

.NET Versions

.NET Framework

- ❖ ده الإصدار الأصلي والأساسي اللي بدأ سنة 2002.
- ❖ فضل يتطور من فيرجن 1.0 لحد ما وصل لآخر إصدار ليه وهو **4.8**.
- ❖ مشكلته انه كان Platform dependent يعني كان شغال على **Windows بس**.
- ❖ مايكروسوفت وقفت تطويره، يعني مفيش فيرجن 4.9. هو لسه مدعوم وهيجيله تحديثات أمان، لكن مفيش مميزات جديدة.

.NET Core (2002-2019)

- ❖ البداية - **.NET Framework 1.0** (2002)
- ❖ (Final Release) آخر إصدار - **.NET Framework 4.8** (2019)

.NET Core

- ❖ سنة 2016، مايكروسوفت عملت نسخة جديده منه ومن اهم مميزاتها:
- ❖ **Cross-Platform:** Windows و Mac و Linux تشتغل على
- ❖ **Open-Source:**

.NET Core (2016-2020)

- ❖ لأول مرة Cross-platform - **.NET Core 1.0** (2016)
- ❖ .NET Standard 2.0 - **.NET Core 2.0** (2017)
- ❖ Desktop Apps اضاف - **.NET Core 3.0** (2019)
- ❖ LTS Version - **.NET Core 3.1** (2019)

Modern .NET (2020-Present)

- ❖ بعد كد سنة 2020، مايكروسوفت قررت تثبيت اسم واحد ليه وهو: **.NET**.
- ❖ عشان الناس متلخبطش بين.NET Framework 4.8 و.NET 4 الجديد، قرروا ينطوا رقم 4 خالص.
- ❖ فيقى عندنا:
- ❖ **.NET Core** و **.NET Framework** وحدوا - **.NET 5** (2020)
- ❖ LTS Version - **.NET 6** (2021)
- ❖ Support سنة واحدة - **.NET 7** (2022)
- ❖ LTS Version (الحالي) - **.NET 8** (2023)
- ❖ جديد خالص - **.NET 9** (2024)
- ❖ وهكذا... كل سنة بينزل إصدار جديد في شهر نوفمبر.

Solution & Projects

Solution

- ❖ الـ **Solution** ده اللي بتحط فيها كل Projects بتاعتك
- ❖ ملف الـ Solution بيكون امتداده **sln**
- ❖ ممكن يكون فيه **Project واحد** أو **Projects كتيرة** بتشتغل مع بعض
- ❖ بيساعدك تنظم شغلك وتدير المشاريع كلها من مكان واحد

Project

- ❖ الـ **Project** ده المشروع الفعلي اللي فيه الكود بتاعك
- ❖ كل Project بيطلع **ملف واحد** في الآخر يا إما **exe** (برنامج) أو **dll** (مكتبة)
- ❖ ملف الـ Project بيكون امتداده **csproj** للـ #C

مثال يوضح العلاقة:

```

  MyECommerceStore.sln (Solution)
  |
  ├── MyStore.Web (Project -> "Website")
  |   |
  |   ├── Controllers/
  |   ├── Views/
  |   └── MyStore.Web.csproj
  |
  └── MyStore.API (Project -> "The API")
      |
```

```
| | | Controllers/
| | | Models/
| | | MyStore.API.csproj
|
└─ 📁 MyStore.Database (Project - "Dealing with Database")
|
└─ Models/
└─ Repositories/
└─ MyStore.Database.csproj
```

Namespace

- ✦ سيساعدك في تنظيم الـ **Classes** و الـ **Methods** بتاعتك
- ✦ بيمنع الـ **Conflict** في الأسماء لو اتنين كتبوا Classes بنفس الاسم

A real-life example:

- ✦ لو عندك مثلا 2 صاحب اسمهم احمد ف انت لازم تفرق بينهم ب شئ معين ث مثلا ممكن نقول "أحمد" صحتبي في **الشغل** و "أحمد" صاحبتي في **الجامعة**
- ✦ لما بتتكلم عنهم بتقول "أحمد اللي في الشغل" أو "أحمد اللي في الجامعة"
- ✦ الـ Namespace نفس الفكرة بالظبط!

مثال في الكود:

```
namespace MyBank.Accounts
{
    public class Customer
    {
        public string Name { get; set; }
        public int AccountNumber { get; set; }
    }
}

namespace MyStore.Users
{
    public class Customer
    {
        public string Name { get; set; }
        public string Email { get; set; }
        public List<Order> Orders { get; set; }
    }
}
```

- ✦ هنا عندنا **اثنين** **Classes** اسمهم **Customer** لكن في **Namespaces** مختلفة
- ✦ فمفيش مشكلة خالص عشان كل واحد ليه "عنوانه"