 초안

윈도우 탐색기에서 exe 파일을 더블 클릭하여 실행할 수도 있지만, 이 경우 출력 콘솔 창이 실행 완료와 함께 사라지므로, 가급적 명령행 도구를 실행한 다음 프로그램을 실행하는 것이 좋습니다.

다음과 같은 결과 가 표시됩니다.

옵션 >

도구 >

솔루션 디렉터리 설명.

디버그/릴리즈 빌드 - 런타임으로 설명하기.

이번에는 온라인 컴파일러에서 실행해 본다. (한번만 설명)

이제 코드를 한줄씩 이해해 봅시다.

Main함수. C언어 프로그램의 시작과 끝.

main함수는 두가지 버전이 있지만, 당분간은 입력이 없는

지시문은 한번에 하나씩. 끝날때마다 세미콜론을 찍어 준다.

printf는 화면에 출력을 하기 위한 함수. stdio.h라는 파일에 정의되어 있음.

왜 헤더 파일을 따로 정의했을까.. 레고 블럭도 연결할 수있는 부분을 미리 알아야 연결할 수 있으니까. 그래서 함수의 모양새를 미리 알려줘야 한다. 그래서 매번 소스코드를 컴파일할 때 알려줄 수 없으니 따로 빼서 #include 로 포함시킨 것.

Retrun

주석. 코드를 작성하는 것은 컴퓨터에 시킬 작업 지시문. 그 작업을 변경하는 것은 결국 사람. 코드를 읽고 바로 의도를 이해할 수 있으면 가장 좋지만 그렇지 않을 경우에는 추가 설명이 필요합니다.

/\* \*/

//

“설명이 필요한 개그는 실패한 개그이다"

하지만 모호할 때는 설명이 필요합니다. -> 클린 코드

빈줄은 실행 파일을 만드는 데는 의미가 없지만, 시각적으로 코드를 분리시켜 주므로 적절히 빈줄로 영역을 나눠 주자.

함수는 작은 함수들로 이루어 진다.

디버깅.

이제 잠시 책을 기존 코드에 “안녕하세요"를 출력해 보자…. 한글도 출력되는 것만 확인. 문자열 인코딩 이런건 나중에 이야기 하자.

한줄씩 실행 -> 로컬 변수 살펴 보기

한줄씩 실행하다간 원하는 부분까지 가다가 죽을꺼야,. 중단점 설정해서 원하는 부분까지 가서 좀 더 살펴보자.

Step over같은 것은 용어만 설명하는 수준에서 넘어갈 것.

—-

드디어 첫번쨰 프로그램을 실행해 보았다. 이제 이 프로그램 골격을 바탕으로 계속 확장시켜 보자.

컴파일러는 개별 소스코드 단위로 동작을 이해하여 기계가 수행할 수 있는 코드로 변환하므로, 표준 라이브러리를 비롯하여 해당 파일에 정의되지 않은 함수에 대해서는 어떤 식으로 사용되는지를 알 수 없습니다. 이 문제를 해결하기 위하여 사용하는 함수의 모양을 선언해 주어야 하는데, 이와 같은 함수 선언을 모아둔 파일이 헤더 파일입니다. 확장자는 .h를 사용하며, 특정 함수를 사용하기 위해서는 include 문을 사용하여 적절한 헤더 파일을 포함시켜주어야 합니다.