[Library vs Framework vs Platform](https://github.com/SaleemAdhamKassab/ProblemSolving/blob/main/Back-End/Programming%20Advices/014_C%20Sharp_Level%201/001_Brief/001_Library%20vs%20Framework%20vs%20Platform" \o "001_Library vs Framework vs Platform)

1- Library vs Framework vs Platform?

1- Library:

- هي مكتبة تتكون من مجموعة من الفانكشنز اللي الهن علاقة مع بعض مشان احسن استدعيهم في أي مشروع من غير ما اتطر ارجع اكتبهم كل مرة

- مثال عليهم المكتبات اللي عملناهم الخاصة بالسترينغ أو الدايت وغيرهم

- فقط بعملها (using) وبستخدمها بكل سهولة

2- Framework:

- هو أوسع من الليبرري

- مكون من مجموعة من المكتبات ومجموعة من القوانين اللي بتحكمك انت كديفلوبركيف تكتب الكود

- كل شركة الها الفريمورك الخاص فيها اللي هي القوانين تبعها في كتابة الكودوتعطيك الستركتشر أو هيكلة البرنامج : يعني شلون بدك تشبك على الداتا بيز

أو شلون بدك تستدعي الداتا(هاد الحكي بخلي كل الديفيلوبرز يشتغلو على نسق واحد ومو كل واحد بيشتغل على كيفه)

يعني لو بدك تبعت ايميل مافيك تبعته على كيفك لازم تتبع الفريم الورك اللي الشركة حاطته مثلا بقلك لازم تستدعي المكتبة الفلانية بالطريقة الفلانية

بالاضافة لالاف الامور الموجودة بالفريم وورك اللي هي بتنظم عملية كتابة الكود

بالمختصر : هو بعرف مجموعة من القوانين والبروتوكولات التي يجب على الديفيلوبر اتباعها عند كتابة الكود

Ensure that all parts of an application work well together and follow a consistent design pattern.

بيئة عمل بتخلي الكل يشتغل على نسق واحد

3- Platform:

- Platform = Programming Languages + Libraries

- Run Apps That written by .Net Languages (like c#, F#..)

- هو أوسع من الفريمورك

- هو الاينفيرومينت اللي منعمل عليها (رن:تشغيل) للسوفت وير

- Library --> Framework --> Platform اللي بيعمل رن لهاد الحكي كله

[**What is .NET, .NET Core vs .NET Framework, what can we do with Dot NET**](https://github.com/SaleemAdhamKassab/ProblemSolving/tree/main/Back-End/Programming%20Advices/014_C%20Sharp_Level%201/001_Brief/002_What%20is%20Dot%20NET_Dot%20NET%20Core%20vs%20Dot%20NET%20Framework_What%20can%20we%20do%20with%20Dot%20NET)**?**

1- What is .Net ?

- It is a comprehensive development platform that provides tools and technologies for building all types of applications,

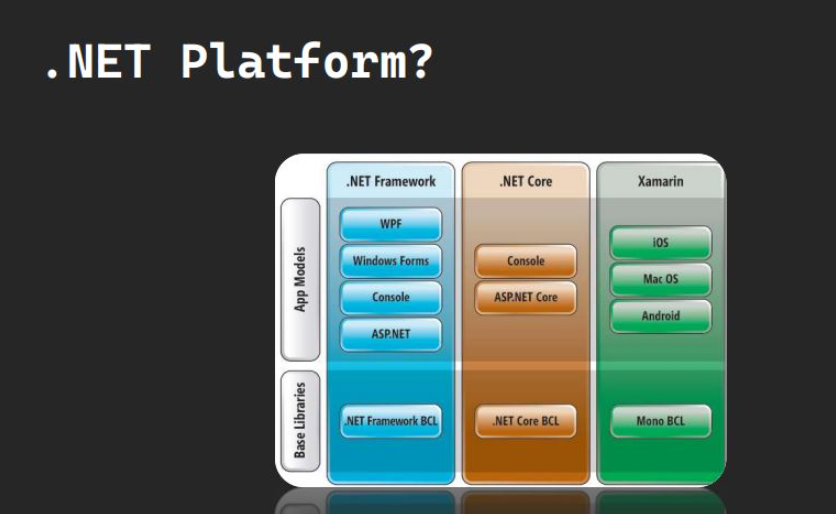
including web, desktop, mobile, gaming, IoT, AI, and cloud.

- يحتوي الدوت نت على العديد من لغات البرمجة المدعومة من قبل الدوت نت بلاتفورم

(c#, Visual Basic, F#..)

- كل اللغات تستخدم نفس المكتبات وكل اللغات الموجودة بالدوت نت هي أوجه لعملة واحدة يعني كلهم نفس البيرفورمنس

- .Net Platform Components:



1- .NET Framework: released in 2002 (Windows forms, Console, Asp.Net)

2- .NET Core: (Console, ASP.NET Core)

3- Xamarin: (IOS, Android)

In short, .NET is a platform, while .NET Framework is one of its components.

[**Compilation in .NET**](https://github.com/SaleemAdhamKassab/ProblemSolving/blob/main/Back-End/Programming%20Advices/014_C%20Sharp_Level%201/001_Brief/003_Compilation%20in%20Dot%20NET.docx)

1. كل لغات الدوت نت بلاتفورم هي (Managed Code) أي يتم ادارتها من قبل مدير وهو

(CLR: Common Language Runtime)

1. Compilation Steps
2. كل لغة من لغات الدوت نت الها الكومبايلر الخاص بها ومهمة الكومبايلر لكل لغة هي تحويل الكود المكتوب بهذه اللغة الى Intermediate Language
3. الان تأتي مهمة المانجر المسؤول عن كامل هذه اللغات وهو ال CLR الذي يحتوي بداخله على مايسمى بال JIT: Just-In-Time Compiler فينا نعتبره متل ال Interpreter مسؤوليته أنه الكود اللي انت بحاجته بيعمله كومبايليشن بالرن تايم للماشين كود (الخطوة اللي بعده)

مثلا لو أنا عندي شاشة فيها زرين:

لما اكبس ع أول زر ال JIT بجييب الكود اللي استخدمته بالزر الأول

في الرن تايم --< بيعمله كومبايلايشن بالرن تايم لمرة واحدة --< وبخزنه في الميموري

يعني لوعندي شاشة فيها 100 زر وانت بس كبست على زر واحد --< فقط الفانكشنز اللي بهاد الزر هنن اللي بينعملهن كومبايليشن في الميموري (يعني خلاصة ال JIT هي بعمل كومبايل فقط للشغلات اللي بحتاجها بال runtime)

هاد الشي اله مساوء واله حسناته : بال Startup بكون بطيء (لآنه أول مرة بيعمل كومبايليشن) بعدين بصير سريع

فإذا انت بدك شي بال Startup يكون سريع بدك تغير على مود ثاني وهو ال:

Head Of Time Compilation: اللي هي بتعملك كومبايل لكامل الكود وساعتها هون ال JIT بيطفي وبتصير متلها متل ال Compiled Language

منستنتج من هاد الحديث أنه .Net بتدعم ال Both ال (Compiled and Interpreted Languages)

1. Machine Code

[**.NET Framework Architecture**](https://github.com/SaleemAdhamKassab/ProblemSolving/blob/main/Back-End/Programming%20Advices/014_C%20Sharp_Level%201/001_Brief/004_Dot%20NET%20Framework%20Architecture.docx)

.Net Framework Architecture has two major components:

**.Net Framework لايعتبر فريم وورك وانما يعتبر Platform لأنه بيعمل Run لل Applications**

1. CLR:

هو ال engine اللي بيعملك run للتطبيق

It provides services like **thread management, garbage collection, type-safety, exception handling, and more**.

رح نفصله بالدروس القادمة

1. Class Library:

* ميزة الدوت نت انه بيزودك بمجموعة هائلة من الكلاسات اللي بتمكنك تبرمج بسلاسة وسهولة (وهذه الكلاسات مدعومة بكامل لغات الدوت نت )
* هذه الكلاسات هي مجموعة من ال APIs وهي بالالاف وهي غنية جدا تسرع عملية البرمجة بشكل كتير كبير Rich Classes
* نذكر منها:

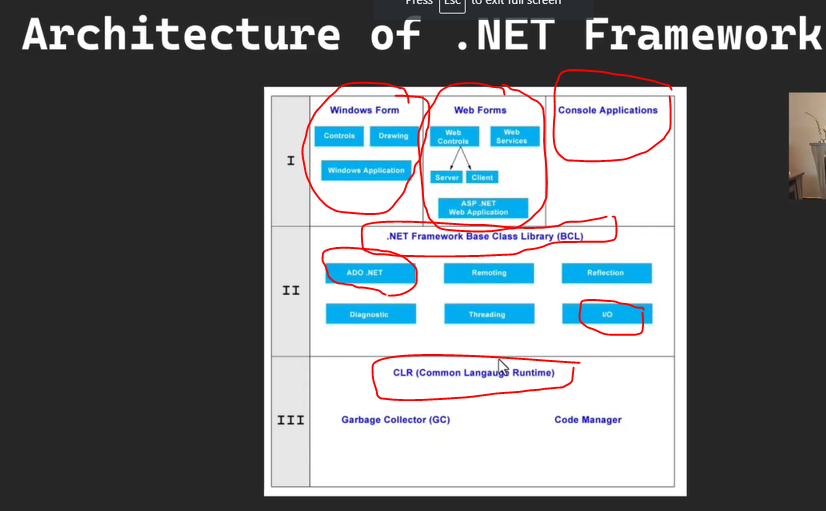
Database Class Library

Handling Files Class Library and More…

Another Major Component is : Languages

Latest Versions

* Latest version of .NET framework is: .NET Framework 4.8.1
* Latest version of .NET is: 6.0 تدعم ال Core

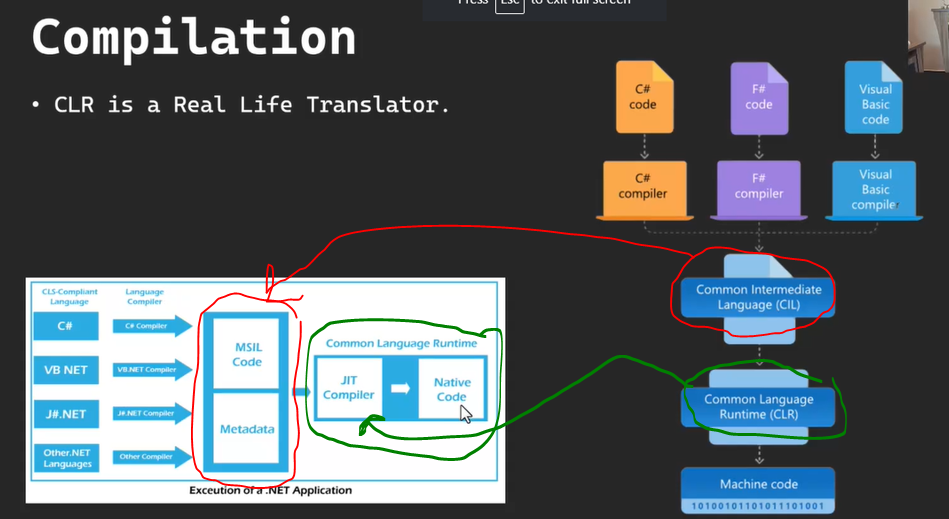


[**What is CLR**](https://github.com/SaleemAdhamKassab/ProblemSolving/blob/main/Back-End/Programming%20Advices/014_C%20Sharp_Level%201/001_Brief/005_What%20is%20CLR.docx)

Common Language Runtime (CLR)

1. CLR is a real-life translator

متل ما منعرف سابقا انه بحول من ال Intermediate Language الى ال Machine Code عن طريق ال JIT



1. Responsible of execution .Net Applications
2. Provide a number of services to .Net Applications like:

سنشرحهمم لاحقا بالتفصيل

* 1. Memory Management
  2. Type safety
  3. Exception handling
  4. Other…

1. بمعن اخر يعتبر CLR بمثابة جسر يعبر عليه الكود تبعك لحتى يصل لل Operation System

CLR Main Components

1. Common type System (CTS):

لنختصر الموضوع:

بما انه الدوت نت فيه عدة لغات برمجة --< فبالتالي كل لغة الها DataTypes خاصة بها مثلا:

VB: Integer

C#: int

ومتل مامنعرف سابقا انه الكومبايلر بكل لغة بحول الكود الى Intermediate language التي تحتوي ضمنها على

Common Type System

يعني النوعين السابقين باللغتين سيتم تحويلهم الى Int32 (بينعملهن Mapping) مشان هيك هي اسمها Type Save

Type Save: data types are managed by (clr: cts)

1. Common Language Specification (CLS):

متل مامنعرف انه الدوت نت فريم وورك يحتوي على مايقارب ال 70 لغة (c#, vb, f#....)

وحتى ترتقي هي اللغة لحتى تصير من ضمن لغات الدوت نت فريم وورك **لازم تتبع قواعد وقوانين الفريم وورك**

ليش؟

**لأنه الفريم وورك بخليك تعمل توافقية integration/ interoperability بين هذه اللغات**

يعني هي اللغة تحكي مع هي اللغة , هي اللغة تدعم ال Type system تبع اللغة الأخرى وهكذا...

حيث أنه يوجد الالاف القوانين والقواعد اللي لازم اللغة تتبعها مشان تصير ضمن اطار الفريم وورك

**السؤال هنا مين اللي بيعطي هي القوانين والقواعد؟؟ هو ال CLS: Common Language Specification**

لما كل هاللغات عم تتبع نفس قوانين الدوت نت فبصير أسهل عليها انها تتكامل مع بعضها:

يعني تخيل انك عم تشتغل على vb وتستخدم فنكشن بال C#

طيب هاد التكامل وانت عم تعمل debugging (بتكون بال vb وبتنتقل على ال C# وبكمل معك) هاد الحكي كله مين بحققه؟

كمان ال **CLS: Common Language Specification**

**كل لغة الها ال Syntax الخاص بها لكن جميعن يتبعوا قوانين ال cls**

1. Garbage Collector (GC):

في ال c++ لازم نضف ال pointers اللي تركتها وراي لكن في اللغات الحديثة عندي ال gc يعني لما يشوف ال objects في الميموري غير مستخدمة يقوم بحذفها بشكل تلقائي

It also ensures the safety of objects by not allowing one object to use the content of another object.

بساعدك بموضوع ال

**Automatic Memory management** فانا كديفيلوبر مابضل مشغول بالي بإدارة الميموري

1. Just In Time Compilation (JIT):

ي