

Start In C#

□ ▲ ⚡ 2025-08-06

م

.NET Start

- في البداية 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      internal class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("hi this is my first c# app");
14             Console.ReadLine();
15         }
16     }
17 }
18
19
20 }
21
22 }
23 }
```

م

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

1. library

مجموعه من الفانكشنز تستخدموهم 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 التي تحكم في كتابه الكود

ممكن نقدر نقول بربو ان دي بيئه العمل بتاعت كل مكان بتبقا مخصوصه ليه هو بالتحديد 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

الـ 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

الي هناده دلوقتي بقا هتبقى معلومات عامة عن **NET**. بشكل عام

.NET بشكل عام هي تعتبر platform

.NET Is a platform

- Languages in .Net

- C#
- Visual basic
- F#

كل اللغات دي بتستخدم نفس المكاتب

- platforms

- .NET core
- .NET framework

كانت هي .NET. أول بلات فورم نزلت لما نزل الـ .NET framework
أحدث الـ .NET core.

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)

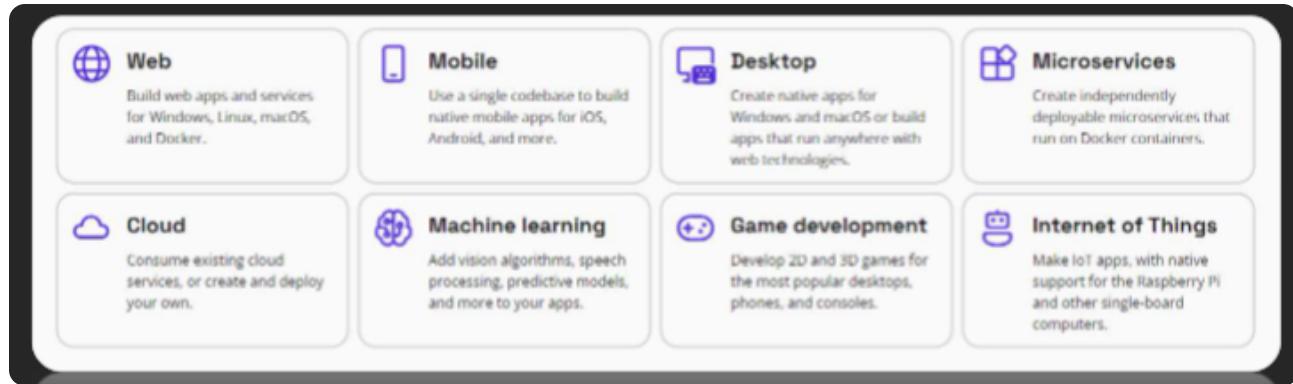
كل دول بيستخدمو نفس المكاتب وتقدر تستخدم كله في كله مش علشان احول من فريم وورك لفريم وورك اتعلم مكاتب جديده

تقدر تعمل اي بقا باـ .NET ؟

- by using .NET YOU Can build

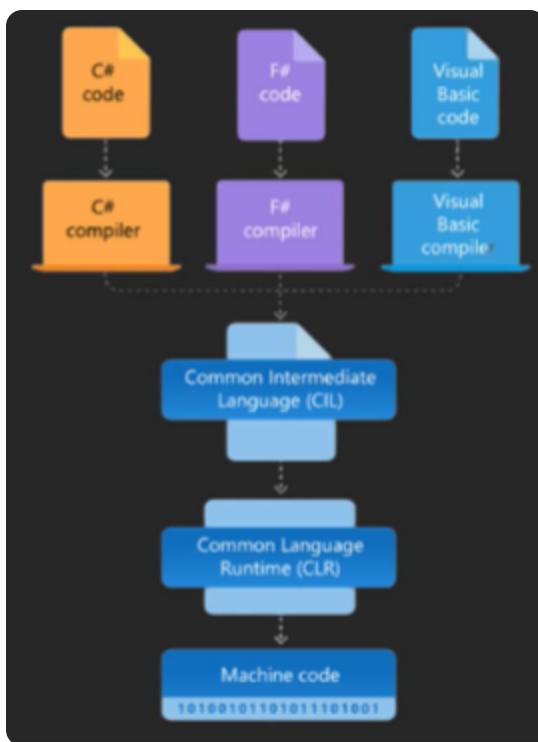
1. desktop apps
2. web apps
3. mobile apps
4. games
5. IOT (Internet of things)

6. AI



احنا هندرس.NET core مش.NET Framework. لان الفريم ووراك اقدم و اغلب الشراكات الان تستخدمه و مغيرتش برامجها للكور لسه الكور احدث لكن لازم تتعلم القديم علشان الجديد بيقا سهل عليك

هعرف بعد حاجه اسمها Compilation in .NET



المختصر

عندى لغات برمجه عديده و كل لغة برمجه ليها كومبایلر خاص بيها الكومبایلر ده مش بيحول الكود بتاعك علطول للغه الكومبيوتر الي هي (1010 باينري - machine) لا الكومبایلر بتاع اللغه بيحول الكود الي انت كتبته لـ CIL الي هي تبقى common code او بمعنى لغه وسطيه و بعد كده يجي الـ CLR الي بيقى intermediate language او language runtime عن كل اللغات يعمل لـ machine code runtime لـ language runtime لـ machine code

علشان كده كل لغات الـ .NET نفس السرعه.

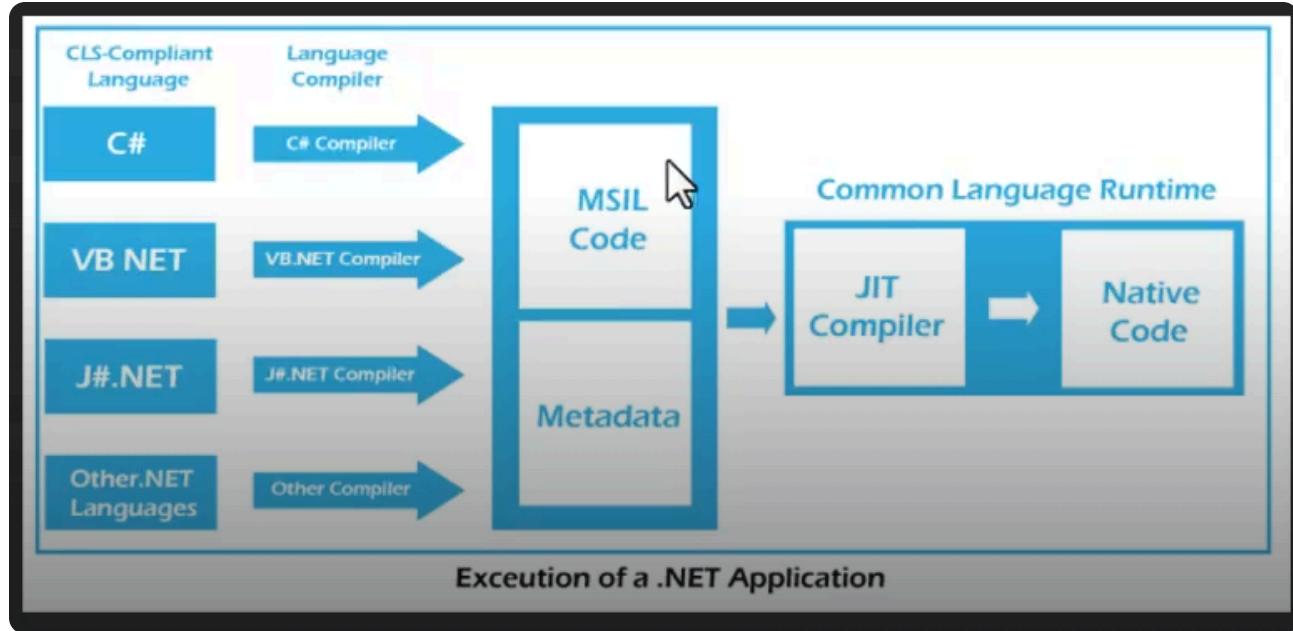
هعرف بقا دلوقتي

.NET Framework Architecture

- two major components
 - CLR لـ APP الي بيعمل :
 - Class library provides :

- what is the CLR :

يحوّل الكود من machine code لـ CIL (Common intermediate language)



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

م

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 كومبایلر خاص بيها ، وظيفة هذه الكومبایلرز تحويل اللغات للغة وسطية موحدة ومن ثم الى لغة التجميع وتكون بملف تنفيذى بعدها يقوم CLR بعمله بتحويل هذه اللغة الوسطية لغة آله، ازاي بقا ؟ اول شيء يجي دور CTS وهذا يقوم بتحويل تابع اللغات الى تابع 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 سلس ومتافق مع البلاتفورم .NET.

3. Garbage Collector (GC) :

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

4. Just in time compiler (JIT) :

. Native code يقوم بتحويل اللغة الوسطية للغة الآلة او بتحول الـ MSIL Code لـ ملاحظه جانبیه - Side Note :

MSIL اختصار لـ Microsoft Intermediate language

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 الكومباير الافتراضي يقوم بترجمة الجزء المطلوب فقط من الكود ويخزنها في الذاكرة في الكاش ميموري، يكون بطىء 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 ايه Assembly : هو ببساطة الملف اللي بيكون بعد ما تكتب كود C # وتعمل له Build. يعني الملف exe أو DLL اللي بنسفله أو نستخدمه. مثلاً: لو عملت برنامج اسمه MyApp هيتكون ملف اسمه MyApp.exe .

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

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

هناخد دلوقتی بقا وظائف CLR اړ

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

3. exception handling
4. provide type-safety
5. memory management
6. security
7. performance

م

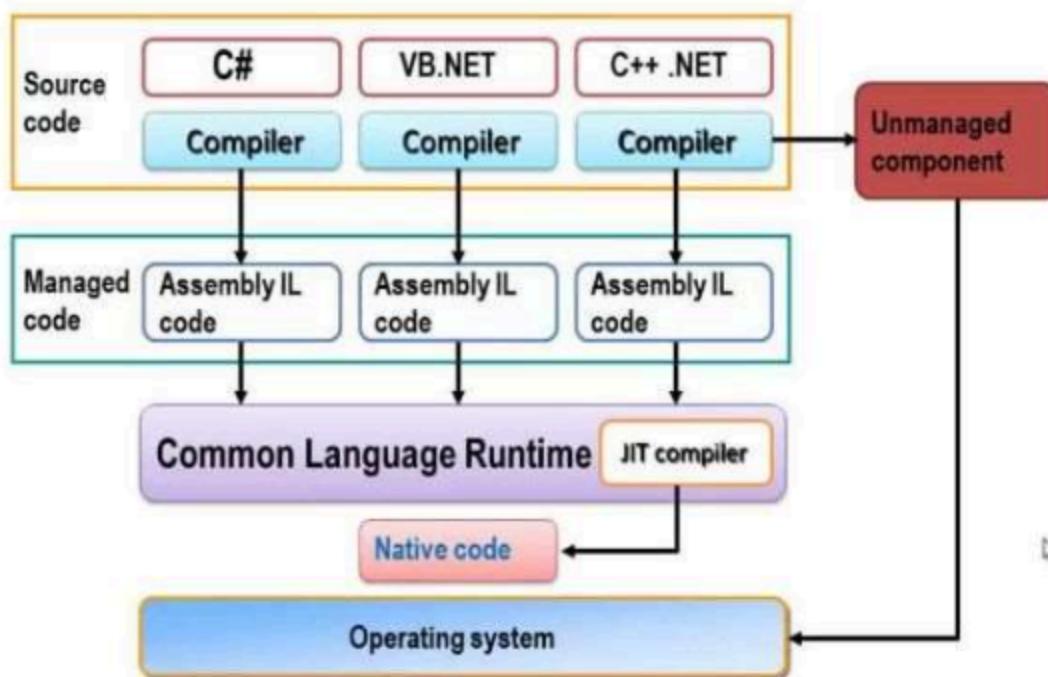
📅 2025-08-08 ⚡

Managed vs Unmanaged Code

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

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

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.

هناخد بقا تركيب الـ CLR - CLR Structure

محتويات CLR في .NET :

1. Base Class Library Support :

يقوم بدعم الكلاسات الموجودة في .NET

2. Thread Support :

تسمح لك بتشغيل امررين في نفس الوقت

مثال : لو كان لديك فانكشن يستدعي 10 الف رقم عشوائي و كنت تريدين ان تشغله مرتين تستطيع تشغيل الفانكشن مرتين في نفس الوقت مع بعض استخدام ميزة تسمى Thread Programming بدلا من تشغيل الفانكشن حتى ينتهي ثم تشغله مرتين اخرى .

3. COM Marshaler :

وبيربط بينهم CLR بخلاف تبادل مكتبات و فانكشنز من الـ

4. Type Checker :

يتاكد ان الداتا تايب او الفانكشنز ماشيء مع قوانين الـ .NET

5. Exception manager :

مسؤول عن الايرور هاندل الي بيحصل

6. Security Engine :

مسؤول عن السيكيورتي في كل لغات الـ .NET

7. Debug Engine :

على فانكشن مكتوبه وانت debug على اكتر من لغه و تستطيع عمل debug محرك يسمح لك بعمل بتكتب بلغه الفريم وورك

8. IL Compilers :

لكي يترجمها الى Native Code Just In Time Compilation (JIT) هي لغة وسيطة ياخذها تتوافق مع بعضها مما يسمح بعمل اللغات المختلفة في .NET. تستخدم لجعل جميع اللغات داخل بالعمل مع بعض .NET.

9. Code Manger :

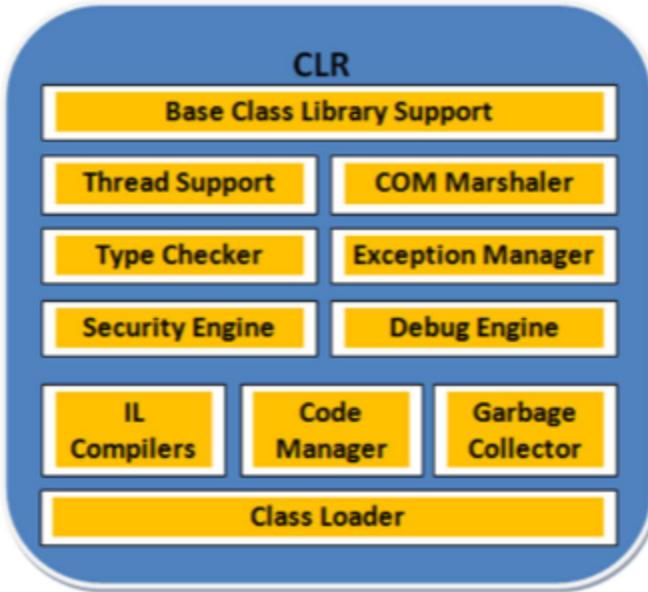
يقوم بادارة كل محتويات داخل Common Language Runtime (CLR)

10. Garbage Collector :

يقوم بعملية ادارة الذاكرة بشكل اوتوماتيك مما يسمح له بتنظيف الذاكرة التي ليس لها استخدام في الكود لاحقا

11. Class Loader :

مسؤول عن رفع الكلاسات على .NET .



.NET Framework Class Library (FCL)

مجموعه من الكلاس (عدها ألف) كل كلاس مسؤول عن شيء 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 واكتэр :

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

وألاف المكاتب

هناخد بقا دلوقتی اي هي لغة (#C)

ولی اصلاح پر تعلمها

هي لغة متعددة الأهداف وهي لغة OOP ليست لغة فانكشن

(C++) كانت لغة تدعم الالذين فانكشن و OOP) والـ C# هي لغه تعتبر حديثه نسبيا و قويه و سريعة

وهي لغه من اشهر اللغات في برمجه تطبيقات الويب و الديسكتوب ، هي مبنيه على لغات الـ . C,C++,JAVA

مميزتها :

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

ممكن نعمل اي بالـ #C ?

كل حاجه بمعنى كل حاجه

