

# Start In C#

📄 🚩 🌐 2025-08-06

📄

## .NET Start

- في البدايه ننزل #C في Visual Studio 2022
- اول حاجه و اول اختلاف هتلاحظه اول ما تدخل اول بروجيكت

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp1
8  {
9      0 references
10     internal class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             Console.WriteLine("hi this is my first c# app");
16             Console.ReadLine();
17         }
18     }
19 }
20
21
22
23
```

📄

- تاني حاجه هنعرفها هي اي الفرق بين Library vs Framework vs Platform

### 1. library

مجموعه من الفانكشنز تستخدمهم بدون ما تكتب الكود كل مره

## What is Library?

A library is a collection of pre-written code that developers can use to add functionality to their applications.

A library typically provides a set of functions or classes that a developer can call from their code to perform specific tasks, without having to write all the code from scratch.

Example: Math Library.

### 2. framework

مجموع من المكاتب و القوانين الي تحكمك في كتابه الكود

ممکن نقدر نقول بر دو ان دي بيئه العمل بتاعت كل مكان بتبقا مخصصه ليه هو بالتحدد زي مثلا شركه

بتستخدم بيئه عمل خاصه بيها

## What is Framework?

A framework is a set of libraries and tools that provide structure and support for building and running applications.

A framework defines a set of rules, protocols, and conventions that developers must follow when writing their code. This helps ensure that all parts of an application work well together and follow a consistent design pattern.

### 3. platform

ال environment الي بتشغل عليه السوفت وير عليها

## What is Platform?

A platform refers to the hardware or software environment in which a piece of software runs.

For example, a computer running the Windows operating system could be considered a platform for running software written for Windows

Platform = Programming Language + Libraries

س

الي هناخده دلوقتى بقا هتبقى معلومات عامه عن **NET.** بشكل عام

!!

## يَا رَبِّ , لَكَ الْحَمْدُ كَمَا يَنْبَغِي لِجَلَالِ وَجْهِكَ , وَلِعَظِيمِ سُلْطَانِكَ.

.NET platform بشكل عام هي تعتبر .NET

.NET Is a platform

- Languages in .Net

- C#
- Visual basic
- F#

كل اللغات دي بتستخدم نفس المكاتب

- platforms

- .NET core
- .NET framework

.NET framework كانت هي .NET اول بلاط فورم نزلت لما نزل الـ

احدث .NET core لآكن الـ

## What is .NET?

Platform = Programming Language + Libraries

.Net is a Platform 😊

Languages in .Net Platform are:

- C#, Visual Basic, and F#...etc.

Platforms:

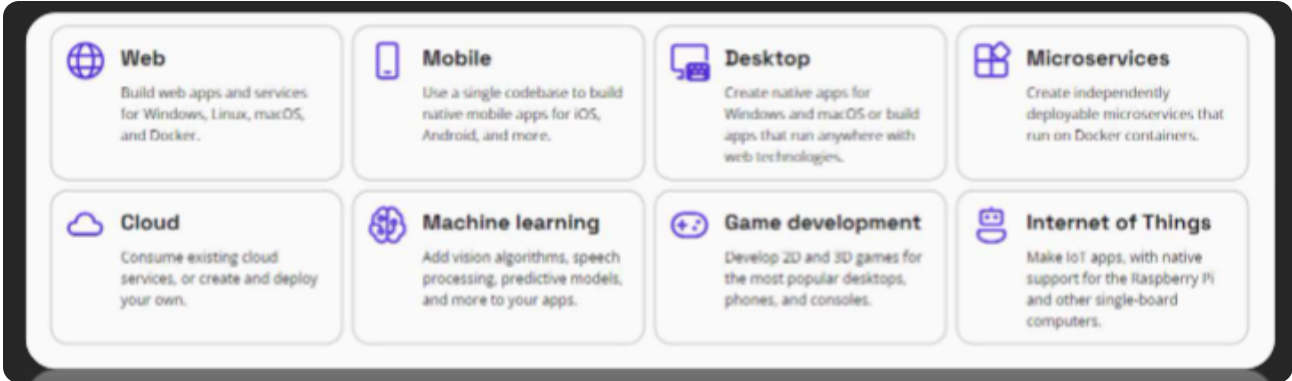
- .Net Core: (runs anywhere) Windows, Linux, and macOS
- .Net Framework: Websites, services, and desktop apps ..etc on windows
- Xamarin /Mono : a .NET for Mobile (Cross Platform runs on android or ios)

كل دول بيستخدمو نفس المكاتب وتقدر تستخدم كله في كله مش  
علشان احوال من فريم وورك لفريم وورك اتعلم مكاتب جديده

تقدر تعمل اي بقا بالـ .NET ؟

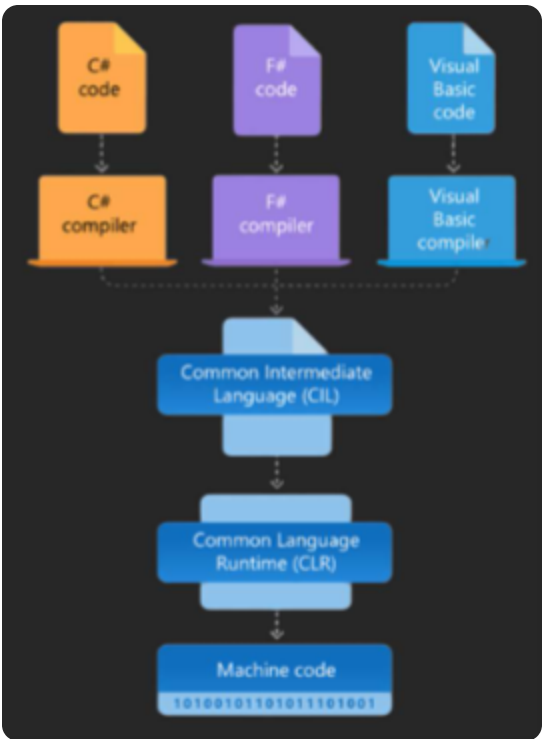
- by using .NET YOU Can build

1. desktop apps
2. web apps
3. mobile apps
4. games
5. IOT (Internet of things)



احنا هندرس .NET Framework مش .NET core لان الفريم وورك اقدم و اغلب الشركات الان تستخدمه و مغيرتش برامجها للكور لسه الكور احدث لכן لازم تتعلم القديم علشان الجديد يبقى سهل عليك

## هنعرف بعد كده حاجه اسمها .NET Compilation



### المختصر

عندي لغات برمجه عديده و كل لغة برمجه ليها كومبايلر خاص بيها الكومبايلر ده مش بيحول الكود بتاعك علطول للغه الكومبيوتر الي هي (1010 باينري - machine code) لا الكومبايلر بتاع اللغه بيحول الكود الي انت كتبتة لـ CIL الي هي تبقى common intermediate language او بمعنى لغة وسطيه و بعد كده يجي الـ CLR الي يبقى common language runtime المسئول عن كل اللغات يعمل runtime لل machine code او للغه الكومبيوتر . علشان كده كل لغات الـ .NET نفس السرعه.

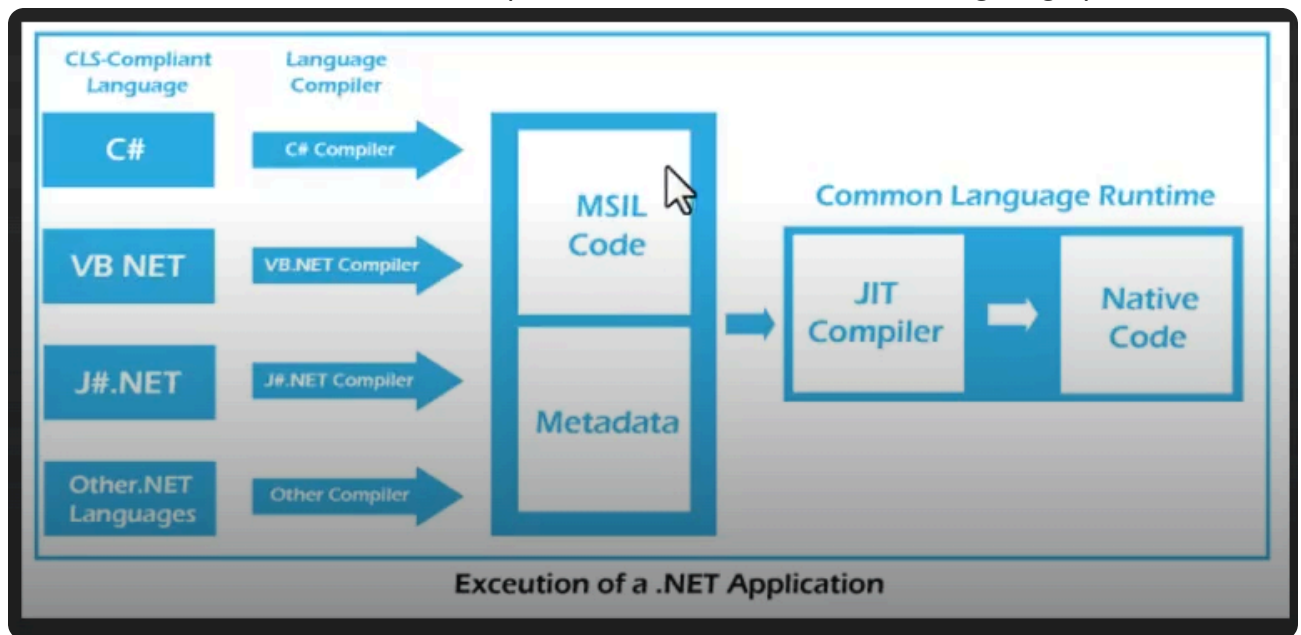
## هنعرف بقا دلوقتي

## .NET Framework Architecture

- two major components
  - APP لل RUN الي بيعمل : CLR
  - مكاتب بشكل عام : Class library provides

- what is the CLR :

يحول الكود من (Common intermediate language) CIL إلى machine code



لو حد كتب بالـ IL (INTERMEDIAT LANGUAGE) اي هتبقى المشكله ؟ المشكله ان الكود هيبقى بطيئ جداً و عما تعمل RUN هياخد وقت كثير .  
الـ JIT سريع 64 بيت

٥

📅 2025-08-07

## هناخد بقا دلوقتي حاجه اسمها CLR Main Components

- اي هي المكونات الرئيسيه في CLR و بعدها هنمسك كل حاجه لوحدها نشرحها

1. Common type system (CTS)
2. Common language specification (CLS)
3. Garbage Collector
4. Just in time compiler (JIT)
5. Metadata and Assemblies

### نبدأ نفصل كل واحد بقا منهم

1. Common type system (CTS) :

كل اللغات الموجودة بالفريم ورك عندها كومبايلر خاص بيها ، وظيفة هذه الكومبايلرز تحويل اللغات ل لغة وسطية موحدة ومن ثم الى لغة التجميع وتكون بملف تنفيذي بعدها يقوم CLR بعمله بتحويل هذه اللغة الوسيطة ل لغة آله، ازاي بقا ؟ اول شيء يجي دور CTS وهذا يقوم بتحويل تايب اللغات الى تايب جديدة موحدة يفهمها ال CLR

2. Common language specification (CLS) :

مجموعة من القواعد والمتطلبات التي يجب على جميع لغات دوت نت اتباعها لضمان التوافق والتشغيل

السليم على البلاتفورم بمعنى اخر : باتباع قوانين CLS يمكن للغات البرمجية المختلفة العمل معا بشكل سلسل ومتوافق ع البلاتفورم .NET

### 3. Garbage Collector (GC) :

الـ GC وهو اختصار الى Garbage Collector وهو ثالث مكون للـ CLR. وظيفة الـ GC يعمل تنظيف للذاكرة بشكل تلقائي. مثلا لديك فانكشن فيها متغيرات وهم ليسو مبعوثين باي رفرنس & لذلك بعد انتهاء سكوب الفانكشن سوف يقوم بحذف هذه المتغيرات وتنظيف الذاكرة خلفك. نفس الشيء اذا عرفت متغيرات داخل لوب او كنترول فلو.. وهو موجود في اللغات الحديثة مثل Python , Java , C# , لكن C++ و C لا تحتوي على GC , الـ GC هو مدير الذاكرة .

### 4. Just in time compiler (JIT) :

كومبايلر يقوم بتحويل اللغة الوسطية للغة الآله او بتحول الـ MSIL Code لـ Native code .  
ملاحظه جانبيه - Side Note :

Microsoft Intermediate language اختصار لـ : MSIL

عندنا بعد كده في انواع للـ JIT :

#### 1. Pre JIT (AOT Ahead Of Time) :

. يقوم بترجمة الكود كامل كله مره واحده ، ده بيستهلك ذاكرة بس يكون اسرع بالتنفيذ

#### 2. Normal JIT :

هذا الكومبايلر الافتراضي يقوم بترجمة الجزء المطلوب فقط من الكود ويخزنه في الذاكرة في الكاش ميموري، يكون بطيء مقارنة بالنوع الاول لكن استهلاكه للذاكرة اقل

#### 3. Econo JIT :

يقوم بترجمة الكود المطلوب لكن بدون تخزينه بالذاكرة ،اي يقوم بترجمته في كل مرة يحتاجه

### 5. Metadata and Assemblies :

أولاً : يعني إيه Assembly : هو ببساطة الملف اللي بيتكون بعد ما تكتب كود C# وتعمل له Build. يعني الملف .exe أو DLL اللي بنشغله أو نستخدمه. مثلاً: لو عملت برنامج اسمه MyApp.exe هيتكون ملف اسمه MyApp .

ثانياً : يعني إيه Metadata او (Manifest) : معلومات عن الكود اللي كتبته او الملف الي لسه مسمينه فوق الي هو .exe او DLL , مش الكود نفسه،  
ونقدر نقول ان الـ Metadata مخزنه :

اسم الـ assembly و الاصدار بتاعه و مين الـ creator و اي الفايلز الي عملته و بقيت المعلومات بتاعت الـ assembly .

وصف اسهل له تخيل إن الـ Assembly هو "علبة"، وجوا العلبة دي فيه حاجتين الكود اللي كتبته (لكن مترجم بلغة اسمها IL) Metadata = و شرح للكود علشان .NET يفهمه ويشغل عليه .

ش

## هناخد دلوقتي بقا وظائف الـ CLR Function

1. convert program into native code
2. manger for all .NET supported Language

3. exception handling : لما يحصل ايرور عندي بين لغتين مثلا في برنامج هو بيصلحه
4. provide type-safety : بيتأكد ان عندي انواع الداتا تاييس متشابه في البرنامج لو يستخدم لغتين مختلفتين
5. memory management
6. security
7. performance

٣

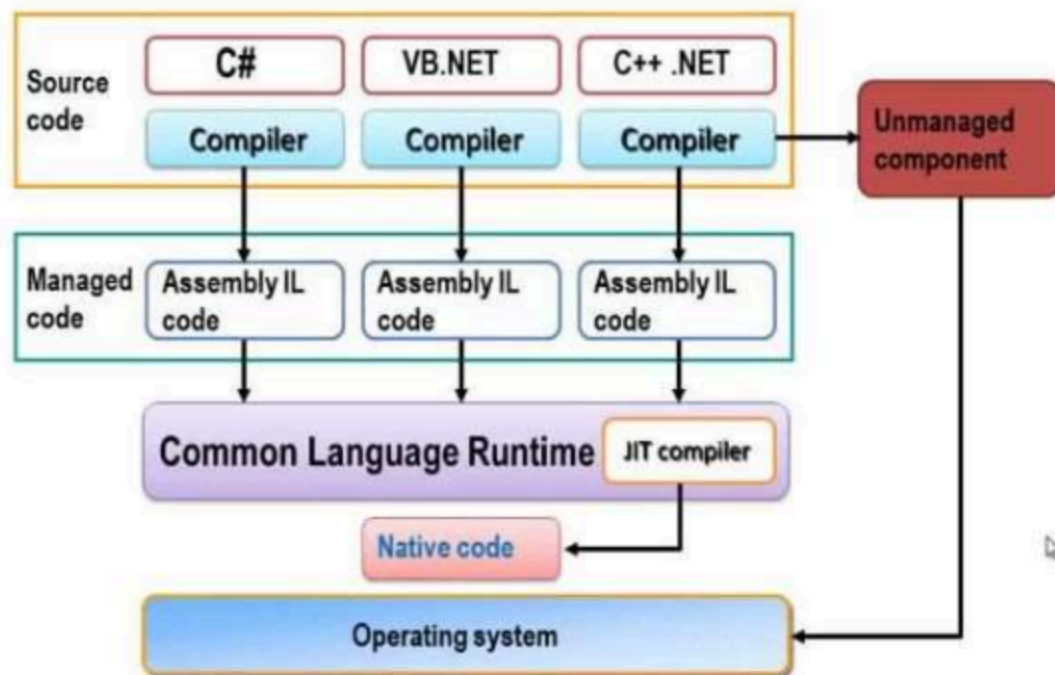
📅 2025-08-08 ▲

## Managed vs Unmanaged Code

كل لغات الـ .NET هي Managed Code لانها تحت مراجعه الـ CLR ولاكن ف مصطلح ثاني وهو **Unmanaged Code** :

يعني احيانا انت بتحتاج تتعامل مع اكواد و مع لغات خارج فريم وورك الـ .NET زي جافا مثلا او ++C  
الـ .NET اي لغه ثانيه دي بقا الي بيبقا اسمها Unmanaged Code  
الـ CLR عنده تحكم بس على الـ Managed Code و لغات البرمجه بس الي تبعه

### The CLR Execution Model



## Summary :

- Managed Code : runs inside CLR, benefiting from automatic memory management, security, and cross-platform execution.
- Unmanaged Code : runs outside CLR,

requiring manual memory management and being more prone to errors.  
✓ .NET provides interoperability features like P/Invoke and COM Interop to communicate with unmanaged code.

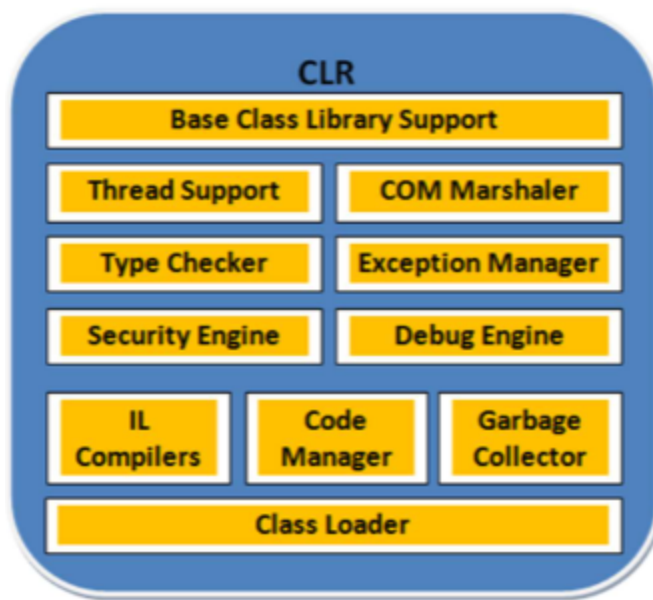


## هناخذ بقا تركيب ال CLR - CLR Structure

محتويات CLR في .NET :

1. Base Class Library Support :  
.NET يقوم بدعم الكلاسات الموجودة في
2. Thread Support :  
تسمح لك بتشغيل امرين في نفس الوقت  
مثال : لو كان لديك فانكشن يستدعي 10 الاف رقم عشوائي و كنت تريد ان تشغله مرتين تستطيع  
Thread Programming تشغيل الفانكشن مرتين في نفس الوقت مع بعض باستخدام ميزة تسمى  
. بدلا من تشغيل الفانكشن حتى ينتهي ثم نشغله مرى اخرى
3. COM Marshaler :  
ويعربط بينهم CLR بيخليك تنادي مكتبات و فانكشنز من ال
4. Type Checker :  
.NET يتأكد ان الداتا تايب او الفانكشنز ماشيه مع قوانين ال
5. Exception manger :  
مسؤول عن الايرور هاندل الي بيحصل
6. Security Engine :  
.NET مسؤول عن السيكيورتي في كل لغات ال
7. Debug Engine :  
على فانكشن مكتوبه وانت debug على اكثر من لغه و تستطيع عمل debug محرك يسمح لك بعمل  
بتكتب بلغه الفريم وورك
8. IL Compilers :  
Native Code لكي يترجمها الى Just In Time Compilation (JIT) هي لغة وسيطة ياخذها  
تتوافق مع بعضها مما يسمح بعمل اللغات المختلفة في .NET. تستخدم لجعل جميع اللغات بداخل  
بالعمل مع بعض .NET.
9. Code Manger :  
Common Language Runtime (CLR) يقوم بادارة كل محتويات بداخل
10. Garbage Collector :  
يقوم بعملية ادارة الذاكرة بشكل اوتوماتيك مما يسمح له بتنظيف الذاكرة التي ليس لها استخدام في الكود  
لاحقا
11. Class Loader :  
.NET مسؤول عن رفع الكلاسات على





٣

## .NET Framework Class Library (FCL)

مجموعه من الكلاسس (عدها آلاف) كل كلاس مسؤول عن شئ معين

.NET Framework Class Library is the collection of classes, namespaces, interfaces and value types that are used for .NET applications.

→ It contains thousands of classes that supports the following functions.

بتعمل support لكل ده واكثر :

- الداتا تاييس - Base and user-defined data types
- الايرور هاندل - Support for exceptions handling
- المدخلات و المخرجات - input/output and stream operations
- تقدر تتواصل مع النظام بتاعك - Communications with the underlying system
- تقدر تربط المشروع بالداتا بيز - Access to data
- الويندوز فورمز - Ability to create Windows-based GUI applications
- الويب فورمز - Ability to create web-client and server applications
- Support for creating web services

و آلاف المكاتب

٣

هناخد بقا دلوقتي اي هي لغة (C#)

## ولي اصلا بنتعلمها

هي لغة متعددة الاهداف وهي لغة OOP ليست لغة فانكشن

( ال-++C كانت لغة تدعم الاتنين فانكشن و OOP) و ال-#C هي لغة تعتبر حديثه نسبيا و قويه و سريعه

وهي لغة من اشهر اللغات في برمجته تطبيقات الويب و الديسكتوب , هي مبنية على لغات الـ C,C++,JAVA .

## مميزاتها :

انها لغة بسيطة Simple تقدر تفهم الكود الي انت شايفه و تقرأه  
هي لغة type safe : يعني متقدرش تضيف مثلا حرف داخل integer ولا العكس ولا اي حاجة شبه كده  
كل داتا تايب خاص بنفسه و هي كتير مشهوره فا ليها مجتمع كبير .

## ممكن نعمل اي بالـ #C ?

كل حاجة بمعنى كل حاجة

