# **.NET Nedir?**

***.NET 4 ana kısımdan oluşur:***

1. **Programlama Dilleri:**

Frameworkler programlama dillerini kullanarak bir şeyleri sıfırdan implement etmek yerine bize hazır kullanabileceğimiz kolaylıklar sunan yapılardır. .NET özellikle C#, F# ve VB.net ve bunlarla sınırlı olmayan dilleri destekler.

1. **Runtime:**

Runtime sistemi hem oluşturulduğu bilgisayarda hem de hedef cihazlarda aynı anda bulunan bir alt sistemdir. Çoğu programlama dili yazılan programların çalıştırılması için bir çeşit runtime sistemine sahiptir ve bu runtime sistemi programların çalışabilecekleri ortamı sunar. Bu ortam hafıza denetimleri, programın değişkenlere ulaşabilmesi, prosedürler arasında parametrelerin iletilmesi ve işletim sistemi ile interface’lerin sağlanması gibi gerekli eylemleri gerçekleştirir.

Kodlar derlendikten sonra runtime’ın sağladığı ölçüde mobil cihazlarda, bilgisayarlarda, server’larda çalıştırılabilir..NET5 bunun da ötesinde derlenmiş kodun herhangi bir platformda çalıştırılabilmesini sağlar.

1. **Kütüphane Koleksiyonu:**

.NET framework içinde çok fazla sayıda kütüphane barındırır ve bu kütüphaneler sundukları hazır yapılarla üretimi hızlandırır ve kolaylaştırır.

1. **Aletler (Tools):**

Farklı uygulamalar için birçok sayıda taslak sunar, projeyi yapılandırır ve dış bağlantılar için kullanması kolay arayüzler sağlar.

**.NET Framework Nedir?**

2002 yılında microsoft tarafından geliştirilmiştir. Örneğin bir C# kodunu derlediğimizde onu bir “ortak ara dil” (common intermadiate language (CIL)) ’e dönüştürürüz. .NET üzerinde derlendiğinde CIL’in .NET’in konuştuğu dil olduğunu söyleyebiliriz. CIL ardından common language runtime (CLR, ortak dil çalışması) tarafından okunur ve makine diline çevirilir. Bu CLR .NET framework içinde bulunur.

20 yıldır var olan .NET framework, doğal olarak teknolojinin de ilerlemesi ile sürekli bir değişikliğe tabi tutuldu. Bu değişiklikler 20 yıl boyunca bir çok şeyi temelli farklı bir yöne sürdü. Ancak bu değişiklikler yapılırken aynı zamanda görece eski kalan teknolojilerin de hala .NET framework ile uyumlu olabilmesi için oldukça fazla sayıda prosedür gerekti. Bu prosedürlerin devamlı üst üste eklenmesi ile .NET framework oldukça yavaşladı.

2016 yılında .NET Core çalışmaları başlatıldı, bu çalışmalar oldukça fazla sayıda geçmiş teknoloji ile uyumluluğu arkada bıraktı ve daha geleceğe dönük yapıldı. Bu .NET Core’un çok daha hızlı olmasını sağladı.

Kafa karışıklığı buradan kaynaklanıyor. Genel yapısı aynı veya fazlasıyla benzer olmasıyla beraber .NET frameworkten daha hızlıydı. .NET Core 3.0’la birlikte .NET framework’ün sağladığı tüm soyutlama işlemlerini gerçekleştirebiliyordu. .NET Core 4.0 versiyonu çıkarılmak yerine 4.0 atlandı ve bundan sonra çıkan yeni .NET Core versiyonu sadece .NET olarak adlandırıldı.

Aynı zamanda .NET standard denen bir platform daha var. .NET standard’ın çıkış nedeni iki farklı platformu olmasıydı. .NET standard bu iki platformu birbirine köprüleyecek ve ikisinin de gereken işlevselliklerini kullanacaktı. Sadece .NET framework ve .NET core’u birbirine bağlamakla kalmadı ve mono, xamarin, unity gibi kendi soyutlama katmanları olan dilleri de köprüledi. Ancak bu sadece kütüphaneler ile sınırlıydı.

.NET 6 ile birlikte bu soyutlama katmanları tamamen tek bir soyutlama katmanına dönüştürülmesi hedefleniyordu ve .NET standard gibi köprüleme yapacak katmanlara ihtiyaç duyulmayacaktı ve çalışmaları .NET 7 ve gelecek olan .NET 8 ile devam ediyor.