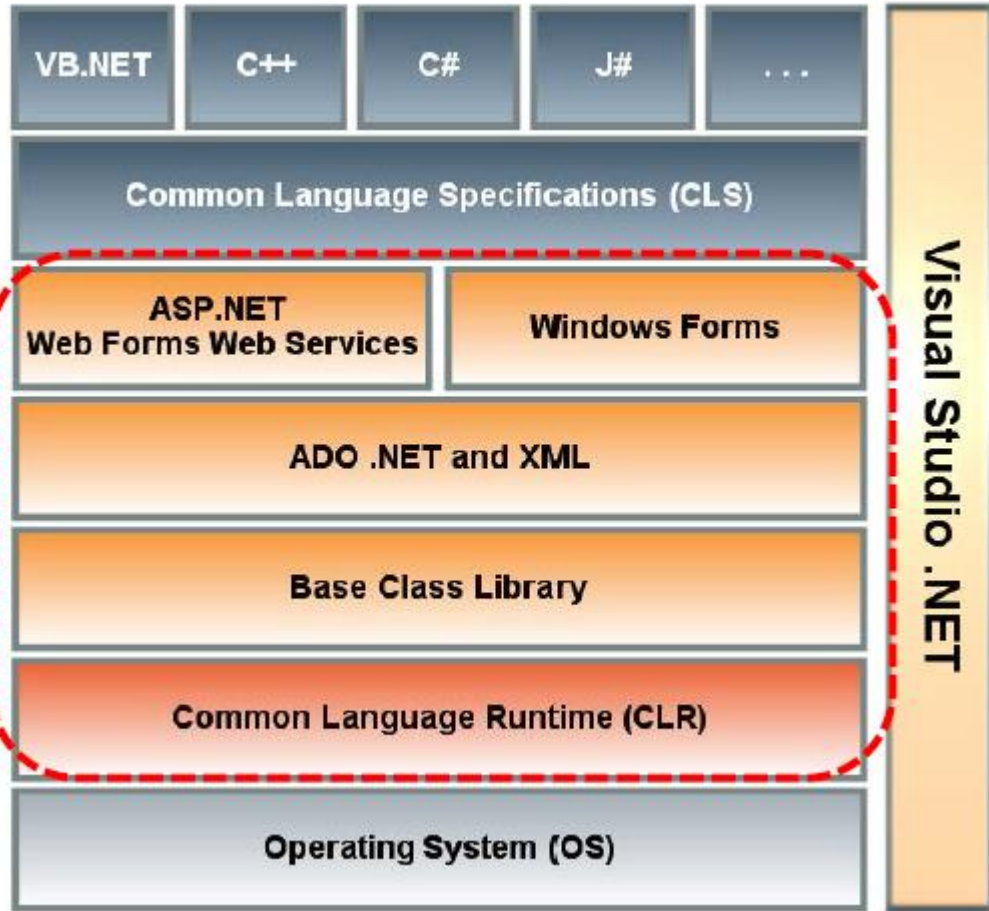


## .NET Framework (닷넷 프레임워크)

C# 이라고 하면 항상 .NET Framework 에 대한 얘기가 나오곤 합니다.

.NET Framework 는 Microsoft에서 개발한 웹 기반, 윈도우 기반 application을 개발할 수 있고, 실행시킬 수 있는 프레임워크이며, .NET 표준에 따르는 프로그래밍 언어로 개발된 프로그램의 실행환경이라고 할 수 있습니다.

.NET Framework의 구성을 그림으로 보면 다음과 같습니다.



위에서 하나씩 살펴보자면

여기서 .NET 표준을 따르는 (CLS; Common Language Specification)을 따르는 여러 언어를 제공합니다.

(대표적으로 C# 외에 VB.NET , Managed C++ , Jscript.NET , j # 등이 있다.)



Class Library 구성을 보면 다음과 같습니다.  
ASP.NET Web Forms Web Services : Web service와 Web apps를 개발  
Windows Forms : Windows 프로그래밍  
ADO .NET and XML : XML 및 데이터베이스 활용  
Base Class Library : CLR 지원을 위한 핵심 클래스 라이브러리

Common Language Runtime (CLR)

.NET 언어로 작성된 프로그램을 실행하고 관리하는 실행환경이며 Java의 virtual machine에 해당한다고 볼 수 있습니다.  
CLR은 .NET에서 동작하는 프로그램을 적재하고, 프로그램의 동적 컴파일, 프로그램의 실행, 메모리관리 (Garbage Collection) ,프로그램의 예외처리, 언어 간의 상속 지원, COM과의 상호 운영성 지원 등을 가능케 합니다.  
.NET 언어라 하면 C#뿐만 아니라 위에서 나열하였던 VB.NET 등 .NET 표준을 따르는 여러언어들은 CLR을 통해 프로그램을 실행할 수 있게 됩니다.



이것이 가능한 이유는 Intermediate Language (IL, 중간언어) 라는 기계어로 변환하기 쉬운 상태의 중간 단계의 언어로 .NET에서 번역되어 실행되기 위해 IL형태로 컴파일을 하는데, IL을 기계어로 바꾸는 번역기만 제공되면 어떤 플랫폼에서도 실행가능합니다. JIT (Just-In-Time) 컴파일러를 통해 IL을 동적으로 컴파일하는데 .NET에서는 이와같이 프로그램을 2번 컴파일 합니다. Assembly는 이 때 IL로 컴파일된 결과 파일들을 패키징 한 것을 말합니다. (.exe는 여러개의 class와 Main프로그램을 포함하고, dll은 class만을 포함한다고 생각하면 된다.)

또 .NET 언어가 지켜야 하는 스펙으로 Common Language Specifications (CLS) 로 다른 .NET 언어로 작성된 것도 호환되어 동작 가능하게 합니다. 이 과정에서 Common Type System (CTS) 라는 .NET 언어마다 Data type이 다를 수 있는데,이를 언어나 시스템 환경에 관계없이 동일한 Data type을 유지하기 위한 규약이 사용되기도 합니다.

쉽게 요약하면, C# 컴파일러는 우리가 작성한 코드를 IL로 작성된 실행파일을 만들게 되고, 이 파일을 실행시키면 CLR이 IL을 읽어 다시 OS에 이해할수 있는 코드로 컴파일하여 실행시킵니다. 이렇게 서로 다른 언어가 만나는 지점이 IL언어이고, CLR이 다시 자신이 설치되어 있는 플랫폼에 최적화시켜 컴파일한 후 실행하게 됩니다. 이렇게 플랫폼에 최적화된 코드를 만들어 내는 장점이 있으나, 단점은 실행시에 이루어지는 컴파일의 비용부담이 되겠습니다.

이상으로 C#이 어떤 방식으로 프로그램을 동작하게 하는지 정리보았습니다.

